شبیه سازی مونت کارلو

محمد رضا اجتهادی

Ali Farnudi iMac
Sony Electronics, Inc.
2/5/2008

Table of Contents

1. مقدمه 6

1.1. شبیه‌سازی در مقایسه با آزمایش 7

1.2. محدودیت‌های شبیه سازی 7

1.3. شبیه سازی با استفاده از آلگوریتم های تصادفی 8

1.4. چند توصیه عمومی 8

5.1. پیش نیازهای این کتاب 10

2. فراکتال‌ها 11

2.1. فراکتال‌های خود شبیه 11

2.2. بعد فراکتال* 13

بعد خود تشابهی 13

بعد جرمی: 14

2.3. شبیه سازی فراکتالهای خود تشابه 14

بیشتر بدانیم: 21

3. لایه نشانی 22

بیشتر بدانیم: 33

4. تراوش 34

4.1. آلگوریتم رنگ کردن 34

4.2. آلگوریتم رشد خوشه ..... **Error! Bookmark not defined.**

4.3. تصحیح بعد محدود 43

4.4. تمرین ..... **Error! Bookmark not defined.**

5. ولگشت 51

5.1. ولگشت ساده 51

5.2. ولگشت ساده با تله 54

5.3. تجمع ..... **Error! Bookmark not defined.**

5.4. ولگشت خود پرهیز ..... **Error! Bookmark not defined.**

5.5. تمرین ..... **Error! Bookmark not defined.**

6. تولید اعداد تصادفی 63

6.1. تابع توزیع 63

6.2. همبستگی 64

6.3. روش تولید اعداد شبه تصادفی با توزیع یک نواخت 64

6.4. روش تولید اعداد تصادفی با توزیع غیر یکنواخت 66

6.5. تمرین ..... **Error! Bookmark not defined.**

7. آلگوریتم های تصادفی برای انتگرال گیری 73

7.1. شلپ 74

7.2. نمونه برداری ساده 75

7.3. محاسبه خطا 77

7.4. نمونه برداری هوشمند 78

7.5. انتگرال چندگانه 80

8. تولید اعداد تصادفی با توزیع دلخواه - روش متروپولیس 81

8.1. فرایند مارکوف ..... Error! Bookmark not defined.

8.2. تعادل 82

8.3. طول همبستگی 88

8.4. نرخ قبول و انتخاب طول قدم 90

9. شبیه سازی آنسامل کانونیNVT 92

9.1. آنسامبل کانونی ..... Error! Bookmark not defined.

9.2. تابع توزیع احتمال بولتزمان ..... Error! Bookmark not defined.

9.3. استفاده از آلگوریتم متروپولیس برای نمونه برداری از آنسامبل کانونی 93

9.4. انتخاب قدم های ارگادیک 95

10. مثال مدل گسسته - آیزینگ 98

10.1. شکست تقارن در مدل آیزینگ 99

10.2. شرایط مرزی پریودیک 99

10.3. شرایط مرزی مارپیچی ..... Error! Bookmark not defined.

10.4. تصحیح ابعاد محدود 106

11. مثال مدل پیوسته - گاز واندر والس 108

11.1. پتانسیل لنارد-جونز 108

11.2. شعاع قطع پتانسیل 108

11.3. لیست همسایگی 108

12. مثال سیستم با قیود پیچیده - ساختارهای پلیمری 109

12.1. مدل درشت دانه پلیمر 109

12.2. برهمکنش ها 109

12.3. حرکت های موضعی 109

12.3.1. گوشه 109

12.3.2. انتها 109

12.3.3. میل لنگی 109

12.4. حرکت های سراسری 109

12.4.1. لولایی 109

12.4.2. میل لنگی بلند 109

12.4.3. خزشی 109

13. شبیه سازی آنسامبل میکروکانونی NVE 110

13.1. آنسامبل میکروکانونی 110

1. مقدمه

سال‌هاست که کامپیوتر وارد زندگی بشر شده است. این ابزار در پیشرفت دانش و سرعت بخشیدن به پژوهش بسیار موثر بوده است. در دانش فیزیک، کامپیوتر کاربردهای فراوانی دارد. از محاسبات عددی، تحلیل داده‌ها و رسم نمودارها گرفته تا کنترل فرآیند آزمایش با کمک یک واسط سخت‌افزاری و یا ثبت داده‌ها. در کنار این کاربردها، شبیه سازی ابزاری است که با کامپیوتر متولد شد و بسرعت جای خود را در دانش و فن‌آوری پیدا کرد. شبیه سازی در پژوهش‌های نظری به ما این امکان را می‌دهد که سیستم‌های پیچیده‌ای که معادلات پایه‌شان را میشناسیم ولی توانایی حل‌شان را نداریم را در فضای مجازی کامپیوتر بازسازی کنیم و رفتار آنها را بررسی کنیم. همچنین این ابزار در فن‌آوری امکان خطا و اشتباه در ساخت را به شدت کاهش می‌دهد. امروزه از این ابزار برای آموزش مهارتهایی که خطای کار آموز می‌تواند بسیار پرهزینه باشد، مانند خلبانی، ناوبری و جراحی استفاده می‌شود. همچنین در شاخه هایی از دانش و فن آوری که نیازمند به پیش بینی آینده در آن زیاد است مانند هواشناسی، اقتصاد، سیاست و جامعه شناسی بسیار کاربرد دارد.

در راه طولانی پیشرفت دانش همواره یک داد و ستد میان آزمایشگاه و نظریه وجود داشته. معمولا به این‌گونه است که در آزمایشگاه پدیده‌ای مشاهده می‌شود و بر آن اساس نظریه‌ای ارائه می‌شود. ولی این پایان داستان نیست که بلکه آغاز کار است. حال نظریه بایستی توانایی خود را در پیشبینی آزمایشهای انجام نشده نشان دهد. آزمایشگران با انجام آزمایش‌های پیشنهادی صحت و یا ضعف نظریه را آشکار می‌کنند. این داد و ستد هیچگاه قطع نمی‌شود و همیشه منجر به تصحیح و یا تکمیل نظریه‌ها می‌شود. ولی نکته مهم این است که هر نظریه‌ای دارای محدوده ی اعتبار خاصی است و از طرف دیگر آزمایشگر هم در آزمایشگاه با محدودیت های تکنیکی مواجه است. در شرایط ایده‌آل نظریه و آزمایشگاه به یکدیگر نزدیک هستند ولی در موارد متعدد آنها از یکدیگر آنقدر دور هستند که برای برقراری این داد و ستد به یک پل ارتباطی نیازمندیم. این پل ارتباطی شبیه سازی می‌باشد.

متناسب با نوع مسائلی که با آنها مواجه هستیم روش‌های متفاوتی برای شبیه سازی وجود دارد. اینکه کدام یک از این روش ها کارا تر و مناسب مسئله است، یکی از مهمترین نکاتی است که یک پژوهشگر باید قبل از شروع به کار در باره آن تصمیم بگیرد. در بعضی از مسائل منظور از شبیه سازی در حقیقت یک محاسبه عددی یا تقریبی برای حل معادلات پیچیده حرکت است. در مسائل دیگری و در غیاب هرگونه معادله حرکتی و فقط با در دست داشتن شکل برهمکنش‌ها، سیستم شبیه سازی می‌شود. یکی از امکانات شبیه سازی این است که در مقیاس‌هایی می‌توان کار کرد که خارج از دسترس آزمایشگران است. تا چندی پیش بررسی سیستم ها در مقیاس‌های چند نانو متری فقط آرزویی برای آزمایشگران بود، در صورتی که در شبیه سازی‌ها براحتی قابل دسترس بود. شاید فن‌آوری نانو در حال حاضر مدیون این شبیه سازی ها باشد.

تعداد بیشماری نرم‌افزارهای آماده به منظور به کارگیری در شبیه سازی مدل های متفاوت تهیه شده اند. در میان آنها نرم افزارهای بسیار خوبی هم هستند که به صورت مجانی در اختیار کاربران قرار می‌گیرد و تنها انتظار تولید کنندگان از کاربران، ارجاع مناسب به کار آنها می‌باشد. در مواردی هم نرم‌افزارهایی به صورت متن باز[1] در اختیار عموم قرار می‌گیرد تا کاربران ضمن استفاده از این نرم‌افزارها بتوانند در تصحیح، تکمیل و یا گسترش آنها مشارکت داشته باشند. ولی معمولا این نرم افزارها دقیقا با مسئله مورد نظر شما هماهنگ نیستند و زمان لازم برای فراگیری کار با آنها و ایجاد تغییرات متناسب با مسئله مورد نظر بیش از نوشتن یک برنامه جدید وقت می‌برد. هرچند من همواره

[1] Open scource

توصیه میکنم اگر منظور از شبیه سازی یاد گیری این دانش نیست و مقصود اصلی حل یک مسئله خاص است، بهتر است که بعد از تصمیم در مورد نوع شبیه سازی، در ابتدا برای یافتن نرم افزار آماده و مناسب تلاش کنیم و در صورت عدم موفقیت شروع به نوشتن برنامه کنیم.

این کتاب سعی دارد که خوانندگان خود را با یکی از پرکاربرد ترین روش‌های شبیه سازی که به مونت کارلو[2] (metropolice) معروف است، آشنا کند. کتاب به گونه این مرتب شده که با آلگوریتم‌های ساده و مسائل جذاب شروع می‌کند و به تدریج خواننده را برای نوشتن برنامه‌های پیچیده‌تر آماده می‌کند. این کتاب به دانشجویان رشته‌های علوم و مهندسی توصیه می‌شود. حجم مطالب این کتاب متناسب با یک درس 2 واحدی(یا نیمی از یک درس 4 واحدی) در مقطع کارشناسی یا کارشناسی ارشد است (Ejtehadi, 2005) .

1.1. شبیه‌سازی در مقایسه با آزمایش

قواعد و مراحل یک کار شبیه سازی بسیار به قواعد و مراحل یک کار آزمایشگاهی نزدیک است. به دلیل همین شباهت گاهی از شبیه سازان می‌شنوید که از لفظ آزمایش برای اجراهای متفاوت برنامه خود استفاده می‌کنند. در آزمایشگاه یک نمونه داریم و در شبیه سازی ما مدلی داریم که آنرا قرار است بیازماییم. در آزمایشگاه یک چیدمان (یا دستگاه) برای آزمایش آماده می‌کنیم و درشبیه سازی از یک برنامه کامپیوتری (یا آلگوریتم) استفاده می‌کنیم. در آزمایشگاه تعدادی آزمایش اولیه برای اطمینان از صحت کار دستگاه و تنظیم (کالیبراسیون) آن انجام می‌دهیم و در شبیه سازی هم تعدادی اجرای اولیه برای رفع نقایص نرم افزار (دی‌باگ) و اطمینان از صحت نتایج انجام می‌دهم. بعد از اطمینان از سلامت کار شروع به انجام آزمایش در آزمایشگاه و یا اجرای برنامه در شبیه سازی می‌کنیم. از اینجا به بعد همه چیز دقیقا یکسان است، جمع آوری داده‌ها، تحلیل داده‌ها، ، گزارش نتایج و نتیجه‌گیری.

جدول 1: مقایسه مراحل انجام یک کار آزمایشگاهی با شبیه سازی

آزمایشگاه	شبیه سازی
نمونه	مدل
چیدمان	برنامه
تنظیم	دی باگ
آزمایش	اجرا
ثبت داده ها	
تحلیل داده‌ها	
محاسبه خطا	
نتیجه گیری	

1.2. محدودیت‌های شبیه سازی

تا اینجا به نظر می‌رسد که هر پدیده‌ای را می‌توان شبیه سازی کرد. ولی در حقیقت اینگونه نیست و محدودیت‌های متفاوتی هم در ابزار و روشها، و هم نظریه وجود دارد که اثر خود را بر ابعاد، زمان و دقت شبیه سازی‌های ما تحمیل می‌کند. حجم حافظه، و قدرت محاسبات کامپیوترها محدود است. به این دلیل ابعاد سیستمهای که شبیه سازی می‌کنیم محدود می‌شود. به طور مثال برای شبیه سازی یک ماده مولکولی ما هرگز نمی‌توانیم یک مول از این ماده را شبیه سازی کنیم و حتی قادر نیستیم که به نزدیکی عدد آووگادرو برسیم. حتی برای سیستم‌های بسیار

[2] نام شهری در شاهزاده نشین موناکو در جنوب فرانسه که به دلیل داشتن قمارخانه‌ای به همین نام معروف است.

کوچکتر شبیه سازی چند صد نانو ثانیه یک شبیه بسیار بزرگ به حساب می‌آید. یا برای شبیه سازی یک کهکشان بایستی از ستاره هایی کوچکتر از خورشید صرفنظر کنیم. شاید این گونه تصور شود که با افزایش قدرت کامپیوتر ها در آینده و یا موازی کردن آنها می‌توان بر این مشکلات غلبه کرد. در پاره‌ای از مسائل این حرف صحیح است ولی بیشتر مسائلی که ما با آنها مواجه هستیم با آلگوریتم هایی شبیه سازی می‌شوند که به آنها NP گفته می‌شود. در این آلگوریتمها پیچیدگی مسئله به صورت نمایی با اندازه سیستم رشد می‌کند و در نتیجه امیدی به حل کامل این مسائل با کمک شبیه سازی وجود ندارد. پیشرفت های نظری در فیزیک و همچنین ابداع روشها و الگوریتم‌های کارا تر می‌تواند در جابجا کردن مرز این محدودیت موثر باشد ولی هیچگاه نمی‌تواند ما را قادر به شبیه سازی یک بازی فوتبال و تعیین نتیجه بکند.

1.3. شبیه سازی با استفاده از آلگوریتم‌های تصادفی

آلگوریتم‌های متفاوتی برای شبیه سازی یک پدیده فیزیکی وجود دارد. انتخاب آلگوریتم مناسب به نوع مسئله و پاسخهای مورد توجه بستگی دارد. معمولا در آلگوریتم‌های متفاوتی که ما استفاده می‌کنیم جایی برای یک مولد اعداد تصادفی وجود دارد. در ساده ترین شکل در بسیاری از شبیه سازی ها شرایط اولیه بصورت تصادفی انتخاب می‌شود. ولی دسته ای از آلگوریتم‌ها وجود دارند که تصادف اساس آلگوریتم است. این آلگوریتم‌ها به آلگوریتم‌های مونت کارلو معروف هستند. در این کتاب ما خود را به آشنایی با این الگوریتم‌ها، که در مقابل آلگوریتم‌های تعینی آنها را تصادفی می‌نامیم، محدود می‌کنیم.

شبیه سازی‌های مونت کارلو را می‌توان به دو گروه عمده تقسیم کرد. در گروه اول تصادف قسمتی از مدل و نظریه مورد استفاده می‌باشد همانطور که در بسیاری از مدلهای فیزیکی نیروها و یا پتانسیل‌های تصادفی قسمتی از مدل هستند. به طور مثال در مورد شبیه سازی فرضی یک بازی فوتبال نی‌متوان از دخالت دادن عواملی مانند وضعیت هوا و زمین، سرعت باد، سلامت بازیکنان و احتمال آسیب دیدگی آنها چشم پوشید. دسته دیگر آلگوریتم هایی هستند که برای حل یک سیستم کاملا تعینی با معادلات تعینی بکار می‌روند ولی در این کار از یک روش کاملا تصادفی استفاده می‌کنند.

1.4. چند توصیه عمومی

میتوانم حدس بزنم که خوانندگان این کتاب را می‌توان در دو گروه دسته بندی کرد. آنهایی که میخواهند فقط با شبیه سازی آشنا شوند و آنهایی که قصد دارند در آینده شبیه ساز شوند و اکنون در قدمهای اول هستند. در اینجا توصیه هایی برای گروه دوم دارم. رعایت این نکات مانند رعایت نکاتی در دست خط یا لهجه آدم است. اگر این نکات را از اول کار رعایت کنید بسیار موفق تر خواهید بود.

1. شما می توانید به زبان های برنامه نویسی متفاوتی برنامه بنویسید ولی به دلایلی زبانهای FORTAN و C برای شبیه سازی توصیه می‌شود. یکی از این دلایل سرعت بالای اجرای برنامه است که فکر می‌کنم همین یک دلیل کافی باشد. تقریبا هیچ نرم افزار جدی را پیدا نمی‌کنید که به یکی از این دو زبان نوشته نشده باشد. پس حتی اگر به یکی از این دو زبان مسلط هستید بد نیست که دیگری را نیز در حد آشنایی بدانید. امکان دارد در آینده لازم باشد تغییرات جزئی در یک برنامه نوشته شده به این زبان‌ها بدهید. همچنین از آنجایی که زبان Python به سرعت در تمام نرم افزارها فراگیر شده است بخصوص در مورد کسانی که برا ی اولین بار برای یادگیری زبان برنامه نویسی اقدام می‌کنند توصیه

می‌شود. این زبان کتابخانه‌های مجهز و زیادی به خصوص برای انجام کارهای علمی در اختیار کاربر قرار می‌دهد.

2. خیلی خوب است که برنامه نویسی شی گرا را بیاموزید. روشهای متفاوتی برای برنامه نویسی وجود دارد. ساده ترین شکل برنامه نویسی، برنامه نویسی ساختار یافته است. در این نوع برنامه نویسی برنامه خط به خط کمپایل می‌شود. این نوع برنامه نویسی اگرچه بسیار ساده است مناسب نوشتن برنامه های بزرگ نیست. زیرا هم از نظر سرعت در اجرا و هم از نظر روانی در نوشتن و دنبال کردن آلگوریتم بسیار نا کارآمد هستند. در سوی مقابل این گونه برنامه نویسی، برنامه نویسی شی گرا وجود دارد. در این نوع برنامه نویسی برنامه با معرفی اشیا و روابط حاکم بر آنها بسیار کارا تر خواهد بود. برنامه نویسی ساختار یافته و برنامه نویسی شی گرا دو سر طیف هستند. در این میان برنامه نویسی بسته‌ای وجود دارد. در این گونه برنامه نویسی، برنامه به بسته های کوچکتری تقسیم می‌شود. هر بسته بطور مستقل می تواند کامپایل شود. اگر شی گرا نمی‌نویسید، حداقل سعی کنید که بسته‌ای بنویسید. مطمئن باشید که زمانی که برای فراگیری آن می‌گذارید در مقایسه با صرفه جویی که در وقت دی‌باگ کردن می‌کنید قابل مقایسه نیست.

3. حتما عادت کنید که در برنامه خود بدون هیچ خسته‌گی و تا جایی که می‌توانید از نام گذاری معنی دار متغیرها و توابع استفاده کنید و از توضیح گذاشتن در کنار برنامه پرهیز نکنید. امکان دارد که لازم باشد که بعد از مدتی مانند چند ماه یا سال در برنامه‌ای که نوشته‌اید تغییر بدهید. مطمئن باشید که حتی یک خط از آنرا به یاد نخواهید آورد. تمام متغیرهایی که تعریف می‌کنید، توابعی که استفاده می‌کنید، شرط ها و قیودی که در برنامه می‌گذارید را شرح دهید. این کار برنامه را برای خودتان و دیگران خوانا خواهد کرد.

4. اگر تصمیم گرفتید که در یک پروژه شبیه سازی بزرگ مشارکت داشته باشید لازم است که روشهایی که امکان کار مشترک بر روی یک برنامه را می‌دهد بیاموزید. از این روشها می توان به CVS و git اشاره کرد. در این روش هر فرد تغییرات خود را بر روی نسخه ای از کد که در اختیار دارد می‌دهد و بعد از اطمینان از صحت تغییرات این نسخه را در مخزنی که قابل دسترس همه است قرار می‌دهد. به همراه این کار لیستی از تغییراتی که اعمال کرده نیز می‌گذارد تا دیگران کار خود را بر روی این نسخه ادامه دهند. نکته مهم این است که در فضای عمومی تمامی نسخه های قبلی با تاریخچه تغییرات حفظ می‌شود. زیرا امکان دارد که بعضی از تغییرات مخرب باشند. این روش برنامه نویسی حتی وقتی که به تنهایی کار می‌کنید نیز بسیار مفید است. امروزه فقط برنامه نویس ها از این زیر ساخت ها استفاده نمی کنند بلکه کسانی که هر داده ی متنی بر روی رایانه ایجاد می کنند (نویسندگان، دانشجویان، ...) و هر لحظه ممکن است نیاز به دسترسی به زحمات ماه ها قبلشان داشته باشند از یان زیر ساخت استفاده می کنند.

5. سعی کنید که برنامه شما با کاربر ارتباط خوبی برقرار کند و یا به اصطلاح کاربر دوست باشد. همیشه فرض کنید که کد شما امکان دارد به کار دیگران هم بیاید. تا می‌توانید از واسط گرافیکی در خروجی و ورودی برنامه هایتان استفاده کنید. به خصوص وجود یک واسط گرافیکی برای تصویر کردن سیستم برای درک سیستم و همچنین پیدا کردن خطاهای برنامه بسیار مفید است.

6. در انتهای هر شبیه سازی نتایج را ذخیره کنید. همیشه فرض کنید که به نتایج هر شبیه سازی در آینده نیاز خواهید داشت. مخصوصا اگر از نتایج شبیه سازی در یک مقاله نشر یافته استفاده کرده باشید. در این صورت ممکن است سال‌ها پس از نشر مقاله برای بررسی مجدد نتایج به شما مراجعه شود. مثلا هنگام انجام شبیه سازی دینامیک ملکولی مهم است که در مراحل مختلف شبیه سازی مکان و سرعت ذرات را ذخیره سازی کنید. نیازی به ذخیره ی تمام قدم‌های شبیه سازی نیست و از آنجایی که ممکن است فضای بسیار زیادی اشغال کند توسیه نمی شود. در بخش مربوطه بیشتر در این مورد بحث خواهد شد. همچنین

بهتر است که برای هر نتیجه یک فایل متنی گزارش گونه جهت ذخیره تمام پارامترهای مورد نیاز برای باز تولید نتایج شبیه سازی نیز تهیه شود.

1.5. پیش نیازهای این کتاب

هرچند نکاتی که در بالا اشاره شد به هر خواننده ای که تمایل به کار شبیه سازی به طور حرفه ای دارد توصیه می‌شود ولی برای همراهی با این کتاب دانش برنامه نویسی ابتدایی با یک زبان ساده و میانی مانندPython، Matlab، Mathematica و یا BASIC کافیست.

زبان پایتون به علت سادگی و امکانات گرافیکی و همچنین کتابخانه های بسیار جذاب در سال‌های اخیر بین قشر علمی محبوبیت زیادی پیدا کرده است. بنابرین در این کتاب برای تمرین‌ها نمونه حل‌های ساده به زبان Python ارائه شده است.

همچنین برای همراهی با بخش‌هایی از کتاب دانستن مکانیک آماری و ترمودینامیک در حد مقدماتی بسیار مفید خواهد بود. در قسمت هایی از کتاب برای خوانندگان علاقه‌مند اطلاعات بیشتر نظری داده شده است که در صورت عدم تمایل می‌توان از این بخش‌ها بدون برخورد با مشکلی در دنباله کتاب گذر کرد. این بخشها با ستاره مشخص شده اند.

2. فراکتال‌ها (Fractals)

همانگونه که در مقدمه اشاره شد از خوانندگان این کتاب انتظار می‌رود که حداقل به یک زبان برنامه نویسی کامپیوتر مسلط باشند. از آنجا که امکان دارد بعضی از ایشان برای مدتی از برنامه نویسی فاصله گرفته باشند و یا تمایل داشته باشند که همراه با این کتاب توانایی برنامه نویسی خود را افزایش دهند، این بخش را میتوان به عنوان تمرینی برای برنامه نویسی تلقی کرد.

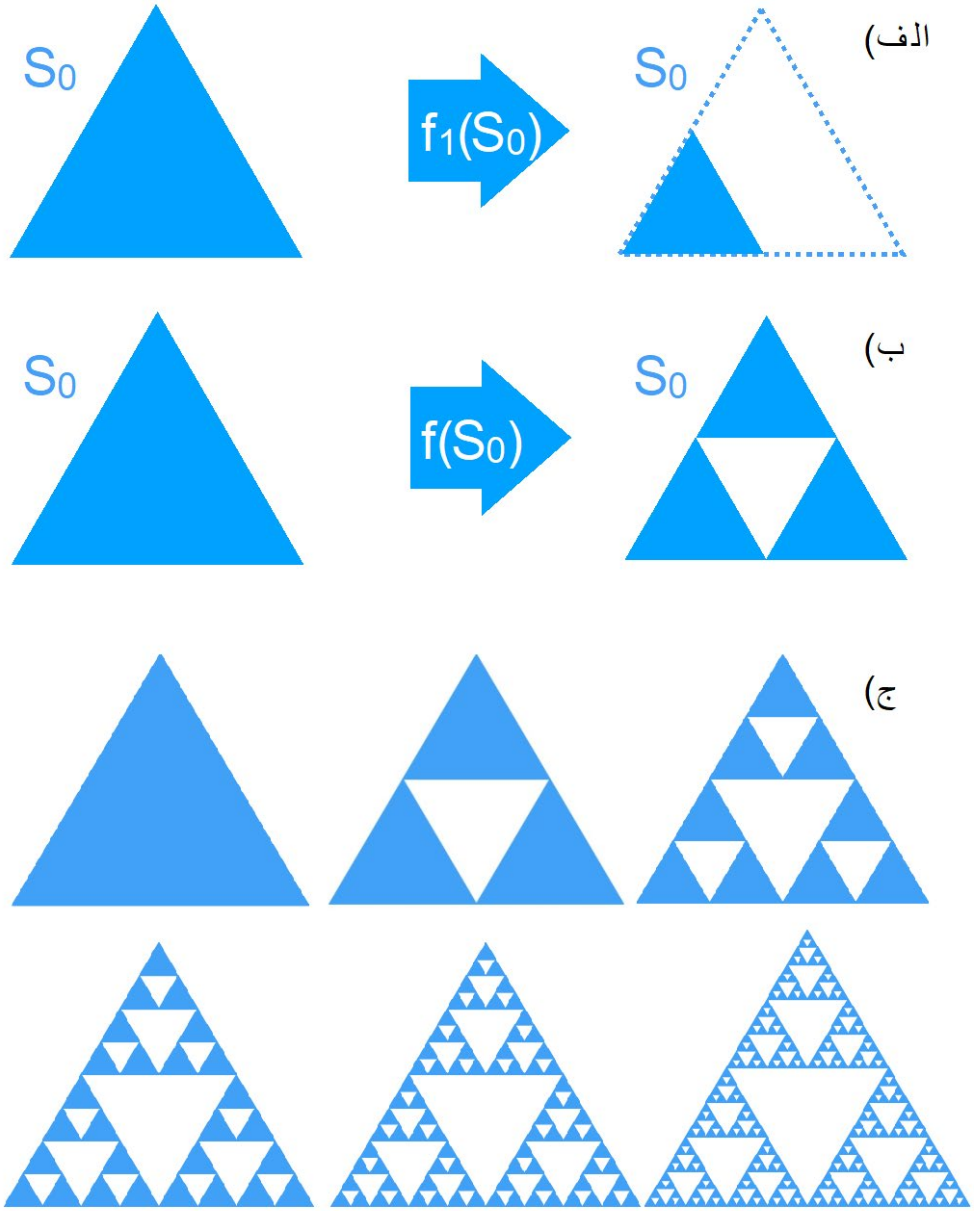
در این بخش به شبیه سازی ساختارهای فراکتالی می پردازیم. امید دارم که جذابیت نظری و تصویری این موجودات زیبا باعث شود که شروع شیرین‌تری داشته باشید. نکته جالب در این بخش این است که برای این شبیه‌سازی‌ها نیازی به نوشتن برنامه های طولانی ندارید. اکثر آلگوریتم های معرفی شده در این بخش بسیار ساده هستند و میتوانند به عنوان تمرینی برای افزایش مهارت‌های برنامه نویسی به کار آیند. به اضافه اینکه نمی‌توان از زیبایی این فراکتال‌ها بی آنکه آنها را به تماشا بنشینیم لذت بریم. (به نظرم تکرار داریم) پس برای شبیه سازی آنها باید بیاموزیم که چگونه آنها را نمایش دهیم. در حقیقت تنها خروجی تمرینات این بخش نمایش تصاویر فراکتالهای زیبا بر روی نمایشگر و یا چاپ آنهاست. در نتیجه این نیز تمرین خوبی برای استفاده از توانایی نمایش و بکارگیری واسط گرافیک است که در بخش‌های دیگر این کتاب به آن نیازمندیم.

از دیدگاه ریاضی فراکتال‌ها مجموعه‌های با توپولوژی غیر بدیهی هستند. این مجموعه‌ها می‌توانند ابعاد غیر صحیح داشته باشند. هر یک از این مجموعه‌ها در فضای بزرگتری قرار دارند و به عبارتی زیر مجموعه آن فضای بزرگتر هستند. این فضای بزرگتر را فضای غوطه‌وری فراکتال می‌نامیم. بُعد هر فراکتال کوچکتر یا مساوی بعد توپولوژی فضای غوطه وری آن است. برای تعریف بعد توپولوژی از پیچیدگی‌های ریاضی صرفنظر می‌کنیم و با تعریف سادهٔ مرز آنرا توصیف می‌کنیم. یک نقطه، یک مجموعه با بعد توپولوژی صفر است. حال هر مجموعه‌ای که برای محدود کردن حرکت بر روی آن بتوان از مرزهای صفر بُعدی (نقطه) استفاده کرد دارای بُعد یک است. حرکت بر روی یک پاره خط می‌تواند به کمک دو مرز نقطه‌ای محدود شود. در نتیجه پاره خط یک مجموعه یک بعدی است. به همین ترتیب یک زندان دو بعدی دیوارهای یک بعدی دارد و یک زندان سه بعدی دیوارهای دو بعدی.

گاهی برای توصیف بعد توپولوژی از مفهوم حجم d بعدی استفاده می‌شود. برای یک مجموعه بسته d بعدی، فقط حجم d بعدی خوش تعریف، محدود و غیر صفراست. یک پاره خط دارای طول (حجم یک بعدی) محدود و سطح (حجم دوبعدی) صفر است. یک صفحه متناهی طول نامحدود، سطح محدود و حجم صفر دارد. ولی در مورد فراکتالها با وجود اینکه مجموعه‌های متناهی هستند ولی امکان دارد حجم d بعدی (برای d های صحیح) متناهی نداشته باشند. به همین دلیل این مجموعه‌ها به موجوداتی با بعد غیر صحیح معروف هستند (که همیشه درست نیست).

2.1. فراکتال‌های خود شبیه (Self-Similar Fractals)

ساده ترین مثال‌هایی که درمورد فراکتال‌ها می‌توان زد مربوط به فراکتال‌های خود شبیه است. با مثال معروف مثلث سرپینسکی (Sierpinski) شروع می‌کنیم.



شکل 2-1. مراحل تولید مثلث سرپینسکی

مثلث متساوی الاضلاع S_0 و تابع تجانس f_1 با ضریب تجانس $r=\frac{1}{2}$ را در نظر بگیرید. اثر این تابع بر S_0 از آن مثلث کوچکتری با ابعاد $\frac{1}{2}$ مثلث اول می‌سازد (شکل ۲-الف). حال توابع f_2 و f_3 را نیز در نظر بگیرید که مشابه با f_1 مثلث را کوچک میکند ولی یک جابجایی هم در صفحه می‌دهد، به گونه‌ای که اثراجتماع این سه تابع، $f=f_1\cup f_2\cup f_3$، بر روی این مثلث، مجموعه $S_1=f(S_0)$ که زیر مجموعه S_0 می‌باشد را نتیجه می‌دهد (شکل ۲-ب). با تکرار اثر تابع f بر روی آن مجموعه های زیبای

$$S_1 = f(S_0)$$
$$S_2 = f(S_1) = f^2(S_0)$$
$$\vdots$$
$$S_n = f^n(S_0)$$

بدست می‌آید. تکرار این عمل در حد $n \to \infty$ به مجموعه خود شبیه $S = \lim_{n\to\infty} S_n$ منجر می‌شود که اثر تابع f بر روی آن خودش را نتیجه می‌دهد،

$$S = f(S)$$

مجموعه S یک فراکتال خود شبیه است که به مجموعه سرپینسکی معروف است (شکل ۲-ج).

اگر مساحت مثلث ابتدایی را A_0 بگیریم، هر یک از مثلثهای تولید شده به وسیله توابع تجانس مساحتی برابر با $\frac{A_0}{4}$ دارند. در نتیجه مساحت مجموعه S_1 برابر$\frac{3}{4}A_0$ است و به همین ترتیب برای S_n مساحت $A_n = \left(\frac{3}{4}\right)^n A_0$ بدست می‌آید. در نتیجه در حد $n \to \infty$ مساحت مجموعه سرپینسکی صفر است. از اینجا می‌توان نتیجه گرفت که بعد توپولوژی مجموعه سرپینسکی کمتر از 2 است.

با نگاهی به شکل ۱-۲ می‌توان دید که تمام نقاط روی مرز مثلث اولیه عضو مجموعه نهایی خواهند بود. به همین ترتیب تمام اضلاع مثلثهای میانی نیز عضو مجموعه نهایی خواهند ماند. پس مجموعه‌ی محیط تمام مثلث‌ها (اضلاع آنها) زیرمجموعه‌ای از مجموعه نهایی خواهند بود. اگر طول هر ضلع مثلث اولیه را l بگیریم، برای محیط مجموعه S_1 داریم: $p(S_1) = 3\left(\frac{l}{2}\right) \times 3$. و به همین ترتیب برای محیط S_n داریم، $p(S_n) = 3l \times \left(\frac{3}{2}\right)^n$. این عبارت برای $n \to \infty$ واگرا می‌شود. در نتیجه حجم یک بعدی زیر مجموعه‌ای از مجموعه سرپینسکی بینهایت است. این نشان می‌دهد که بعد توپولوژی این مجموعه بزرگتر از 1 است. پس مجموعه سرپینسکی مجموعه‌ای با بعد توپولوژی بزرگتر از 1 و کوچکتر از 2 است. پس بعد توپولوژی برای این گونه مجموعه ها خوش تعریف نیست و اگر بخواهیم بعدی به این مجموعه نسبت دهیم باید یک عدد غیر صحیح باشد.

2.2. بعد فراکتال

تعریف‌های متفاوتی برای بعد فراکتال‌ها ارائه شده‌است که در رابطه با فراکتال‌های خود تشابه همگی با هم همخوانی دارند و پاسخ یکسانی دارند. به طور کلی بُعد یک فراکتال همواره کوچکتر یا مساوی بعد توپولوژی فضای غوطه وری آن است. برای مثال بعد فراکتال سرپنسکی باید کوچکتر مساوی 2 باشد.

بعد خود تشابهی:

در مورد فراکتال‌های خود تشابه

$$S = f(S)$$

که در آن f اجتماع توابع $f_i\,(i = 1 \cdots n)$ با ضرایب تجانس (تراکم) $r_i \leq 1$ است، بعد خود تشابهی d_{ss} براحتی از رابطه زیر قابل محاسبه است.

$$\Sigma_{i=1}^{n} r_i^{d_{ss}} = 1\,.$$

مثال:
بعد خود تشابهی مثلث سرپینسکی را بدست آورید.
پاسخ:

$$3\left(\frac{1}{2}\right)^{d_{ss}} = 1 \;\rightarrow\; d_{ss} = \frac{\log 3}{\log 2}$$

همانگونه که انتظار داشتیم بعد این فراکتال عددی بزرگتر از 1 و کوچکتر از 2 است.

بعد جرمی:

این‌تعریف که به مفهوم بعد در فیزیک نزدیکی بیشتری دارد برای هر نوع فراکتالی قابل تعریف است. به زبان ساده رفتار مقیاسی رشد جرم (حجم) با ابعاد سیستم را بررسی می‌کند. برای درک بیشتر با مثال‌هایی از اجسام متعارف شروع می‌کنیم. برای یک جسم سه بعدی ساده در صورت تغییر ابعاد با یک ضریب 2 حجم جسم و در نتیجه جرم آن (جسم را همگن فرض کنید) $2^3 = 8$ برابر می‌شود. در نتیجه می‌توان گفت که برای اجسام 3 بعدی

$$M(r) \sim r^3.$$

و به طور کلی برای اجسام d_m بعدی

$$M(r) \sim r^{d_m}.$$

حال با همین رویه رفتار مقیاسی جرم در مثلث سرپینسکی را بدست می‌آوریم. فرض می‌کنیم که جرم کوچکترین مثلثی که در این مجموعه با قدرت تفکیک ما (r_0) قابل تشخیص است برابر با واحد جرم (m_0) باشد. به این ترتیب با دو برابر کردن ابعاد، ما 3 مثلث داریم پس جرم سه برابر می‌شود. با ادامه این روند دیده می‌شود که با 4 برابر کردن ابعاد، جرم 9 برابر می‌شود و در حالت کلی

$$\frac{r}{r_0} = 2^n \Rightarrow \frac{m}{m_0} = 3^n$$

با حذف n از روابط بالا

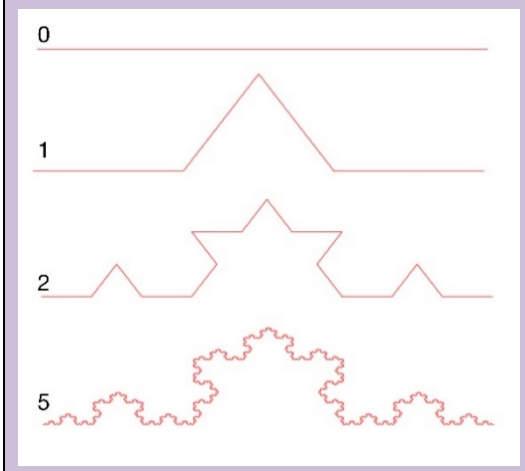
$$\frac{m}{m_0} = (r/r_0)^{\frac{\log 3}{\log 2}}$$

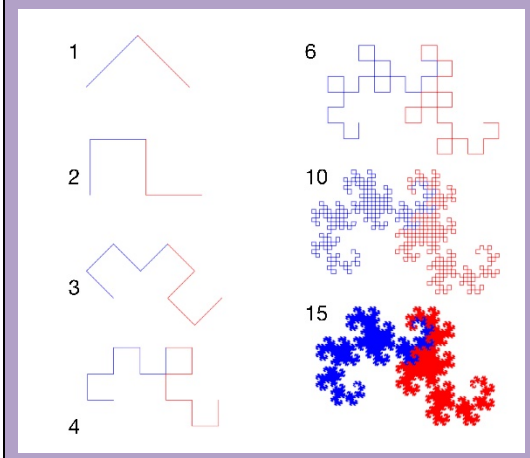
در نتیجه بعد جرمی این فراکتال $d_m = \frac{\log 3}{\log 2}$ است که همانطور که قبلا اشاره شد با بعد خود تشابهی برابر است. در ادامه این بخش منظور از بعد فراکتالی که آنرا با d_f نشان می‌دهیم، همان بعد جرمی d_m است مگر اینکه صریحا غیر از این اشاره شده باشد.

2.3. شبیه سازی فراکتالهای خود تشابه

برای شبیه سازی فراکتال‌های خود تشابه روشهای متعددی وجود دارد. ساده ترین آلگوریتمی که در ابتدا به نظر می‌رسد استفاده از توابع خود تشابهی است. به طور مثال کافی‌است که شما بدانید چگونه یک خط را نمایش دهید. برای این کار باید مختصات دو نقطه انتهای خط را بدانید. حال با اعمال توابع خود تشابه بر این پاره خط، پاره خطهای جدیدی تولید می‌شود که قابل رسم به همان طریق هستند. ادامه این کار در یک چرخه و اعمال توابع خود تشابه به مجموع پاره خطهای جدید، فراکتال را تولید می‌کند. البته بدیهی‌است که ما نمی‌توانیم این کار را بینهایت بار انجام دهیم. ولی در حقیقت نیازی به این کار نیز وجود ندارد. اگر منظور از این شبیه سازی تولید تصویری از فراکتال باشد، به دلیل محدودیت قدرت تفکیک نمایشگر بعد از تعدادی تکرار دیگر نمی‌توان جزییات

بیشتری بر تصویر نمایان کرد. این یک نمونه قابل نمایش از محدودیت‌های عددی و محاسباتی‌است که در آینده بیشتر در باره‌ی آن صحبت خواهیم کرد.

.2.1 تمرین	مجموعه کوخ: ساختار فراکتالی زیر را پله به پله تولید کرده و تصویر آنرا بر روی نمایشگر نشان دهید. توجه کنید که برای ساختن این فراکتال به چهار تابع نیاز دارید که هر مجموعه $\frac{1}{3}$چهار تابع با نسبت کنند ولی اولیه را کوچک می در دو تابع بغیر از تجانس و انتقال، دوران نیز سهیم است. بهتر است که برای تبدیلات نقاط مورد نظر از ماتریس‌های انتقال استفاده کنید.	

.2.2 تمرین	اژدهای هی وی: ساختار فراکتالی زیر را پله به پله تولید کرده و تصویر آنرا بر روی نمایشگر نشان دهید. توجه کنید که برای ساختن این فراکتال فقط به دو تابع نیاز دارید. یکی از نکات جالب این فراکتال این است که هرگز خودش را قطع نمی کند یا از روی یک یال دوبار رد نمی شود. برای دیدن این خاصیت می توانید از طیف رنگی برای نمایش جهت حرکت استفاده کنید. این کار خروجی را بسیار جذاب می کند.	

.2.3 تمرین	مثلث سرپینسکی: مثلث سرپینسکی را به استفاده از توابع خود تشابه تولید کنید. توجه کنید که در اینجا باید اول فرا بگیرید چگونه یک مثلث را با استفاده از مختصات رئوس آن تولید کنید.

برای تولید فراکتال‌های خود تشابه می‌توان الگرویتم‌های متفاوتی به کار برد. بعضی از این الگوریتم‌ها کاملا مبتکرانه و معمولا قابل استفاده برای

تولید فراکتال خاصی هستند. تمرین زیر یکی از این الگوریتم‌های مبتکرانه را معرفی می‌کند.

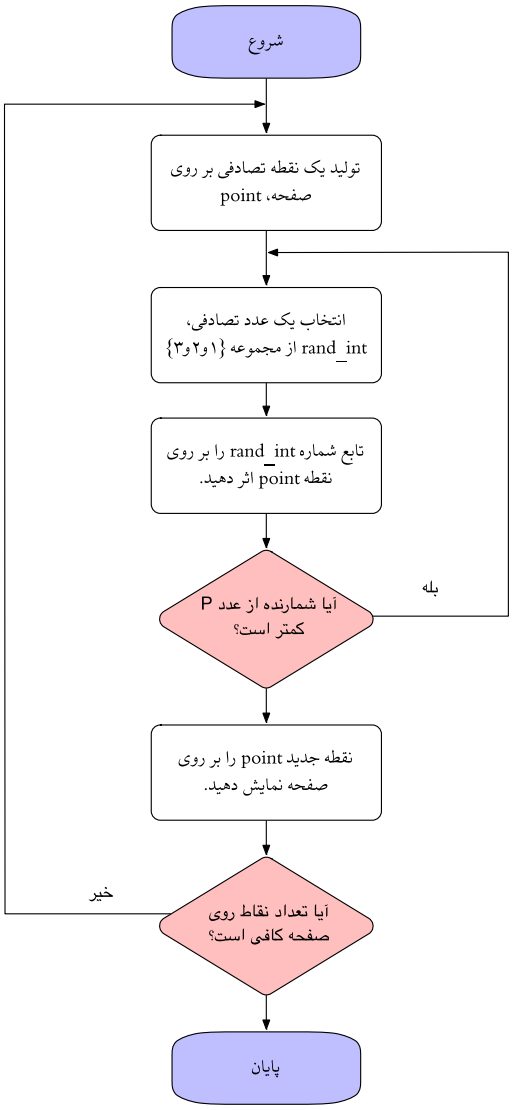
.2.4 تمرین	مثلث خیام: مطمئنا با مثلث خیام که در بدست آوردن ضرایب بست دو جمله‌ای به کار می رود آشنا هستید. هر درایه این مثلث مجموع دو عدد بالای آن است. برنامه ای برای تولید این مثلث بنویسید. در نمایش مثلث بر روی نمایشگر به جای نشان دادن اعداد به هر عدد یک پیکسل اختصاص دهید. تمام عددهای فرد را با رنگ سبز و اعداد زوج را با رنگ قرمز نشان دهید. آیا نتیجه آشنا نیست؟	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1

آلگوریتم‌های بالا برای تولید فراکتال‌ها، ساختاری کاملا تعینی دارند. به این معنی که در این آلگوریتم‌ها نیازی به اعداد تصادفی نیست. البته این جای تعجب ندارد زیرا ساختارهایی که مورد توجه بودند نیز کاملا غیر تصادفی هستند. ولی هدف این کتاب این است که نشان دهد برای حل مسائل تعینی نیز می‌توان از آلگوریتم‌های تصادفی استفاده کرد. منظور از آلگوریتم تصادفی، آلگوریتمی است که یک مولد اعداد تصادفی (random generator) نقش اساسی در آلگوریتم داشته باشد. اکنون نشان می‌دهیم چنین آلگوریتمی چگونه می‌تواند در تولید فراکتال‌های خود تشابه به کار آید.

همانطور که در توصیف مثلث سرپینسکی بیان شد، این فراکتال با اعمال متمادی 3 تابع خود تشابه تولید می‌شود. البته همانطور که گفته شد به دلیل محدودیت قدرت تفکیک نمایشگر تعداد دفعاتی که باید این توابع بر مثلث اولیه اثر کند تا شکل قابل قبولی بدست بیاید محدود است. فرض کنید بعد از P بار تاثیر توابع، این شکل(؟) بدست آمده است پس هر نقطه از آن در اثر تاثیر متوالی رشته‌ای از این توابع با طول P تولید شده است. حال می‌توان از این نکته برای معرفی یک آلگوریتم تصادفی ساده استفاده کرد. در زیر این آلگوریتم برای مثلث سرپینسکی با 3 تابع خود تشابه معرفی می‌شود. ولی می‌توان آن را برای هر فراکتال دیگری نیز به کار برد.

1. یک نقطه دلخواه را با استفاده از مولد اعداد تصادفی در صفحه نمایش انتخاب کنید.
2. یک عدد به طور تصادفی از مجموعه‌ی $\{1,2,3\}$ انتخاب کنید و تابع متناظر با آن را بر روی نقطه اثر دهید.
3. قدم 2 را P بار تکرار کنید و بعد نقطه نهایی را بر روی نمایشگر نشان دهید.
4. به قدم اول برگردید و تا زمانی که تصویر مطلوبی بر روی خروجی بدست آید این آلگرویتم را تکرار کنید.

این الگوریتم مستقل از سادگی (فقط دو چرخه‌ی ساده‌ی تو در تو) قابلیت به کار گیری برای تولید هر فراکتال دیگری را نیز دارد.

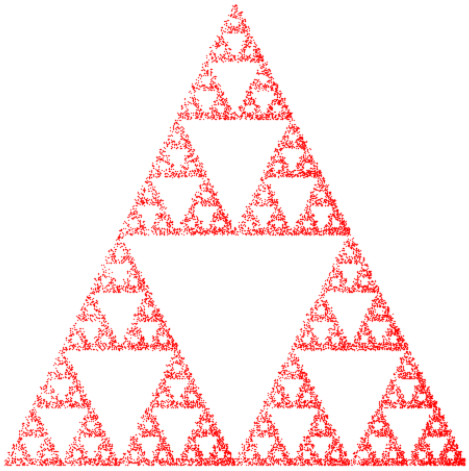


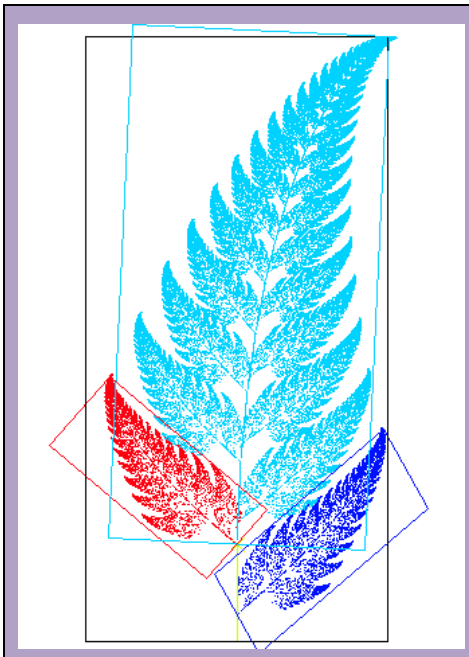
نمودار شناور 1 رسم مثلث سرپینسکی با استفاده از آلگوریتم غیر تعیینی.

برای اثر تابع بر روی نقاط و انتقال آنها می‌توان به سادگی از رابطه

$$\overrightarrow{x'} = r.R.\vec{x} + \vec{a}$$

استفاده کرد، که در اینجا r ضریب تشابه R ماتریس انتقال و $\vec{a}$ بردار انتقال است. با تغییر این پارامترها می‌توان اشکال متفاوتی تولید کرد.

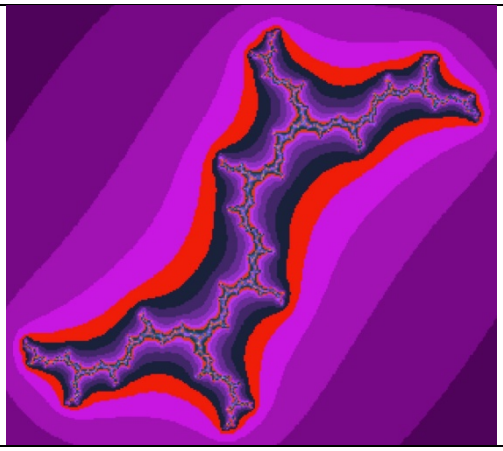
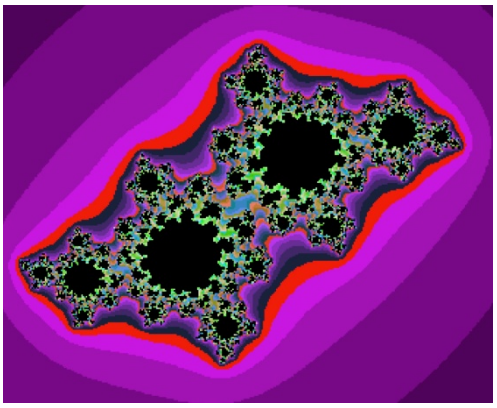
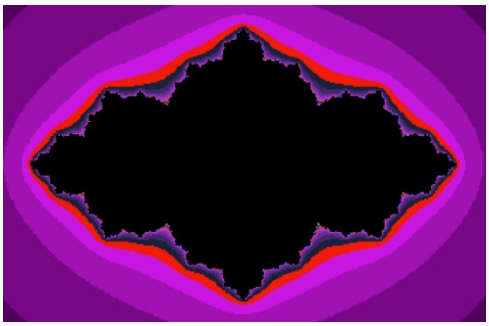
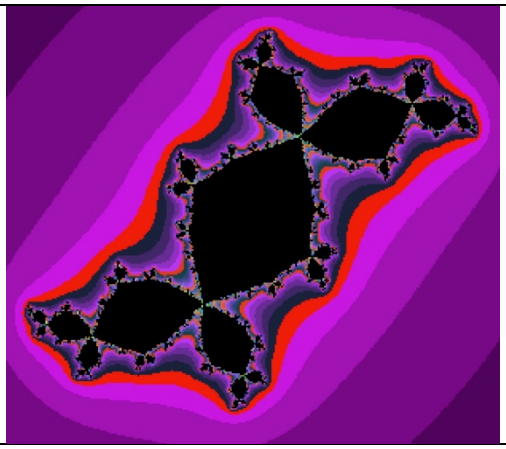
.2.5 تمرین	بازهم مثلث سرپینسکی: مثلث سرپینسکی را به استفاده از توابع خود تشابه و آلگوریتم تصادفی معرفی شده در اینجا تولید کنید.	

.2.6 تمرین	برگ سرخس: یکی از فراکتالهای بسیار زیبا، فراکتالی است که به دلیل شباهتش به برگ گیاه سرخس به این نام معروف است. اگر به این فراکتال نگاه کنید میتوانید توابع خود تشابه آن را بیابید. با استفاده از یک خطکش و نقاله میتوانید نسبت های تجانس و زوایای انتقال این توابع را بیابید. این امکان وجود دارد که ضرایب تشابه توابع انتقال در تمام راستاها یکسان نباشد و به عبارتی به جای آنکه تابع خود تشابه باشد خود آفین (self-affine) باشد. با استفاده ازتوابعی که بدست آورده اید و آلگوریتم تصادفی معرفی شده در متن، این فراکتال را بسازید. با توابعی که بدست آورده اید بازی کنید و با تغییر پارامترهای آن ببینید چگونه میتوانید شکلهای متفاوتی از فراکتال سرخس را تولید کنید.	

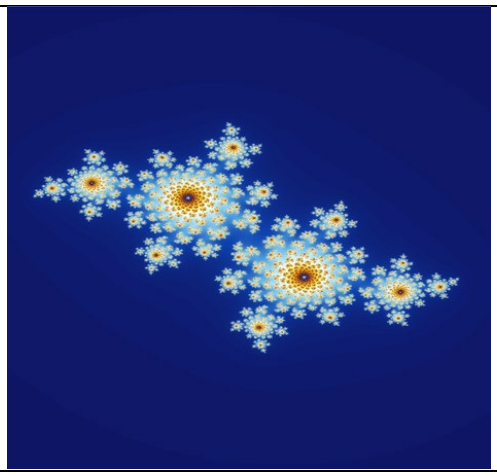
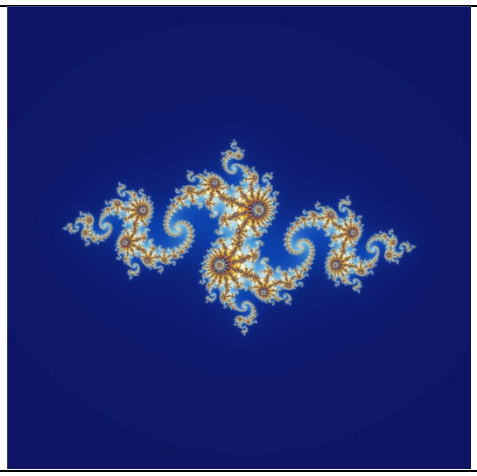
.2.4 مجموعه ی ژولیا (Julia Set)

نقطه ی $z_0 = x + i\,y$ را در فضای مختلط در نظر بگیرید. با اثر تابع $F(z)$ بر روی آن، نقطه‌ی جدید $z_1 = F(z_0)$ بدست می‌آید. اگر به همین ترتیب تابع را n بار اثر دهیم، فاصله $z_{n+1} = F(z_n)$ از مبدا تغییر خواهد کرد. مجموعه نقاطی که فاصله‌ی آنها از مبدا با تکرار اعمال این تابع بر روی آنها به بینهایت میل نمی‌کنند را مجموع ژولیا می‌نامند. به عبارت دقیقتر مجموعه‌ی ژولیا مرز نقاطی‌است که تحت تاثیر تابع F بر روی آنها، به بینهایت می‌روند و آنهایی که نمی‌روند. به طور مثال برای تابع $F(z) = z^2$

به وضوح این مرز یک دایره به شعاع واحد است. ولی درصورتی که این تابع به صورت $F(z) = z^2 + c$ تعریف شود برای مقادیر غیر صفر c این مرز یک فراکتال است[3] که به مجموعه ی ژولیا معروف است. در نمایش این شکلهای زیبا می‌توان بعد از اثر تابع F به تعداد متناهی بر روی هر نقطه از فضا رنگی به این نقطه، بر حسب فاصله ی نقطه ی نهایی از مبدا مختصات، تخصیص داد. در این صورت طیفی رنگی از دینامیک فرار نقاط از مرکز بدست می‌آید. جدول 1 تعدادی از این فراکتالها را با مقادیر c متناظر آنها نشان می‌دهد.

	
$c = -i$	$c = -0.4 - 0.6\,i$
	
$c = -0.6$	$c = -0.12 - 0.75\,i$

[3] برای مقدار $c = -2$ نیز شکل بدست آمده فراکتال نیست.

	
$c = -0.4 + 0.6\,i$	$c = -0.8 + 0.16\,i$

جدول 2-1 - مجموعه های متفاوت ژولیا به ازای مقادیر متفاوت پارامتر c

.2.7	مجموعه های ژولیا:
تمرین	شکلهای جدول 1 را تولید کنید. با تغییر پارامتر c سعی کنید شکلهای زیبای دیگری خلق کنید.

بیشتر بدانیم:

کتاب‌های زیادی در مورد فراکتال‌ها وجود دارد. برای دانشجویان علاقه‌مند به پایه‌های ریاضی این مبحث کتاب Measure, Topology and Fractals نوشته Gerald Edgar توصیه می‌شود. این کتاب با ریاضیات دقیق به شکلی ساده، قابل درک، و خود آموز تنظیم شده است.

3. لایه نشانی (Ballistic Deposition)

در بخش قبل با فراکتال‌های قاعده مند و خود تشابه آشنا شدیم. هر چند این فراکتال‌ها بسیار زیبا هستند و از دید نظری هم مطالعه‌شان بسیار ساده است ولی در طبیعت بسیار نادراند. در عوض خواص خود تشابهی در بسیار دیگری از پدیده‌های طبیعت به طور آماری مشهود است. به طور مثال وقتی که شما به ساحل دریا نگاه می‌کنید مستقل از اینکه با چه جزییاتی به ساحل نگاه کنید، آنرا خطی ناهموار با خمیدگی‌های تصادفی می‌بینید. اگر به تصویر ماهواره‌ای این ساحل نگاه کنید، خمیدگی‌ها در مقیاس چند کیلومتر هستند ولی وقتی در کنار ساحل ایستاده اید مقیاس متر دارند. هر چند این ناهمواری‌ها دقیقا یکسان نیستند ولی در هر دو مقیاس شکلهای مشابهی دارند. این شباهت به حدی است که اگر در حاشیه تصویر به مقیاس شکل اشاره نشود امکان تخمین این مقیاس برای بیننده وجود ندارد. البته شدت ناهمواری در بسیاری ازپدیده‌های طبیعی می‌تواند به مقیاس مشاهده نیز بستگی داشته باشد. مثلا کره زمین از روی ماه سیاره ای هموار و تقریبا صاف دیده می‌شود و با تصویری که شما در موقع عبور از دامنه های البرز می‌بینید بسیار متفاوت است. ناهمواری سطوح یکی از مباحث بسیار جذاب و مهم در فیزیک است. به طور مثال در فیزیکِ ماده چگال یکی از مهمترین مسایل، چگونگی رشد مواد و دینامیک سطح لایه نشانی شده می‌باشد. مدل‌های متفاوتی برای توصیف فرآیندهای رشد آرایه می‌شود. این پدیده در علم و فناوری به دلیل اهمیتی که دارد بسیار مورد توجه است، و مطالعات نظری و شبیه سازی بسیاری در این باره وجود دارد.

توجه ما به این موضوع در بخش‌های ابتدایی این کتاب به دلیل جذابیت های فرآیندهای رشد و نیز سادگی الگوریتم ها و مدل‌های معتبر برای مطالعه ی این فرآیندهاست. با توجه به اینکه مقصود این کتاب معرفی آلگوریتم های تصادفی است، شاید شبیه سازی فرآیندهای تصادفی مثال خوبی برای ورود به موضوع شبیه سازی سیستم‌های فیزیکی باشد. در حقیقت رفتار تصادفی این سیستم‌ها ناشی از افت و خیزهای حرارتی در طی فرآیند رشد است. بدیهی است که فیزیک رشد وابستگی زیادی به دما و شدت افت وخیزها دارد ولی ما از این جزییات فعلا چشم پوشی می‌کنیم.
یکی از فرایندهای بسیار پرکاربرد رشد، "لایه نشانی" است. فیزیک لایه‌های نازک کاربرد بسیار زیادی در شاخه‌های مختلف فیزیک و فناوری از قبیل اپتیک و الکترونیک دارد. برای تولید لایه‌های نازک از روش‌های متفاوت لایه نشانی استفاده می‌شود. در ساده‌ترین این روش‌ها بخاری از یک ماده در مجاورت زیر لایه قرار داده می‌شود تا ذرات بخار فرصت نشست بر روی زیر لایه را بیابند. به این ترتیب با گذشت زمان لایه‌ای از اتم‌های (مولکول‌های) بخار بر روی زیر لایه می‌نشینند. عوامل زیادی بر ساختار، شکل لایه و دینامیک رشد اثر می‌گذارند. جنس و ساختار کریستالی زیر لایه، فشار بخار، دما، و برهمکنش بین ملکول‌های بخار و زیر لایه از جمله عوامل موثر هستند. به دلیل وجود عوامل مختلف و مهمتر از همه افت و خیزهای حرارتی در حرکت ذرات در محفظه ی لایه نشانی، فرآیند نشست ذرات به شدت تصادفی است و سطح لایه دارای ناهمواری است. ساختار این ناهمواری و دینامیک آن نیز بستگی به مواد و روش رشد دارد. در طی فرآیند رشد نه

تنها ضخامت لایه افزایش می‌یابد بلکه ناهمواری سطح آن نیز رشد می‌کند. اگر سطح زیر لایه را هموار فرض کنیم و این سطح ایده‌آل را در مبدا مختصات قرار دهیم، در ابتدا ارتفاع لایه در تمام نقاط صفر است. با شروع نشست، ارتفاع در نقاط مختلف شروع به تحول می‌کند و با زمان تغییر می‌کند. ارتفاع لایه در نقطه $\vec{r}$ و در زمان t را با $h(\vec{r},t)$ نشان می‌دهیم و مقدار متوسط مکانی آن یعنی ضخامت متوسط لایه در زمان t را $\bar{h}(t)$ نشان می‌دهد. مقدار انحراف از میعار ارتفاع نیز نشان دهنده زبری (ناهمواری) سطح است.

در ادامه سعی می‌کنیم با تمرکز بر مسئله‌ی لایه نشانی راهکاری کلی در شبیه سازی مسایل فیزیکی را طی کنیم. در اولین قدم هدف یافتین مدلی است که حد اکثر انطباق را با مسئله مورد نظر داشته باشد. قدمهایی که برای مدل سازی این مسئله مطرح می‌شود بسیار عمومی است و در بسیاری از مسائل دیگر در فیزیک نیز قابل استفاده بوده و کاربرد دارند.

- گسسته سازی فضایی:
  در این مسئله‌ی خاص به دلیل اهمیت ساختار میکروسکوپی زیر لایه و حتی خود لایه، گسسته سازی مختصات شاید کاملا معقول و در جهت صورت مسئله باشد. به طور مثال وجود ساختار کریستالی زیر لایه و برهمکنش‌های مولکولی امکان نشست مولکول‌های لایه در هر نقطه را از آن می‌گیرد و جایگاه هایی را برای نشست تعیین می‌کند. ولی حتی اگر این گونه نبود ویا اگر ما نیاز به شبیه سازی در مقیاسی داشته باشیم که توان تفکیک ما از اندازه‌ی مولکولی بسیار ضعیفتر باشد و محیط برایمان پیوسته باشد، باز هم در بسیاری از شبیه سازی‌ها ترجیح می‌دهیم برای سادگی مدل سازی، مسئله را در فضای گسسته در نظر بگیریم. به این گونه ما مقیاس مکانی سیستم را تعیین می‌کنیم.
- گسسته سازی زمانی:
  فرآیند گسسته سازی فقط به مکان محدود نمی‌شود. هر چند در حقیقت هیچ قیدی میان فاصله زمانی نشستن ذرات بر روی زیر لایه وجود ندارد و توزیع زمانی این وقایع یک کمیت پیوسته است، باز هم برای راحتی کار می‌توان فرض کرد که برای رشد یکنواخت و با نرخ ثابت، این ذرات با فاصله های زمانی مساوی بر روی سطح می‌نشینند. با این فرض مقیاس زمانی شبیه سازی به کمک نرخ نشست داده می‌شود.
- تقلیل بعد فضایی مسئله:
  هرچند ما همیشه تمایل داریم که مدلی که می‌سازیم تا حد امکان شبیه به مسئله واقعی باشد ولی بعضی مواقع به دلایل فنی مجبور به تقلیل بعد فضایی مسئله می‌شویم. مطمئنا این امر در نتایج تاثیر می‌گذارد ولی گاهی می‌تواند در درک فیزیک مسئله بسیار کمک کند. در مورد مسئله مورد نظر ما که رشد یک رویه دو بعدی است، تصمیم می‌گیریم که مسئله را به رشد یک رویه یک بعدی تقلیل دهیم. دلیل واقعی این امر نیز از هیچ پشتوانه‌ی فیزیکی برخوردار نیست. این کار را تنها به دلیل این که نمایش رویه یک بعدی بر روی نمایشگر ساده‌تر است و مشاهده این پدیده ارزش آموزشی بالایی دارد، انجام می‌دهیم. البته می‌توان تعداد زیادی فرآیند رشد یک بعدی را مثال زد تا توجیه کنیم که انتخاب ما کار خیلی بدی هم نیست و نتیجه شبیه‌سازی ما به فیزیک مسئله

بسیار نزدیک است. مثال‌هایی از فرایندهایی وجود دارد که با اینکه در ظاهر تفاوت زیادی با پدیده‌ی رشد دارند ولی رویه تولید شده توسط آنها در کلاس فرآیندهای رشد می‌نشینند. برای مثال فرآیند تر شدن کاغذی عمودی که در ظرف آبی قرار داده شده است و سطح جدایی کاغذ تر شده و خشک با زمان به سمت بالا رشد می‌کند.

- محدود کردن مسئله:

به دلیل محدودیت های محاسباتی امکان شبیه سازی‌های بسیار بزرگ وجود ندارد. پس لازم داریم که ابعاد سیستم را محدود کنیم. در کارهای پژوهشی باید نشان دهیم که این محدودیت تاثیری بر نتایج ندارند و یا درک درستی از میزان تاثیر آن و یا روش‌های اصلاح نتایج داشته باشیم.

3.1.نمای رشد دینامیکی

با در نظر گرفتن نکات بالا تصویری از مدل شبیه سازی داریم. یک شبکه یک بعدی به طول L که در هر قدم زمانی، ذره‌ای بر آن سقوط می‌کند و بر یکی از نقاط این شبکه، یا ذراتی که قبلا بر روی آن نشسته، قرار می‌گیرد و ارتفاع آن نقطه را یک واحد بالا می‌برد. پس ارتفاع متوسط لایه را می‌توان اینگونه محاسبه کرد:

$$\bar{h}(t) = \frac{1}{L}\sum_{i=1}^{L} h(i,t) = \frac{t}{L}$$

که $h(i,t)$ ارتفاع در نقطه i در زمان t است. با توجه به واحد زمان انتخابی مسئله حاصل جمع که برابر با تعداد تمام ذرات نشسته تا این زمان است با زمان‌برابر است. در این بخش علامت بار بر روی متغیرها به معنی متوسط مکانی است.

ناهمواری سطح نیز با اندازه گیری افت وخیز ارتفاع بدست می‌آید.

$$w(t) = \sqrt{\overline{h^2}(t) - \bar{h}^2(t)}$$

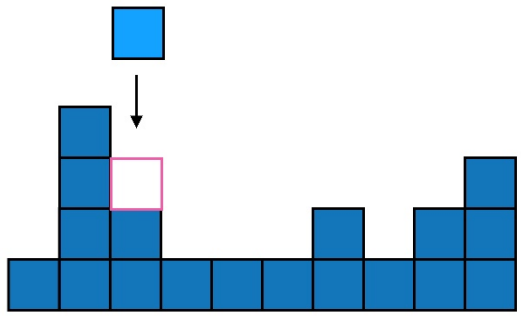
هرچند آهنگ تغییر ارتفاعِ متوسط در یک فرآیند نشستِ یکنواخت همواره ثابت است ولی در مورد ناهمواری این نکته درست نیست. بطور تجربی مشاهده می‌شود که ناهمواری با توانی از زمان رشد می‌کند،

$$w(t) \sim t^{\beta},$$

که β در این رابطه نمای رشد دینامیکی نامیده می‌شود. نکته جالب اینجاست که هرچند مکانیزم‌های متفاوتی برای رشد وجود دارد و فیزیک حاکم بر آنها نیز بسیار متفاوت است، ولی در حد ترمودینامیکی (ابعاد بسیار بزرگ) گزینه‌های بسیار محدودی برای نمای رشد دینامیکی وجود دارد. این نما یکی از نماهایی است که برای دسته بندی فرآیندهای رشد در کلاس‌های جهانشمولی استفاده می‌شود. در ادامه این بخش و همچنین در بخش‌های دیگر با نماهای دیگری نیز آشنا خواهیم شد و به موضوع جهانشمولی باز می‌گردیم.

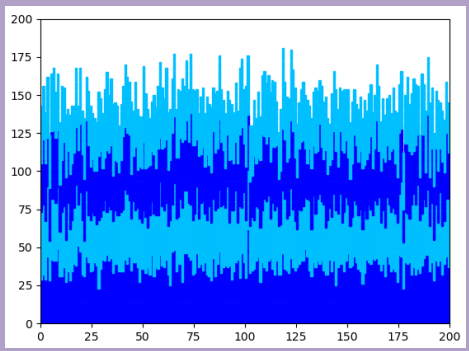
3.2. ول نشست (Random Ballistic Deposition)

ساده ترین مدلی که برای نشست می‌توان فرض کرد آن است که ذرات کاملا به طور کتره ای از ارتفاعی بالای زیر لایه بر روی آن سقوط کنند و در همان نقطه به لایه بچسبند و سطح زیر لایه را در آن نقطه یک واحد بالا بیاورند. این نشست را ول نشست می‌نامیم.



شکل 1 نمایی فرضی از فرآیند ول نشست که نحوه‌ی رشد را نشان می‌دهد.

.3.1 تمرین	- مدل ول نشست را در یک بعد شبیه سازی کنید. برای اینکار خطی افقی به طول 200 واحد به عنوان زیر لایه در نظر بگیرید و ذرات را به طور کاتوره ای بر روی آن بنشانید. - دینامیک مدل را بر روی نمایشگر نشان دهید. برای درک بهتری از رفتار زمانی رشد بهتر است که رنگ ذرات فرودی را با زمان (هر 10×L) به طور تناوبی تغییر دهید. - مقدار متوسط ارتفاع و ناهمواری را در بازه های زمانی متوالی محاسبه کنید. - منحنی تغییرات ناهمواری بر حسب زمان را رسم کنید. - β را برای ول نشست محاسبه و گزارش کنید. - آیا تصوری از دقت عددی که در بالا گزارش کرده اید دارید؟



در اینجا قبل از مقایسه‌ی مدل ساده‌ی ول‌نشست با فرایندهای واقعی رشد و هرگونه تلاشی برای تکمیل یا تعمیم آن به بحث در مورد آلگوریتم مناسب برای حل مسئله بالا می‌پردازیم.

شاید در ابتدا بنا به نحوه‌ای که مسئله‌ی بالا فرآیند ول نشست را توصیف می‌کند تصویری از یک انیمیشن در ذهن خواننده ایجاد شود. از طرف دیگر تصویری که برای ساختار لایه در یک زمان نمایش داده شده نیز تصویری دو بعدی از آن ارایه می‌دهد. به این معنی که هر ذره در این فضا با دو مختصات x (افقی) و z (عمودی) داده می‌شود. مقدار x مقداری تصادفی کوچکتر از طول سیستم را می‌گیرد و دیگر در طی فرآیند تغییر نمی‌کند. در صورتیکه

مقدار z باید مقداری به اندازه کافی بزرگ داشته باشد و طی آلگوریتم همراه با سقوط ذره این عدد شروع به کاهش می‌کند تا به مقداری که در حافظه‌ای برای ارتفاع لایه در این x برسد و در ارتفاعی بالاتر متوقف می‌شود.
اگر آلگوریتمی که در ذهن شما است کوچکترین شباهتی به سناریوی بالا دارد باید گفت که شما برای تولید یک انیمیشن توانایی خوبی دارید ولی تا شبیه ساز شدن مدل‌های فیزیکی فاصله زیادی دارید. واقعیت این است که شبیه سازی این مدل بسیار سادهتر از این حرف‌هاست. کافی است که خط زیر را در یک حلقه قرار دهیم تا همه چیز بخوبی پیش رود.

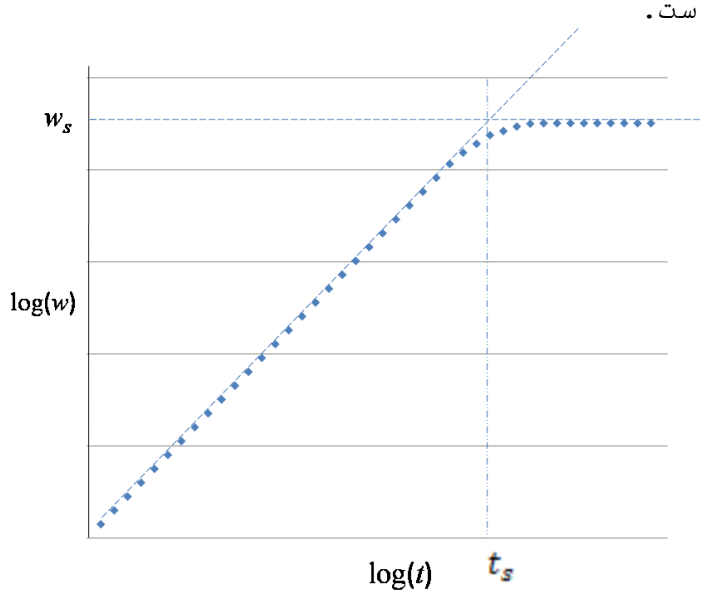
- h[randint(1,200)]+=1;

در اینجا و در ادامه‌ی این کتاب هرگاه نیاز به نوشتن برنامه یا قسمتی از آن باشد از زبان Python3 استفاده می‌کنیم. برای خوانندگان آشنا به برنامه نویسی نیازی نیست که توضیح داده شود که آرایه h باید قبلا با مقدار اولیه‌ی صفر معرفی شده باشد و شمارنده‌ی این حلقه نقش زمان را بازی می‌کند. به وسیله‌ی همین برنامه‌ی یک-خطی ولنشست شبیه سازی می‌شود و در هر زمان مقدار h در هرنقطه نشان دهنده‌ی ارتفاع لایه در آن نقطه است. این مقدار در هر زمان نیز می‌تواند برای نمایش، نقطه‌ای بر نمایشگر را مشخص کند. برای انجام پیشنهاد مسئله در تغییر رنگ نیز کارهای متفاوتی می‌شود کرد. شاید ساده ترین کار تقسیم حلقه بالا به دو حلقه تو در تو است که شمارنده حلقه خارجی می‌تواند به عنوان کد رنگ استفاده شود.
نکته مهم این مسئله که آنرا با مسایلی که تا کنون در این کتاب دیده اید متفاوت می‌کند این است که این مسئله به یک مشاهده ختم نمی‌شود و از شما می‌خواهد که کمیت‌های عددی خاصی را محاسبه و گزارش کنید. به طور خاص از شما خواسته است که مقادیر ارتفاع متوسط و ناهمواری را در بازه‌های زمانی خاصی بدست آورید و گزارش کنید. در آینده خواهید دید که کار اصلی شبیه سازان گزارش این گونه عددهاست و نمایش‌های زیبا فرع آنرا تشکیل می‌دهند. برای گزارش این عددها می‌توان از دو حلقه‌ی تو در تویی که برای نمایش ساختیم استفاده کنیم. کافی است در حلقه خارجی مقادیر خواسته شده را بدست آوریم و آنها را گزارش کنیم.
قسمت‌های آخر تمرین بالا خارج از فضای شبیه سازی هستند. این قسمت‌ها همان چیزی هستند که در فصل اول "تحلیل داده‌ها" نامیده شد. برای انجام این قسمت‌ها باید شبیه سازی به پایان برسد و نتایج عددی مورد نظر استخراج شده باشند. حال ما تعدادی جدول داریم که می‌توانیم آنها را نمایش بدهیم. مقدار ارتفاع متوسط بر حسب زمان مطمئنا یک منحنی خطی بدون هرگونه انحرافی خواهد بود. این را از مدل می‌دانیم ولی رسم این منحنی خالی از لطف نیست. همیشه یادتان باشد که در بیشتر شبیه سازی‌ها خروجی‌هایی داریم که به دلایل نظری مقدار عددی آنها را انتظار داریم. این خروجی‌ها برای اطمینان از درستی کار و اعتماد به آن نتایجی که از درستی شان مطمئن نیستیم خیلی مهم هستند.
حال می‌رسیم به گزارش β که نتیجه نهایی این مسئله است. برای این کار یک مدل داریم که به ما می‌گوید ناهمواری با زمان چگونه رفتار می‌کند. پس برای بدست آوردن β باید نتایج را بر مدل برازش داد. یکی از بهترین روشهای برازش، که بیشتر نرم افزارهای تحلیل داده به آن مجهز هستند،

استفاده از روش کمترین متوسط مجذور فاصله[4] است. برای این کار بهتر است داده‌ها به گونه‌ای رسم شود که مدل به یک خط تبدیل شود. در مورد مسئله‌ی ما اگر $\log(w)$ برحسب $\log(t)$ ترسیم شود، انتظار می‌رود خروجی خطی باشد با شیب β. به این طریق شما می‌توانید نمای دینامیکی را گزارش کنید. نکته‌ی مهم در اینجا این است که هرگاه شما عددی را گزارش می‌دهید باید دقت آنرا نیز گزارش دهید. به بحث دقت و خطا در بخش‌های بعدی مفصل می‌پردازیم. در اینجا برای داشتن تصویری از دقت عددی که گزارش می‌کنید کافی است که شبیه سازی را چند بار تکرار کنید و تعدادی β بدست آورید. به این روش شما می‌توانید متوسط این اعداد را به عنوان عدد نهایی و انحراف از معیار آنها را به عنوان معیاری از خطا گزارش کنید.

3.3.دیگر نماهای بحرانی در فرآیند نشست

شاید به نظر برسد که یک راه برای کاهش خطای عدد گزارش شده برای نمای دینامیکی افزایش زمان اجرای برنامه باشد. این ایده در مورد ول نشست و خیلی دیگر از پدیده های فیزیکی درست است ولی نمی‌شود به عنوان یک اصل به آن نگاه کرد. در بسیاری از فرآیندهای رشد، رفتار خطی منحنی $\log(w)$ برحسب $\log(t)$ یک روند دائمی نیست و بعد از گذشت زمانی (که آنرا زمان اشباع، t_s، می‌نامیم) اشباع می‌شود و ناهمواری سطح به یک مقدار حدی می‌رسد و دیگر رشد نمی‌کند. این رفتار به طور شماتیک در شکل زیر نشان داده شده است.



شکل 2 رشد ناهمواری با زمان به یک اشباع می‌رسد

[4] Least Mean Square (LMS)

ولی مقدار زمان اشباع به ابعاد سیستم بستگی دارد. این رابطه نیز به صورت مقیاسی‌است و از رابطه زیر تبعیت می‌کند:

$$t_s \sim L^z$$

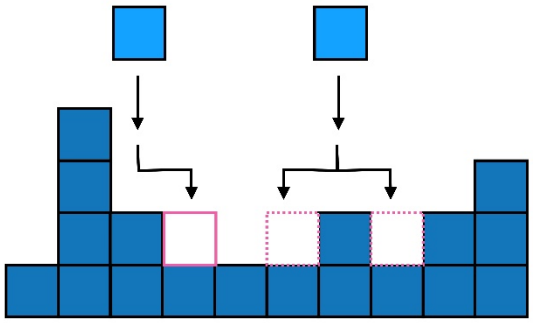
z یک نما دیگر از مجموعه نماهایی است که کلاس جهانشمولی فرآیند رشد را تعیین می‌کند. w_s، مقدار ناهمواری در زمان اشباع هم با توجه به اینکه ناهمواری تا نقطه اشباع بر روی خط مدل است با ابعاد سیستم باید رفتاری مقیاسی نشان دهد،

$$w_s \sim t_s^{\beta} \sim L^{z\beta} \sim L^{\alpha}$$

که α نیز یک نمای دیگر است که همانطور که از رابطه بالا پیداست مستقل از دو نمای دیگر نیست.

3.4. پایین نشست (Ballistic Deposition with Relaxation)

در بیشتر فرآیندهای نشست، ذرات بر روی سطح، آزادی حرکتی محدودی دارند. به این ترتیب دلیلی وجود ندارد که در اولین جایگاهی که سقوط می‌کنند متوقف شوند. این جابجایی می‌تواند سطح را هموارتر کند چون ایجاد اختلاف ارتفاع زیاد در جایگاه‌های مجاور خیلی مطلوب نیست. برای مدل کردن این فرآیند مجددا فرض می‌کنیم که ذرات به صورت کتره‌ای از ارتفاعی بالای زیر لایه بر روی آن سقوط کنند. ولی بعد از رسیدن به سطح، امکان جابجایی به اندازه‌ی یک واحد، برای پیدا کردن جایگاهی در ارتفاع پایین‌تر به ذره داده شود. به این ترتیب اگر ذره در همسایگی جایگاه اولیه فرود خود جایگاهی با ارتفاع پایین‌تر بیابد به آنجا سقوط می‌کند. در صورتی که هر دو همسایه در ارتفاع پایین تری باشد همسایه گوتاه تر را انتخاب می‌کند و در صورتی که ارتفاع ها برابر باشند به طور تصادفی به یکی از آنها خواهد رفت. این نشست را "پایین نشست" می‌نامیم.



شکل 3 نمایی فرضی از فرایند ته نشست که نحوه‌ی رشد را نشان میدهد

آلگوریتم این شبیه سازی بسیار شبیه مسئله قبل است. تغییراتی جزیی نیاز است تا بعد از انتخاب یک جایگاه به طور کاتوره‌ای مقادیر ارتفاع در این جایگاه و دو جایگاه مجاور مقایسه شوند و به این وسیله جایگاهی

که ارتفاعش باید افزایش یابد بدست آید. ولی این مدل یک تفاوت اساسی با ول نشست دارد و آن وجود همبستگی است. در مدل ول نشست هر جایگاه مستقل از همسایگانش رشد می‌کرد و هیچ سازوکاری که بتواند بین جایگاه های مختلف همبستگی ایجاد کند وجود نداشت. در ته نشست همسایه‌ها همدیگر را می‌بینند. هر جایگاه نمی‌تواند خیلی بیش از همسایه اش رشد کند. این خاصیت دلیل اصلی وجود اشباع در این مدل است. هر چند در نگاه اول هر جایگاه فقط همسایه اولش را می‌بیند ولی با تحول این سیستم طول همبستگی بین جایگاه‌ها نیز رشد می‌کند و ارتفاع یک جایگاه به ارتفاع همسایه‌های دور تر نیز همبسته می‌شود. در فصل‌های بعد تعریف دقیقی از طول همبستگی ارایه خواهیم کرد. ولی در این جا کافیست که به این واقعیت که طول همبستگی با زمان نشست رشد می‌کند، اعتماد کنیم. ولی محدود بودن طول شبیه سازی، امکان رشد نامتناهی را به طول همبستگی نمی‌دهد. یک حد اشباع برای طول همبستگی وجود دارد و این همان عاملی است که باعث می‌شود ما در این سیستم یک اشباع برای ناهمواری ببینیم. عدم وجود همبستگی در ولنشست باعث شده بود که در آنجا ناهمواری اشباع نشود.

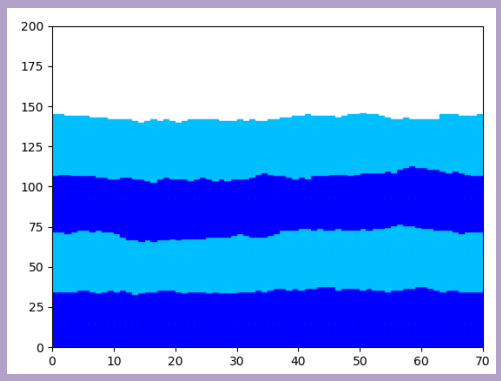
3.5. شرایط مرزی

مستقل از فیزیک جدیدی که وجود همبستگی برایمان به همراه دارد، این خاصیت یک مشکل تکنیکی هم برای شبیه سازی ایجاد می‌کند. اگر بخواهیم قوانین بازی را برای تمام جایگاه‌ها اجرا کنیم در مورد دو جایگاه انتهایی با مشکل روبرو می‌شویم. این دو جایگاه با بقیه متفاوت هستند. برای حل این مشکل دو راه وجود دارد.

- شرط مرزی دیواره‌ی صلب
  در این حالت قبول می‌کنیم که این دو جایگاه با بقیه متفاوت هستند و فقط یک همسایه دارند. این کار همگنی مسئله را مخدوش می‌کند. همسایه‌های این دو جایگاه شانس بیشتری برای دریافت ذرات فرودی بر همسایگانشان را دارند. به طور کلی هر گونه فرضی که نقاط مرزی را متمایز از دیگر نقاط کند باعث می‌شود که مرز در نتایج تاثیر مشهودی داشته باشد و این به طور کلی بسیار نامطلوب است مگر در مواردی که واقعا تمایلی بر مطالعه مرزهای فیزیکی باشد.
- شرط مرزی تناوبی
  فرض کنیم که در مجاورت این سیستم مجموعه‌هایی کاملا مشابه با آنچه ما شبیه سازی می‌کنیم وجود دارد. به دلیل آنکه فقط همسایه اول مورد توجه است نیاز به تکرار کل سیستم نیست وفقط می‌توان تصویری از دو جایگاه انتهایی در انتهای دیگر در نظر گرفت. در حقیقت این عمل مانند این است که دو انتهای شبیه سازی را مانند یک حلقه به هم متصل کرده‌ایم. ضمن این که این حلقه همچنان طول محدودی دارد ولی در یک حلقه هیچ تفاوتی بین نقاط شبکه وجود ندارد. این پیشنهاد به دلیل همگن نگه داشتن فضا بر شرط مرزی دیواره صلب ارجح است.

	.3.2

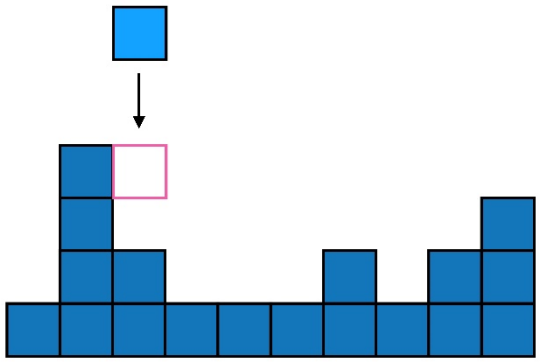
تمرین



- مدل پایین نشست را در یک بعد شبیه‌سازی کنید. برای اینکار خطی افقی به طول 200 واحد با مرز متناوب به عنوان زیر لایه در نظر بگیرید و ذرات را بر روی آن بنشانید.
- دینامیک مدل را بر روی نمایشگر نشان دهید. برای درک بهتری از رفتار زمانی رشد بهتر است که رنگ ذرات فرودی را با زمان به طور تناوبی تغییر دهید. پس از مشاهده‌ی زیرلایه‌ها، بخش نمایش گرافیکی برنامه را غیرفعال کنید.
- مقدار متوسط ارتفاع و ناهمواری را در بازه‌های زمانی متوالی محاسبه کنید. برای مشاهده رفتار شکل 2 چند ذره باید نشانده شود؟
- منحنی تغییرات ناهمواری بر حسب زمان را رسم کنید. برای مشاهده نقطه‌ی تغییر نما در شکل 2 باید چند بار شبیه سازی را تکرار کرده و متوسط ناهمواری را در هر نقطه نمایش دهید.
- β، α و z را برای پایین نشست محاسبه و گزارش کنید.
- آیا تصوری از دقت عددی که در بالا گزارش کرده‌اید دارید؟

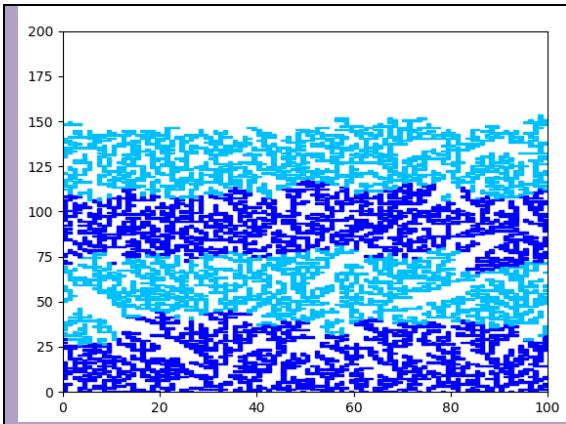
3.6.کنار نشست

اجازه حرکت بر روی سطح تنها راه برای ایجاد همبستگی میان همسایگان نیست. می‌شود ساز و کارهای دیگری را نیز پیشنهاد داد که مکان نشست ذرات فرودی وابسته به ارتفاع لایه در همسایگی باشد. یکی از این مدل‌ها کنارنشست است. مدلی که امکان رشد عرضی را نیز به لایه می‌دهد و برای نشست ذرات می‌توانند از کنار نیز به ذرات دیگر بچسبند. در این مدل مجددا فرض می‌کنیم که ذرات به صورت کتره‌ای از ارتفاعی بالای زیر لایه بر روی آن سقوط کنند. ولی به محض رسیدن به جایگاهی که در همسایگی آن ذره‌ایی قبلا نشسته باشد متوقف می‌شوند. با این روش وجود حفره در مدل امکان پذیر است و لایه‌ای متخلخل ایجاد می‌کند. رشد عرضی و وجود تخلخل، لایه را از پایین نشست متفاوت می‌کند، هرچند به دلیل وجود همبستگی میان همسایگان رفتاری مشابه ولی با نماهایی متفاوت دارد.



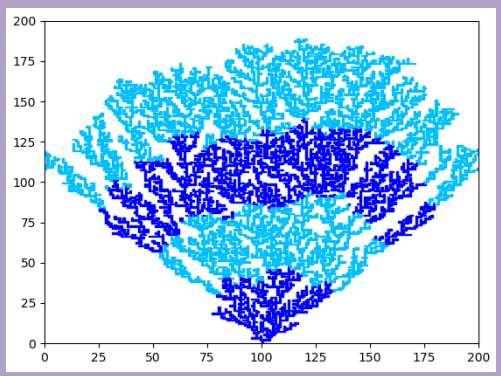
شکل 4 نمایی فرضی از فرایند کنارنشست که نحوه‌ی رشد را نشان میدهد

مشابه حالت قبل کافی است که بعد از انتخاب یک جایگاه به شکل کاتوره‌ای، ارتفاع این جایگاه با ارتفاع همسایگان مقایسه شود. در پایین نشست این مقایسه برای یافتن جایگاه و ارتفاع نهایی ذره بود و در کنار نشست به منظور یافتن ارتفاع نهایی ذره است. همچنین مشابه حالت قبل شرایط مرزی نیز اهمیت دارد.

.3.3 تمرین	 - مدل کنارنشست را در یک بعد شبیه سازی کنید. برای اینکار خطی افقی به طول 200 واحد با شرایط مرزی متناوب به عنوان زیر لایه در نظر بگیرید و ذرات را بر روی آن بنشانید. - دینامیک مدل را بر روی نمایشگر نشان دهید. برای درک بهتری از رفتار زمانی رشد بهتر است که رنگ ذرات فرودی را با زمان به طور تناوبی تغییر دهید. مشابه تمرین قبل پس از مشاهده‌ی رفتار لایه‌ها قسمت خروجی گرافیک برنامه را غیرفعال کنید. - مقدار متوسط ارتفاع و ناهمواری را در بازه های زمانی متوالی محاسبه کنید. در این شبیه‌سازی نیز تعداد ذرات نشانده شده، همچنین متوسطگیری نتایج اجراهای متعدد بر نتیجه اثر می‌گذارد. - منحنی تغییرات ناهمواری بر حسب زمان را رسم کنید. - β، α و z را برای کوتاه نشست محاسبه و گزارش کنید.

در این مدل نیز شاهد زمانی خواهیم بود که پس از آن ناهمواری به حالت پایا می‌رسد. چرا در ولنشست ناهمواری پایا نیست؟ هنگامی که در مدل برهمکنش با خانه‌های همسایه‌ی نزدیک معرفی می‌شود وضعیت خانه‌ها بر خانه‌های فراتر از همسایگی خود تاثیر خواهد گذاشت که رفتاری شبیه به

طول همبستگی سیستم است. یکی از جذابیت‌های مدل کنار نشست این است که قبل از تعریف دقیقی از طول همبستگی، تصویری از آن برایمان ایجاد می‌کند. تمرین بعد به شما کمک می‌کند که این تصویر را بدست آورید.

3.4 تمرین	طول همبستگی در کنار نشست: تمرین قبل را با یک تغییر کوچک تکرار کنید. در این تمرین به جای یک زیر لایه مسطح، با یک نقطه شروع کنید. یعنی فقط یک نقطه در فضا وجود دارد که اگر ذره ای بر آن یا همسایگان آن سقوط کند به آن می‌چسبد ولی بقیه ذرات از سیستم خارج می‌شوند. با پیشرفت فرآیند این نقطه مانند بذر یک درختچه رشد می‌کند. هم در ارتفاع و هم در عرض. پهنای عرضی این درختچه را بر حسب زمان رسم کنید. آیا رفتار مقیاسی مشاهده می‌شود؟ نمای این مقیاس چیست؟ 

امکان دارد با تغییر قوانین بازی مدل‌های دیگری نیز برای رشد بتوان معرفی کرد. در مرجعی که در انتهای این بخش معرفی شده است تعدادی از این مدل‌ها معرفی شده است. ولی نکته‌ی جالب این مدل‌ها این است که همگی در کلاس جهانشمولی یکی از سه مدلی که در بالا معرفی شد قرار می‌گیرند و رفتار مقایسی آنها از مجموعه نماهایی که شما بدست آوردید تبعیت می‌کند.

3.7. مدلهای نشست رقابتی

در بعضی از مدلهای نشست، رقابتی میان جایگاه‌های نشست ذرات در جذب ذرات جدید وجود دارد. در مدل‌های با برهمکنش همسایه نزدیک که در بالا معرفی شد (پایین نشست و کنار نشست) هر جایگاه در صورت رشد سعی در بالا بردن همسایه‌های خود داشت. این رفتار یک نوع رفتار جمعی است که همبستگی میان همسایگان را افزایش می‌دهد. ولی مدل‌هایی وجود دارد که به جای همیاری رقابتی بین جایگاه‌ها ایجاد می‌کنند. در این مدل‌ها هر جایگاهی که به نحوی موفق به رشد شود سایه‌ای بر همسایگان خود می‌اندازد که از رشد آنها جلوگیری کند. به راحتی می‌تواند تصور کرد که چنین رقابتی باعث می‌شود که افت خیزها رشد کند و تفاوت میان جایگاه های کوتاه و بلند بیشتر شود. در بخش‌های بعدی به این موضوع برگشته و بیشتر صحبت خواهیم کرد. در این جا فقط بر آشنایی مقدماتی به مدلی بسیار ساده در تمرین زیر اشاره می‌شود.

3.5 تمرین	ول نشست رقابتی (رشد سوزنی): در مدل ول نشست فرض کنید که ذراتی که برای نشستن بر روی زیر لایه به سمت آن حرکت می‌کنند به جای اینکه در راستای خط قائم (عمود بر زیر لایه) سقوط کنند در راستایی که با خط قائم زاویه می‌سازد حرکت می‌کنند. در این حرکت بعد از برخورد ذرات به اولین ستونی که در مسیر راهش قرار دارد جذب آن ستون شده و ارتفاع آنرا یک واحد افزایش می‌دهد. توجه کنید که این ذره امکان دارد

	به میان یک ستون برخورد کند. در این حالت نیز فرض بر این است که ارتفاع ستون افزایش می‌یابد. - نمایشی از سیستم ارایه کنید. - آیا رشد دینامیکی این سیستم با ول نشست مشابه است؟ - در بازهای زمانی مختلف فاصله‌ی دورترین نقاطی که در سمت چپ و راست روی شاخه قرار دارند را بر حسب زمان رسم کنید.

بیشتر بدانیم:

برای آشنایی با مدلهای مختلف لایه نشانی و فراگرفتن روشهای تحلیلی مطالعه‌ی این فرایندها کتاب "Fractal concepts in surface growth" نوشته‌ی Albert-Laszlo Barabasi و Harry Eugene Stanley بسیار کتاب جذاب و مفیدی است. در ضمن این کتاب مدلهای دیگری برای رشد را نیز معرفی میکند.

4. تـراوش (Percolation)

فرض کنید می‌خواهید پوششِ رنگیِ رسانا داشته باشید. یک پیشنهاد ساده شاید مخلوط کردن ریز براده های فلزی یا پودر ماده رسانایی مانند گرافیت در رنگ باشد. ولی یک سوال ساده این است که «چه مقدار ماده رسانا باید در رنگ ریخت تا حاصل ماده‌ای رسانا شود؟». اگر مخلوط را خوب به هم بزنیم تا کاملا همگن شود برای مشاهده‌ی رسانش در محصول باید شبکه ای از ذرات رسانای متصل به هم در آن وجود داشته باشد که ضمن اتصال به یکدیگر در درون رنگ، بین نقاط مختلف نیز ارتباط برقرار کند. مثال‌های دیگری نیز می‌توان زد که در حقیقت فیزیکی مشابه مثال بالا داشته باشد. به طور مثال اگر جامد متخلخلی داشته باشیم که آب بتواند در درون خلل و فرج آن نفوذ کند، چه مقدار تخلخل برای اینکه آب بتواند از یک سوی این جامد به سوی دیگر آن تراوش کند نیاز است؟ اسفنج ظرفشویی مثالی از چنین جامد متخلخل و تراوایی است. این مثال دلیل انتخاب نام "تراوش" برای این پدیده را به خوبی توجیه می‌کند، هر چند که مثال‌های کاملا متفاوتی در ظاهر می‌توان یافت. حتی می‌توان مثال‌هایی از جامعه شناسی یا بهداشت زد. مثلا در یک شبکه اجتماعی یک شایعه یا لطیفه چقدر باید جذاب باشد تا در این شبکه منتشر شود؟ یا در مسئله ای بسیار مشابه ولی با عواقبی کاملا متفاوت، احتمال سرایت یک بیماری چقدر باشد تا در یک جامعه همه گیر شود؟ برای چه تراکمی از درختان در یک جنگل یک آتش سوزی کوچک می‌تواند به یک فاجعه تبدیل شود؟

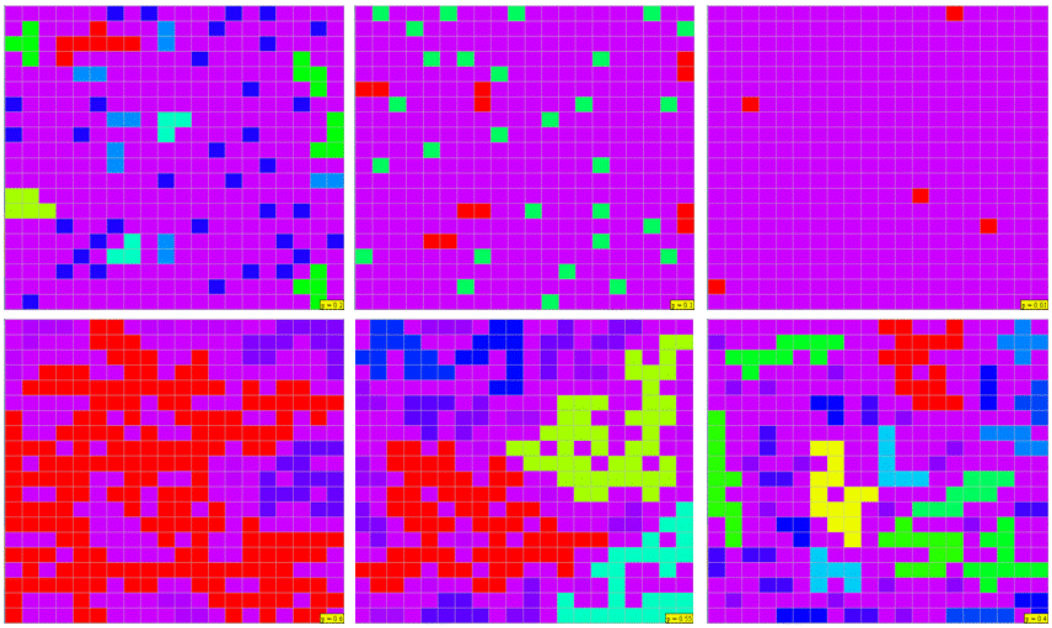
شکل 1 آتش سوزی در جنگل‌های آمریکا. عکس برگرفته از نشریه نشنال جیوگرافیک.

مستقل از اهمیت کاربردی تراوش، این مدل از نظر آماری نیز بسیار غنی است. بسیاری از مفاهیم اساسی در مبحث سیستم های پیچیده مانند تغییر فاز و رفتار بحرانی را می‌تواند در آن مطالعه کرد. در این بخش با شبیه سازی تراوش سعی می‌کنیم که ما هم با این مفاهیم آشنا شویم. در اینجا نه تنها با کمک این مدل با شبیه سازی تغییر فاز در سیستم های آماری آشنا می‌شویم، بلکه با یکی از مهم ترین محدودیت‌های شبیه سازی، مسئله‌ی ابعاد محدود، درگیر می‌شویم.

4.1. معرفی مدل

برای معرفی تراوش با یک مثال ساده بر روی یک شبکه‌ی مربعی دو بعدی شروع می‌کنیم. فرض کنید که بر روی هر یک از خانه‌های این مربع چراغی قرار دارد که با احتمال p می‌تواند روشن شود. اگر p عددی کوچک باشد فقط تعداد خیلی کمی از خانه‌ها روشن می‌شوند. برای p های بزرگتر تعداد خانه‌های رنگ شده بیشتر می‌شود. با افزایش p احتمال اینکه بعضی از این خانه‌های روشن در مجاورت هم ظاهر شوند بیشتر می‌شود. این مجموعه‌ها را خوشه (یا جزیره) می‌نامیم. کوچکترین خوشه‌ها فقط از دو خانه‌ی مجاور هم تشکیل می‌شود. در p های بزرگتر خوشه‌هایی با مساحت بیشتر دیده می‌شود. حال می‌توان سوال کرد که p باید چه مقدار باشد تا حداقل یک خوشه وجود داشته باشد که دو سمت این شبکه را به هم متصل کند. اگر پاسخ عجولانه ی شما $p = 1$ است در اشتباهید. چون حتی برای بعضی p های کوچکتر از یک نیز این واقعه می‌تواند با یقین اتفاق بیافتد.

می‌توان داستان را جور دیگری نیز تعریف کرد. از شبکه خاموش شروع می‌کنیم و یکی یکی خانه‌های خاموش را به طور کتره‌ای انتخاب می‌کنیم و آنها را روشن می‌کنیم. با ادامه این کار کم کم خوشه‌های پراکنده به هم متصل می‌شوند و خوشه‌های بزرگتری را تشکیل می‌دهند. حال سوال این است که بعد از روشن کردن چه تعدادی از خانه‌ها تراوش اتفاق می‌افتد. به زبان دیگر چه وقت خوشه بینهایت تشکیل می‌شود. در این جا به خوشه‌ای که دو سوی شبکه را به هم متصل می‌کند خوشه‌ی بینهایت می‌گوییم.



شکل 2 یک شبکه‌ی مربعی با احتمال‌های متفاوت برای روشن شدن جایگاه‌ها را نشان می‌دهد. در مقادیر کوچکِ احتمال، خوشه‌ها کوچک و پراکنده هستند. با افزایش احتمال، ابعاد خوشه‌ها رشد می‌کند وخوشه‌ی بینهایت تشکیل می‌شود.

با تصویری که اکنون بدست آوردید به مثال اول این فصل برمی‌گردیم. فرض کنید خانه‌های روشن نماینده‌ی ذرات رسانا و خانه‌های خاموش نشانگر ماده رنگی نارسانا باشد. به این ترتیب می‌توانید ببینید که تا قبل از یک زمان خاص، جسم فوق توان رسانش را ندارد ولی ناگهان و در ازای روشن شدن تعدادی جایگاه، به طور ناپیوسته به یک جسم رسانا تبدیل می‌شود. در این لحظه در ساختار ماکروسکوپی

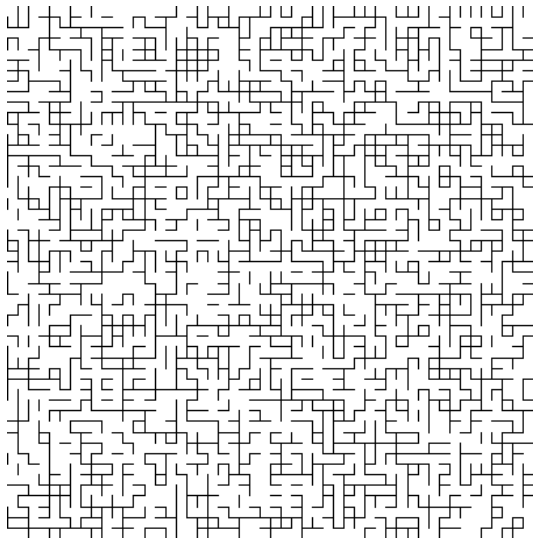
ما یک تغییر فاز ناپیوسته از نارسانا به رسانا اتفاق افتاده است. پارامتر کنترل در این تغییر فاز، احتمال وجود ریز ذرات رسانا در جایگاه‌ها (روشن بودن خانه‌ها) است.

پس اگر احتمال تراوش، که به معنی وجود حداقل یک خوشه بینهایت در سیستم است را با Q نشان دهیم انتظار داریم که Q در مقادیر کوچک p **صفر** و برای مقادیر بزرگ آن **یک** باشد. نکته جالب در مسئله‌ی تراوش این است که برای یک سیستم بینهایت بزرگ (حد ترمودینامیکی) یک مقدار بحرانی برایp وجود دارد که در $p = p_c$، فاز سیستم به طور ناپیوسته تغییر می‌کند. یعنی احتمال داشتن خوشه بینهایت برای احتمال‌های کوچکتر از مقدار بحرانی صفر، و برای بالاتر از آن یک است.

$$Q = \begin{cases} 0 & p < p_c \\ 1 & p \geq p_c \end{cases}$$

البته در شبکه محدود و غیر بینهایتِ مثال ما، این کمیت به طور پیوسته از صفر به یک تغییر می‌کند. زیرا برای یک شبکه محدود برای هر احتمال کوچکی نیز احتمال تولید خوشه بینهایت غیر صفر است. البته شیب این تغییرات در نزدیکی مقدار بحرانی شدیدتر است و این شیب با بزرگ شدن شبکه رشد می‌کند. این مثالی از تاثیر محدودیت‌های شبیه سازی بر نتایج است. در اینجا اندازه‌ی محدود سیستم رفتار فیزیکی سیستم را تحت تاثیر خود قرار می‌دهد. در ادامه‌ی این فصل خواهیم دید که چگونه باید این مشکل را کنترل کرد.

تراوش را میتوان برای پیوندهای بین جایگاه‌ها نیز تعریف کرد. فرض کنید که هر راس شبکه‌ی مربعی قابلیت اتصال به همسایگان خود به وسیله‌ی پیوندهایی را دارد. ولی در ابتدا تمام این پیوند ها قطع هستند. حال به طور تصادفی و با احتمال p آنها را متصل می‌کنیم. مجددا می‌توان انتظار داشت که برای مقادیر بزرگ p و در صورت اتصال تعداد زیادی از پیوندها، دو سوی شبکه از طریق راه‌های ارتباطی به یک دیگر متصل شوند، یا به عبارت دیگر خوشه بینهایت تشکیل شود.



شکل 3 نمایی از یک تراوش پیوندی بر روی شبکه مربعی که تصویری مانند یک هزارتو را تداعی می‌کند.

مقدار p_c نه تنها به ابعاد مدل بستگی دارد بلکه به جزئیات شبکه و مدل نیز وابسته است. به طور مثال مقدار آن برای دو شبکه‌ی دو بعدیِ مربعی و مثلثی متفاوت است. همچنین برای تراوش جایگاهی و تراوش پیوندی بر روی شبکه های یکسان می‌تواند متفاوت باشد. در نتیجه احتمال بحرانی یک کمیت جهان‌شمول نیست. در ادامه با بعضی کمیتهای‌های جهان شمول تراوش آشناه می‌شویم. این کمیتها به جزییات مدل حساس نیستند ولی به بُعد سیستم بستگی دارند. یعنی تغییر در بُعد مسئله کلاس جهانشمولی را تغییر می‌دهد. چنین حساسیتی به بعد در تمام مسائل فیزیک وجود دارد.

مسئله ی تراوش بر روی شبکه یک بعدی جواب بدیهی $p_c = 1$ دارد. چون حتی یک ناپیوستگی در کل سیستم تراوش را غیر ممکن می‌کند. حل این مسئله در شبکه‌هایی که امکان ایجاد حلقه بر روی شبکه وجود ندارد مانند درخت کایلی[5] نیز ساده است. روش‌های مختلفی هم برای حل در دو بعد پیشنهاد شده است که در بعضی از شبکه‌ها حل دقیق را می‌دهد، ولی مسئله برای بعد 3 حل دقیق ندارد. در حقیقت دقیق‌ترین نتایجی که در بعد 3 وجود دارد از شبیه سازی‌های کامپیوتری بدست آمده‌اند و صحت حل‌های تقریبی با مقایسه با این نتایج تایید می‌شود. این خود مثالی است که اهمیت شبیه سازی را نشان می‌دهد. در این مورد خاص با وجود اینکه حل تحلیلی مسئله بسیار مشکل است، شبیه سازی آن بسیار ساده و ابتدایی است.

.4.1 تمرین	تراوش: – یک شبکه‌ی دوبعدی و مربعی $L \times L$ (با قابلیت انتخاب L در ورودی) تولید کنید. – با احتمال p خانه‌های شبکه را روشن کنید. برای این کار کافی است که برای هر خانه یک عدد کاتوره‌ای بین صفر و یک تولید کنید. درصورتی که این عدد از p کوچکتر بود آن خانه را روشن کنید. در این جا اشکالی ندارد که این کار را با ترتیب خاصی از یک خانه شروع کنید و یک به یک جلو بروید. – برای برنامه یک خروجی عددی دوگانی[6] (0 و 1) اختصاص دهید که در صورت وجود خوشه‌ی بینهایتی که دو ضلع چپ و راست شبکه را به هم متصل کند (تراوش عرضی)، عدد 1 ودر غیر این صورت عدد 0 را گزارش کند.

4.2. *آلگوریتمی برای تشخیص تراوش*

با انجام تمرین قبلی متوجه می‌شوید که شبیه سازی تراوش بسیار ساده است. در حقیقت آن چیزی که مشکل است و در مسئله‌ی بالا بسیار زمان‌بر است، نه قسمت تولید شبکه تراوشی بلکه آخرین قسمت مسئله یعنی **تشخیص تراوش** است. برای تولید یک شبکه تراوش با یک احتمال مشخص $N \sim L^d$ محاسبه برای تشخیص روشن یا خاموش بودن پیوندها (یا جایگاه‌ها) نیاز است. در اینجا L اندازه‌ی شبکه و d بُعد

[5] Cayley tree
[6] Binary

مسئله است. ضریب تناسب رابطه بالا بستگی به جزئیات شبکه دارد. به طور مثال در یک شبکه 2 بعدی مربعی این ضریب برای تراوشِ جایگاهی 1 و برای تراوش پیوندی 2 است. به این ترتیب کُد تولید یک شبکه تراوشی یک کُد مرتبه‌ی L^d است.

برای تشخیص وجود خوشه‌ی بینهایتی که چپ و راست شبکه را به هم وصل می‌کند شاید ساده‌ترین روشی که بنظر برسد این است که نقطه‌ای بر روی ضلع چپ انتخاب شود و سعی شود که از روی مسیرهای تولید شده به سمت دیگر رسید. در این راه اگر راه برایمان باز باشد و از سوی دیگر شبکه سر در آوریم که جای خوشبختی است و مسئله تمام است، ولی در صورتی که به بن بستی مواجه شویم باید برگردیم و از آخرین نقطه‌ای که حق انتخاب در مسیر (دوراهی یا چند راهی) داشتیم مسیر دیگری را انتخاب کنیم. این کار باید آنقدر ادامه یابد یا به سوی دیگر برسیم ویا مطمئن شویم که راهی برای عبور از نقطه‌ای که شروع کردیم وجود ندارد. در این صورت باید نقطه دیگری در سمت چپ را برداشته و کار را به همین روش تکرار کنیم. شاید این روش برای مقادیر خیلی بزرگ p که احتمال برخورد با بن بست کم است و یا مقادیر خیر کوچک p که خوشه ها بسیار کوچک است راه بدی نباشد ولی برای بازه‌ی وسیعی از مقادیر p روشی بسیار وقتگیر است. فرض کنید در نزدیکی مقدار بحرانی p باشیم و خوشه‌ی بینهایت تشکیل شده باشد. در این روش باید تمام راه‌های ممکن عبور از شبکه برای یافتن تعداد محدودی راه عبور کنترل شود. این مانند یافتن سوزنی در انبار کاهی است. اگر در جستجوی خود برای راه‌های ممکن در هر نقطه شبکه b انتخاب وجود داشته باشد، تعداد راه‌های ممکن از مرتبه b^N است. b عددی است که به بعد و نوع شبکه بستگی دارد و در ابعاد بالاتر از 1 مطمئنا بزرگتر از واحد است. در نتیجه تعداد راه‌های ممکن بسیار سریع با N رشد می‌کند. برای درکی از بزرگی این عدد مثالی می‌زنیم. برای یک شبکه نسبتا محدود 100×100 و با فرض $b \approx \sqrt{10}$ تعداد راه های ممکن از مرتبه 10^{5000} است. اگر بررسی تمام این راه‌ها برای سریع‌ترین ابرکامپیوترهای دنیا نیز بسیار بیشتر از عمر جهان ($\sim 10^{17}$ s) وقت می‌گیرد. به این گونه برنامه‌ها که زمان اجرای آن با نمایی از N رشد میکند NP-پیچیده می‌گویند. به زبان ساده اینها مسائلی هستند که از نظر محاسباتی غیر قابل حل هستند.

این مثال خوبی است که نشان می‌دهد که در یک برنامه کامپیوتری قسمت‌های مختلف برنامه از نظر وقت گیری می‌توانند بسیار متفاوت باشد. بدیهی است که زمان اجرای نهایی برنامه را کندترین قسمت آن تعیین می‌کند. مشکلاتی مانند مثال بالا با تغییرات جزیی قابل حل نیستند ونیازمند آلگوریتم‌های جدید هستند. معمولا مرتبه‌ی کُد را به آلگوریتم آن اختصاص می‌دهند. برای مثال آلگوریتمی که در اینجا برای ساختن شبکه تراوشی معرفی شد یک آلگروتیم مرتبه 1 ($\sim N^1$) است ولی الگوریتمی که برای تشخیص تراوش معرفی شد یک آلگوریتم NP-پیچیده است. به این ترتیب ترکیب بالا کمکی به حل مسئله تراوش نمی‌کند.

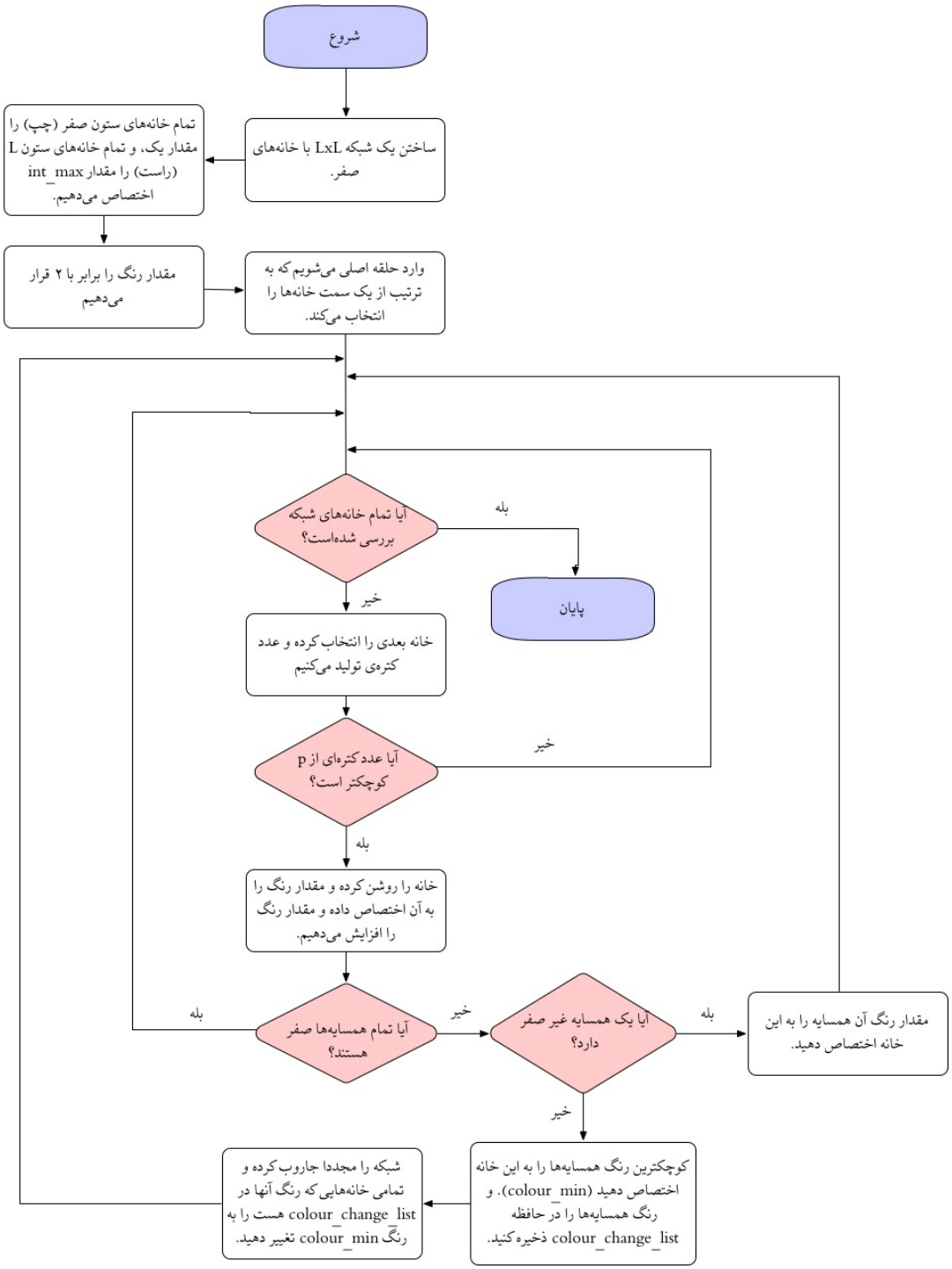
خوشبختانه مشکل بالا با معرفی یک آلگوریتم مبتکرانه حل شده است. این الگوریتم به الگوریتم رنگ آمیزی معروف است. در زیر این الگوریتم برای مسئله‌ی تراوش پیوندی[7] معرفی می‌شود.

[7] کافی است که عبارت های "پیوند" به "جایگاه" تبدیل شود تا از الگوریتم برای شبیه‌سازی تراوش جایگاهی استفاده شود.

آلگوریتم رنگ آمیزی

1. به تمام خانه‌ها یک حافظه با مقدار اولیه 0 اختصاص دهید. (عدد صفر به معنی خاموش بودن خانه است.)
2. تمام خانه‌های روی مرز چپ را با دادن مقدار 1 روشن کنید.
3. تمام خانه‌های روی مرز راست را نیز با اختصاص یک عدد صحیح خیلی بزرگ، int_max، روشن کنید.
4. (شروع حلقه‌ی اصلی) پیوندها را یک به یک انتخاب کنید.
   4.1. با احتمال p به آن یک عدد غیر صفر کوچکتر از int_max که قبلا استفاده نشده باشد اختصاص دهید.
   4.2. تمام پیوندهای همسایه ی این پیوند را بررسی کنید. سه امکان وجود دارد:
      a) تمام همسایه ها 0 هستند: به (4) برگردید.
      b) فقط یک همسایه غیر 0 دارد: مقدار آن همسایه را به این پیوند بدهید و به (4) برگردید.
      c) بیش از یک همسایه غیر 0 دارد: کوچک ترین این اعداد را به این پیوند و تمام پیوندهایی که در کل شبکه شماره ای برابر با عددهای بزرگتر دارند بدهید و به (4) برگردید
5. مقدار عددی یکی از پیوندهای مرز راست را کنترل کنید. اگر مقدار 1 را دارد، تراوش اتفاق افتاده است.

در آلگوریتم بالا می‌توانید از عددهای اختصاص داده شده به خانه‌ها به عنوان کد رنگ در نمایش شبکه استفاده کنید. برای همین به آن "الگوریتم رنگ آمیزی" می‌گویند. در این الگوریتم زمان‌گیرترین قسمت بند (C) در (4.2) است که باید به کمک یک حلقه‌ی شرطی تمام پیوندهایی که قبلا مقدار خاصی به آنها اختصاص داده شده است را یافته و مقدار عددی آنها را جایگزین کنیم. ولی توجه کنید با وجودی که در این فرآیند باید تمام نقاط شبکه کنترل شوند، بازهم فرآیندی از مرتبه‌ی $N \sim L^d$ است. چون این قسمت خودش در درون حلقه‌ای با هزینه محاسباتی از مرتبه L^d نشسته است پس مرتبه الگوریتم نهایی L^{2d} می‌شود.



نمودار شناور 2 آلگوریتم رنگ آمیزی جهت تشخیص تراوش در شبکه را نشان می‌دهد.

.4.2	الگوریتم رنگ آمیزی: – کُدی که برای تمرین 4.1 نوشتید را به آلگوریتم رنگ آمیزی برای تشخیص تراوش مجهز کنید.
تمرین	

	– شبکه تراوش را نمایش بدهید.

4.3. آلگوریتم هُشِن-کُپلِمَن (Hoshen-Keoplman)

پس از پیاده سازی آلگوریتم رنگ آمیزی می‌توان پیش بینی کرد که برای تشخیص تراوش در شبکه‌های بزرگتر به آلگوریتم بهینه‌تر نیاز است. الگوریتمی که در پایین معرفی شده است را هشن و کپلمن برای تشخیص تراوش در شبکه‌های مربعی معرفی کرده‌اند. از مزایای مهم این الگوریتم این است که برخلاف آلگوریتم رنک آمیزی نیازی به بازنگری و تصحیح رنگ‌های خانه‌های همسایه نیست. در این آلگوریتم کل شبکه در یکه مرحله ساخته و تراوش آن مشخص می‌شود.
هنگام پایان این آلگوریتم تمامی خانه‌ها در یک بار بررسی شبکه برچسب گذاری شده و اندازه تمامی خوشه‌ها مشخص شده است. لازم به تاکید است که برای تشخیص تراوش نیازی به بررسی شبکه برای بار دوم نیست ولی می‌توان شبکه را برای بار دوم بررسی کرد و رنگ خانه‌ها را طبق برچسب گذاری تغییر داد. این کار تنها در زمانی انجام می‌شود که نیاز به نمایش شبکه است و با بررسی مجدد شبکه می‌توانیم رنگ خانه‌ها را نمایش دهیم یا برای عیب یابی کُد خروجی حاصل از بررسی مجدد شبکه مطالعه می‌شود.

آلگوریتم هشن-کپلمن

1. به تمام خانه‌ها یک حافظه با مقدار اولیه 0 اختصاص دهید.
2. تمام خانه‌های روی مرز چپ را با دادن مقدار 1 روشن کنید.
3. از خانه بالا سمت چپ شروع کرده و به صورت ستونی (بالا به پایین) خانه‌های شبکه را بررسی می‌کنیم.
4. هر خانه را با احتمال p روشن کرده و با توجه به آرایه یک بعدی L به شکل زیر برچسب گذاری می‌کنیم. رنگ هر برچسب، k است به طوری که L(k)=k.
   - 4.1. اگر همسایه بالا و چپ خاموش بود، خانه برچسب L را اختیار می‌کند (که در ابتدا برابر با 1 است).
   - 4.2. در صورتی که یکی از همسایه‌ها روشن بود برچسب آن همسایه اختیار می‌شود.
   - 4.3. در صورتی که دو همسایه روشن بود، خانه و همسایه بالا (k2) برچسب همسایه چپ (k1) را همزمان اختیار می‌کنند. یعنی L(k2)=k1 و L(k1)=k1.
5. همچنین اندازه خوشه‌ها نیز در آرایه یک بعدی S زخیره می‌شود.
   - 5.1. هر بار که خانه‌ای برچسب گذاری می‌شود اندازه خوشه آن برچسب به اضافه 1 می‌شود.
   - 5.2. اگر دو همسایه غیر صفر بود اندازه خوشه‌ها با یکدیگر جمع زده و به اضافه 1 (برای خانه جدید) می‌شود.

برای فهم بهتر آلگوریتم معرفی شده قسمتی از شبکه زیر را به کمک این آلگوریتم بررسی می‌کنیم. فرض کنید شبکه الف در اختیار ماست. از خانه بالا سمت چپ شروع می‌کنیم. رنگ اولین خانه روشن برابر با 1 است پس L(1)=1 قرار می‌دهیم. از آنجایی که اولین عضو این خوشه مشخص شده است پس اندازه خوشه نیز برابر با 1 است، S(1)=1. دو خانه روشن بعد هم رنگ یک دارند چون دارای یک همسایه در بالا هستند. پس تا به اینجا S(1)=3 شده است. به همین ترتیب خانه روشن چهارم و و پنجم اولین خانه با رنگ جدید 2 و 3 است پس L(2)=2، S(2)=1، L(3)=3، و S(3)=1. در ستون دوم با ستون اولین خانه روشن برچسب رنگ سمت چپ خود را ختیار می‌کند پس L(1)=1 و S(1)=4. دو خانه روشن بعدی نیز به همین ترتیب برچسب رنگ 2 را اختیار کرده S(2)=3. حال به خانه آخر از ستون دوم می‌رسیم. اول از همه رنگ این خانه را برچسب رنگ خانه سمت چپ آن قرار می‌دهیم، یعنی L(3)=3. حال برای تعیین اندازه خوشه‌ها S(L(3))=S(L(3))+S(L(2))+1=5 و S(L(2))=0. سپس برچسب رنگ همسایه بالا را برابر با برچسب رنگ همسایه چپ قرار می‌دهیم، پس L(2)=3. در شکل زیر این کار برای تمامی خانه‌ها

انجام شده. تغییرات حاصل از یک بار بررسی شبکه در شکل ب و رنگ‌ها پس از بررسی مجدد در شکل ج نشان داده شده است.

الف

1	1	0	0	1	0	1
1	0	1	0	0	1	1
1	0	0	1	1	0	0
0	0	1	0	0	0	0
1	1	1	1	1	0	1
0	1	0	1	1	1	0
1	1	1	1	0	1	1

ب

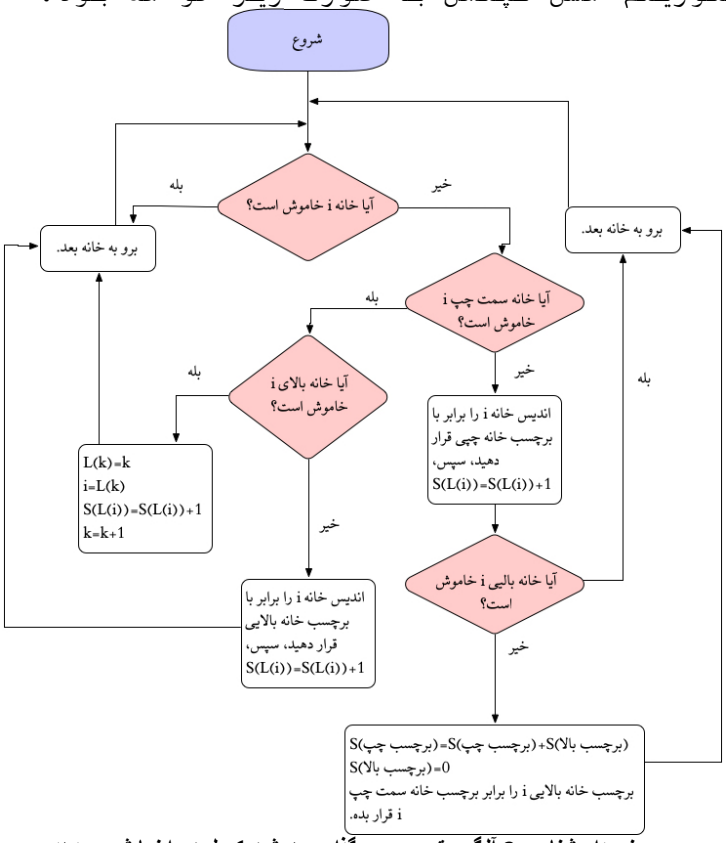
1	1	0	0	7	0	9
1	0	4	0	0	8	8
1	0	0	6	6	0	0
0	0	5	0	0	0	0
2	2	3	3	3	0	10
0	2	0	3	3	3	0
3	3	3	3	0	3	3

ج

1	1	0	0	7	0	8
1	0	4	0	0	8	8
1	0	0	6	6	0	0
0	0	3	0	0	0	0
3	3	3	3	3	0	10
0	3	0	3	3	3	0
3	3	3	3	0	3	3

شکل5 الف) یک شبکه 7 در 7 را نشان می دهد که با احتمال p روشن است. ب) شبکه‌ی الف را نشان می‌دهد که با آلگوریتم هوشن-کوپلمن برچسب گذاری شده‌است. ج) در صورتی که نیاز به نمایش شبکه باشد می توان برچسب‌های شبکه را با رنگ هر برچسب نمایش داد.

نمودار شناور الگوریتم هشن کپلمن به صورت زیر خواهد بود.



نمودار شناور 3 آلگوریتم برچسب گذاری هوشن-کوپلمن را نمایش می‌دهد.

4.4. مکانیک آماری تراوش و تصحیحات اندازه‌ی محدود

همانطور که قبلا اشاره شد پدیده‌ی تراوش به دلیل سادگی برای درک مفاهیم آماری بسیار بکار برده میشود. برای ورود به این بحث لازم است که کمیت‌های مورد نظر را معرفی کنیم.

احتمال وجود شاخه بینهایت Q

این کمیت قبلا معرفی شد. در حد ترمودینامیکی شبکه تراوش یک تغییر فاز ناپیوسته از نارسانا به رسانا در $p = p_c$ نشان می‌دهد. با سیستم های با اندازه محدود این تغییرفاز نرم‌تر می‌شود. برای بدست آوردن این مقدار در شبیه سازی لازم است که برای هر مقدار p شبیه سازی تکرار شود و مقدار Q با متوسط گیری بر روی نتایج بدست می‌آید.

.4.3 تمرین	احتمال ایجاد خوشه بینهایت برای شبکه‌ی محدود: – برنامه آماده شده در تمرین 4.2 را برای طول $L = 10$ آماده کنید. – در داخل برنامه حلقه ای بسازید که بازه ی $0 \leq p \leq 1$ را با قدم‌های $\Delta p = 0.05$ جارو کند و برای هر مقدار p برنامه را 100 بار اجرا کند و با متوسط گیری بر روی دفعاتی که تراوش اتفاق می‌افتد مقدار Q را بدست آورید. – همین کار را مجددا برای $L = 100$ و $L = 200$ اجرا کنید – نتایج بدست آمده برای Q را برای هر سه شبکه بر روی یک منحنی بر حسب p رسم کنید.

احتمال اتصال به خوشه بینهایت Q_∞

اگر به طور تصادفی یک خانه روشن انتخاب شود چه قدر احتمال دارد که این خانه به یک خوشه بینهایت متصل باشد. بدیهی است که Q_∞ تا قبل از تشکیل خوشه بینهایت بنا به تعریف صفر است. بعد از تراوش این کمیت با افزایش p به یک نزدیک می‌شود.

.4.4 تمرین	احتمال اتصال به خوشه بینهایت: – برنامه آماده شده در تمرین 4.3 را به گونه ای تکمیل کنید که Q_∞ را در هر اجرا محاسبه کند و مقدار متوسط آن را گزارش کند. – این کار را برای $L = 10$ ، $L = 100$ و $L = 200$ اجرا کنید. – نتایج بدست آمده برای Q_∞ را برای هر سه شبکه بر روی یک منحنی بر حسب p رسم کنید.

طول همبستگی ξ

شعاع ژیراسیون هر خوشه را می‌توان به عنوان معیاری از اندازه‌ی هر خوشه در نظر گرفت. متوسط اندازه ی تمام خوشه‌های **غیر بینهایت** "طول متوسط اتصال" یا طول همبستگی نامیده می‌شود. شعاع ژیراسیون مشابه مکانیک تحلیلی تعریف می‌شود و مساوی با جذر متوسط مجذور فواصل عناصر خوشه از مرکز جرم خوشه است. در حد ترمو دینامیکی انتظار می‌رود که طول **همبستگی** در نقطه تغییر فاز واگرا شود. ولی برای مدل با اندازه محدود به یک مقدار بیشینه می‌رسد. در قبل از نقطه‌ی بحرانی هیچ خوشه بینهایتی در شبکه مشاهده نمی‌شود. با بزرگ شدن احتمال روشن شدن خانه‌ها و نزدیک شدن سیستم به نقطه‌ی تراوش، متوسط اندازه خوشه‌ها بزرگتر و بزرگتر می‌شود. به طور متوسط نزدیکی نقطه‌ی بحرانی انتظار داریم که خوشه‌های بزرگ شبکه تبدیل به خوشه بینهات شوند. در نتیجه بعد از نقطه‌ی بحرانی خوشه‌های بزرگ اکثرا به خوشه بینهایت می‌پیوندند و هرچقدر احتمال روشن شدن خانه‌ها بیشتر شود، اندازه خوشه‌های غیر بینهایت کوچکتر و کوچکتر می‌شود. این رفتاری است که در شکل 6 مشاهده می‌کنید. با افزایش اندازه شبکه قله‌ی این منحنی تیزتر و تیزتر خواهد شد تا در حد شبکه با طول بینهایت این تابع بسیار تیز و واگرا خواهد بود.

.4.5	طول **همبستگی** : – برنامه آماده شده در تمرین 4.4 را به گونه ای تکمیل کنید که ξ را در هر اجرا محاسبه کند و مقدار متوسط آن را گزارش کند. – این کار را برای $L = \{10, 20, 40, 80, 160\}$ اجرا کنید. – نتایج بدست آمده برای ξ را برای هر سه شبکه بر روی یک منحنی بر حسب p رسم کنید. – در نزدیکی نقطه بحرانی برنامه را برای گام های کوچکتر p تکرار کنید تا دقت منحنی در اطراف این نقطه افزایش یابد. – قلعه ی منحنی ξ مقدار بحرانی احتمال نشان میدهد. همانطور که نتایج میبینید این مقدار به طول شبکه بستگی دارد، $p_c(L)$. آیا میتوانید با برون یابی مقدار $p_c(\infty)$ را بیابید؟
تمرین	

واگرایی ξ در نزدیکی نقطه بحرانی یک رفتار نمایی با نمای $-\nu$ دارد.

$$\xi \sim |p - p_c|^{-\nu}$$

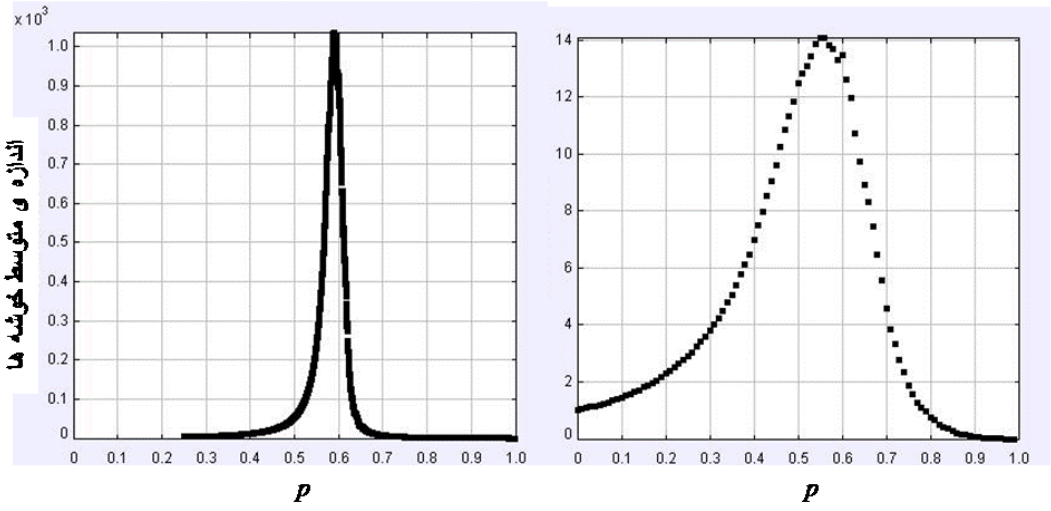
در شبکه‌های کوچک تروا‌ش در احتمال‌های کوچک نیز اتفاق می‌افتد. مثلا در یک شبکه 2 در 2 با p=0.5 نیز می‌توان شاهد تراوش بود. پس قلهای که در شکل 6 مشاهده میکنید با بزرگتر شدن اندازه شبکه به سمت راست منتقل خواهد شد. از طرفی در یک سیستم با اندازه محدود این مقدار هیچگاه نمی‌تواند از L بزرگتر شود در نتیجه رفتار واگرایی تابعی از اندازه شبکه خواهد بود. به این وسیله می‌توان اثر اندازه محدود در شبیه سازی را اصلاح کنیم. این به این معنی است که

$$|p_c(L) - P_c(\infty)| \sim L^{-\frac{1}{\nu}}$$

در حد ترمودینامیکی، یعنی اندازه سیستم به سمت بینهایت میل می‌کند، در نقطه‌ی بحرانی اندازه طول همبستگی به سمت بینهایت میل می‌کند. زمانی که این اتفاق میافتد مقیاس طول سیستم از دست رفته است. فرض کنید که در سیستمی طول همبستگی مشخصی وجود داشته باشد. تا زمانی که این طول مشخص باشد می‌توان کوچکی یا بزرگی آن را نسبت به ابعاد سیستم اندازه گرفت. در نقطه‌ی بحرانی هنگامی که این طول واگرا می‌شود در سیستم رفتار خود تشابه‌ی ایجاد می‌شود. یعنی با نگاه کردن به سیستم در مقیاس‌های مختلف نمی‌توان اندازه‌ی شبکه را تشخیص داد.

.4.6	نمای بحرانی ν:
تمرین	– میدانیم که مقدار p_c برای تراوش پیوندی بر روی شبکه مربعی $\frac{1}{2}$ است. با این اطلاعات و با استفاده از نتایج تمرین 4.5 مقدار ν را بدست آورید.

رد پای واگرایی متوسط طول **همبستگی** را می‌توان در کمیت‌های دیگری نیز مشاهده کرد. یکی از کمیت‌هایی که به راحتی می‌توان واگرایی را در آن مشاهده کرد اندازه‌ی (مساحت یا جرم) متوسط خوشه‌های محدود است. در این متوسط گیری مانند کمیت متوسط طول اتصال باید خوشه ی بینهایت را مستثنی کرد.



شکل 6 اندازه متوسط خوشه ها برای تراوش جایگاهی در یک شبکه مربعی دوبعدی. شکل سمت چپ برای یک شبکه به ضلع 128 و شکل سمت راست برای شبکه ای به ضلع 10 است. بیشینه منحنی برای شبکه بزرگتر دو مرتبه بزرگی بلندتر است و قله پهنای کمتری دارد.

در جدول زیر برخی از نماهای بحرانی برای شبکه‌های مربعی در 2 و 3 بعد گزارش شده است.

جدول 1 نماهای بحرانی در شبکه‌ی مربعی در 2 و 3 بعد				
کمیّت در زمان تراوش	رفتار بحرانی	نمای بحرانی	d=2	**d=3**
انرژی آزاد	$F \sim \lvert p-p_c \rvert^{2-\alpha}$	α	-2/3	
پارامتر نظم	$P_\infty \sim \lvert p-p_c \rvert^{\beta}$	β	5/36	**0.4**
اندازه متوسط خوشه‌های غیر بینهایت، S	$S \sim \lvert p-p_c \rvert^{-\gamma}$	γ	43/18	**1.8**
طول اتصالی	$\xi(p) \sim \lvert p-p_c \rvert^{-\nu}$	ν	4/3	**0.9**
تعداد خوشه‌ها	$n_s(p) \sim S^{-\tau}$	τ	187/91	**2.2**

4.5. آلگوریتم رشد خوشه

برای بررسی خصوصیات خوشه های تراوش یک راه استخراج این خوشه ها از درون شبیه سازی هایی مشابه آنچه در تمرین های قبلی انجام دادید میباشد. ولی در اینجا آلگوریتمی برای تولید یک تک خوشه ی تراوش جایگاهی معرفی میشود.

آلگوریتم رشد خوشه ی تراوش

1. یک جایگاه روشن را در نظر بگیرید.
2. تمام همسایه های این جایگاه را (چهار جایگاه در شبکه مربعی دوبعدی) به ترتیب انتخاب کنید. با احتمال p آنها را روشن کنید و در غیر اینصورت آنها را مسدود کنید.
3. در صورت روشن شدن جایگاه جدیدی همسایه های جدید به خوشه نیز معرفی میشود. درصورتی که این همسایگان قبلا مسدود نشده باشند مشابه (2) به آنها شانس روشن شدن بدهید.
4. قدم (3) را تا زمانی که تمام همسایگان خوشه مسدود باشند یا اندازه خوشه به یک حد بالا برسد ادامه دهید.

4.7.	بعد جرمی (فراکتالی) خوشه های تراوش: – با استفاده از آلگوریتم بالا کُدی برای تولید خوشه های تراوش در یک شبکه دوبعدی مربعی آماده کنید. – برای سه مقدار $p = \{0.5, 0.55, 0.59\}$ خوشه هایی تولید کنید و مقدار ξ و مساحت آنها، s، را برای این خوشه بدست آورید. – در یک نمودار مقدار $\log(s)$ را بر حسب $\log(\xi)$ برای این خوشه ها رسم کنید. – آیا میتوان خطی بر این نقاط عبور داد؟
تمرین	

4.6. تراوش در ابعاد بالاتر

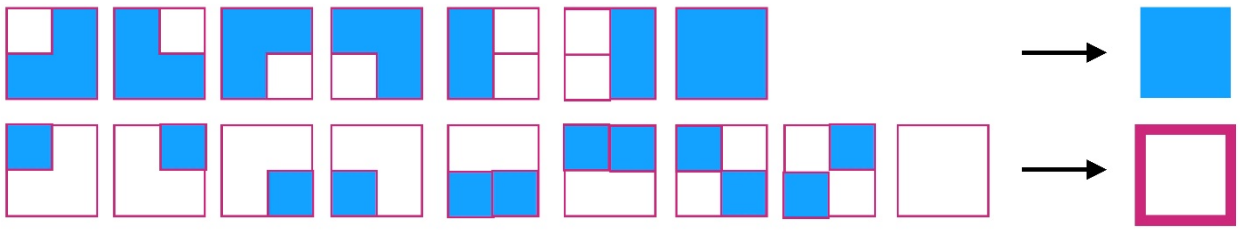
بعد فضایی در مسئله تراوش بسیار در رفتار آن تاثیر میگذارد. از دید نظری حل مسئله در بعد 3 بسیار پیچیده تر از ابعاد پایینتر است. در حقیقت این مسئله در بعد 3 حل دقیق ندارد. ولی در روش شبیه سازی این مسئله اختلاف زیادی بین بعد 2 و 3 وجود ندارد. بیشتر اطلاعاتی که ما از تراوش در بعد 3 داریم از شبیه سازی های کامپیوتری بدست امده است.

یکی از اختلافهای مهم در تراوش در بعد 3 با تراوش در بعد 2 این است که مقدار بحرانی احتمال کوچکتر از $\frac{1}{2}$ است. این به این معنی است که هم جایگاه های روشن و هم خاموش همزمان میتوانند تراوش کنند و خوشه ی بینهایت داشته باشند. به مثال اول این فصل برگردیم، اگر رنگ رسانا را بخواهیم در دو بعد بسازیم سیستم متلاشی میشود. به محض تشکیل خوشه ی بینهایت شبکه به دو قسمت پاره خواهد شد. ولی در بعد 3 هم ذرات رسانا و هم رنگ میتوانند به طور همزمان خوشه بینهایت داشته باشند و این امکان داشتن رنگ رسانا را میدهد.

4.7. گروه باز به هنجارش(Renormalization Group)

در بخش‌های قبل مشاهده کردیم که مقدار احتمال بحرانی تراوش، p_c و همچنین محل واگرایی طول همبستگی با بزرگ شدن ابعاد شبکه تغییر می‌کند. حالا می‌خواهیم با استفاده از گروه باز به هنجارش خواص مقیاسی شبکه‌ها را بررسی کنیم. برای این کار فرض کنید که از یک فاصله‌ی دور به یک شبکه نگاه می‌کنید. در فاصبه دور تشخیص خانههای مجزا از یکدیگر کمی دشوار میشود و تنها مشاهده دسته‌های از خانه‌های چسبیده به هم ممکن خواهد بود. اگر جایگاه‌های این شبکه با احتمالی کمتر از احتمال بحرانی، $p < p_c$ روشن شده باشد هر چه از شبکه دورتر می‌شویم تشخیص روشن و یا خاموش بودن خانه‌های نزدیک به هم دشوارتر خواهد شد. در فواصل خیلی دور تنها اطلاعاتی که برای ما اهمیت خواهد داشت دسته‌ی خانه‌هایی است که به یکدیگر متصل هستند. خوشه‌های کوچک نیز به شکل یک نقطه‌ی روشن دیده می‌شوند و اهمیت خود را ازدست می‌دهند. به طور مشابه زمانی که به یک شبکه‌ی دارای تروواش از دور نگاه کنیم، خوشه‌های بینهایت حتما مشخص‌تر دیده می‌شوند و در فواصل حتی دورتر شبکه به شکل یک خانه‌ی پردیده خواهد شد.

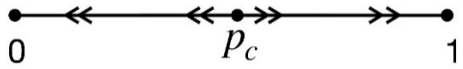
این مثال شما را با روش باز به هنجار کردن فضای حقیقی[8] آشنا کرد. میخواهیم دگرگونی[9] مقیاسی را بررسی کنیم که احتمال روشن و خاموش بودن جایگاه های شبکه را به طور موضعی تغییر ندهد. برای این کار فرض می‌کنیم زمانی که از دور به شبکه نگاه می‌کنیم هر b در b خانه کنار هم را به شکل یک جایگاه روشن یا خاموش می‌بینیم. در شکل 7 دستهای از قوانین را نشان می‌دهد که مجموعه های مربع های 2 در 2 را به یک خانه روشن یا خاموش تبدیل می‌کند.



شکل 7 این دسته از قوانین برای شبکه‌های مربعی طراحی شده که خواص تراوش شبکه را در جهت عمودی حفظ می‌کند.

در واقع هرگاه در شبکه های کوچکتر 2 در 2 تراوش در جهت عمودی اتفاق بیافتد آنرا با یک خانه روشن و هنگامی که تراوش نباشد آنرا با خانه خاموش جایگذین می‌کنیم. اگر یک شبکه‌ی L در L که در جهت عمودی تراوش دارد را با این قانون به یک شبکه‌ی (L-b) در (L-b) تبدیل کنیم همچنان در جهت عمودی تراوش خواهد داشت. البته این جمله همیشه صحیح نیست. فرض کنید ک یک شبکه دارای دو خوشه‌ی بسیار بزرگ است که در صورتی که با یک خوشه‌ی افقی به یکدیگر متصل می‌شوند تشکیل خوشه بینهایت می‌دهند. این شبکه تحت این تبدیل ممکن است ارتباط حاصل از خوشه افقی و درنتیجه خاصیت تراوش را از دست بدهد. مثالهای دیگری نیز میتوان در نظر گرفت که با این قوانین خوشهی بینهایتی تولید شود که در شبکهای اصلی وجود نداشته است. اما در شبکه های بسیار بزرگ و در تقریب ترمودینامیک این اثرات بسیار موضعی است و مقدار بحرانی تروا‌ش قابل محاسبه است.

اگر دسته قوانین نشان داده شده در شکل 7 را پشت سر هم بر روی شبکه های مختلف اعمال کنیم، شبکه هایی که تراوش دارند در نهایت به یک خانه روشن و شکبه هایی که تراوش ندارند به یک خانه خاموش تبدیل می‌شود. به شکل 8 نگاه کنید. در صورتی که از نقطه‌ای کمتر از حد بحرانی تراوش باز به هنجارش را شروع کنیم حتما به خانه خاموش یا احتمال 0 می‌رسیم و در صورتی که از نقطه‌ای بالاتر از یا برابر با حد بحرانی تراوش باز به هنجارش را شروع کنیم حتما به خانه‌ی روشن یا احتمال 1 می‌رسیم. به رفتاری که شکل 8 نمایش می‌دهد جریان باز به هنجارش گفته می‌شود که با این قوانین دارای 3 نقطه ثابت[10] است که دو تا از آنها یعنی 0 و 1 جاذب هستند.



شکل 8 جریان باز به هنجارش را برای قانون باز به هنجارش شبکه مربعی نشان میدهد. در این باز به هنجارش 3 نقطهی ثابت وجود دارد که دوتای آن جاذب هستند یعنی 0 و 1.

[8] Real Space Renormalization Group
[9] Transformation
[10] Fixed point

بیشتر بدانیم:

شکلهای این بخش با استفاده از کُد های موجود در وب گاه http://openscourcephysics.org تولید شده است. در این وب گاه کدهای بسیار جالبی برای شبیه سازی مسایل مختلف فیزیک یافت میشود.

برای آشنایی بیشتر با مبحث تراوش کتاب "Introduction to percolation theory" نوشته ی Dietrich Stauffer و Ammon Aharony مرجعی بسیار کامل و مناسب است. زبان کتاب بسیار روان و آلوده به طنز است که باعث جذابیت بیشتر کتاب میشود با این وجود چیزی از گفتار علمی و دقیق کتاب نمیکاهد. روند پیشرفت کتاب بسیار با دقت انتخاب شده است و خواننده را به آرامی از دنیای زیبا و ساده ی تراوش به محاسبات پیچیده ی آن میبرد. با وجود اینکه کتاب کمی قدیمی است و بیش از 20 سال از چاپ اول آن میگذرد ولی به دلیل پوشش وسیع موضوع به مرجعی برای این مبحث تبدیل شده است.

برای آشنایی بیشتر با آلگوریتم هوشن-کوپلمن می‌توانید مقالات زیر را مطالعه کنین:

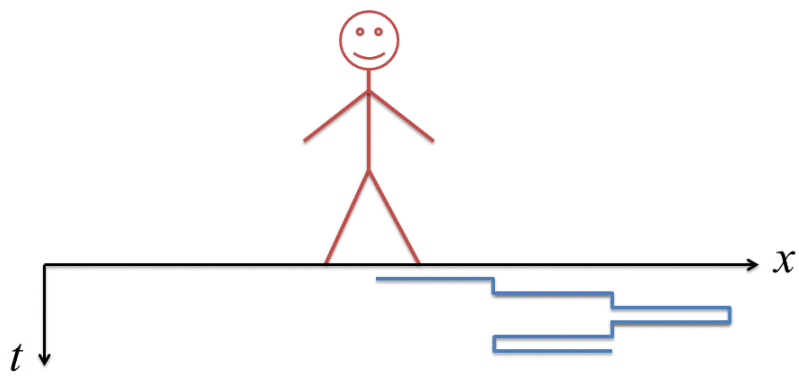
[1] "Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm", J. Hoshen and R. Kopelman, Phys. Rev. B 14, 3438 – Published 15 October 1976. DOI:https://doi.org/10.1103/PhysRevB.14.3438

[2] "Percolation and cluster distribution. II. layers, variable-range interactions, and exciton cluster model", J. Hoshen, R. Kopelman, and E. M. Monberg, 1978-09. DOI:http://dx.doi.org/10.1007/BF01011724

[3] "Recent advances in percolation theory and its applications", Abbas AliSaberi, Physics Reports Volume 578, 24 May 2015, Pages 1-32. https://doi.org/10.1016/j.physrep.2015.03.003

5. ولگشت[11]

ول گردی را در نظر بگیرید که در لحظه‌ی t_0 در مکان x_0 است. این ول گرد در بازه‌های زمانی ثابت که برابر با واحد زمان فرض می‌کنیم به طور کاملا کتره‌ای قدمی به سمت چپ یا راست بر می‌دارد. برای سادگی فرض کنید که ول گرد یک سکه دارد که در هر قدم با انداختن سکه و شیر یا خط بودن آن انتخاب می‌کند که قدم خود را در کدام جهت بردارد. برای سادگی فرض کرده‌ایم که ول گرد فقط در یک بعد قدم می‌زند ولی در ادامه مسئله را به ابعاد بالا تر تعمیم می‌دهیم. سوال این است که در زمان t ول گرد کجاست. بدیهی است که ساختار تصادفی مسئله امکان پاسخ گویی دقیق به این سوال را نمی‌دهد ولی می‌توان احتمال حضور او در هر نقطه از فضا و در هر زمان خاص را بدست آورد.



شکل 9 حرکت یک ولگرد در یک بعد

5.3. ولگشت یک بعدی

فرض کنید که در مثال بالا احتمال این که ول گرد در هر قدم به سمت راست برود p و احتمال رفتن به چپ $q = 1 - p$ باشد و در لحظه $t_0 = 0$ در مبدا مختصات، $x_0 = 0$، باشد. جدول زیر احتمال حضور این ولگرد در مکان‌های دیگر و در زمان‌های بعدی را می‌دهد.

جدول 2 احتمال حضور ولگرد در نقاط شبکه در زمان های متفاوت. در هر قدم ولگرد با احتمال p به سمت راست و با احتمال q به سمت چپ می‌رود.

x									t
-4	-3	-2	-1	0	1	2	3	4	
0	0	0	0	1	0	0	0	0	0
0	0	0	q	0	p	0	0	0	1
0	0	q^2	0	$2pq$	0	p^2	0	0	2

[11] Random Walk

0	q^3	0	$3pq^2$	0	$3pq$	0	p^3	0	3

برای ولگرد یک بعدی به راحتی می‌توان این احتمال را در حالت کلی محاسبه کرد. احتمال اینکه بعد از پیمایش N قدم، N_+ قدم به سمت راست و $N_- = N - N_+$ قدم به سمت چپ بردارد از بست دوجمله ای بدست می‌آید.

$$\boldsymbol{P(N_+;N) = \frac{N!}{N_+!N_-!}\, p^{N_+} q^{N_-}} \qquad (1)$$

بنابر این احتمال اینکه این ولگرد در لحظه $t = N\tau = (N_+ + N_-)\tau$ در $x = (N_+ - N_-)l$ قرار داشته باشد به راحتی با جایگزینی مقدار x و t به جای تعداد قدم‌ها بدست می‌آید. در اینجا l و τ به ترتیب طول قدم و واحد زمان برای برداشتن قدم‌های این گشت هستند. بنا به قضیه حد مرکزی[12] می‌دانیم که در حد N های بزرگ به یک تابع توزیع گوسی میل می‌کند. درنتیجه احتمال حضور ولگرد در زمان و مکان با رابطه‌ی

$$\boldsymbol{P(x;t) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\langle x\rangle)^2}{2\sigma^2}}} \qquad (2)$$

داده می‌شود که در رابطه بالا $\langle x \rangle$ و σ مقادیر متوسط آنساملی مکان و انحراف میعار آن هستند و هر دو تابعی از زمان هستند. برای محاسبه می‌توان از رابطه‌ی برگشتی استفاده کرد.

$$\boldsymbol{x(t) = x(t-\tau) + al} \qquad (3)$$

که در اینجا a یک متغیر کاتوره‌ای است که با احتمال p مقدار 1+ و با احتمال q مقدار -1 را می‌گیرد. از طرفین رابطه بالا متوسط می‌گیریم

(4)

$$\begin{aligned} \langle \boldsymbol{x(t)} \rangle &= \langle \boldsymbol{x(t-\tau)} \rangle + \langle \boldsymbol{a} \rangle \boldsymbol{l} \\ &= \langle \boldsymbol{x(t-\tau)} \rangle + (\boldsymbol{p-q})\boldsymbol{l} \\ &= \langle \boldsymbol{x(t-2\tau)} \rangle + (\boldsymbol{p-q})\boldsymbol{l} + (\boldsymbol{p-q})\boldsymbol{l} = \frac{\boldsymbol{l}}{\boldsymbol{\tau}}(\boldsymbol{p-q})\,\boldsymbol{t} \end{aligned}$$

به روشی مشابه می‌توانیم انحراف میعار را نیز محاسبه کنیم که نتیجه می‌شود

[12] Central Limit Theorem

$$\sigma^2 = < x^2 > - < x >^2 = \frac{4l^2}{\tau} pq\, t. \quad (5)$$

.4.1	رابطه 5 را اثبات کنید.
تمرین	

.4.2	– یک برنامه برای ولگشت یک بعدی بنویسید و صحت روابط 5 و 4 را برای چند مقدار مختلف p بررسی کنید. یکی از این مقادیر $p = 1/2$ باشد.
تمرین	

5.4. معادله پخش برای ولگشت

مسئله‌ی ولگشت شاید در نگاه اول چیزی شبیه یک بازی به نظر برسد ولی در فیزیک سرو کله‌ی آن در بسیاری از مسائل دیگر پیدا می‌شود. به زبان دیگر بسیاری از مسئله‌های فیزیک و حتی در علوم دیگر در کلاس عمومی ولگشت می‌نشینند. یکی از مهم‌ترین و پرکاربردترین این مسئله‌ها، پدیده‌ی پخش است. در پخش، هر ذره به طور کاتوره‌ای و به خاطر ضربه‌های گرمایی (افت و خیز حرارتی) یک حرکت تصادفی می‌کند. پس بدیهی است که پخش را می‌توان ولگشتی در فضا تصور کرد. در اینجا برای بدست آوردن احتمال ذره در فضا راه دیگری را در پیش می‌گیریم و برای تابع احتمال $P(x;t)$ معادله تحولی می‌نویسیم که به معادله‌ی مادر[13] معروف است. ذره فقط در صورتی می‌تواند در لحظه‌ی t در نقطه ی x باشد که در قدم قبل در یکی از همسایگی‌های این نقطه باشد. برای سادگی فرض می‌کنیم که $p = q = \frac{1}{2}$، در این صورت

$$P(x;t) = \frac{1}{2}\big(P(x-l;t-\tau) + P(x+l;t-\tau)\big). \quad (6)$$

از طرفین رابطه بالا مقدار $P(x;t-\tau)$ را کم می‌کنیم و آنرا بر $\tau\, l^2$ تقسیم می‌کنیم.

[13] Master equation

$$\frac{P(x;t)-P(x;t-\tau)}{\tau}=\frac{l^2}{2\tau}\frac{\frac{P(x+l;t-\tau)-P(x;t-\tau)}{l}-\frac{P(x;t-\tau)-P(x-l;t-\tau)}{l}}{l} \quad (7)$$

در حد پیوسته، یعنی وقتی $\tau \to 0$ و $x \to 0$ ولی $D = l^2/2\tau$ محدود و غیر صفر است، رابطه بالا به رابطه معروف پخش تبدیل می‌شود،

$$\frac{\partial P(x;t)}{\partial t} = D\ \frac{\partial^2 P(x;t)}{\partial x^2} \quad (8)$$

که دارای پاسخ آشنای گوسی است که در قسمت قبل بدست آوردیم. یکی از نکات آموزنده‌ی این بحث این است که حتی برای مسایلی که به طور صریح تصادف در فرآیند آن نقش دارد نیز می‌توان به معادلاتی رسید که قابل حل با روش‌های معمول عددی و بدون نیاز به استفاده از آلگوریتم‌های تصادفی باشد.
مقایسه ی روابطه 5 و 8 می‌تواند ظریب پخش ول گشت را تعریف کند.

$$\sigma^2 = 2Dt \quad (9)$$

معادله 8 معادله پخش است که نه تنها در توصیف پدیده های فیزیکی بسیار معادله‌ی مهمی است بلکه در علوم اجتماعی و اقتصادی نیز بسیار پر کاربرد است. سرعت پخش بیماری یا شایعه در یک اجتماع یا پخش ثروت در یک مجموعه تنها چند مثال ساده از کاربرد این رابطه است. می‌توانیم مثالی ملموستر را بررسی کنیم. فرض کنید که در یک بعد حرکت ول گشت انجام می‌دهید. اگر در هر ثانیه یک متر قدم بردارید، پس از گذشت یک ساعت شما 3600 قدم برداشته‌اید و طبق رابطه‌ی 9 احتمالا در فاصله 60 متری مکان اولیه خود قرار گرفته‌اید. در صورتی که جهتمند به سمت این نقطه حرکت می‌کردید کافی بود که یک دقیقه به سمت نقطه‌ی مورد نظر قدم بر می‌داشتید.

5.5. ولگشت با تله

فرض کنید که در یک مسئله ساده‌ی یک بعدی ولگشت، شرایط مرزی جاذب باشند. یعنی اینکه درصورت رسیدن ولگرد به مرزها، در آنجا متوقف می‌شود. دراین مدل می‌توان مرزها را تله‌هایی فرض کرد که باعث نابودی ولگرد می‌شوند.

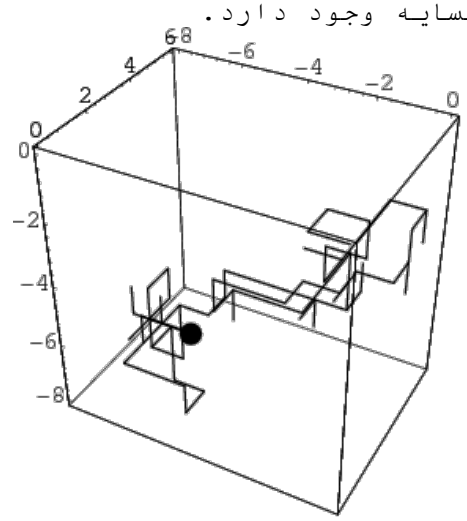
.4.3	– شرایط مرزی جاذب برای مسئله قبل قرار دهید و برنامه را تا زمان به دام افتادن ولگرد در تله ها ادامه دهید. زمان متوسط زندگی ولگرد در این شبکه را با متوسط گیری بر روی تعدا د زیادی اجرا محاسبه کنید. فرض کنید که شبکه دارای 20 خانه است. – بستگی متوسط عمر ولگرد به مکان اولیه آنرا نشان دهید.
تمرین	

یک راه برای حل مسئله بالا استفاده از الگوریتم سرشماری به جای شبیه سازی تصادفی است. جدولی مانند جدول (1) را برای محاسبه‌ی احتمال حضور در نقاط مختلف تولید کنید. به دلیل وجود داشتن تله در انتهای شبکه احتمال‌هایی که به این خانه‌ها وارد می‌شوند دیگر در سیستم منتشر نمی‌شوند. در هر زمان مجموع کل احتمال‌ها باید برابر واحد باشد و مجموع احتمال‌ها بدون در نظر گرفتن مقادیر مربوط به تله‌ها احتمال زنده ماندن ولگرد تا آن زمان را می‌دهد. همچنین جمع احتمال‌های دوخانه‌ی تله احتمال مرگ را می‌دهد. با توجه به داشتن این مقدار می‌توان متوسط عمر ولگرد را محاسبه کرد.

.4.4	با استفاده از روش سرشماری مسئله قبل را حل کنید و نتیجه دو روش را با هم مقایسه کنید.
تمرین	

5.6. ولگشت در ابعاد بالاتر

مشابه حرکت در یک بعد می‌توان ولگشت را در ابعاد بالاتر نیز تعریف کرد. در این حالت متحرک در هر قدم تعداد بیشتری حق انتخاب دارد. البته تعداد همسایگان نه تنها به بعد فضایی که به نوع شبکه نیز بستگی دارد. برای سادگی یک شبکه ساده مربعی d را فرض کنیم. بطور مثال در یک شبکه مربعی، چهار همسایه و در یک شبکه مکعبی ساده 6 همسایه وجود دارد.



شکل 10 حرکت ولگشت در فضای سه بعدی و بر روی شبکه مکعبی ساده

حرکت در این شرایط را می‌توان به d حرکت یک بعدی تجزیه کرد. هر کدام از این حرکت‌ها دقیقا یک حرکت ولگشت یک بعدی هستند. باز هم برای سادگی فرض می‌کنیم که احتمال رفتن به هر یک از جهات برابر باشد. پس برای حالت سه بعدی برای حرکت بر روی هر بُعد داریم:

$$tD\,2 = > {}^{2}z <=> {}^{2}y <= > {}^{2}x < \qquad (10)$$

$$< \boldsymbol{r^2} > = < \boldsymbol{x^2} + \boldsymbol{y^2} + \boldsymbol{z^2} > = \mathbf{3} \times \mathbf{2}\ \boldsymbol{Dt} \qquad \mathbf{(11)}$$

در حالت کلی می‌توان رابطه را برای هر بُعد به صورت

$$< \boldsymbol{r^2} > = \mathbf{2}\boldsymbol{d}\ \boldsymbol{Dt} \qquad \mathbf{(12)}$$

نوشت. این رابطه نشان می‌دهد که رفتار مقیاسی شعاع ژیراسیون با زمان برای ول گشت ساده بستگی به بُعد ندارد و همواره داریم

$$\boldsymbol{R_g} = \sqrt{< \boldsymbol{r^2} >} \sim \boldsymbol{t^{\nu}} \qquad \mathbf{(13)}$$

که ν نمای بحرانی ولگشت است و در ولگشت ساده مستقل از بُعد است و همواره برابر با $\frac{1}{2}$ است.

.4.5	
تمرین	– یک برنامه برای ولگشت دو بعدی بر روی شبکه مربعی بنویسید. فرض کنید که احتمال قدم بر داشتن در تمام جهت‌ها برابر است. – صحت رابطه‌ی 12 را برای این ولگرد بررسی کنید.

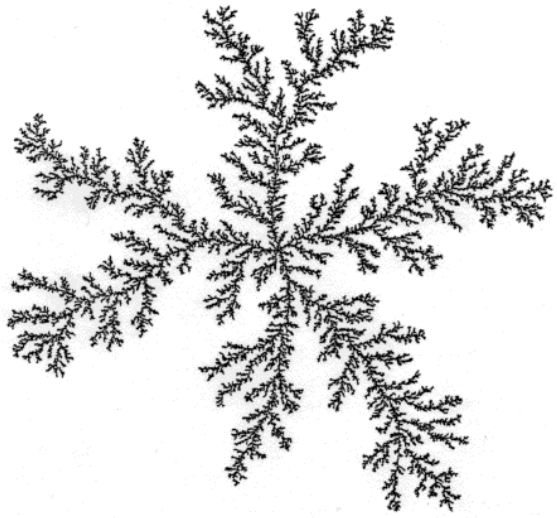
5.7. تجمع پخش محدود[14]

در اینجا به موضوع رشد که در فصل قبل با آن آشنا شدیم بر می‌گردیم. دسته گسترده‌ای از فرآیندهای رشد به فرآیندهای تجمعی معروف هستند. در این گونه فرآیندها، رشد از یک بذر اولیه شروع می‌شود و با تجمع ذرات در اطراف آن یک خوشه تشکیل می‌شود. این خوشه با ادامه فرآیند رشد می‌کند. شکل خوشه و خواص هندسی و فیزیکی آن بستگی به نوع فرآیند رشد دارد.
دسته‌ی خیلی مهمی از این گونه فرآیندهای رشد به فرآیندهای "تجمع با پخش محدود" معروف هستند. در این گونه فرآیندها ذرات در محیط یک حرکت پخشی یا ولگشت آزاد دارند. این ذرات در صورتی که طی حرکت پخشی خود به بذر اولیه یا خوشه‌ی تشکیل شده برسند به آن می‌چسبند و متوقف می‌شوند. این پدیده را

[14] Diffusion limited Aggregation (DLA)

می‌توان مشابه ولگشت در حضور تله فرض کرد با این تفاوت که تله دراینجا خود دینامیک دارد.
برای شبیه سازی این فرآیند می‌توان از برنامه‌ای که برای ولگشت تولید کردید استفاده کرد.

.4.6	برنامه ای برای تولید خوشه‌های تجمع پخش محدود در دو بعد و با یک بذر خطی تهیه کنید. – شرایط اولیه خوشه (بذر) را خطی افقی به طول 200 در نظر بگیرید. – ولگردی را از فاصله‌ای بالاتر از خوشه رها کنید و اجازه دهید که در صفحه گشت کند و در صورت اتصال به خوشه به آن بچسبد. – فرآیند را بر روی نمایشگر نمایش دهید. از کد رنگ برای تصویر کردن دینامیک فرآیند استفاده کنید.
تمرین	



شکل 11 یک خوشه تولید شده در فرآیند تجمع با پخش محدود

یکی از مشخصه‌های مهم فرآیند تجمع پخش محدود رفتار رقابتی در رشد خوشه‌هاست. این رقابت کاملا ناپایدار است. در ابتدا تمام نقاط (بذر خطی) با هم یکسان هستند و هم ارتفاع. به محض اینکه یک ذره به نقطه‌ای بچسبد و ارتفاع آن نقطه بیشتر شود، شانس این نقطه برای جذب ذرات دیگر از نقاط دیگر بذر بیشتر خواهد شد.

شکل 12 خوشه تجمع با پخش محدود با بذر خطی. شکل سمت راست: شبه فسیل‌های اکسید منیزیم بر روی سنگی از کوهستان درکه در شمال تهران.

شکل سمت چپ: خوشه تولید شده با شبیه سازی کامپیوتری

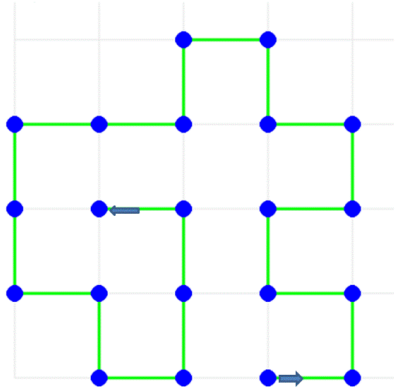
شکل بالا عکسی از یک خوشه واقعی که بر شکاف سنگها تولید می‌شود را با تصویری شبیه سازی شده از این فرآیند مقایسه می‌کند. در تصویر شبیه سازی می‌بینید که حتی در مراحل نهایی شبیه سازی نیز این امکان برای ذرات پخشی وجود دارد که از بین درختچه های بزرگتر عبور کرده و خود را به درختچه های کوچکتر برسانند، هر چند این اتفاق نادر و با احتمال کمی صورت می‌گیرد. این خاصیت رقابتی در تولید درختچه ها در این فرآیند دلیل اصلی رفتار مقیاسی درختچه ها و ساختار فراکتالی آنهاست.

5.8. ول گشت خود پرهیز[15]

تغییرات جزیی رفتار مقیاسی ول گشت را تغییر نمی‌دهد. به طور مثال تغییر نوع شبکه یا حتی بعد می‌تواند هندسه‌ی ولگشت را تغییر دهد ولی مقدار ν تحت تاثیر این گونه تغییرا ت ناوردا است. ولی وجود دارد تغییراتی که می‌توانند رفتار مقیاسی را تحت تاثیر قرار دهند. در این میان یکی از مهم‌ترین تغییرات اختصاص حافظه‌ای بلند برد برای ولگرد است که اجازه ورود به خانه هایی که قبلا دیده است را به او ندهد. به این ترتیب ول گرد در طی مسیر تمام نقاطی که می‌گذرد را به خاطر می‌سپارد و از ورود به آنها اجتناب می‌کند.
این تغییر به ظاهر جزیی تاثیر زیادی در رفتار مقیاسی ولگشت می‌گذارد. از سوی دیگر خواهیم دید که از نظر محاسباتی نیز شبیه سازی را بسیار پیچیده و مشکل می‌کند.
یکی از تفاوت‌های آشکار بین ولگشت معمولی و خود پرهیز وابستگی شدید به بُعد در مدل خود پرهیز است. برای مثال اگر یک سیستم یک بعدی شروع کنیم کاملا بدیهی است ولگرد فقط در قدم اول امکان انتخاب جهت را دارد. بعد از اولین انتخاب، قید خود پرهیزی راهی به جز ادامه مسیر در همان جهت را به ولگرد

[15] Self-Avoiding Random Walk (SAW)

نمی‌دهد. در نتیجه در بعد $d = 1$، $v = 1$ است. ولی داستان در $d = 2$ کار به این سادگی نیست. اینجا مسیرهای بیشماری برای ولگرد وجود دارد.
اگر به برنامه‌ای که در قسمت قبل برای ول گشت تهیه کرده‌اید حافظه‌ای برای نقاطی که قبلا ولگرد از آن عبور کرده اختصاص دهید می‌توانید از آن برای تولید گشت‌های خود پرهیز استفاده کنید. ولی به دلیل محدودیت خود پرهیزی تعداد مسیرها خود پرهیز بسیار کمتر از ولگشت ساده است. به همین دلیل در این برنامه تولید مسیرهای با طول های بزرگ تقریبا ناممکن است. ولگرد به کرات در مسیرهایی گیر می افتد که امکان پیش روی ندارد.



شکل 13 یک مسیر نوعی برای ول گشت خود پرهیز که بعد از 12 قدم به بن بست رسیده است.

به این دلیل امکان تولید تعداد قابل توجهی گشت در طول‌های بلند با روش گشت تصادفی وجود ندارد و به این طریق نمی‌شود مقدار دقیقی برای v بدست آورد. یکی از روش‌هایی که می‌توان به عنوان جایگزین پیشنهاد داد روش سرشماری تمام مسیرهاست. در این روش متحرک بعد از رسیدن به بن بست تعدادی قدم به عقب برگشته تا راه جدیدی پیدا کند. با تکرار این روش متحرک می‌تواند مسیر خود را به انجام برساند. این درست است که تعداد مسیرهای گشت خود پرهیز بسیار کمتر از ولگشت ساده است، ولی این تعداد هنوز آنقدر زیاد است که آلگوریتم سرشماری را یک آلگوریتم NP- پیچیده کند و امکان حل مسئله به این طریق را سلب کند.

.4.7	– برنامه‌ای برای تولید تمامی گشت‌های خود پرهیز بر روی یک شبکه مربعی دو بعدی تهیه کنید. – منحنی تعداد گشت‌های ممکن بر حسب N را رسم کنید. – برای یک ولگشت آزاد به طول N بر روی این شبکه تعداد راه‌های ممکن 4^N است. نسبت گشت های خودپرهیز بر گشت های آزاد را بر حسب N رسم کنید.
تمرین	

اختلاف اصلی ولگشت خود پرهیز با گشت‌های آزاد عدم وجود حلقه بسته در این گشت‌هاست. در نتیجه حل این مسئله معادل با امکان شماردن تعداد حلقه‌های بسته

در شبکه است. این مسئله در دو بُعد حل دقیق دارد. به این دلیل در $d = 2$ مسئله حل دقیق دارد و مقدار $\nu = 3/4$ است. در ابعاد بالاتر از چهار هم به دلیل پایین بودن احتمال برخورد گشت با خودش نقش این حلقه ها خیلی مهم نیست و اختلاف زیادی میان ولگرد خود پرهیز و ساده وجود ندارد و در حد ترمودینامیکی می‌توان نشان داد که برای $d \geq 4$ مقدار ν برابر با مقدار آن برای ولگشت ساده، $\nu = 1/2$ است. اما در $d = 3$ مسئله حل دقیقی ندارد و تمام اطلاعات ما از شبیه سازی‌های کامپیوتری و یا روش های تقریبی می آید. به طور مثال تقریبی که به تقریب فلوری[16] معروف است

$$\boldsymbol{\nu = \frac{3}{d+2}} \tag{14}$$

مقدار $\nu = 0.6$ را به ازای $d = 3$ می‌دهد که خیلی به مقدار بدست آمده از شبیه سازی ها $\nu = 0.58$ نزدیک است.

5.9. حذف حلقه ها

این بخش باید اضافه شود. در حال حاضر تمرین زیر فقط برای یاد اوری به شخص خود من است نه برای انجام توسط دانشجو.

تمرین: تعداد حلقه های که برای رسیدن به یک گشت N قدمی باید حذف کنید را بر حسب N بیابید.

5.10. ول گشت جهت دار[17]

قطره بارانی را در نظر بگیرید که در حال سقوط است. در حین سقوط تحت تاثیر ضربه‌های تصادفی از مولکول‌های هوا و افت و خیزهای جریان هوا این قطره حرکات تصادفی عرضی دارد. این گونه جابجایی‌ها که در یک راستا حرکت جهت دار و بدون امکان بازگشت است و در جهت‌های دیگر تصادفی است. این گونه گشت‌ها بنا به تعریف امکان قطع خود را ندارد و خود پرهیز می‌باشند ولی اگر بیشتر دقت شود می‌توان آنرا به دو حرکت قاعده‌مند رو به جلو و یک حرکت ول گشت ساده در راستای عمود بر جهت محور سقوط تجزیه کرد.
اگر مکان یک ولگشت ساده را بر حسب زمان رسم کنیم نموداری که حاصل می‌شود با توجه به جهت دار بودن حرکت در زمان یک ولگشت جهت دار است. این گشت یک فراکتال خود آفین است. به این معنی که برای دیدن رفتار خود تشابهی در این فراکتال باید تابع مقیاسی دارای ضرایب مقیاس متفاوت در راستاهای مکان و زمان باشد.

5.11. ول گشت محافظه کار[18]

ول گشت ساده و پرهیز کار دو سوی طیف گسترده‌ای از گشت‌های با حافظه محدود هستند. نکته مهم این است که هر گونه گشتی با حافظه محدود در طول‌هایی قابل

[16] Flory
[17] Directed Random Path
[18] Persistent Random Walk

مقایسه با حافظه شاید رفتاری مشابه ولگشت خود پرهیز داشته باشد یا حتی رفتار تصادفی آن قابل مشاهده نباشد ولی در حد طول‌های بسیار بلند حتما در کلاس جهانشمولی ول گشت ساده خواهد بود. یک مثال ساده ول گشت با تمایل بر ادامه مسیر در قدم قبل است. این ولگشت را محافظه کار می‌نامیم. به این معنی که هر گاه قدمی در راستای خاصی برداشت احتمال ادامه مسیر در همین راستا بیش از دیگر مسیرهاست. هرچند این تمایل بیشتر باشد ولگرد مسیرهای مستقیم طولانی تری را طی خواهد کرد تا شانس تغییر مسیر بیابد. هرچه این تمایل کمتر باشد رفتار گشت تصادفی‌تر است. پس می‌توان برای گشت طولی به نام طول پایسته تعریف کرد که معیاری از متوسط تداوم مسیر توسط ولگرد است. در طول‌های بسیار بلند تر از طول پایسته ولگشت مشابه ولگشتی ساده با قدم‌هایی با طول پایسته است.

5.12. مدلهای پیوسته

تا کنون فرض کرده‌ایم که ول گشت محدود به نقاط یک شبکه است. این نکته هم می‌تواند به دلایل شباهت با بعضی از مسایل باشد و هم می‌تواند ناشی از تمایل ما به گسسته سازی باشد که در فصل ؟؟ قبل توضیح داده شد. ولی در واقعیت این امکان برای ولگشت وجود دارد که در محیطی پیوسته اتفاق بیافتد. این پیوستگی هم می‌تواند در راستای حرکت باشد یا در طول قدم‌ها و یا هردو. حتی فاصله زمانی قدم برداشتن ها هم می‌تواند پیوسته باشد و از یک تابع توزیع تبعیت کند. نکته جالب توجه این است که چنین تعمیمی رفتار مقیاسی ولگرد را عوض نمیکند و فقط کافی است در تمام روابط بالا در این فصل هرجا که از طول قدم یا زمان قدم استفاده شده مقدار متوسط این کمیت‌ها را قرار دهیم.

شکل 14 حرکت یک باکتری ای-کلای[19] در محیط آبی مثالی از یک گشت پیوسته است

5.13. مثالهایی از ولگشت

[19] e-colie

1- پخش

یکی از آشناترین مثال‌ها برای ول گشت، پدیده‌ی پخش است. در بالا تصویری از حرکت پخشی یک باکتری نشان داده شده است. این گونه حرکت‌ها بخوبی در کلاس ول گشت می‌نشینند. ولی مثال‌های جالب دیگری نیز برای ولگشت وجود دارد.

2- بازارسهام

رفتار قیمت‌ها در بازار سهام یک گشت تصادفی را تداعی می‌کند. در حقیقت رفتار سهام با ولگشتی که تا کنون با آن آشنا شدید یک تفاوت اساسی دارد. در این فرآیند طول قدم‌های با مقدار آنها متناسب است. احتمال آنکه قیمت کالایی به ارزش 100 تومان جهشی در قیمت به اندازه 10 تومان (بالا یا پایین) داشته باشد با احتمال تغییر قیمت کالای با ارزش 100 هزار تومان به اندازه ی 10 هزار تومان برابر است. این گونه ولگشت را ولگشت هندسی [20] می‌نامند.

3- فیزیک پلیمرها

یکی از پرکاربردترین و جالب ترین مثال‌ها برای ولگشت خود پرهیز ساختار هندسی پلیمرها است. یک پلیمر از زنجیره‌ای از اتم‌های مشابه که بطور تناوبی در کنار هم قرار گرفته‌اند تشکیل می‌شود. این زنجیره‌ها در محیط محلول و در دمای غیر صفر ساختارهای تصادفی به خود می‌گیرند. اگر یک ساختار را در یک لحظه نگاه کنید این ساختار به مشابه یک گشت در زمان است. لازم به توجه است که در اینجا این گشت در زمان انجام نشده بلکه در مکان وجود دارد. یعنی به جای اینکه مکان یک ولگرد در زمان به یک گشت تعبیر شود، کل ساختار پلیمر مشابه این گشت است. به دلیل عدم امکان قرار گرفتن دو اتم به طور هم زمان در یک مکان این گشت را خود پرهیز می‌کند.

در حقیقت تقریبی که در بالا برای نمای ν در ولگشت خود پرهیز معرفی شد، اولین بار توسط فلوری برای بررسی خواص مقیاسی پلیمرها ارایه شده است. در این تقریب او عبارت‌هایی برای انرژی و آنتروپی یک پلیمر در محلول ارایه می‌کند و کمینه کردن انرژی آزاد رابطه بین طول ژیراسیون و طول پلیمر بدست می‌آورد.

بیشتر بدانیم:

در مورد ول گشت منابع بسیار وجود دارد. تقریبا هیچ کتاب آماری پیدا نمی‌کنید که در این باره صحبت نکرده باشد. برای آشنایی با بعضی کاربردهای آن در بیوفیزیک و یا فیزیک پلیمرها نیز می‌توانید به کتاب Random Walks in Biology نوشته‌ی Howard C. Berg یا کتاب Scaling concepts in polymer physics نوشته‌ی P.G. de Gennes مراجعه کنید. کتاب Biological Physics نوش4ته‌ی Philip Nelson به موضوع پخش و مثال‌های آن تمرکز خوبی دارد.

[20] Geometrical Random Walk

6. تولید اعداد کتره‌ای[21]

در تمام فصل‌های گذشته با پدیده‌هایی آشنا شدید که رفتار کاتوره‌ای داشتند. همچنین در شبیه سازی آنها نیز مولد اعداد کتره‌ای نقش اساسی را در کُد و شبیه سازی پدیده داشته‌اند. احتمالا شما برای انجام تمرین‌های خود از مولدی که نرم افزار مورد استفاده در اختیارتان قرار می‌داده استفاده کرده‌اید. ولی فراموش نکنید که کامپیوتر شما یک ماشین کاملا منطقی است و تصادف و احتمال در آن جایی ندارد. در حقیقت این جای خوشبختی است که این ماشین همیشه برای یک سری فرآیندهای منطقی پاسخی قاطع و تکرار پذیر دارد. ولی چگونه می‌توان با چنین ماشینی اعداد کتره‌ای ساخت؟ جواب ساده است: **نمی‌توان.** پس این مولدها چه چیزی به ما تحویل می‌دهند و ما چگونه بر خروجی برنامه‌های خود اعتماد کنیم وقتی می‌دانیم که این مولدها ما را گول زده‌اند و یک رشته اعداد تکرار پذیرکه با یک دیگر رابطه منطقی دقیقی نیز دارند را بجای اعداد کتره‌ای جا زده اند؟ برای پاسخ این سوال در ابتدا تعریفی از یک رشته‌ی کاتوره‌ای خوب باید داشته باشیم.

6.3. تابع توزیع احتمال[22]

یک مولد اعداد کتره‌ای، رشته‌ای از اعداد را به ما تحویل می‌دهد و ادعا می‌کند که این رشته مجموعه مناسبی است برای آنکه آنها را کتره‌ای بدانیم. ولی این رشته چه خصوصیتی باید داشته باشد. اولین خاصیتی که انتظار می‌رود تبعیت از تابع توزیع مورد نظر است. به طور مثال اگر شما تابعی در زبان برنامه نویسی خود دارید که ادعا می‌کند اعداد کتره‌ای با توزیع یکنواخت بین صفر و یک تولید می‌کند، ساده‌ترین انتظاری که می‌توان داشت این است که در رشته‌ای که این مولد به ما تحویل می‌دهد احتمال ظهور عددی بیش از دیگری نباشد.

.6.1 تمرین	– در زبان برنامه نویسی که استفاده می‌کنید تابع مولد اعداد کتره‌ای که اعداد صحیح بین 0 تا 9 تولید می‌کند را صدا بزنید. این تابع را در حلقه‌ای به طول N قرار دهید و منحنی فرآوانی اعداد خروجی را رسم کنید. اگر این مولد سالم باشد انتظار دارید هر یک از اعداد $N/10$ بار ظاهر شود. آیا این گونه است؟ – نشان دهید که انحراف نسبی از عدد بالا با جذر عکس N به سمت صفر می‌رود. یعنی $$\frac{\sigma}{N} \sim \frac{1}{\sqrt{N}}$$ – آیا شباهتی میان این تمرین و تمرین ول نشست مشاهده می‌کنید؟

[21] Random Number Generator
[22] Probability Distribution Function

در ادامه خواهیم دید که می‌توان مولدهایی با تابع توزیع غیریکنواخت نیز داشت که امتیازی برای استفاده از مولد به حساب می‌آید. نکته مهم این است که مولد باید بخوبی تابع توزیعی که ادعا می‌کند از آن تبعیت می‌کند را تولید کند.

6.4. همبستگی[23]

وقتی یک تاس را به بالا پرت می‌کنید انتظار دارید که در صورت سلامت تاس، احتمال نشستن هر یک از اعداد 1 تا 6 برابر باشد. به عبارت دیگر تابع توزیع ظهور اعداد باید یکنواخت باشد. ولی این تنها انتظاری نیست که از یک تاس سالم می‌رود. نکته دیگری نیز وجود دارد که به همین اندازه مهم است. انتظار می‌رود که مقداری که تاس در هر پرتاب نشان می‌دهد از مقدارهایی که قبلا نشان داده است مستقل باشد[24]. یعنی همبستگی بین اعداد وجود نداشته باشد و تاریخچه تاثیری بر آینده نداشته باشد.

.6.2	تمرین قبل را تکرار کنید با این تفاوت که این دفعه از رشته اعداد تصادفی فقط اعدادی را بردارید که عدد قبلی آنها 4 بوده باشد. مثلا اگر اعداد تولید شده، 1 4 6 7 8 9 4 4 8 3 5 7 1 4 9 باشد،تابع توزیع اعداد قرمز را نمایش دهید.آیا تابع توزیع این اعداد یکنواخت است؟
تمرین	

6.5. تولید اعداد شبه کتره ای[25] با توزیع یک نواخت

دوباره به سوال اول این بخش بر می‌گردیم. چگونه یک ماشین کاملا منطقی مانند کامپیوتر می‌تواند اعداد کتره‌ای تولید کند؟ به هر حال نتیجه‌ی دو تمرین قبل ظاهرا در تایید کیفیت کتره‌ای بودن اعداد این ماشین‌هاست. در حقیقت رشته تولید شده توسط کامپیوتر اصلا کتره‌ای نیست بلکه یک رشته ی کاملا منطقی است ولی الگوریتم تولید این رشته توانایی تولید رشته‌ای را دارد که در ظاهر کتره‌ای به نظر بیاید. برای همین به اینها مولد اعداد **شبه کتره‌ای** می‌گویند. آلگوریتم‌های تولید اعداد شبه کتره‌ای انگشت شمار هستند. یکی از معروفترین ومتداول‌ترین این آلگوریتم‌ها روش استفاده از هم‌نهشتی است. این روش به دلیل عدم نیاز به حافظه زیاد و نیز به دلیل سرعت بالا در تولید اعداد، بسیار پر کاربرد است. در این روش رشته اعداد با یک مقدار اولیه که به بذر[26] معروف

---

[23] Correlation

[24] بعضی مواقع در زندگی روزمره با بی توجهی به این اصل، انتظار نامعقولی داریم. مثلا گفته می‌شود که مدتی است که اصلا 6 نیامده، پس این دفعه حتما می‌آید. این یعنی انتظاری وجود دارد که اگر مثلا در 30 پرتاب قبل 6 ظاهر نشده در پرتاب 31ام احتمال ظهور 6 کمی بیشتر از بقیه اعداد باشد.

[25] Pseudorandom

[26] seed

است شروع می‌شود. با یک معادله بازگشتی، دیگر عددهای رشته قدم به قدم تولید می‌شود. این برنامه بازگشتی به شکل

$$x_{n+1} = (a\, x_n + c) \mod m \tag{1}$$

است. در اینجا a ضریب، c جابجایی و m عامل هم‌نهشتی است و هرسه اعداد مثبتی هستند و به پارامترهای الگوریتم معروفند. بدیهی است که این رابطه ساده نمی‌تواند یک رشته‌ی کتره‌ای را تولید کند. اولین اشکالی که به این الگوریتم وارد است خاصیت تناوبی آن است. از آنجا که پارامترهای مدل ثابت هستند، یک رابطه منطقی میان x_n و x_{n+1} وجود دارد. در نتیجه اگر در رشته‌ی اعداد، عددی تکرار شود دنباله نیز تکرار خواهد شد.

از آنجا که x ها باقیمانده‌ی تقسیم بر m هستند، پس همواره $x_i < m$. پس در بهترین حالت دوره تناوب این رشته نمی‌متواند از m بزرگتر باشد. البته بسته به انتخاب مجموعه‌ی پارامترها، این تناوب می‌تواند بسیار کوچکتر نیز باشد. بنابراین کیفیت رشته اعداد تولیدی به شدت به انتخاب پارامترهای مدل بستگی دارد. داشتن دوره تناوب تنها اشکال این الگوریتم نیست. بلکه همبستگی‌های دیگری نیز در رشته‌های تولیدی می‌تواند کیفیت ظاهر کاتوره‌ای رشته را خدشه دار کند. روش همنهشتی برای تولید اعداد کتره‌ای به خوبی مطالعه شده است و مجموعه پارامترهایی برای تولید رشته پیشنهاد می‌شوند که نقاط ضعف را بهتر بپوشانند[27].

به طور خاص می‌توان نشان داد که اگر $m = 2^k$ باشد، در صورتی که c عددی اول باشد و $(a-1)$ مضربی از 4 باشد، دوره تناوب این رشته به مقدار بیشینه خود، یعنی m می‌رسد. بیشتر توابع کتره‌ساز در نرم افزارهای آشنا از چنین مجموعه پارامترهایی استفاده می‌کنند. در بیشتر این نرم افزار ها $k = 32$ است و پارامترهای دیگر به گونه ای انتخاب شده‌اند که در شرط بالا صدق کرده و دوره تناوب مقدار بیشنه خود را داشته باشد. به طور مثال در Borland C++، $\{a, c, m\} = \{\mathbf{22695477},\ \mathbf{1},\ \mathbf{2^{32}}\}$ و در MS Visual C++، $\{a, c, m\} = \{\mathbf{214013},\ \mathbf{2531011},\ \mathbf{2^{32}}\}$ است.

خروجی این مولدها عددهای صحیحی از 0 تا $2^{32}-1$ هستند. این عددها کمتر به همین شکل مورد استفاده قرار می‌گیرند. به طور مثال در بیشتر مواقع ما به خروجی بین 0 تا 1 نیازمندیم. برای این منظور توابع واسطی در کُدها وجود دارند که خروجی را متناسب با نیاز ما در اختیار ما قرار دهند. برای این مثال در صورتی که خروجی صحیح بر $2^{32}-1$ تقسیم شود، حاصل همان چیزی است که ما به دنبالش هستیم. ولی باید توجه کرد که بدلیل این که این اعداد کاملا کاتوره ای نیستند در کار کردن با آنها باید احتیاط کرد. مثال زیر موضوع را روشن تر خواهد کرد.

به منظور انجام تمرین‌های بالا به مولدی احتیاج دارید که اعداد صحیح بین 0 تا 9 را تولید کند. همچنین فرض کنید که خروجی خام شبه کتره‌ساز را در اختیار دارید که عددی بین 0 تا $2^{32}-1$ است. یک راه ساده این است که از اعداد تولید شده یک رقم را انتخاب کنیم. مثلا عدد یکان را بر داریم. حال اگر تمرین (6.2) را با این مولد انجام دهید نتیجه جالبی خواهید گرفت. تمام عددهایی که بعد از 4 ظاهر می‌شوند فرد هستند. اصلا جای تعجب ندارد. کافی است به معادله بالا و مجموعه پارامترهای معرفی شده نگاه کنید تا به دلیل آن پی ببرید. m عددی زوج و a, c عددهای فرد هستند. پس خروجی این مولدها به تناوب فرد و زوج خواهند بود. ولی من بعید می‌دانم که شما چنین نتیجه ای از تمرین (6.2) گرفته باشید. زیرا برنامه نویسان به خوبی با این مشکل آشنا هستند و در الگوریتم خود عدد دیگری را برای گزارش به جای عدد یکان انتخاب کرده اند.

---

[27] برای آشنایی با مجموعه های مناسب پارامترها به مرجع معرفی شده در **"بیشتر بدانیم"** در انتهای این فصل مراجعه کنید.

مثال ساده‌ی فوق به خوبی نشان می‌دهد اگر چه مولدهای اعداد شبه کاتوره‌ای می‌توانند در بسیاری مواقع بسیار مفید باشند ولی استفاده‌ی درست از آنها نیازمند اطلاعات کاملی از رفتار آنها است. وجود همبستگی منطقی بین اعداد رشته و همچنین دوره تناوب محدود می‌تواند درنتایج بعضی از شبیه سازی‌ها، بخصوص شبیه سازی‌هایی که نیاز به صدا کردن مکرر مولد دارند تاثیر جدی بگذارد. تمرین (6.1) به ما نشان داد که تابع توزیع این اعداد یکنواخت است و در کل فضای یک بعدی احتمال ظهور اعداد برابر است. ولی اگر از مجموعه های d تایی از اعداد متوالی در رشته برای مشخص کردن نقاط کاتوره ای در فضای d بعدی استفاده شود، قضیه ی مارساگلیا[28] نشان می‌دهد که به دلیل همبستگی اعداد، این نقاط بر روی تعداد محدودی صفحه ی $d-1$ بعدی می‌نشینند.

6.6. بذر و کاتوره‌گر[29]

آلگوریتم همنهشتی یک مزیت خیلی مهم از نظر محاسباتی دارد و آن هزینه پایین محاسبات است. آخرین عدد تولید شده را می‌گیرد و عدد بعدی را تحویل می‌دهد. ولی این داستان باید سر آغازی داشته باشد. اولین عددی که برای شروع دنباله به آلگوریتم باید تحویل داد را به اصطلاح بذر می‌نامند. ساختار منطقی آلگوریتم الزام می‌کند که با مشخص شدن بذر کل رشته مشخص می‌شود. اگر بذر در یک برنامه تعیین شود و مقدار ثابتی داشته باشد، اجرای مجدد برنامه دقیقا تکرار اجرای قبل خواهد بود. ولی در صورت تغییر بذر رشته کاتوره‌ای و در نتیجه خروجی برنامه تغییر خواهد کرد. یکی از راه‌های افزایش عامل تصادف در دنباله‌ی تولیدی، استفاده از بذر کاتوره‌ای است. همیشه عواملی وجود دارد که نقش تصادف در آنها بسیار بالا است. هرچند به دلیل محدود بودن این گزینه‌ها امکان تولید رشته کاتوره‌ای با آنها وجود ندارد ولی می‌توان به مقدار محدود از آنها سود جست. یکی از این مناسبت‌ها انتخاب بذر است. در بیشتر برنامه‌ها تابعی به عنوان **کاتوره‌گر** وجود دارد که چنین نقشی دارد. این تابع از **زمان** به عنوان عنصر تصادف استفاده می‌کند. اگر زمان اکنون (ساعت + دقیقه + ثانیه) را به صورت یک عدد صحیح نشان دهیم می‌توانیم ادعا کنیم که این عدد کاملا ساختار تصادفی دارد. لحظه‌ای که کاربر کلید اجرا را فشار می‌دهد به خیلی عوامل انسانی و محیطی بستگی دارد و در نتیجه وقتی برنامه به خطی می‌رسد که کاتوره‌گر را صدا می‌کند این عدد قابل پیش بینی نیست. استفاده از کاتوره‌گر نمی‌تواند کیفیت دنباله شبه تصادفی را بالا ببرد، ولی این امکان را به ما می‌دهد که قطعه‌ی مورد استفاده از دنباله را به طور تصادفی انتخاب کنیم.

[28] Marsaglia's Theorem
[29] Randomizer

نکته 1:

در شبیه‌سازی‌هایی که از آلگوریتم تصادفی استفاده می‌شود به طور معمول هرچه آمار بالاتر باشد، نتایج دقیق‌تر خواهد بود. برای همین گاهی بعد از اتمام اجرا در صورتی که شبیه ساز متوجه ضعف آماری نتایج شود به فکر تکرار برنامه برای بالا بردن آمار می افتد. اگر برنامه از کاتوره‌گر استفاده نکرده باشد و با بذر ثابت کار بکند این تکرار هیچ چیز جدیدی تولید نخواهد کرد.

نکته 2:

یکی از مشکلات برنامه‌هایی که در آنها آلگوریتم‌های تصادفی وجود دارد این است که در صورت وجود مشکل یا باگ (bug) در برنامه، این مشکل نیز خود را به طور تصادفی نشان می‌دهد. یعنی در اجراهای متفاوت در زمان‌های متفاوتی ظاهر می‌شود. این باعث می‌شود که دیباگ یا عیب یابی کردن برنامه بسیار مشکل شود. برای همین توصیه می کنم که در مرحله دیباگ کردن برنامه، از بذر ثابت استفاده کرده و کاتوره‌گر را خاموش کنید. به این ترتیب با تکرار برنامه مشکل در زمان ثابتی آشکار می‌شود. این به شما این امکان را می‌دهد که به داخل برنامه رفته و در قدم ما قبل از خطا به وارسی برنامه و متغیرها بپردازید تا عامل خطا را بیابید. این کار را می توانید با چند بذر متفاوت نیز تکرار کنید. بعد از اطمینان از

نکته 3:

بعضی مواقع شبیه سازان ترجیح می‌دهند که برای کاهش همبستگی دنباله‌ی اعداد، کاتوره‌گر را به دفعات در جای جای برنامه صدا کنند. مطمئنا این کار در کاهش همبستگی دنباله موثر است ولی باید توجه کرد که کثرت این کار می‌تواند نتایج مخربی داشته باشد. همانطور که گفته شد، کاتوره‌گرها معمولا از زمان با دقت ثانیه به عنوان بذر استفاده می‌کنند. اگر فاصله دو بار صدا زدن کاتوره‌گر کمتر از ثانیه باشد، این کار نه تنها کمکی به کاهش همبستگی نمی‌کند، بلکه تاثیر کاملا معکوس دارد و منجر به تکرار مجدد رشته می‌شود. برای امتحان می‌توانید کاتوره‌گر و دستور چاپ یک عدد تصادفی را در درون یک حلقه گذاشته وبرنامه را اجرا کنید.

مستقل از عدم کتره ای بودن واقعی اعداد مشکل مهم دیگر تناوب محدود رشته است. این نکته باعث می‌شود که امکان اجرای برنامه‌های طولانی از شبیه‌ساز گرفته شود.

برای مثال اگر قصد بدست آوردن متوسط یک کمیت آماری را داشته باشید، به خوبی می‌دانیم که با افزایش آمار دقت اندازه‌گیری (محاسبه) افزایش می‌یابد. ولی این جمله تا جایی درست است که نمونه برداری‌های جدید از قبلی‌ها مستقل باشد. مطمئنا دوبرابر کردن نمونه‌ها با تکرار آنها و بدون ورود آمار جدید هیچ ارزشی ندارد.

اگر در تمرین ول نشست در بخش (3.2) طول سیستم برابر با m، دوره تناوب مولد اعداد شبه کاتوره‌ای انتخاب شود، در هر m قدم تمام خانه های شبکه یک ذره دریافت می‌کنند (البته نه به ترتیب). بنابر این ناهمواری این ول نشست صفر خواهد شد. در این گونه مواقع باید راهی برای افزایش دوره تناوب پیدا کرد. در زیر یک روش ساده برای این کار معرفی می‌شود.

آلگوریتم بُر زدن دنباله‌ی اعداد کتره‌ای

1. دو تابع مولدR1 و R2 که از مجموعه پارامترهای متفاوتی استفاده می‌کنند ایجاد کنید (یکی از این‌ها می‌تواند همان مولد استاندارد نرم افزار شما باشد)
2. آرایه ای به طول مثلا 200 بسازید و آنرا با صدا کردن R1 مقدار دهی کنید.
3. با استفاده از R2 عددی کاتوره‌ای $1 \le p \le 200$ را تولید کنید.
4. عنصر pام آرایه را به عنوان خروجی مولد جدید خود گزارش کنید و دوباره با صدا کردن R1 این عنصر را مقدار جدیدی بدهید.

تمام خروجی‌های الگوریتم ترکیبی بالا بوسیله‌ی R1 تولید شده‌اند و در نتیجه این آلگوریتم تابع توزیع یکنواخت R1 را مخدوش نمی‌کند. تنها کاری که این آلگوریتم می‌کند استفاده از R2 برای بُر زدن دنباله تولید شده توسط R1 است. این کار باعث می‌شود که دوره تناوب رشته خروجی بسیار بزرگتر شود. اگر دوره‌های تناوب R1 و R2 نسبت به هم اول باشند، دوره تناوب خروجی آلگوریتم بالا برابر با حاصل ضرب دوره‌های تناوب دو مولد می‌باشد. این کار حتی در شبیه‌سازی‌های کوتاه‌تر که نگرانی برای مشاهده تکرار دنباله وجود ندارد نیز کار بسیار خوبی است، زیرا باعث می‌شود که همبستگی اعداد متوالی در دنباله مخدوش شود.

6.7. تولید اعداد کتره ای با توزیع غیر یکنواخت

در بسیاری از مواقع ما تمایل داریم که مولد ما اعداد کاتوره‌ای با توزیع دلخواه ما تولید کند. مثلا با بسیاری از پدیده‌های تصادفی در فیزیک آشنا هستیم که عامل تصادف توزیع طبیعی (گوسی) دارد. در چنین مواقعی شبیه‌سازی این پدیده‌ها نیازمند مولدی است که رشته اعداد کاتوره‌ای آن از توزیع گوسی تبعیت کند. در ابتدا با مثالی نشان می‌دهیم که می‌توان با استفاده از مولد کاتوره‌ای با توزیع یکنواخت، مولدی دیگر را آماده کرد.

6.7.1. قضیه حد مرکزی[30]

این قضیه یکی از قضایای مهم در آمار است و کاربردهای فرآوانی در مسایل مختلف دارد. به زبان ساده این قضیه می‌گوید هر کمیت تصادفی که در حقیقت مجموع تعداد زیادی کمیت تصادفی مستقل باشد، از یک تابع توزیع احتمال گوسی پیروی می‌کند[31]. شکل ساده شده‌ی این قضیه می‌تواند به منظور تولید اعداد با توزیع گوسی استفاده شود. فرض کنید که مولدی با توزیع دلخواه $P(x)$ برای تولید دنباله اعداد کتره ای $\{x_1, x_2, \dots, x_i, \dots\}$ با مقدار میانگین

$$< x > = \int x\, P(x)\, \mathrm{d}x, \tag{2}$$

و مقدار مجذور افت و خیز (انحراف از معیار)

$$\sigma_x^2 = \int x^2\, P(x)\, \mathrm{d}x - < x >^2, \tag{3}$$

در اختیار داریم. متغیر y را اینگونه تعریف می‌کنیم:

$$y = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{4}$$

بنا به قضیه حد مرکزی y متغیر کاتوره‌ای است که تابع توزیع آن برای مقادیر $N \gg 1$ به تابع توزیع طبیعی

$$P(y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\, e^{-\frac{(y-y_0)^2}{2\sigma_y^2}} \tag{5}$$

میل می‌کند که در آن

$$< y > = y_0 = < x > \tag{6}$$

و

$$\sigma_y = \frac{\sigma_x}{\sqrt{N}} \tag{7}$$

است.

.6.3	برای تحقیق قضیه‌ی حد مرکزی با استفاده ازمولد اعداد کاتوره‌ای در نرم افزار مورد استفاده‌ی خود، تابع توزیع اعدادی که از جمع N تولید می‌شوند را بدست آورید. این کار را برای مقادیر $N = \{5, 10, 100, 1000\}$ انجام دهید.
تمرین	

[30] Central Limit Theorem (CLT)

[31] این قضیه به شکل دقیق بعضی شرایط را برای کمیتهای کاتوره ای که در جمع وارد میشوند قایل است که در اینجا از ذکر این جزییات میگذریم. خواننده ی علاقمند میتواند به مراجع انتهای این بخش مراجعه کند.

چه شباهتی میان این تمرین و تمرین ولگشت و ولنشست می‌بینید؟	

6.7.2.تغییر تابع توزیع با استفاده از تابع تبدیل

فرض کنید که مولدی با تابع توزیع $p(x)$ در اختیار داریم. در نتیجه احتمال اینکه این مولد عددی در بازه $(x, x+\mathrm{d}x)$ را تحویل دهد، $p(x)\mathrm{d}x$ است. موضوع این بخش این است که به دنبال تابعی هستیم که با اعمال آن بر روی x متغیری مانند y تولید کند $(y=f(x))$ که از تابع توزیع دلخواه $g(y)$ تبعیت کند. در این صورت احتمال داشتن عددی در بازه‌ی $(y, y+\mathrm{d}y)$ باید برابر با $g(y)\mathrm{d}y$ باشد. در نتیجه داریم:

$$p(x)\mathrm{d}x = g(y)\mathrm{d}y. \tag{8}$$

بدون اینکه چیزی از کلیت بحث کاسته شود و فقط به دلیل اینکه در عمل برای تولید اعداد معمولا مولدِ یکنواختِ استانداردِ نرم افزارهای خود را در اختیار داریم، فرض می‌کنیم که

$$p(x) = p_u(x) = \begin{cases} 1 & 0 \le x < 1 \\ 0 & \text{else where} \end{cases} \tag{9}$$

از طرفین رابطه (8) انتگرال می‌گیریم

$$\int_{-\infty}^{x} p_u(x)\mathrm{d}x = \int_{-\infty}^{y=f(x)} g(y)\mathrm{d}y. \tag{10}$$

سمت چپ این تابع به راحتی با استفاده از شکل تابع $p_u(x)$ قابل محاسبه و برابر با x است. فرض می‌کنیم انتگرال سمت راست را قادر به محاسبه هستیم و حاصل تابع تجمعی $G(y)$ است. درنتیجه داریم

$$x = G(y). \tag{11}$$

اگر این تابع معکوس پذیر باشد می‌توانید به راحتی y را برحسب x بنویسیم،

$$y = f(x) = G^{-1}(x). \tag{12}$$

چند مثال می‌تواند به روشن شدن توانایی این روش در تولید اعداد کاتوره‌ای با توزیع دلخواه کمک کند. در ابتدا با مثال بسیار ساده‌ای شروع می‌کنیم. فرض کنید

$$g(y) = \begin{cases} A & a \le y < b \\ 0 & \text{else} \end{cases} \tag{13}$$

مقدار $A = \frac{1}{b-a}$ با فرض‌یکه بودن تابع توزیع تثبیت می‌شود. با استفاده از رابطه (10) به سادگی داریم:

$$x = G(y) = \frac{y-a}{b-a}. \tag{14}$$

که به رابطه نه چندان غیر قابل پیش بینی

$$y = a + (b-a)x \tag{15}$$

می‌رسیم.
مثال کمی پیچیده‌تر می‌تواند تابع توزیع نمایی باشد. فرض کنید که مولد قرار است اعدادی مثبت و با تابع توزیع زیر را بدهد.

$$g(y) = \begin{cases} a\, e^{-\frac{y}{a}} & x \geq 0 \\ 0 & \text{else} \end{cases} \tag{16}$$

در این حالت نیز به سادگی با محاسبه انتگرال داریم:

$$x = G(y) = 1 - e^{-\frac{y}{a}} \tag{17}$$

و معکوس آن به رابطه

$$y = -a \ln(1-x) \tag{18}$$

منجر می‌شود. در نتیجه این رابطه‌ی ساده می‌تواند از مولد یکنواخت کامپیوتر ما مولد کاتوره‌ای بسازد که خروجی آن تابع توزیع نمایی داشته باشد. توجه به این نکته که $(1-x)$ همان تابع توزیع x را دارد می‌تواند کمک کند تا رابطه بالا باز هم ساده تر شود

$$y = -a \ln x \tag{19}$$

یکی از مورد توجه‌ترین توابع توزیع در فیزیک، تابع توزیع نرمال یا گوسی است. ولی در مورد این تابع کار به سادگی مثال‌های بالا نیست. پاسخ برای انتگرال محدود تابع گوسی وجود ندارد. در این حالت می‌توان از ترفند آشنای استفاده از فضای دو بعدی استفاده کرد. تابع توزیع گوسی

$$g(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{y^2}{2\sigma^2}} \tag{20}$$

را در نظر می‌گیریم. دو متغیر y_1 و y_2 را که هر دو تابع توزیع بالا را دارند را در نظر می‌گیریم . در این حالت احتمال داشتن جفت (y_1, y_2) از رابطه زیر بدست می‌آید.

$$g(y_1, y_2) = g(y_1)g(y_2) = \frac{1}{2\pi\sigma^2} e^{-\frac{y_1^2 + y_2^2}{2\sigma^2}} \tag{21}$$

اگر متغیرهای (y_1, y_2) مختصات دکارتی یک نقطه در نظر بگیریم با تغییر متغیر به مختصات قطبی داریم:

$$g(y_1, y_2)dy_1dy_2 = \frac{1}{2\pi\sigma^2} e^{-\frac{\rho^2}{2\sigma^2}} \rho \, \mathrm{d}\rho \, \mathrm{d}\theta \tag{22}$$

از رابطه بالا به راحتی می‌توان تابع توزیع ρ و θ را خواند.

$$g_\rho(\rho) = \frac{1}{\sigma^2} e^{-\frac{\rho^2}{2\sigma^2}} \rho \tag{23}$$

$$g_\theta(\theta) = \frac{1}{2\pi} \tag{24}$$

فاکتور ρ در رابطه اول کمک می‌کند که تابع نمایی انتگرال پذیر شود. در مورد θ کار حتی ساده‌تر نیز است. پس به این روش به راحتی قادر هستیم جفت اعداد (ρ, θ) را با توابع توزیع بالا تولید کنیم. با تغییر متغیر از مختصات قطبی به دکارتی به ازای هر جفت مختصات قطبی یک جفت مختصات دکارتی به دست می‌آید. نکته جالب توجه این است که هر دو مولفه‌ی مختصات دکارتی از تابع توزیع گوسی تبعیت می‌کنند. به این ترتیب به ازای هر دو بار صدا کردن مولد یکنواخت یک جفت عدد کاتوره‌ای با توزیع گوسی داریم. یعنی یک عدد به ازای هر بار صدا کردن.

.6.4	با روشی که دربالا توصیف شد مولدی با توزیع گوسی بسازید و با رسم فرآوانی خروجی‌های آن نشان دهید که مولدتان خوب کار می‌کند.
تمرین	

بیشتر بدانیم:

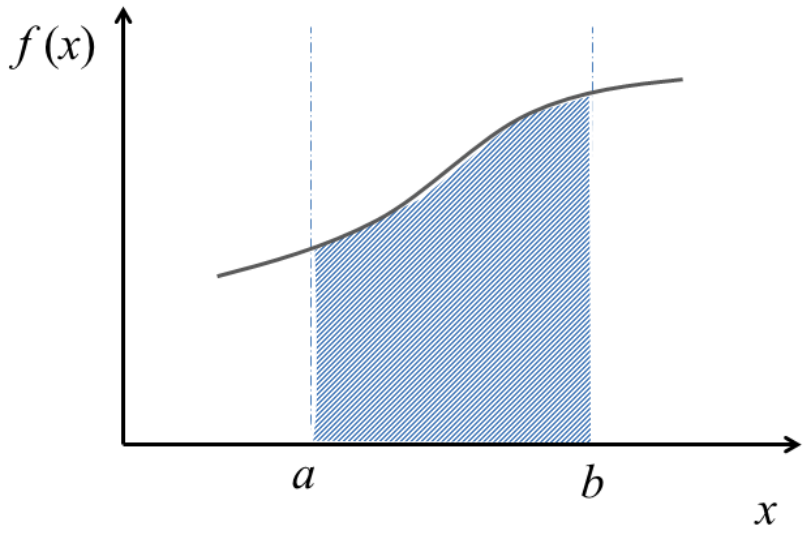
برای آشنایی با الگوریتم های تولید اعداد کاتوره ای و نیز مجموعه پارامترهای مناسب برای استفاده دراین آلگوریتم ها میتوانید به کتاب Number Theory for Computing نوشته ی Song Yan مراجعه کنید. برای اطلاع بیشتر از قضیه حد مرکزی نیز متوانید به کتاب Statistical Physics of Particle نوشته یی Meharn Kardar مراجعه کنید.

7. انتگرال گیری

تا کنون با فرآیندهایی آشنا شدیم که در فیزیک آنها تصادف نقشی اساسی بازی می‌کرد. استفاده از آلگوریتم‌هایی که در آنها یک مولد اعداد کاتوره‌ای نقش اصلی را داشته باشد برای شبیه سازی این گونه فرآیندها کاملا موجه و قابل انتظار است. در این فصل می‌خواهیم توانایی آلگوریتم‌های تصادفی در شبیه سازی و محاسبات عددی را با یک مثال کاملا جبری و تعینی نشان دهیم. فرض کنید که بخواهیم انتگرال،

$$\boldsymbol{I} = \int_{\boldsymbol{a}}^{\boldsymbol{b}} \boldsymbol{f}(\boldsymbol{x})\mathbf{d}\boldsymbol{x} \qquad (1)$$

محاسبه کنیم. کاملا واضح است که در اینجا نه تنها از تصادف صحبتی نیست بلکه با یک محاسبه کاملا دقیق روبرو هستیم. شاید برای ادامه بحث بهتر باشد که تصویری هندسی از انتگرال فوق داشته باشیم. در حقیقت این انتگرال برابر است با مساحت زیر منحنی $f(x)$ در بازه ی انتگرال گیری.



شکل 15 محاسبه‌ی انتگرال معادل با محاسبه سطح زیر منحنی است.

برای حل چنین انتگرال‌هایی روش‌های اجزای محدود[32] بسیار کاربرد دارند. در این گونه روش‌های عددی بازه‌ی انتگرال گیری به اجزای بسیار کوچکی تقسیم می‌شود و مساحت زیر منحنی با جمع مساحت مستطیل‌هایی به عرض این اجزا و طول مقدار تابع تخمین زده می‌شود. ولی منظور ما از روش اجزای محدود نیست و به دنبال روشی هستیم که از تصادف برای تخمین انتگرال استفاده کند. برای این منظور با مثالی جالب نشان می‌دهیم این کار امکان پذیر است.

[32] Finite element

7.3. آلگوریتم شِلِپ

فرض کنید که در بیرون باغی با دیوارهای بلند ایستاده‌اید و به هیچ وجه امکان مشاهده‌ی درون باغ را ندارید. به شما اطلاع می‌دهند که در درون این باغ استخر آبی وجود دارد و از شما خواسته می‌شود که مساحت این استخر را تخمین بزنید. شاید در وهله‌ی اول به نظر برسد که این کار امکان پذیر نیست ولی راه حلی بسیار هوشمندانه برای حل این مسئله وجود دارد. سنگی بردارید و از بالای دیوار آن را به درون باغ پرتاب کنید. اگر سنگ به درون استخر بیافتد صدای شِلِپی خواهید شنید. درغیر این صورت از برخورد سنگ با زمین صدای تِلِپ به گوش می‌رسد.
احتمال سقوط سنگ به درون استخر برابر با نسب مساحت استخر به مساحت باغ است. با تکرار این عمل شما می‌توانید به تخمینی از این احتمال برسید. فرض کنید که از N بار پرتاب سنگ N_s بار صدای شلپ بشنوید. در نتیجه احتمال سقوط سنگ در استخر $\frac{N_s}{N}$ است. با توجه به اینکه شما قادرید مساحت کل باغ را با اندازه‌گیری هندسه‌ی باغ از خارج آن بدست آورید، مساحت استخر به راحتی با رابطه‌ی،

$$\boldsymbol{A_p = A\frac{N_s}{N}} \qquad (2)$$

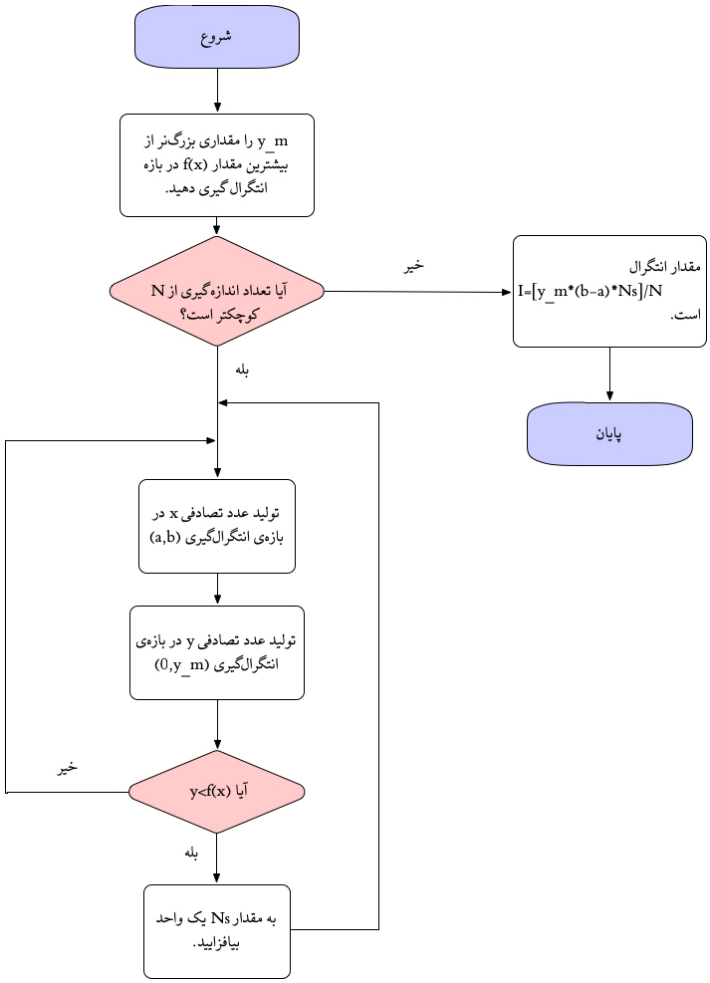
داده می‌شود که در اینجا A و A_p به ترتیب مساحت باغ و استخر هستند. بدیهی است که دقت تخمین فوق به دو عامل مهم بستگی دارد. تخمین احتمال سقوط در استخر با پرتاب یک یا دو سنگ امکان پذیر نیست. برای داشتن تخمین قابلِ قبولی نیاز به آمار بیشتر هستیم. نکته‌ی دیگر در پرتاب‌یکنواخت سنگ‌هاست. اگر سنگ‌ها متمایل به سقوط در نقاط خاصی از باغ باشند نیز نمی‌توان تخمین دقیقی از مساحت استخر داشت.
حال به مسئله انتگرال گیری بر می‌گردیم. با توجه به اینکه این مسئله در واقع محاسبه یک مساحت است می‌توان روش فوق را برای آن به کار برد. در زیر این روش در قالب یک آلگوریتم برای بدست آوردن مساحت معرفی می‌شود.

آلگوریتم شلپ برای انتگرال گیری:

1- y_m را مقداری بدهید که مطمئن هستید بزرگتر یا مساویِ بزرگترین مقدار تابع در بازه انتگرال گیری است.
2- دو متغیر کاتوره‌ای $x \in (a,b)$ و $y \in (0,y_m)$ را انتخاب کنید.
3- اگر $y < f(x)$ به N_s یکی بیافزایید.
قدم های 2 و 3 را در حلقه‌ای به طول N قرار دهید.
در پایان حلقه مقدار انتگرال را از رابطه

$$I = y_m(b-a)\frac{N_s}{N}$$

بیابید.



نمودار شناور 1 آلگوریتم شلپ را نشان می‌دهد. در این روش با انتخاب نقاط تصادفی قادر به محاسبه‌ی انتگرال هستیم.

کمی جلوتر به موضوعِ مهمِ دقتِ این گونه روش‌ها خواهیم پرداخت ولی بدیهی است که در صورت کیفیتِ مناسب و یکنواخت بودن اعداد کاتوره‌ای مورد استفاده، با افزایش تعداد نقاط مورد بررسی می‌توانیم، به هر دقتی که بخواهیم، انتگرال را به کمک این الگوریتم محاسبه کنیم.

7.4. نمونه برداری ساده[33]

در بالا دیدیم که محاسبه ی انتگرال با روش کاملا تصادفی امکان پذیر است ولی مانند همیشه ساده‌ترین راه برای حل مسئله بهترین راه نیست. در اینجا یک بار دیگر به مسئله‌ی انتگرال گیری بر میگردیم. انتگرا ل (1) را به شکل زیرباز نویسی می‌کنیم:

[33] Simple sampling

$$\boldsymbol{I = \int_{-\infty}^{\infty} f(x)p(x)\mathrm{d}x} \tag{3}$$

که در اینجا

$$\boldsymbol{p(x) = \begin{cases} 1 & a \le x < b \\ 0 & \text{else where} \end{cases}} \tag{4}$$

این تابع یکه نیست و

$$\boldsymbol{\int_{-\infty}^{\infty} p(x)\mathrm{d}x = b - a} \tag{5}$$

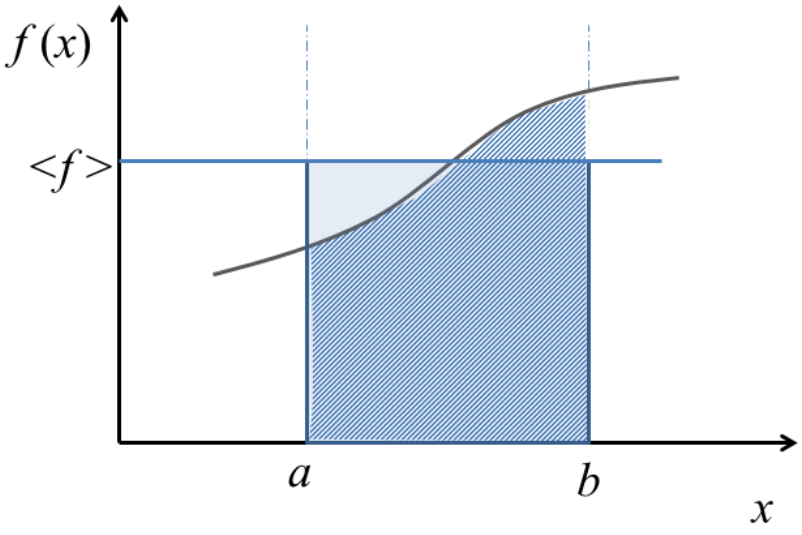
پس می‌توانیم انتگرال مورد نظر را به صورت

$$\boldsymbol{I = (b-a)\,\frac{\int_{-\infty}^{\infty} f(x)p(x)\mathrm{d}x}{\int_{-\infty}^{\infty} p(x)\mathrm{d}x}} \tag{6}$$

نوشت. ولی عبارت کسری در سمت راست چیزی نیست به غیر از تعریف متوسط تابع $f(x)$ در بازه a تا b. در نتیجه

$$\boldsymbol{I = (b-a) < f >.} \tag{7}$$

این نتیجه را می‌توان با توجه به تصویر هندسی انتگرال که در زیر داده می‌شود بهتر درک کرد.



شکل 16 مساحت زیر تابع برابر است با مساحت مستطیلی با عرض برابر و ارتفاع متوسط تابع

در نتیجه برای محاسبه‌ی انتگرال کافی است که مقدار متوسط تابع در فاصله انتگرال گیری را داشته باشیم. این کار را می‌توان به راحتی با استفاده از

روش نمونه برداری آماری انجام داد. یعنی کافی است که مقادیر تابع را در نقاطی که کاملا به طور تصادفی انتخاب می‌کنیم بخوانیم و بعد با میاین‌گیری از آنها مقدار متوسط تابع را تخمین بزنیم. اگر رشته اعداد کاتوره ای $\{x_1, x_2, \dots, x_n\}$ را در اختیار داشته باشیم. مقدار متوسط تابع به راحتی قابل محاسبه است،

$$< f > = \frac{1}{n} \sum_{i=1}^{n} f(x_i). \qquad (8)$$

و با داشتن مقدار متوسط انتگرال را می‌توانیم از رابطه (7) محاسبه کنیم. این روش از روش شلپ کاراتر است زیرا برای نمونه برداری کافی است که بر روی یک محور نمونه برداری شود و نیازی به جستجو در فضای دو بعدی ندارد. این روش محاسبه انتگرال را ***مونت کارلو***[34] می‌نامند. به دلیل یکنواختی در نمونه برداری آنرا مونت کارلو با نمونه برداری ساده می‌نامند.

7.5. محاسبه خطا

حال وقت آن رسیده که به موضوع مهمِ دقت و خطای محاسبات بپردازیم. واضح است که دقت انتگرال به دقت در محاسبه‌ی متوسط تابع بر می‌گردد. دقت در محاسبه‌ی متوسط هم دو عامل مهم دارد.

عامل اول - شکلِ تابع بشدت اهمیت دارد. متوسط یک تابع هموار و نسبتا افقی که افت و خیز کمی در بازه انتگرال گیری داشته باشد به راحتی و با نمونه برداری بسیار کم و محدود با دقت خوبی قابل محاسبه است. پس خطای محاسبه به میزان افت و خیز تابع، $\sigma = \sqrt{< f^2 > - < f >^2}$ ، بستگی دارد.

عامل دوم - اگر برای محاسبه‌ی متوسط تابع متوسط گیری را با مجموعه دیگری از اعداد تصادفی تکرار کنیم مقدار جدید با مقدار قدیم متفاوت خواهند بود و این معیاری از خطای محاسبه است. بنابراین $< f >$ خود تابع توزیعی دارد که پهنای آن (انحراف از میعار) میعاری از خطای محاسبه‌ی آن است. بر اساس قضیه‌ی حد مرکزی این تابع توزیع گوسی است و پهنای آن یا خطای محاسبه از رابطه‌ی

$$\Delta = \frac{\sigma}{\sqrt{n}} \qquad (9)$$

بدست می‌آید. با چند مثال بحث خطا را ادامه دهیم.

فرض کنید که می خواهیم متوسط قد ایرانی‌ها را بدست بیاوریم. مطمئننا نمی‌توانیم قد تمام ایرانی‌ها را اندازه بگیریم، پس ناچاریم نمونه برداری کنیم. پس قد چند ایرانی را اندازه می‌گیریم و متوسط قد آنها را گزارش می‌کنیم. دقت ما در گزارش این عدد به چه عواملی بستگی دارد؟ بدون شک مهم‌ترین عامل تعداد ایرانی‌هایی است که در نمونه برداری شرکت کرده اند. اگر متوسط قد را با یک میلیون ایرانی گزارش کنیم دقتمان بیشتر از اندازه گیری هزار ایرانی است. ولی آیا می‌توان اختلاف دقت این دو اندازه‌گیری را مشخص کرد؟

فرض کنید که یک کیلو نخود و یک ترازوی بسیار دقیق در ختیار داریم و می‌خواهیم متوسط وزن نخودها را اندازه بگیریم. حال دوباره این سوال مطرح می‌شود که دقت وزن کردن 100 نخود بیشتر است یا 1000 نخود. فرض کنید که با یک انباری از نخود روبرو هستیم و می‌خواهیم وزن متوسط نخودها را گزارش کنیم. پس تعداد کل نخودها مهم نیست بلکه تعداد نمونه‌ها مهم است. از آنجایی که همه نخودها

[34] Monte Carlo

دارای وزن هستند، وزن متوسط نخودها یک کمیت فیزیکی قابل اندازه گیری است. اگر این عدد را با یک بار نمونه برداری گزارش دهیم قطعا خطای زیادی خواهیم داشت. با اندازه‌گیری دو نخود دقت بیشتر خواهد شد. خطا با تعداد نمونه برداری چگونه کاهش می‌یابد؟ در واقع با افزایش تعداد نمونه چه کمیتی کاهش می‌یابد که ما آنرا به عنوان خطا گزارش می‌کنیم؟ فرض کنید که وزن هر نخود را با n_i نشان دهیم و تابع توزیع نمونه‌ها را بدست آوریم. متوسط وزن به این تابع توزیع بستگی خواهد داشت. مثلا اگر دو جنس نخود مرغوب و نامرغوب در انبار باشد احتمالا تابع توزیع حول دو وزن نخودها قله‌ای خواهد داشت. خطای اندازه گیری وزن نخودها در اصل به پهنای تابع توزیع وزن نخودها بستگی خواهد داشت. اما پهنای تابع توزیع وزن نخودها مقدار مشخصی است که به جنس نخودها انبار بستگی دارد، پس با افزایش تعداد نمونه‌ها چه کمیتی کاهش پیدا می‌کند؟ فرض کنید پیمانه‌های داریم که میتوانیم با استفاده از آن نخودها را 100 تا 100 تا وزن کنیم. پس $M = \sum_{i=1}^{100} n_i$ وزن 100 نخود است. فرض کنید چندین بار با این پیمانه نخودها را وزن کنیم و این بار تابع توزیع Mها را رسم کنیم. Mها جمع 100 عدد تصادفی هستند و طبق اصل حد مرکزی تابع توزیع Mها یک تابع گوسی است و حتما تیزتر از تابع توزیع nها خواهد بود. در واقع پهنای نسبی این تابع توزیع با جذر تعداد نمونه‌ها کاهش می‌یابد (معادله 9).

نکته 1:

یکی از مزایای محاسبه انتگرال به روش مونت کارلو امکان محاسبه خطا به همراه انتگرال و کنترل توقف برنامه بعد از رسیدن به دقت مطلوب است. شما می‌توانید همراه با متوسط تابع مقدار انحراف از میعار آن را نیز محاسبه کنید. با داشتن مقدار انحراف میعار شما می‌توانید تخمین از تعداد نمونه‌ها برای اینکه دقت مورد نظر را بدست آورید تخمین بزنید.

برای این کار کافی است که بعد از مثلا چند هزار نمونه‌ای که می‌گیرید مقدار انحراف از میعار را بدست آورید. بدیهی است که در مقدار انحراف از میعار هم مانند متوسط خطا دارید، ولی این کمیت برای تخمین دقت بکار می‌رود و خطای آن اهمیت زیادی در این تخمین ندارد.

7.6. نمونه برداری هوشمند[35]

روش مونت کارلو با نمونه برداری ساده، روشی بسیار کارا برای انتگرال گیری از توابع در بازه‌ی محدود و همچنین با اُفت و خیز کم است. ولی استفاده از این روش در دو حالت به مشکل برخورد می‌کند:

-انتگرال گیری با حدود بینهایت.

-انتگرال گیری از توابعی که اُفت خیز زیادی دارند،مثلا اگر انتگرالده در بیشتر محدوده‌ی انتگرالگیری مقدار خیلی کوچکی داشته باشد و فقط در یک ناحیه‌ی بسیار کوچک (در مقایسه با حدود انتگرال) غیر صفر است.

در هر دو حالت بالا اگر از روش نمونه برداری ساده استفاده کنیم در بیشتر زمان نمونه‌هایی را داریم که ارزشی در محاسبه‌ی ما ندارند. برای درک این مشکل به تصویر مساحت زیر انتگرال برگردید. فرض کنید که تابع فقط در حوالی یک نقطه ماکزیمم بسیار بلند دارد و در بقیه نقاط مقداری بسیار کوچک و نزدیک

[35] Important sampling

به صفر دارد. حال اگر در نمونه برداری ساده این ماکزیمم را از دست بدهیم مطمئنا مقدار مساحت زیر منحنی را بسیار کمتر از آنچه هست بدست می‌آوریم. برای رفع این مشکل باید مطمئن باشیم که نقاطی که تابع مقدار قابل توجهی دارد حتما در متوسط گیری لحاظ شود. این داستان را می‌توان به شکل ریاضی نیز توضیح داد. به انتگرال

$$\boldsymbol{I} = \int_{a}^{b} \boldsymbol{f(x)\mathrm{d}x} \tag{10}$$

برمی‌گردیم. فرض کنید که تابع $g(x)$ که در بازه‌ی انتگرال گیری به تابع $f(x)$ شباهت دارد را بشناسیم. لازم است که انتگرال این تابع را در بازه فوق را نیز بدانیم. در این صورت انتگرال را می‌توانیم به صورت

$$\boldsymbol{I} = \int_{-\infty}^{\infty} \frac{\boldsymbol{f(x)}}{\boldsymbol{g(x)}}\, \boldsymbol{g(x)\mathrm{d}x} \tag{11}$$

یا با کمی تغییرات به شکل

$$\boldsymbol{I} = \left(\int_{-\infty}^{\infty} \boldsymbol{g(x)\mathrm{d}x}\right) \frac{\int_{-\infty}^{\infty} \frac{\boldsymbol{f(x)}}{\boldsymbol{g(x)}}\, \boldsymbol{g(x)\mathrm{d}x}}{\int_{-\infty}^{\infty} \boldsymbol{g(x)\mathrm{d}x}} \tag{12}$$

نوشت. کسر انتهایی در سمت چپ چیزی جز متوسط تابع $f(x)$ با تابع توزیع احتمال

$$\frac{\boldsymbol{g(x)}}{\int_{-\infty}^{\infty} \boldsymbol{g(x)\mathrm{d}x}} \tag{13}$$

نیست. پس انتگرال را می‌توانیم به صورت

$$\boldsymbol{I} = \left(\int_{-\infty}^{\infty} \boldsymbol{g(x)\mathrm{d}x}\right) < \frac{\boldsymbol{f}}{\boldsymbol{g}} >_{\boldsymbol{g(x)}} \tag{14}$$

است. در اینجا منظور از $<\cdots>_{g(x)}$ متوسط گیری بر روی اعداد کاتوره‌ای x است که از تابع توزیع $g(x)$ تبعیت کند.

$$< \frac{\boldsymbol{f}}{\boldsymbol{g}} >_{\boldsymbol{g(x)}} = \frac{\boldsymbol{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{\boldsymbol{f(x_i)}}{\boldsymbol{g(x_i)}} \tag{15}$$

توجه کنید که برای محاسبه متوسط این کسر با تابع توزیع $g(x)$ فقط کافی است که اعداد کاتوره‌ای x با این تابع توزیع تولید شده باشند. مشابه آلگوریتم قبل خطای محاسبه به افت و خیز تابع $f(x)/g(x)$ بستگی دارد. اینجاست که اهمیت شباهت f و g مشخص می‌شود. اگراین دو تابع به یکدیگر شبیه باشند حاصل کسر تابعی هموار و کم اُفت و خیز است. در نتیجه با تعداد کمتری نمونه‌گیری در مقایسه با روش نمونه برداری ساده به دقت مطلوب می‌رسیم.
از شباهت f و g نکته‌ی دیگری نیز حاصل می‌شود. هرجا که f کوچک باشد g هم کوچک است و در نتیجه احتمال داشتن xی در این ناحیه کم است. در مقابل در جاهایی که f و g بزرگ هستند، احتمال نمونه برداری بیشتر است. برای همین به این روش نمونه برداری هوشمند گفته می‌شود.

.7.1	انتگرال $I = \int_0^2 e^{-x^2}\mathrm{d}x$ را به دو روش نمونه برداری ساده و هوشمند بدست آورید و نتیجه را باهم مقایسه کنید. برای انتگرال گیری هوشمند از $g(x) = e^{-x}$ استفاده کنید. در فصل پیش آموختید که چگونه مولد اعداد کتره‌ای با این تابع توزیع را درست کنید. در جدولی مقدار انتگرال، خطای آماری، خطای واقعی (در مقایسه با نتیجه انتگرال در نرم افزارهایی مانند Matlab یا Mathematica) ، و زمان اجرا را برای هر دو روش و برای مقادیر مختلف تعداد نمونه ها مقایسه کنید.
تمرین	

7.7. انتگرال چندگانه

یکی از مزایای خیلی مهم انتگرال‌گیری مونت کارلو سادگی تعمیم آن به ابعاد بالاتر و حل انتگرال‌های چندگانه است. فقط کافی است که متوسط تابع محاسبه شود. البته این بار این تابع بیش از یک متغیر دارد و برای نمونه برداری باید از مولد اعداد کتره‌ای برای تمام متغیرها استفاده کرد.

.2.1	چگالی جرمی کره‌ای در راستای عمودی آن از با لا تا پایین به صورت خطی کم می‌شود، به گونه‌ای که کمترین چگالی نصف چگال‌ترین نقطه است. مرکز جرم این کره کجاست؟
تمرین	

بیشتر بدانیم:

8. تولید اعداد کتره ای با هر توزیع دلخواه - روش متروپولیس

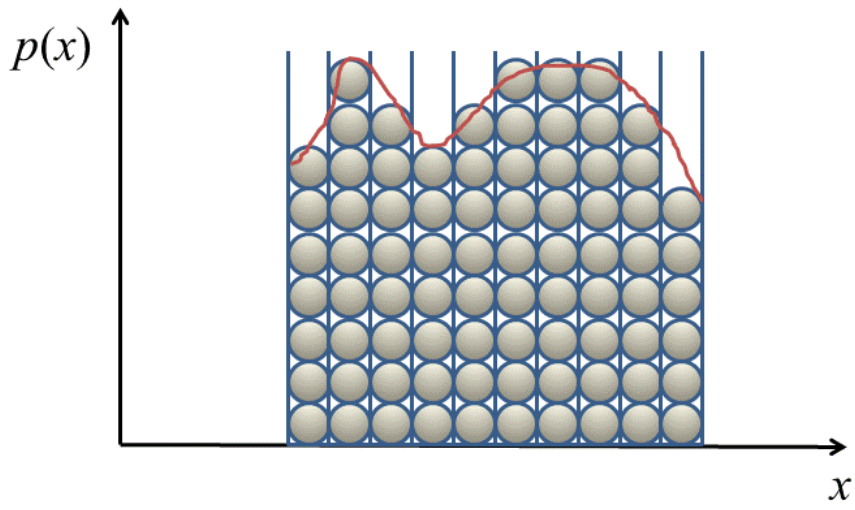
در فصل 6 با آلگوریتم‌هایی که امکان تولید اعداد کتره‌ای با توزیع غیر یکنواخت آشنا شدیم. درمورد توابع توزیع انتگرال پذیر و معکوس پذیر مشکلی وجود ندارد و با روش ارایه شده در آن بخش به راحتی می‌توان تابع تبدیلی برای تولید اعداد کاتوره‌ای با توزیع دلخواه بدست آورد. حتی در مورد تابع توزیع گوسی که در یک بعد انتگرال پذیر نیست، ترفندی به ما کمک کرد که تابع تبدیل را در فضای دو بعدی بیابیم. بدینوسیله برای تولید اعدادی با تابع توزیع گوسی دو روش معرفی شد.

الف) استفاده از تابع تبدیل
ب) استفاده از قضیه حد مرکزی

اجازه بدهید که نگاهی دقیق تر بر روش دوم بیاندازیم. دراین روش جمع تعدادی عدد کاتوره‌ای به عنوان خروجی معرفی می‌شود. در ساده‌ترین شکل می‌توان این اعداد را رشته‌ای از عددهای 1+ و 1- در نظر گرفت که با احتمال برابر امکان ظهور دارند. در نتیجه این آلگوریتم هم ارز آلگوریتم ول گشت است. در حقیقت در این روش مکان ولگرد بعد از تعدادی قدم به عنوان خروجی داده می‌شود. می‌دانیم که مکان بعد از N قدم از تابع توزیع گوسی با پهنای $\sigma = \sqrt{N}$ تبعیت می‌کند.
نتیجه‌ای که از آلگوریتم (ب) می‌گیریم امکان طراحی یک بازی است که در یک فرآیند تصادفی خروجی‌هایی با تابع توزیع دلخواه ما تولید کند. در ادامه این فصل می‌خواهیم این روش را تعمیم دهیم و نشان دهیم که با فرآیند مشابهی می‌توان هر تابع توزیع دلخواهی را تولید کرد.

8.3. مهره ها در جعبه

فرض کنید که تابع مطلوب ما برای مولد اعداد کاتوره‌ای تابع توزیع احتمال $p(x)$ است. برای سادگی کار فعلا فرض می‌کنیم که دامنه‌ی این تابع محدود است. متناسب با توان تفکیک مورد نظر محور x را به اجزایی تقسیم می‌کنیم. هر یک از جزء‌ها را مانند جعبه‌ای در نظر می‌گیریم. فرض کنید که این جعبه‌ها را با گلوله‌هایی پر کنیم. تعداد گلوله‌ها در هر جعبه متناسب با مقدار تابع در آن نقطه است.



شکل 17 گلوله‌ها را متناسب با تابع توزیع در جعبه‌ها پخش می‌کنیم.

همچنین فرض کنید که هر گلوله شماره‌ای دارد. حال به کمک مولد اعداد کاتوره‌ای یکنواخت یک گلوله را بطور کاملا تصادفی انتخاب می‌کنیم و مقدار x آن گلوله را گزارش می‌کنیم. به این ترتیب مقدار گزارش شده کاملا تصادفی و کاتوره‌ای است. از طرف دیگر احتمال گزارش هر مقدار x متناسب با تعداد گلوله‌ها در آن جعبه یا به عبارت دیگر مقدار تابع توزیع است. پس به همین راحتی می‌توان مسئله را حل کرد.
شاید برای مثال ساده، محدود، و یک بعدی که در بالا زدیم این پایان ماجرا باشد ولی استفاده از این روش در حالت کلی با مشکلاتی همراه است. اگر تابع توزیع دامنه‌ای نامحدود داشته باشد که به دلیل محدود بودن تابع توزیع احتمال باید در نقاط خیلی دور به صفر هم میل کند نمی‌توان آن را با روش فوق تولید کرد مگر اینکه مقدار بیشماری گلوله داشته باشیم. حتی در این صورت نیز مجبور به قطع تابع توزیع هستیم اگر بخواهیم این عدد محدود باشد. حتی اگر دقت خود را پایین بیاوریم و تابع را برای مقادیر کوچکتر از حد دقت ما صفر فرض کنیم، بازهم این روش برای توابع چند بعدی نیاز به ثبت حافظه‌ی بسیار زیادی برای نگه داشتن تعداد گلوله‌های بسیار زیاد دارد. و در نهایت نیاز به جارو کردن کل فضای فاز برای چیدن گلوله‌ها نیست که این کار معمولا در مسایل فیزیکی غیرممکن است. ولی نباید نا امید شد. در ادامه راهی برای چیره شدن بر این مشکلات ارایه می‌شود. ولی در ابتدا دینامیکی را معرفی می‌کنیم که در قالب یک بازی بتوان ازآن برای چیدن مناسب گلوله‌ها در جعبه‌ها برای هر تابع توزیع دلخواهی استفاده کرد.

8.4. دینامیک گلوله ها د رجعبه ها

مثال بالا را مجددا، ولی این بار با امکان جابجایی گلوله‌ها در جعبه‌ها نظر می‌گیریم. این جابجایی دینامیکی به گلوله‌ها می‌دهد که نه تنها باعث تغییر در چینش گلوله‌ها در جعبه‌ها می‌شود، بلکه می‌تواند توزیع آنها را نیز تغییر دهد. اگر جابجایی باعث شود که ارتفاع یک ستون پایین بیاید و ارتفاع ستون دیگری بالا برود، شکل تابع توزیع عوض می‌شود. حال سوال این است که آیا این امکان

وجود دارد که قوانین این بازی را به گونه‌ای گذاشت که مستقل از شرایط اولیه بعد از گذشت زمان کافی توزیع گلوله‌ها در جعبه‌ها به تابع $p(x)$ میل کند؟ پاسخ مثبت است. برای یافتن قوانین چنین بازی‌ای اول فرض می‌کنیم که در طی این فرآیند سیستم به توزیع دلخواه ما رسیده باشد. برای اینکه این توزیع پایا باشد انتظار داریم که ادامه‌ی بازی نتواند این توزیع را بهم بزند. یعنی در ادامه‌ی این بازی به طور متوسط باید همانقدر گلوله از هر ستون خارج شود که در طی همان زمان به آن وارد می‌شود. اگر به هر ستون نگاه کنیم انتظار داریم جریان خروج گلوله از این جعبه برابر با جریان ورود به آن جعبه باشد. پس دینامیک این بازی را این گونه معرفی می‌کنیم. گلوله‌ای را به تصادف انتخاب می‌کنیم. این گلوله مثلا در خانه‌ی iام نشسته است. خانه ی دیگری مانند j را نیز به تصادف انتخاب می‌کنیم. اجازه می‌دهیم که گلوله‌ی انتخابی با احتمال w_{ij} به خانه جدید خود برود. چون در هر واحد زمان یک بار این کار را تکرار می‌کنیم، w_{ij} در حقیقت نرخ احتمال گذر (احتمال گذر در واحد زمان) است. بدیهی است که دینامیک این بازی به w_{ij} ها بستگی شدیدی دارد. با توجه به اینه w_{ij} نرخ انتقال گلوله از خانه‌ی i به خانه‌ی j است، نرخ خروج ذره از خانه‌ی i در واحد زمان به دو عامل بستگی دارد؛ (الف) احتمال انتخاب گلوله‌ای در جعبه‌ی iام، $p(x_i)$ و (ب) نرخ انتقال به جعبه‌های دیگر، w_{ij}. از آنجا که گلوله‌هایی که در جعبه‌ی i نشسته‌اند به هر جعبه‌ی دیگری می‌توانند بروند، پس نرخ خروج ذره از این جعبه برابر است با $\sum_j w_{ij}\, p(x_i)$ که جمع بر روی تمام جعبه‌هاست.
از طرف دیگر نرخ ورود گلوله به همین جعبه در همین بازه‌ی زمانی برابر با جمع گلوله‌های ورودی از تمام جعبه‌های دیگر به این جعبه است، $\sum_j w_{ji}\, p(x_j)$. مجددا جمع بر روی تمام جعبه‌هاست. مطابق فرضی که کردیم، سیستم را در زمانی در نظر می‌گیریم که تابع توزیع گلوله‌ها $p(x)$ است. برای اینکه این توزیع پایدار باشد، نرخ ورود و خروج گلوله به هر جعبه باید برابر باشد. در نتیجه شرط ثبات تابع توزیع

$$\sum_j \boldsymbol{w_{ij}\, p(x_i)} = \sum_j \boldsymbol{w_{ji}\, p(x_j)} \qquad (1)$$

است. این شرط به شرط **"توازن"** معروف است. شرط توازن، شرط لازم و کافی است که ثبات تابع توزیع را تضمین می‌کند. ولی می‌توان شرط قوی‌تری که ازنظر کاربردی ساده‌تر است برای این منظور معرفی کرد. کافی است که جمع‌ها را از طرفین رابطه ی بالا حذف کنیم.

$$\boldsymbol{w_{ij} p(x_i) = w_{ji} p(x_j)} \qquad (2)$$

واضح است که در صورت درستی شرط فوق شرط توازن نیز بر قرار خواهد بود. در حقیقت این شرط قوی‌تر از شرط توازن است و برای ثبات تابع توزیع لازم نیست، ولی البته کافی است و آنرا شرط ***توازن جزیی***[36] می‌نامند. توجه کنید که در معادله‌ی توازن جزیی مقادیر $p(x)$ معلوم هستند و سوال یافتن نرخ‌های انتقال بین جعبه‌های متفاوت است. همانطور که می‌بینید یک معادله و دو مجهول داریم. در نتیجه مجموعه نامتناهی از پاسخ وجود دارد. ولی برای هر دسته پاسخی که برای نرخ‌های عبور انتخاب کنیم، تابع توزیع $p(x)$ نقطه‌ی ثابت تحول ما در فضای توابع خواهد بود.

ولی برای اینکه $p(x)$ تابع توزیع پایدار این بازی باشد، باید این نقطه‌ی ثابت جاذب باشد. مطابق معمول برای تست پایداری تعادل فرض می‌کنیم که توزیع گلوله‌ها

[36] Detailed Balance

کمی از $p(x)$ دور شود. به طور مثال در i تعداد گلوله‌ها پایین بیاید و در j تعداد آنها بالا برود. با نگاهی به نرخ انتقال خواهیم دید که در این شرایط احتمال انتخاب و جابجایی گلوله‌ای از جایگاه i کمتر می‌شود و احتمال ورود گلوله‌ها به این جعبه افزایش می‌یابد. در نتیجه با انحرا ف از نقطه‌ی ثابت، تغییرات در جریان گلوله‌ها به گونه‌ای است که سعی در تصحیح این انحراف دارد. این نشان دهنده‌ی پایداری این نقطه ثابت است. پس مستقل از تابع توزیع اولیه گلوله‌ها در جعبه‌ها با شروع بازی و بعد از مدتی سیستم در این نقطه‌ی ثابت جاذب متمایل می‌شود و به تابع توزیع تعادلی خود میل می‌کند و برای ادامه‌ی بازی نیز آن را حفظ می‌کند.
این نشان می‌دهد که لازم نیست در ابتدا گلوله‌ها در جعبه‌ها به شکل خاصی چیده شده باشند. گلوله‌ها را با توزیع دلخواه در بین جعبه‌ها توزیع می‌کنیم و بازی را مطابق توصیف بالا شروع می‌کنیم. بعد از گذشت زمان کافی توزیع گلوله‌ها به توزیع مورد نظر ما نزدیک خواهد شد.
حال اجازه بدهید به موضوع انتخاب پاسخ مناسب برای w_{ij} که باید در شرط توازن جزیی صدق کند، باز گردیم. همانطور که گفته شد یک معادله و دو مجهول داریم که پاسخ‌های بسیاری دارد. ساده ترین پاسخی که به ذهن می‌رسد

$$\boldsymbol{w_{ij} \sim p(j)} \tag{3}$$

است. یعنی نرخ گذر فقط تابعی از جایگاه مقصد است. درست است که این رابطه بسیار ساده است، ولی انتخاب خوبی نیست. به دلیل کوچک بودن نرخ عبور د ر اینجا، دینامیک حرکت ذرات بسیار کند است و در نتیجه زمان خیلی زیادی باید منتظر شد تا سیستم بتواند نقطه ثابت خود را بیابد و به تابع توزیع تعادلی برسد.
انتخاب بسیار پرکاربرد و مهمی که می‌توان برای پاسخ‌ها ارایه کرد، انتخاب متروپولیس[37] است؛

$$\boldsymbol{w_{ij} = \min\left\{1,\ \frac{p(x_i)}{p(x_j)}\right\}.} \tag{4}$$

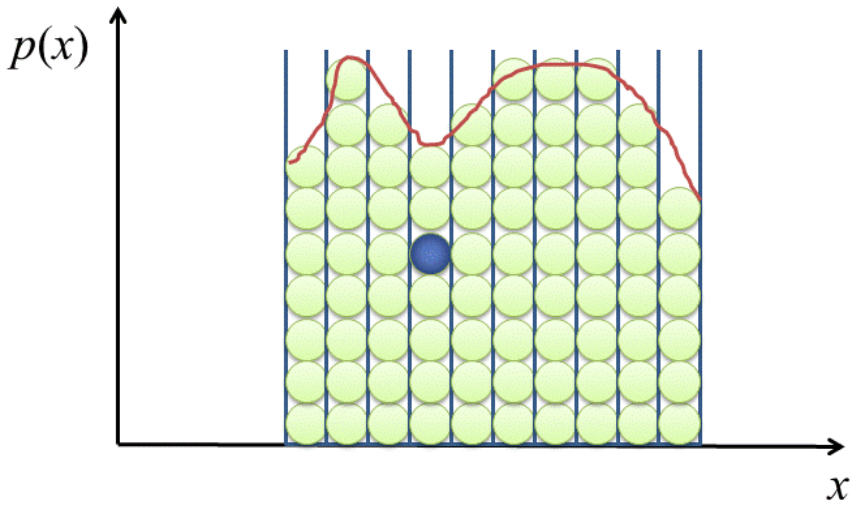
مطابق با این انتخاب، در آلگوریتم متروپولیس یک گلوله انتخاب می‌شود و بعد جایگاه دیگری به تصادف انتخاب می‌شود. اگر مقدار تابع در خانه جدید بالاتر بود گلوله حتما به آن خانه خواهد رفت. در غیر این صورت با نسبت تابع در خانه جدید به خانه‌ی قدیم به گلوله شانس جابجایی داده خواهد شد. همانطور که می‌بینید در اینجا نرخ گذر متناسب با نسبت تابع در این خانه‌هاست. در نتیجه گلوله‌ها شانس بیشتری برای جابجایی دارند و سیستم دارای دینامیکی سریع‌تر است.

8.5. تقلیل گلوله‌ها به یکی

درست است که ما راهی را یافتیم که بعد از گذشت زمان توزیع گلوله‌ها را به توزیع دلخواه ما نزدیک شود، ولی همانطور که قبلا هم اشاره شد این کار هنوز یک مشکل اساسی دارد و آن اینکه برای بدست آوردن توزیع با دقت و تفکیک مناسب نیاز به تعداد بیشماری گلوله داریم. اما برای این مشکل هم راه حلی وجود دارد. بگذارید که مسئله را یک بار دیگر مرور کنیم. قرار است که گلوله‌ای به

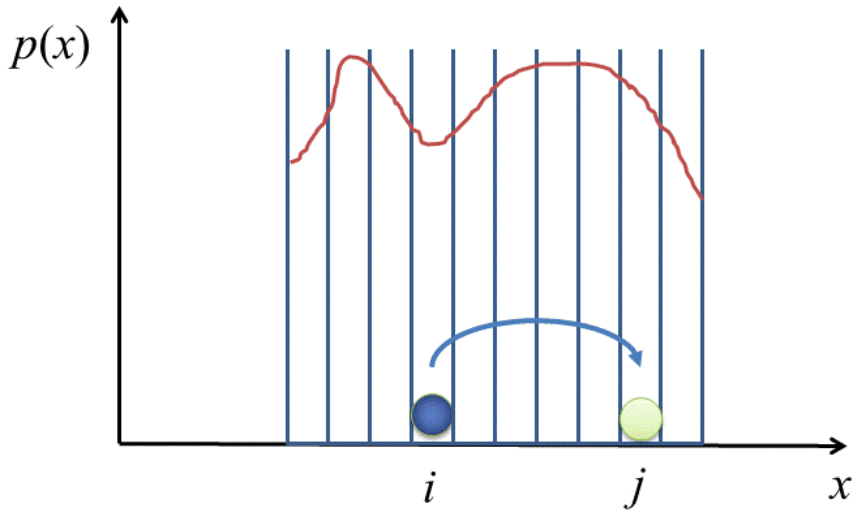
[37] Metropolis

شکل کاملا تصادفی انتخاب شود و مکان آن به عنوان خروجی گزارش شود. اینکه کدام گلوله انتخاب شود اصلا نباید در آمار تاثیری داشته باشد. فرض کنید که در جعبه‌ای که داریم یک گلوله را از بقیه مشخص کنیم. نشان دادیم که اگر بعد از رسیدن به تابع توزیع تعادلی به بازی ادامه دهیم تابع توزیع به هم نمی‌خورد. ولی مطمئنا ادامه بازی در چیدمان گلوله‌ها در خانه‌ها تغییر ایجاد می‌کند. به این معنی که اگر در بازه‌های زمانی نسبتا بزرگ به گلوله‌ها نگاه کنیم این گلوله متمایز در خانه‌ی دیگری یافت خواهد شد. مجددا احتمال حضور این گلوله در هر خانه با تابع توزیع دلخواه ما داده می‌شود. در نتیجه اگر مکان این گلوله در زمان‌های مختلف را گزارش کنیم. دنباله‌ای از اعداد خواهیم داشت که حتما تابع توزیع دلخواه ما را خواهد داشت.



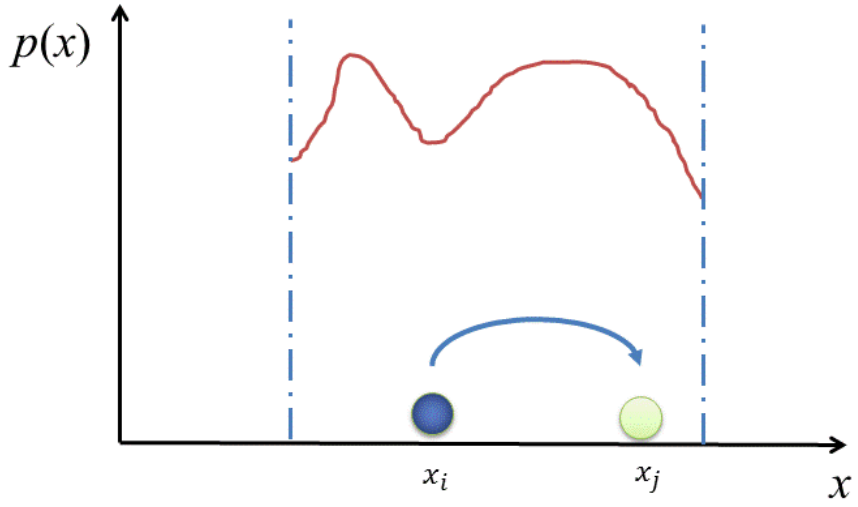
شکل 18 می‌توان از میان تمام گلوله‌ها یکی را نشان کرد و مکان انرا گزارش کرد.

دیدیم که یک مولد اعداد کاتوره‌ای خوب باید دو خاصیت مهم را داشته باشد؛ توزیع مناسب و عدم همبستگی. در مورد تابع توزیع اطمینان بدست آوردیم که جای نگرانی نیست. ولی در مورد عدم همبستگی باید کمی بیشتر دقت کنیم. به این موضوع دوباره باز می‌گیردیم. قبل از آن بهتر است نگاهی به گلوله‌های دیگر هم بیاندازیم. حال که دنباله اعداد کاتوره‌ای با گزارش مکان فقط یک گلوله ساخته می‌شود، چه نیازی به بقیه گلوله‌ها داریم. پاسخ ساده است؛ احتیاجی نداریم.



شکل 19 فقط کافی است که یک گلوله را دنبال کنیم و نیازی به حفظ بقیه نداریم.

فرض کنید که گلوله در لحظه‌ای از زمان در خانه ی i قرار گرفته است. در این بازی فقط کافی است که مکان جدیدی برای گلوله، مثل خانه‌ی j را بطور کاملا تصادفی انتخاب کنیم. احتمال رفتن گلوله از خانه‌ی i به خانه‌ی j با تابع متروپولیس (4) داده می‌شود. این دینامیک ساده باعث می‌شود که احتمال ظهور گلوله در خانه‌ها با تابع توزیع $p(x)$ متناسب باشد. اولین نتیجه ساده و مثبت این فرآیند این است که ما را از حصار خانه بندی رها می‌کند. هیچ دلیلی ندارد که نا پیوستگی خاصی به دلیل وجود خانه‌ها به گلوله‌ی ما تحمیل شود.



شکل 20 لازم نیست که خود را به خانه ها یا شبکه محدود کنیم و گلوله می‌تواند به هر نقطه‌ای در فضا برود.

در هر قدم نقطه‌ای تصادفی در دامنه‌ی تابع توزیع انتخاب می‌شود و گلوله با قاعده‌ی متروپولیس شانس خود را برای رفتن به خانه‌ی جدید امتحان می‌کند.

8.6. آلگوریتم متروپولیس

حال به موضوع عدم همبستگی میان اعداد دنباله بر می‌گردیم. در مثالی که در بالا معرفی کردیم به دلیل اینکه x_j کتره‌ای انتخاب می‌شود پس اعداد دنباله در صورت تغییر باید بدون همبستگی باشند[38]. ولی فراموش نکنیم که همیشه این احتمال وجود دارد که گلوله به مکان جدید نرود. این مشکل برای توابع توزیع با دامنه‌ی نا محدود خیلی جدی‌تر بروز می‌کند. از آنجا که انتگرال یک تابع توزیع باید محدود باشد، برای یک تابع توزیع با دامنه‌ی نا محدود در بیشتر دامنه، تابع توزیع مقداری ناچیز دارد. بدیهی است که برای یک تابع توزیع با دامنه نامحدود انتخاب یک نقطه جدید در دامنه بی معنی است. فرض می‌کنیم که این مشکل را بتوان با قطع کردن دم تابع توزیع در نقاط دور حل کرد. با این وجود گلوله در لحظه‌ای در نقطه‌ای قرار داشته باشد که تابع مقدار قابل توجهی داشته باشد (انتظار نداریم که این اتفاق کم بیافتد). اگر x_j را به صورت کتره‌ای در تمام دامنه انتخاب کنیم در بیشتر نقاط تابع آنقدر کوچک است که هیچ شانسی برای جابجایی گلوله وجود ندارد. در حقیقت گلوله در این نقطه یخ می‌زند و باید مدت‌ها منتظر بماند تا بر حسب تصادف نقطه‌ای که مقدار تابع در آن قابل توجه است انتخاب شود که شاید بتواند به آن نقطه برود.

در قبل ثابت کردیم که آلگوریتم متروپولیس سیستم ما را به تابع توزیع مورد نظرمون هدایت می‌کند. ولی نگفتیم که این کار را در چه زمانی انجام می‌دهد. بدیهی است که برای یافتن نقطه ثابت (تعادل) در فضای توابع، سیستم باید بتواند این فضا را جستجو کند. یعنی برای یافتن این نقطه باید دینامیک داشته باشد. این انتظار که هرچه دینامیک سیستم سریع‌تر باشد، رسیدن به نقطه ثابت نیز راحتتر خواهد بود کاملا معقول است. پس با این حساب در صورت داشتن یک تابع توزیع متمرکز با دامنه‌ی گسترده، دینامیک سیستم آنقدر کند است که برای همگرا شدن به نقطه ثابت زمانی بسیار طولانی لازم است، و این اصلا مطلوب نیست. برای حل این مشکل می‌توان با کوتاه کردن طول قدم‌ها دینامیک سیستم را تسریع کرد. در صورتی که تابع مورد نظر تابعی هموار باشد (که معمولا برای توابع توزیع احتمال فرض درستی است)، انتظار می‌رود که مقدار تابع در همسایگی خیلی تغییر نکند. پس اگر قدم‌هایی که برای یافتن مکان جدید بر می‌داریم در همسایگی نقطه‌ی توقف فعلی گلوله باشد، نسبت $p(x_j)/p(x_i)$ آنقدر مقدار دارد که شانس جابجایی را به گلوله بدهد. به این ترتیب گلوله به حرکت در می‌آید و به سیستم این شانس را می‌دهد که به سوی نقطه ثابت در فضای توابع جذب شود. پس می‌توان شکل نهایی آلگوریتم متروپولیس را نوشت.

---

[38] بهتر است بگوییم که عدم همبستگی معادل با مولد مورد استفاده را دارد.

آلگوریتم متروپولیس:

1. $x = x_0$
"مقدار دهی اولیه (بهتر است که نقطه شروع در مکانی باشد که تابع مقدار قابل توجهی داشته باشد)"
2. $y = x + \Delta\, Rand(-1,1)$
طول قدم $= \Delta$
3. If $Rand(0,1) < p(y)/p(x)$ Then $y = x$
" در صورتی که شرط متروپولیس بر آورده شود قدم قبول می‌شود
4. Loop (2) + (3)
"قدم‌های اصلی باید در یک حلقه قرار داده شوند"

آلگوریتم فوق به با تبدیل y,x و Δ به بردارهای d مولفه‌ای می‌تواند راحتی برای توابع توزیع d بُعدی (متغیره) به کار برده شود.

8.7. نرخ قبولی و انتخاب طول قدم

انتخاب طول قدم به شدت بر دینامیک و زمان رسیدن به تعادل تاثیر می‌گذارد. در حالت حدی طول قدم‌های خیلی بزرگ ما را به شکل قبلی مسئله و مشکل یخ زدگی و توقف گلوله می‌رساند. در این حالت بیشتر تلاش‌هایی که متحرک برای جابجایی بر می‌دارد ناکام می‌ماند. در سوی دیگر اگر قدم‌ها خیلی کوتاه انتخاب شود، با وجود اینکه گلوله تقریبا در هر قدم جابجا می‌شود، این جابجایی آنقدر کوتاه است که برای گشتن فضای فاز باز نیاز به زمان خیلی زیاد است. اگر نسبت قدم‌هایی که برای جابجایی قبول می‌شوند به کل تلاش‌های مونت کارلو را "نرخ قبولی"[39] a_r بنامیم، دیدیم که که هیچ یک از دو حد $a_r \to 0$ و $a_r \to 1$ برای سیستم ما مطلوب نیستند. بدون آنکه در اینجا اثبات کنیم می‌پذیریم که بهترین حالت برای سیستم که درسریع‌ترین زمان بتواند به تابع توزیع دلخواه برسد وقتی است که $a_r \approx 0.5$ باشد.
تنظیم نرخ قبول با تنظیم طول قدم امکان پذیر است. برای بالا بردن نرخ قبول باید طول قدم را کوچک کرد و افزایش طول قدم نرخ قبولی را کاهش می‌دهد. این انتظار وجود ندارد که مقدار نرخ قبولی دقیقا بر روی مقدار نیم تنظیم شود. در عمل $0.3 < a_r < 0.7$ دینامیک قابل قبولی به سیستم ما می‌دهد.

8.8. طول همبستگی

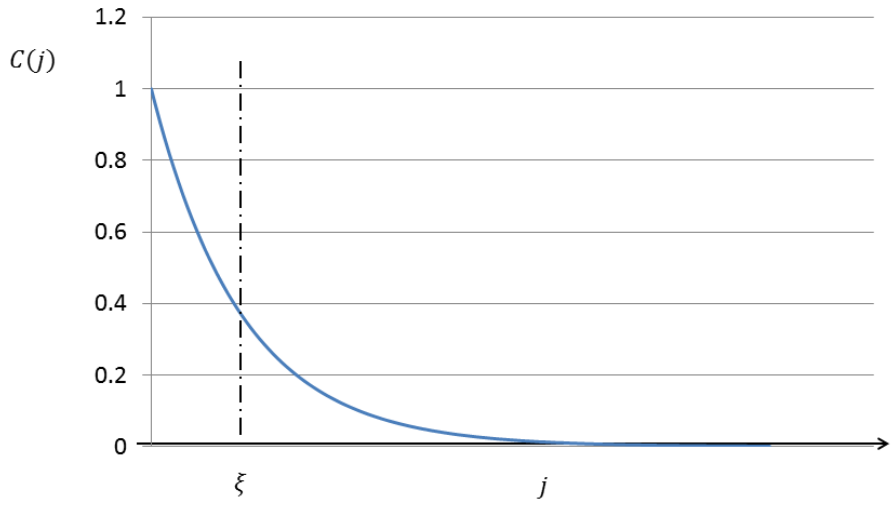
تا کنون صحبت‌های غیر دقیق و کیفی از همبستگی میان دنبالهی اعداد و نیز زمان لازم برای اینکه تابع توزیع تعادلی بدست آید داشته ایم. حال زمان آن رسیده

[39] Acceptance rate

که این مفاهیم را کمی کنیم. اگر خروجی مولد ما دنباله اعداد $\{x_1, x_2, \dots, x_i, \dots\}$ باشد، تابع خود همبستگی به شکل زیر معرفی می‌شود:

$$C(j) = \frac{\langle x_i x_{i+j}\rangle_i - \langle x_i\rangle_i \langle x_{i+j}\rangle_i}{\sigma^2} \quad (5)$$

که در این رابطه $\sigma^2 = \langle x^2\rangle - \langle x\rangle^2$ مجذور انحراف از معیار دنباله است و منظور از $\langle \dots\rangle_i$ متوسطگیری بر روی اندیس i است. به راحتی می‌توان دید که $C(0) = 1$ است. برای j های خیلی بزرگ مقادیر x از یکدیگر مستقل می‌شوند. از آنجا که متوسط حاصل ضرب دو کمیت تصادفی و مستقل برابر است با حاصل ضرب متوسط آن کمیت‌ها، مقدار خود همبستگی در این حد هم قابل پیش بینی است و داریم $C(j \to \infty) \to 0$. برای بیشتر فرآیندهای تصادفی این تابع به صورت نمایی افت می‌کند و می‌توان آن را به خوبی به تابع $C(j) = e^{-\frac{j}{\xi}}$ برازش داد. دراین رابطه ξ زمان واهلش یا زمان همبستگی نامیده می‌شود.



شکل 21 تابع خود همبستگی به صورت نمایی با زمان افت می‌کند و در زمان واهلش به $\frac{1}{e}$ می‌رسد

این نشان می‌دهد که همبستگی عددهایی به فاصله ی ξ در دنباله به مقدار $1/e$ کاهش یافته و با تقریب خوبی می‌توان آنها را مستقل فرض کرد. از طول همبستگی دو استفاده‌ی مهم می‌کنیم. اول اینکه می‌توانیم تعداد عددهای کاتوره‌ای مستقل ازهم در دنباله را بدست آوریم. مثلا اگر دنباله 10^6 عدد داشته باشد ولی زمان همبستگی 100 باشد یعنی در این دنباله 10^4 عدد مستقل وجود دارد. نکته دیگر این است که زمان همبستگی میعاری از عدم وابستگی به گذشته است. یعنی این عدد میعاری است از اینکه چقدر باید صبر کرد تا خروجی های مولد مستقل از شرایط اولیه بشود. این همان چیزی است که دنبال آن می‌گشتیم؛ زمان واهلش یا تعادل سیستم. این زمانی است که باید منتظر بشویم تا گلوله‌ها به توزیع دلخواه ما برسند. در عمل ما بیش از یک زمان همبستگی برای اطمینان به رسیدن به تعادل منتظر میشویم.

.8.1	1. با روش متروپولیس مولدی برای تولید اعداد کتره‌ای با توزیع گوسی بسازید. 2. طول قدم‌ها را به گونه‌ای تعیین کنید که نرخ قبولی مقادیر $\{0.1, 0.2, \ldots, 0.9\}$ را داشته باشد. 3. برای تمام نرخ‌های قبولی فوق طول همبستگی را بیابید.
تمرین	

نکته 1:

برای این که شرط تعادل جزیی به درستی اعمال شود لازم است که احتمال تلاش در رفتن از یک نقطه به نقطه‌ی دیگر با احتمال تلاش برای برگشت برابر باشد. اهمیت این نکته در مثال‌های پیچیده‌تر در بخش‌های بعدی بیشتر مشخص می‌شود. ولی به عنوان ساده‌ترین مثال برای زمانی که با دستگاه‌های قطبی یا کروی کار می‌کنیم باید به سهم ژاکوبی در عنصر حجم برای تلاش در جابجایی ها دقت کرد.

بیشتر بدانیم:

روش مونت کارلو و متروپولیس در بیشتر کتاب‌های مقدماتی شبیه سازی به خوبی بحث می‌شوند. یکی ازمعروفترین پیشگامان در استفاده از این روش و معرفی کاربردهای آن در علوم مختلف Kurt Binder استاد پیشکسوت دانشگاه ماینز است که کتاب‌های متعددی در این رابطه نوشته است. این کتاب‌های از مقدمات مونت کارلو تا کاربردهای تخصصی و پیشرفته‌ی آن را پوشش می‌دهند. به خوانندگانی که می‌خواهند در این موضوع بیشتر بدانند، با جستجو در کتاب‌های این پژوهشگر متناسب با دانش و علاقه خود مطمئنا می‌توانند مباحث جالبی را بیابند.
شاید مقاله متروپولیس و همکارانش:

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller,

"Equation of State Calculations by Fast Computing Machines". *Journal of Chemical Physics* **21,** page:

1087–1092, Year 1953,

بعد از گذشت بیش از نیم قرن کمی قدیمی به نظر بیاید ولی مطالعه آن هنوز هم ارزشمند است. شاید ایده‌ای که در این مقاله مطرح شده است به نظر ساده بیاید ولی اگر تعداد مقالاتی که در آنها نام متروپولیس آورده شده است را جستجو کنید شاید تعجب کنید که ببینید این تعداد هم مرتبه با تعداد مقالاتی است که در آن‌ها نام افرادی مانند شرودینگر یا بوهر یا دیراک آمده است. شاید به این طریق بتوانید به عمق اثر این مقاله‌ی تاریخی در علم پی ببرید.

9. شبیه سازی آنسامل کانونی NVT[40]

اکنون به جایی رسیده ایم که میتوانیم به موضوع اصلی کتاب که شبیه سازی سیستم های آماری بپردازیم. از دید مکانیک آماری، مشاهده پذیرهای فیزیک را میتوان با متوسط گیری آنسامبلی[41] بدست آورد. احتمال حضور سیستمی که در تعادل ترمودینامیکی با یک منبع گرمایی به دمای T است در یکی از ریز حالت[42] های آن با تابع احتمال بولتزمن[43] داده میشود. فرض کنید نقطه‌ی در فضای فاز این سیستم را با $\vec{x}$ نشان دهیم. دراین صورت $\vec{x}$ برداری با بعد فضای فاز است. به طور مثال اگر سیستم ما یک گاز ایده آل با N ذره باشد، این فضا $6N$ مولفه دارد. 3 درجه آزادی برای مکان و سه درجه آزادی برای سرعت هر یک از ذرات گاز. در این صورت کمیت فیزیکی $A(\vec{x})$ تابعی از مختصات سیستم در فضای فاز است. اندازه گیری این کمیت در زبان مکانیک آماری معادل با متوسط گیری از آن بر روی تابع توزیع کانونیک است.

$$\langle A \rangle = \frac{\int_\Omega A(\vec{x})\, e^{-\beta E(\vec{x})} \mathrm{d}v}{\int_\Omega e^{-\beta E(\vec{x})} \mathrm{d}v}$$

که در اینجا $\beta = 1/k_B T$ است که درآن $k_B = 1.38 \times 10^{-23}\ \mathrm{m^2 kg\, s^{-2} K^{-1}}$، ثابت بولتزمن است. انتگرال بر روی تمام فضای فاز سیستم، Ω، است. عبارت مخرج تابع پارش[44] سیستم یا در واقع ضریب یکه سازی تابع توزیع بولتزمن است. به این شکل می‌توان شکل یکه‌ی تابع توزیع بولتزمن را به صورت زیر معرفی کرد:

$$p(\vec{x}) = \frac{e^{-\beta E(\vec{x})}}{\int_\Omega e^{-\beta E(\vec{x})} \mathrm{d}v}$$

به این ترتیب متوسط گیری بالا به شکل آشنای

$$\langle A \rangle = \int_\Omega A(\vec{x})\, p(\vec{x})\, \mathrm{d}v$$

[40] Canonical ensemble
[41] Ensemble averaging
[42] Micro state
[43] Boltzmann
[44] Partition Function

در میاید. در نتیجه اگر بتوانیم فضای فاز را با تابع توزیع $p(\vec{x})$ نمونه برداری کنیم، اندازه گیری کمیت A به سادگی با متوسط گیری بر روی آن بدست میاید. اما این کاری است که اکنون میدانم که چگونه انجام دهیم. با استفاده از روش متروپولیس ما قادریم در فضای فاز به گونه ای گشت و گذر کنیم که احتمال حضور در هر نقطه ای با وزن تابع بولتزمن متناسب باشد.

با توجه به بحثی که داشتیم باید برای اینکه دینامیک مونت کارلو به ما اجازه دهد که در زمان قابل دسترس به تابع توزیع تعادلی بولتزمن برسیم، باید قدمهایی با طولهای کوتاه برای رفتن به همسایگی ها برداریم.

9.3. استفاده از آلگوریتم متروپولیس برای نمونه برداری از آنسامبل کانونی

همانطور که دیدیم برای بدست آوردن هر کمیت مشاهده پذیر فیزیکی در یک آنسامبل کانونی کافی است که این آنسامبل را به کمک تابع توزیع بولتزمن بسازیم و از آن نمونه برداری کنیم. اولین نکته ای که باید در ذهن نگه دارید این است که در شبیه سازی های متروپولیس، منظور شبیه سازی دینامیک سیستم نیست. یعنی قدم هایی که برداشته میشود هیچ ربطی به معادلات حرکت که دینامیک سیستم را تعیین میکند ندارد. این قدمها فقط برای جابجا شدن در فضای فاز به منظور نمونه برداری از آن است. در نتیجه این قدمها نه تنها رابطه دینامیکی با هم ندارند بلکه امید ما این است که هیچ گونه همبستگی نداشته باشند تا مولد کاتوره ای ما برای نمونه برداری قابل باشد.

در این گونه شبیه سازیها تعداد ذرات سیستم، N، حجم سیستم، V، و دمای آن، T، ثابت است. برای همین به آن شبیه سازی در آنسامبل NVT گفته میشود. این ساده ترین آنسامبلی است که میتوان با روش متروپولیس شبیه سازی کرد. شبیه سازی آنسامبل های دیگر نیز امکان پذیر است ولی برای دیگر آنسامبل ها نیاز به تمهیداتی است که کمیت ها ی مورد نظر را در طی شبیه سازی ثابت نگه دارد. به طور مثال در آینده به شبیه سازی آنسامبل میکرو کانونی[45]، NVE، خواهیم پرداخت و نشان میدهیم که برای ثابت نگه داشتن انرژی باید چه تغییراتی در الگوریتم داد.

مشابه الگوریتم یک بعدی، در اینجا نیز باید یک گشت را تولید کرد. البته این بار گشت ما در فضای حقیقی نیست و در فضای فاز است. در هر نقطه از این فضا، مقدار تمام متغیر های مربوط مسئله مشخص است و در نتیجه میتوانیم هر اندازه پذیر را نیز نمونه برداری کرد. در نتیجه میتوانیم آلگوریتم را به شکل زیر باز نویسی کنیم.

[45] Microcanonical ensemble

آلگوریتم متروپولیس برای آنسامبل NVT:

1. $\vec{x} = \vec{x}_0$
   "یک آرایش ابتدایی برای سیستم پیش نهاد میکنیم."
2. $\vec{y} = \vec{x} + \vec{R}$
   $\vec{R}$ برداری کاتوره ای با حد اکثر طول برابر با طول قدم است.
3. If $Rand(0,1) < p(\vec{y})/p(\vec{x})$ Then accept
   " در صورتی که شرط متروپولیس بر آورده شود قدم قبول میشود"
4. Loop (2) + (3)
   "قدم های اصلی باید در یک حلقه قرار داده شوند"

همانطور که در قدم سوم در بالا مشاهده می‌کنید، برای پذیرش یا عدم پذیرش ما به نسبت تابع توزیع احتمال نیاز داریم. این به این معنی است که مقدار تابع پارش که در مخرج تابع توزیع در معاله ی ؟؟ ظاهر شد عنصر مهمی در شبیه سازی ما نیست و به آن نیازی نیست. این نکته به شدت اهمیت دارد. برای مشخص شدن اهمیت آن توجه به این نکته ضروری است که منظور نهایی در مکانیک آماری مانند هر جای دیگر در فیزیک اندازه گیری است. یعنی ابزار مکانیک آماری برای محاسبه مقادیر کمیت های مشاهده پذیر فیزیکی به کار میرود که در دیدگاه مکانیک آماری معادل با متوسط گیری از کمیت بر روی آنسامبل متناظر با مسئله است. از مکانیک آماری میدانیم که با دانستن تابع پارش ما قادریم که دیگر کمیت های فیزیکی را نیز به دست آوریم. به طور مثال متوسط انرژی با مشتق گیری از تابع پارش نسبت به β بدست می آید. برای همین در شکل تحلیلی مکانیک آماری حل یک مسئله فیزیکی معادل با محاسبه ی تابع پارش آن است. مسایل زیادی وجود دارند که ما میتوانیم تابع پارش انها را محاسبه کنیم، ولی از طرف دیگر تعداد مسایلی که برای آن ها حل دقیق نداریم نیز کم نیستند.

در روش متروپولیس ما تابع پارش را محاسبه نمیگنیم. در حقیقت نیازی به محاسبه ی ان نداریم. آن چیزی که نیاز مندیم نمونه هایی با تابع توزیع بولتزمن است، و برای بدست اوردن آنها نیز فقط به نسبت آنها نیاز است،

$$\frac{p(\vec{y})}{p(\vec{x})} = \frac{e^{-\beta E(\vec{x})}}{e^{-\beta E(\vec{y})}}$$

و در این محاسبه نیز فقط باید بتوانیم انرژی سیستم را در هر نقطه فضای فاز محاسبه کنیم. از آنجا که برای هر مسئله در فیزیک اولین داده ی ما و در واقع آن چیزی که سیستم با آن مشخص میشود، هامیلتونی سیستم است، پس ما مطمئنا انرژی را میتوانیم بدست اوریم. در نتیجه در روش شبیه سازی متروپولیس هیچ مسئله غیر قابل حلی وجود ندارد. این مزیت مهم این ابزار نسبت به روشهای تحلیلی است. ولی فراموش نکنیم که خودِ انتگرال تابع پارش را نمیتوانیم محاسبه کنیم. برای محاسبه این انتگرال اگر از روش نمونه برداری ساده استفاده کنیم به دلیل بزرگ بودن فضای تابع و ابعاد مسئله به جواب خوبی نمیرسیم. اگر از روش نمونه بردای هوشمند نیز بخواهیم استفاده کنیم باید تابع توزیع را یکه کنیم که در نتیجه به جواب بدیهی $1 = 1$ میرسیم.

9.4. قدم زمانی مونت کارلو

در گذشته دیدیم که بهتر است قدمهای مونت کارلو را در همسایگی نقطه فعلی برداریم. یعنی در هر تلاش برای جابجایی به سیستم اجازه دهیم که نقطه ای در همسایگی نقطه ای که اکنون ایستاده است را امتحان کند. برای سیستم آماری ما که تعداد متغییر های سیستم و در نتیجه بُعد فضای فاز بسیار زیاد است، میتوان همسایگی را با تغییر فقط یکی از متغیر ها در حالی که بقیه ثابت اند تعریف کنیم. این که کدام یک از متغیر ها شانس تغیر را دارد را میتوانیم به طور تصادفی تعیین کنیم. در این صورت یک قدم زمانی مونت کارلو برابر با تعداد تلاش هایی است که به هر یک از متغیر های مسئله به طور متوسط حد اقل یک بار شانس برای تغییر داده شده باشد. توجه کنید که این قدم زمانی هیچ ربطی به زمان واقعی ندارد. دوباره یاد آوری میکنیم که در مونت کارلو ما

دینامیک را شبیه سازی نمیکنیم و فقط در تلاشیم از یک سیستم در تعادل ترمودینامیکی با یک حمام گرمایی نمونه برداری صحیح کنیم تا بتوانیم کمیت های فیزیکی مربوط را به درستی متوسط بگیریم. در نتیجه یک واحد زمان مونت کارلو فقط تعداد عملیات محاسباتی را تعیین میکند.

9.5. اصل ارگادیک و قدم های مونت کارلو

مطابق با اصل ارگادیک که بنیادی ترین اصل مکانیک آماری است، سیستم در زمان محدود از همسایگی هر نقطه ای در فضای فاز میگذرد. به زبان ساده تر، تمام نقاط فضای فاز برای سیستم قابل دسترس اند. پس ما باید دقت کنیم که قدم هایی که برای تلاش های مونت کارلو تعریف میکنیم این اصل را نقض نکنند. یعنی این قدمها به سیستم اجازه بدهند که در طی شبیه سازی احتمال حضور درنقاط مختلف فضای فاز غیر صفر باشد.

9.6. محاسبه ی تغییر انرژی به جای انرژی

مطابق با رابطه ؟؟ ما برای تعیین احتمال قبول در هر تلاش باید دو مقدار انرژی را داشته باشیم، انرژی سیستم در حال حاضر و در جابجایی فرضی. از این دو انرژی نیازی به محاسبه ی انرژی سیستم در حال حاضر نداریم زیرا آنرا قبلا محاسبه کرده ایم. ولی باید انرژی سیستم را در نقطه جدید محاسبه کنیم. اگر فرض کنیم که تابع انرژی پتانسیل سیستم فقط شامل جملات انرژی جفت ذرات باشد برای بدست آوردن انرژی باید N^2 محاسبه انجام شود. ولی اگر رابطه ؟؟ را به صورت

$$\frac{p(\vec{y})}{p(\vec{x})} = \frac{e^{-\beta E(\vec{y})}}{e^{-\beta E(\vec{x})}} = e^{-\beta \Delta E}$$

بنویسیم که در آن $\Delta E = E(\vec{y}) - E(\vec{x})$، اختلاف انرژی به ازای جابجایی احتمالی سیستم است. از آنجا که ما فقط یک درجه ی آزادی سیستم را تغییر میدهیم و بقیه درجات آزادی تغییر نمیکند محاسبه ی ΔE از مرتبه N است. این تغییر کوچک در آلگوریتم مرتبه آلگوریتم را یک واحد کاهش میدهد که بسیار مهم است. بدیهی است که برای بدست آوردن انرژی کل سیستم نیز از این محاسبه میتوان کمک گرفت. یعنی کافیست که انرژی حالت قبل با ΔE جمع شود.

نکته 1:

در خیلی از مسایل محاسبه ΔE کمی پیچیده است. در این گونه مسایل توصیه میشود که در مراحل دیباگ کردن کُد انرژی را از هر دو روش محاسبه کنید و در هر قدم با هم مقایسه کنید ودر صورت مشاهده اختلاف پیام خطا بدهید. در صورت عدم بروز مشکل برای اجرای اصلی محاسبه ی انرژی کل را خاموش کنید.

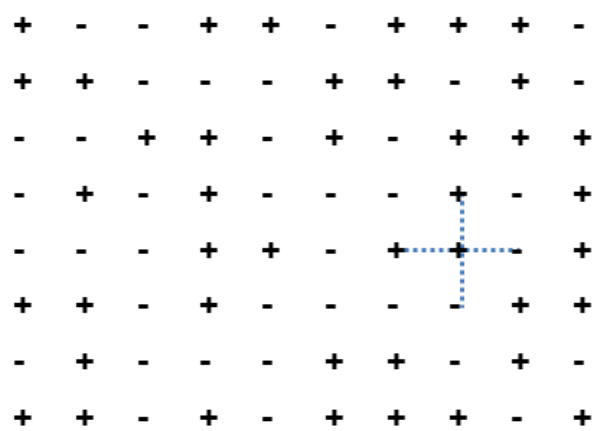
در فصل های بعدی با مثال هایی آشنا میشویم که شبیه سازی مونت کارلو میتواند برای حل مسایل کمک کند.

بیشتر بدانیم:

به بیشتر بدانیم بخش قبل مراجعه شود.

10. مدل آیزینگ[46]

یکی از مدل‌های بسیار پر کاربرد و مهم در مکانیک آماری مدل آیزینگ است. این مدل که اولین بار برای مدل سازی سیستم-های مغناطیسی پیشنهاد شد، برای بسیاری از مسایل دیگر نیز کاربر یافته است. نکته‌ی مهم این مدل اینجاست که به خوبی می‌تواند رفتار بحرانی و تغییر فاز پیوسته را نشان دهد. در این مدل فرض می‌شود بر روی رئوس یک شبکه، دو قطبی‌های مغناطیسی نشسته‌اند. دو قطبی‌ها با میدان دو قطبی‌های دیگر و میدان خارجی برهمکنش دارند. فرض می‌شود که برهمکنش با دیگر دوقطبی‌ها کوتاه برد است و در ساده‌ترین تقریب هر دوقطبی فقط با همسایه‌های اولش برهمکنش می‌کند. این مدل بسیار ساده است و فرض بر آن است که هر دو قطبی فقط می‌تواند دو مقدار 1+ و 1- را اختیار کند. به این علت معمولا این دوقطبی‌ها را اسپین می‌نامند. پس یک نمونه از این سیستم آرایشی از اسپین‌ها با مقادیر 1+ و 1- است.



شکل 22 نمایی شماتیک از یک مدل دوبعدی آیزینگ با آرایشی از اسپین‌ها. هر اسپین فقط با همسایگان نزدیکش بر همکنش دارد (خط چین آبی).

در نتیجه اگر سیستم N اسپین داشته باشد، فضای فاز آن یک فضای گسسته‌ی N بعدی است که هر نقطه‌ی آن با یک بردار N مولفه‌ای، $\vec{s} = \{s_1, s_2, \dots, s_i, \dots, s_N\}$، تعریف می‌شود که مولفه‌ها فقط می‌تواند مقادیر 1+ یا 1- باشد. اگر این بردار را با یک بردار 3 بعدی مقایسه کنیم، هر نقطه‌ی این فضا در یکی از رئوس یک مکعب N بعدی نشسته است. در نتیجه، با مسئله‌ای مواجه هستیم که فضای فاز آن گسسته است.

[46] Ising model

انرژی (یا هامیلتونی) این سیستم به شکل زیر نوشته می‌شود.

$$E(\vec{s}) = -J \sum_{<ij>} s_i\, s_j - h \sum_i s_i$$

در رابطه‌ی بالا J مقیاس انرژی برهمکنش و مقداری مثبت است و $< ij >$ به معنی جمع بر روی تمام همسایه‌های اول است. عبارت دوم برهمکنش میان دوقطبی‌ها با میدان خارجی h را مشخص می‌کند. در ادامه، بدون آنکه از کلیت بحث کاسته شود فرض می‌کنیم که $h = 0$، و فقط عبارت اول را در نظر خواهیم گرفت.

10.3. شکست تقارن در مدل آیزینگ

در ابتدا قبل از شروع بحث شبیه سازی این مدل، اجازه دهید کمی بیشتر با این مدل آشنا شویم. ضریب منفی پشت جمع درعبارت انرژی نشان می‌دهد که اسپین‌های هم جوار ترجیح می‌دهند که در حالت‌های هم سان باشند. به این معنی که هر دو مثبت یا منفی باشند. در نتیجه به راحتی می‌توان حدس زد که آرایش کمینه‌ی انرژی برای این سیستم، آرایشی است که تمام اسپین‌ها مثبت یا منفی باشد. زیرا هر اختلاف در علامت همسایه‌ها باعث افزایش انرژی سیستم خواهد شد. با اینکه این آرایش کمینه‌ی انرژی است، ولی کمینه‌ی انرژی آزاد[47] (F) سیستم نیست. برای روشن شدن این نکته آرایشی را در نظر بگیرید که همه‌ی اسپین‌ها به غیر از یکی مثبت هستند. هر چند این ساختار به اندازه $2J$ انرژی بیشتری نسبت به آرایش کمینه انرژی دارد، ولی به دلیل اینکه محل قرارگیری این تک اسپینِ مخالف می‌تواند در هر کدام از N رأس موجود در سیستم باشد، آنتروپی[48] (S) این سیستم بیشتر است.

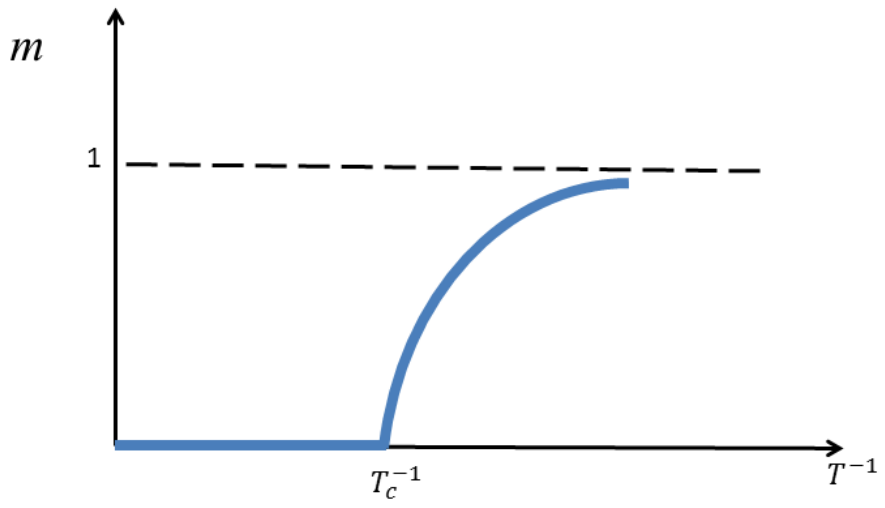
اگر مثال بالا را تعمیم دهیم می‌توان نشان داد که بیشترین آنتروپی برای حالتی است که نیمی از اسپین‌ها مثبت و نیمی منفی است. به این شکل می‌توان حدس زد که در دماهای بالا به دلیل اهمیت دما (T) در انرژی آزاد،

$$F = E - TS$$

کمینه‌ی انرژی آزاد برای آرایش‌های با آنتروپی بالا اتفاق می‌افتد و در نتیجه جمع مغناطش این آرایش‌ها صفر می‌شود. به این ترتیب سیستم در دماهای بالا فاقد مغناطش ذاتی است. اما با کاهش دما سهم عبارت دوم در انرژی آزاد کم شده و کمینه‌ی انرژی آزاد مربوط به حالت‌هایی از سیستم خواهد بود که اندازه‌ی عبارت اول نیز کوچک باشد. در نتیجه ترجیح اسپین‌ها در دماهای پایین این است که

[47] Free energy
[48] Entropy

در آرایشی قرار گیرند که با همسایه‌هایشان هم جهت باشند و در نتیجه مغناطظ سیستم غیر صفر است. نکته‌ی جالب این مدل اینجاست که به خوبی رفتاری مشابه تغییر فاز سیستم‌های مغناطیسی از فاز پارامغناطیس به فاز فِرومغناطیس را نشان می‌دهد.



شکل 23 تغییرات مغناطش بر حسب دما در مدل آیزینگ تغییر فازی از پارامغناطیس (مغناطش ذاتی صفر) به فرومغناطیس نشان می‌دهد

کمیت مغناطش، که در این سیستم می‌تواند معیاری از نظم سیستم باشد، به کمیت نظم[49] معروف است. تغییر فاز در کمیت نظم از مقدار غیر صفر به صفر در دمایی اتفاق می‌افتد که آنرا دمای بحرانی سیستم، T_c، می‌نامند. برای دماهای بالاتر از این مقدار بحرانی، تقارن مثبت – منفی بر سیستم حاکم است ولی در دماهای پایین، سیستم باید یکی از دو حالت ممکن یا همه مثبت یا همه منفی را انتخاب کند. برای همین این مدل به خوبی یکی از مهم‌ترین پدیده‌های فیزیک آماری (و ماده‌ی چگال) یعنی شکست خود به خود تقارن[50] را نشان می‌دهد و از این نظر شبیه سازی آن نه تنها برای ما از نظر محاسباتی ارزش آموزشی دارد، برای درک بهتر فرآیند شکست خود به خود تقارن نیز بسیار مفید است.

یکی از خواص مهم سیستم‌هایی که رفتار بحرانی نشان می‌دهند این است که نه تنها کمیت نظم در نقطه‌ی بحرانی تغییراتی غیر عادی نشان می‌دهد، بلکه در رفتار دیگر مشاهده پذیرهای ترمودینامیک و یا مشتقات آنها نیز می‌توان ناپیوستگی یا تکینگی را مشاهده کرد. در مورد مدل آیزینگ نیز به این گونه است و در نقطه بحرانی سیستم، دو کمیت ظرفیت گرمایی ویژه، C_v، و پذیرفتاری مغناطیسی، χ، دارای تکینگی است.

[49] Order parameter
[50] Spontaneous symmetry breaking

از مکانیک آماری می‌دانیم که دلیل اصلی این تکینگی‌ها بینهایت شدن طول همبستگی، ξ، این سیستم‌ها در نقطه‌ی بحرانی است. این خاصیت را در مورد مدل تراوش در بخش‌های قبلی مشاهده کردیم. طول همبستگی در مدل آیزینگ را کمی جلوتر به شکل ریاضی تعریف خواهیم کرد. در اینجا کافی‌است بدانیم این طول میعاری از همبستگی مکانی اسپین‌ها را مشخص می‌کند. به دلیل نامحدود شدن طول همبستگی در دمای بحرانی، سیستم بدون مقیاس می‌شود و تمام کمیت‌های فیزیک در این نقطه باید رفتاری آزاد-مقیاس[51] یا توانی داشته باشند. به این ترتیب برای رفتار سیستم در نزدیکی نقطه‌ی بحرانی، توان‌های بحرانی زیرمعرفی می‌شود.

$$m \sim (T_c - T)^{\beta}$$

$$\chi \sim |T_c - T|^{-\gamma}$$

$$C_v \sim |T_c - T|^{-\alpha}$$

$$\xi \sim |T_c - T|^{-\nu}$$

مدل آیزینگ در بُعد یک و دو دارای حل دقیق است. این مدل در یک بُعد رفتار بحرانی در دمای محدود نشان نمی‌دهد. ولی در بُعد 2 به خوبی تغییر فاز را نشان می‌دهد. در بُعد 3 حل دقیق ندارد و تمام اطلاعات ما از این مدل یا از شبیه سازی‌های کامپیوتری بدست می‌آید و یا از روش‌های تقریبی که کیفیت تقریب نیز از مقایسه با نتایج شبیه سازی مشخص می‌شود. حل یک بعدی بسیار ساده و بدیهی است، حل دو بعدی بسیار تکنیکی و پیچیده ولی ممکن است. و حل سه بعدی وجود ندارد. اگر به این نکته توجه کنیم که شبیه سازی مدل آیزینگ در اندازه‌ی شبکه‌های متفاوت خیلی شبیه به هم است و تنها طول زمان شبیه سازی برای بدست آوردن پاسخی دقیق با اندازه‌ی شبکه رشد می‌کند، به اهمیت شبیه سازی پی می‌بریم.

10.4. شبیه سازی مدل آیزینگ با استفاده از آلگوریتم متروپولیس

[51] Scale free

در اینجا می‌خواهیم الگوریتم متروپولیس را برای مدل گسسته‌ی آیزینگ بکار ببریم. همانطور که قبلا اشاره شد هر آرایش سیستم در فضای فاز با برداری N بعدی که مولفه‌های آن می‌توانند 1+ یا 1- باشند مشخص می‌شود.

10.4.1. شرایط اولیه

بهتر است که برای شروع شبیه سازی سیستم را با آرایشی آغاز کنیم که خیلی خاص (غیر تصادفی) نباشد. برای همین پیشنهاد می‌کنیم بعد از مشخص کردن اندازه‌ی شبکه، یک آرایش کاملا تصادفی برای شروع کار در نظر بگیریم. در شبیه سازی دو بُعدی کافی است آرایه‌ای دو بعدی متناظر با اندازه‌های شبکه در نظر بگیریم که عناصر آن با مقادیر کاتوره‌ای 1+ و 1- مقدار دهی شده باشد.

نکته 1:

بهتر است که برای ساختار دادن به کُد خود برنامه را به شکل تابعی بنویسیم. برای این کار بهتر است که تابعی با نام ()init معرفی کنید و تمام امور مربوط به آماده سازی شرایط اولیه را در آن قرار دهیم. در آینده در صورتی که نیاز به کمیتی ایجاد شود که حتما باید در ابتدای شبیه سازی معرفی شود یا مقدار اولیه داشته باشد، آن را در داخل این تابع قرار خواهیم داد. در حال حاضر کمیت‌هایی که باید معرفی شوند اضافه بر اندازه های شبکه، L، و مقادیر اولیه اسپین‌ها، دما و ضریب انرژی J است.

10.4.2. قدم مونت کارلو برای مدل آیزینگ

برای ادامه کار باید قسمت اصلی شبیه سازی که حلقه‌ی مرکزی متروپولیس است را ایجاد کنیم. این حلقه با انتخاب یک آرایش در همسایگی نقطه‌ی فعلی برای تلاش درجابجایی در فضای فاز شروع می‌شود. در اینجا به دلیل گسسته بودن فضای فاز، دست ما برای انتخاب طول قدم برای جابجایی باز نیست. بهتر است که کوتاه ترین قدم را در نظر بگیریم. کوتاه ترین قدم معادل است با تغییر یکی از اسپین‌ها. یعنی کافی است که یکی از اسپین‌ها به طور کتره‌ای انتخاب شود و در یک منفی ضرب شود. البته این کار نباید تا امتحان شرط متروپولیس و قبول تلاش نهایی شود.

10.4.3. محاسبه ی انرژی

برای انجام شبیه سازی متروپولیس محاسبه‌ی انرژی یکی از کارهایی که حتما باید انجام داد. برای همین توصیه می‌شود که این محاسبه نیز در تابعی به نام energy() انجام گیرد. برای تحقیق شرط متروپولیس در هر تلاش باید انرژی حالت فعلی با انرژی حالت مورد بررسی مقایسه شود. همانطور که در قبل اشاره شد به دلیل محدود بودن قدم‌های تلاش (اسپین از مثبت به منفی یا برعکس)، محاسبه‌ی تغییر در انرژی به ازای قدمی که قرار است برداشته شود هزینه‌ی کمتری دارد. در مورد مدل آیزینگ محاسبه‌ی انرژی نیاز به $2N$ محاسبه به ازای تمام رابط های بین اسپینی، دارد. در صورتی که تغییر در علامت یک اسپین فقط در چهار جمله از این $2N$ جمله تاثیر می‌گذارد. از آنجا که حلقه‌ی متروپولیس قلب اصلی این شبیه سازی است، کاهش مرتبه‌ی محاسبه در اینجا بسیار اهمیت دارد. برای همین به غیر از تابعی که انرژی را محاسبه می‌کند به تابعی احتیاج داریم که تغییر در انرژی را به ازای جهش علامت یکی از اسپین‌ها گزارش کند.

از آنجا که برای محاسبه احتمال قبول شدن تلاش نیاز به محاسبه عامل بولتزمن $e^{-\beta\Delta E}$ است، در تمام محاسبات ضریت $\beta J = J/k_BT$ با هم ظاهر می‌شود. برای کاهش تعداد عملیات ضرب و تقسیم می‌توان به جای استفاده از واحدهای فیزیکی از واحدی استفاده کنیم که در آن این عامل به یک پارامتر کاهش یابد. مثالا اگر واحد انرژی را k_BT بگیریم، J تنها پارامتر موثر مسئله خواهد بود (البته در این واحد). امکان دیگر تثبیت J/k_B به عنوان واحد انرژی و ورود دما به مسئله در این واحد است. این گونه واحد های انتخابی که در شبیه سازی به به منظور کاهش عملیات ریاضی انتخاب می‌شوند را واحدهای کاهیده[52] می‌نامند.

نکته 2:

در مورد مسئله آیزینگ می‌توان محاسبه‌ی عامل احتمال متروپولیس را به شکل بسیار بهینه‌ای نوشت. فرض کنید که در قدمی از شبیه سازی، تلاش برای تغییر علامت s_{ij} (اسپین در مختصات i و j) مورد بررسی باشد. در این حالت تغییر در انرژی به شکل

$$\Delta E = -2\, s_{ij}\, (s_{i+1j} + s_{i-1j} + s_{ij+1} + s_{ij-1})$$

داده می‌شود. عبارت داخل پرانتز عددی مضرب دو خواهد بود. پس $\Delta E \in \{-8, -4, 0, 4, 8\}$ است. به دلیل محدودیت در مقادیر ممکن ΔE می‌توان ضریت بولتزمن را برای این پنج مقدار در ابتدای شبیه سازی در یک آرایه ی پنج عضوی ذخیره کرد و بر حسب مورد فرآخوانی کرد. این کار به مراتب از محاسبه‌ی تابع نمایی در هر قدم از آلگوریتم کم هزینه تراست.

10.4.4. شرایط مرزی دوره ای

[52] Reduced units

در تمام مسایلی که بر همکنش‌های چند ذره‌ای (تقریبا تمام مسایل فیزیک) در هامیلتونی سیستم وجود دارد، سوال مهمی مطرح است و آن اینکه با ذراتی که بر روی مرز سیستم می‌نشینند چه کنیم. توجه کنید که در سیستم‌های ترمودینامیک (تعداد ذرات به بینهایت میل کند) تعداد این ذرات در مقایسه با ذراتی که در حجم قرار دارند انقدر کوچک است که از اثر آنها می‌توان چشم پوشی کرد و هر گونه همکنشی برای آنها فرض شود تاثیری بر فیزیک نخواهد داشت. ولی در مسایلی که ما شبیه سازی می‌کنیم، به دلیل محدودیت‌های محاسباتی، اندازه‌ی سیستم بسیار کوچک است و این ذرات نقش موثر و قابل مشاهده‌ای را خواهند داشت. پس نتایج شبیه سازی به نحوه‌ی برخورد با آنها وابسته خواهد بود.

روش‌های متفاوتی برای در نظر گرفتن ذرات سطحی وجود دارد، ولی یکی از پر کاربرد ترین آنها که اثر سطح را تا حد امکان کاهش می‌دهد استفاده از شرایط مرزی دوره‌ای است. مثال زیر می‌تواند درک این شرایط مرزی را آسان کند.

فرض کنید که سیستم محدودی با اندازه ای 5×5 داریم؛

$$\begin{matrix} + & - & - & + & + \\ + & + & - & - & - \\ - & - & + & + & - \\ - & + & - & + & - \\ - & - & - & + & + \end{matrix}$$

همانطور که در این تصویر می‌بیند 16 عدد از 25 اسپین این مدل، یعنی بیش از نیمی از آنها، بر روی مرز نشسته‌اند. در این آرایش عناصر مرزی کمتر از 4 همسایه دارند و در صورت عدم تمهیدی برای از بین بردن اثرات مرزی این اسپین‌ها بر فیزیک مسئله تاثیر قابل توجی می‌گذارند.

در واقع در این تصویر 16 اسپین از 9 اسپین دیگر متمایز هستند. برای از بین بردن این تمایز می‌توان این آرایش را در فضا تکرار کرد. یعنی تصاویر این سیستم را از راست وچپ، بالا و پایین در فضا تکثیر کرد به گونه ای که بنظر برسد که سیستم ما تا بینهایت از هر سو ادامه دارد. البته به خوبی می‌دانیم که این تصویر نا متناهی از تناوب سیستم محدود ما تولید شده است.

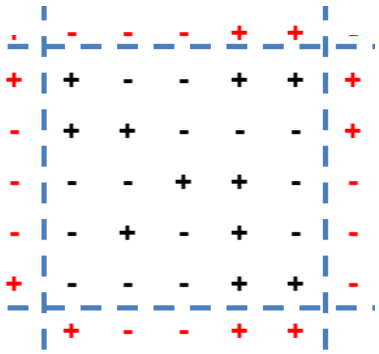
+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
-	-	+	+	-	-	-	+	+	-	-	-	+	+	-
-	+	-	+	-	-	+	-	+	-	-	+	-	+	-
-	-	-	+	+	-	-	-	+	+	-	-	-	+	+
+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
-	-	+	+	-	-	-	+	+	-	-	-	+	+	-
-	+	-	+	-	-	+	-	+	-	-	+	-	+	-
-	-	-	+	+	-	-	-	+	+	-	-	-	+	+
+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
-	-	+	+	-	-	-	+	+	-	-	-	+	+	-
-	+	-	+	-	-	+	-	+	-	-	+	-	+	-
-	-	-	+	+	-	-	-	+	+	-	-	-	+	+

در اینجا برای تمایز سیستم اولیه از تصاویر هم از رنگ استفاده شده است و هم مرزها با خط چین نمایش داده شده است. ولی اگر این علایم ناپدید شود (با توجه به نامحدود بودن این مجموعه) کسی نمی‌تواند محل مرز اولیه و سیستمی که تکثیر شده است را تشخیص دهد. تنها چیزی که قابل مشاهده است وجود دوره تناوب است. ولی سیستم اصلی می‌تواند هر مربع 5×5 دلخواهی باشد.

+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
-	-	+	+	-	-	-	+	+	-	-	-	+	+	-
-	+	-	+	-	-	+	-	+	-	-	+	-	+	-
-	-	-	+	+	-	-	-	+	+	-	-	-	+	+
+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
-	-	+	+	-	-	-	+	+	-	-	-	+	+	-
-	+	-	+	-	-	+	-	+	-	-	+	-	+	-
-	-	-	+	+	-	-	-	+	+	-	-	-	+	+
+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
+	+	-	-	-	+	+	-	-	-	+	+	-	-	-
-	-	+	+	-	-	-	+	+	-	-	-	+	+	-
-	+	-	+	-	-	+	-	+	-	-	+	-	+	-
-	-	-	+	+	-	-	-	+	+	-	-	-	+	+

به این ترتیب هیچ تمایزی بین نقاط این شبکه تناوبی نیست و می‌توان برای انرژی این آرایش، برهمکنش هر ذره با همسیگانش را در نظر گرفت. به این ترتیب اثر مرز حذف شده است، هر چند به دلیل تناوب مسئله اثر طول محدود همچنان پا برجاست و نباید فراموش کنیم که در هر صورت ما در حال شبیه سازی سیستمی به طول 5 هستیم و ما باید انرژی در واحد سیستم محدود را محاسبه کنیم. یعنی فقط برای 25 ذره‌ای که در سیستم وجود دارند.

بدلیل کوتاه برد بودن برهمکنش در مدل آیزینگ نیازی به نگه داشتن تمام تصاویر نداریم. چون هر ذره فقط با همسایه اول خود برهمکنش دارد. در نتیجه سیستم محدود زیر که در آن فقط یک ردیف از مجموعه تصاویر در دو سوی سیستم اصلی نگه داشته شده است معادل سیستم بینهایت گسترده و تناوبی ماست.



10.5. تصحیح اندازه ی محدود

10.6. طول همبستگی

در بخش‌های قبل با تعریف طول همبستگی زمانی یک پارامتر آشنا شدیم. در این مسئله‌ی آیزینگ طول همبستگی مکانی برای ما اهمیت دارد. مفهوم کیفی این طول این است که در صورتی که یک اسپین در مکان مشخصی از شبکه جهت‌گیری خود را تغییر دهد بر جهت گیری چه شعاعی از اسپین‌های اطراف آن تاثیر خواهد گذاشت. در دماهای بالا به علت اینکه افت خیز دمایی سهم زیادی در جهت گیری اسپین‌ها نسبت به برهمکنش همسایگی دارد، با تقریب خوبی می‌توان گفت که طول همبستگی مکانی اسپین‌ها صفر است. یعنی هر اسپین مستقل از آرایش همسایگانش می‌تواند تغییر جهت دهد. با کاهش دما و کوچک شدن سهم جمله‌ی انترپی در انرژی آزاد تعداد آرایش‌های ممکن برای همسایگان هر اسپین محدود می‌شود. محدودیت در تعداد آرایش‌های ممکن اسپین‌ها باعث می‌شود که با تغییر جهت یک اسپین همسایگانی دورتر از همسایگی اول نیز تحت تاثیر قرار گیرند. شعاع همسایگانی که تحت تاثیر یک اسپین قرار می‌گیرد طول همبستگی مکانی سیستم است.

باید اضافه شود.

.9.1	1. کدی برای شبیه سازی مدل آیزینگ دو بعدی با روش متروپولیس آماده کنید. 2. رفتار کمیتهای $\lvert\langle m\rangle\rvert, \chi, C_v$ و ξ را برحسب دما (بخصوص در اطراف دمای بحرانی) بررسی کنید. 3. با فرض اینکه رفتار بحرانی برای ظرفیت گرمایی، تکینگی لگاریتمی دارد، $c_v = c_0 \ln(\lvert T - T_c\rvert)$ با اجرای برنامه برای سیستم هایی با اندازه های متفاوت نماهای بحرانی β, γ, ν و ضریب c_0 را برای آیزینگ دو بعدی بدست آورید.
تمرین	

بیشتر بدانیم

11. مثال مدل پیوسته - گاز واندر والس

11.1. پتانسیل لنارد-جونز

شکل پتانسیل معرفی شود.

تعریف پتانسیل کوتاه برد و بلند برد.

11.2. شعاع قطع پتانسیل

11.3. لیست همسایگی

12. مثال سیستم با قیود پیچیده - ساختارهای پلیمری

12.1. مدل درشت دانه پلیمر

12.2. برهمکنش ها

12.3. حرکت های موضعی

12.3.1. گوشه

12.3.2. انتها

12.3.3. میل لنگی

12.4. حرکت های سراسری

12.4.1. لولایی

12.4.2. میل لنگی بلند

12.4.3. خزشی

13. الگوریتم بهینه سازِ مونت کارلو

14. شبیه سازی آنسامبل میکروکانونی NVE

14.1. آنسامبل میکروکانونی

Ejtehadi, M. (2005). *Computational Physics Course in Sharif*. Retrieved from http://Physics.sharif.edu/~compphys