

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 واسطه 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 شاخه‌هایی از دانش و فن‌آوری 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. معمولاً به این گونه 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 محدوده‌ی اعتبار خاصی است و از طرف دیگر آزمایشگر هم در آزمایشگاه با محدودیت‌های 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 روش‌ها 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 سیستم‌ها در مقیاس‌های چند نانو متری فقط آرزویی برای آزمایشگران بود، در صورتی که در شبیه سازی‌ها بر راحتی قابل دسترس بود. شاید فن‌آوری نانو در حال حاضر مدیون این شبیه سازی‌ها باشد.

تعداد بیشماری نرم‌افزارهای آماده به منظور به کارگیری در شبیه سازی مدل‌های متفاوت تهیه شده‌اند. در میان آنها نرم‌افزارهای 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 باز¹ در اختیار عموم قرار می‌گیرد تا کاربران ضمن استفاده از این نرم‌افزارها بتوانند در تصحیح، تکمیل و یا گسترش آنها مشارکت داشته باشند. ولی معمولاً این نرم‌افزارها دقیقاً با مسئله مورد نظر شما هماهنگ نیستند و زمان لازم برای فراگیری کار با آنها و ایجاد تغییرات

¹ Open source

متناسب با مسئله مورد نظر بیش از نوشتن یک برنامه جدید وقت می‌برد. هرچند من همواره 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 کارلو² (metropolis) معروف است، آشنا کند. کتاب به گونه این مرتب شده که با الگوریتم‌های 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. شبیه سازی 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 در شبیه سازی از یک برنامه کامپیوتری (یا الگوریتم) 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 دقیقاً 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

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

سازی کنیم و حتی قادر نیستیم که به نزدیکی عدد آووگادرو برسیم. حتی برای سیستم‌های بسیار کوچکتر شبیه سازی چند صد نانو ثانیه یک شبیه بسیار بزرگ به حساب می‌آید. یا برای شبیه سازی یک کلهکشان بایستی از ستاره هایی کوچکتر از خورشید صرفنظر کنیم. شاید این گونه تصور شود که با افزایش قدرت کامپیوترها 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 الگوریتم‌هایی شبیه سازی می‌شوند که به آنها NP گفته می‌شود. در این الگوریتمها 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. پیشرفت‌های 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 نکند.

1.3. شبیه سازی با استفاده از الگوریتم‌های تصادفی

الگوریتم‌های متفاوتی برای شبیه سازی یک پدیده فیزیکی وجود دارد. انتخاب الگوریتم مناسب به نوع مسئله و پاسخهای مورد توجه بستگی دارد. معمولا در الگوریتم‌های 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 الگوریتم‌ها وجود دارند که تصادف اساس الگوریتم است. این الگوریتم‌ها به الگوریتم‌های مونت کارلو معروف هستند. در این کتاب ما خود را به آشنایی با این الگوریتم‌ها، که در مقابل الگوریتم‌های 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 می‌توان از دخالت دادن عواملی مانند وضعیت هوا و زمین، سرعت باد، سلامت بازیکنان و احتمال آسیب دیدگی آنها چشم پوشید. دسته دیگر الگوریتم‌هایی 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 توصیه‌هایی 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 می‌توانید به زبان‌های برنامه نویسی متفاوتی برنامه بنویسید ولی به دلایلی زبانهای FORTRAN 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.

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 آلوگوریتم 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 اشیاء 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 بسته ای 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 بسته ای 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 خستگی و تا جایی که می توانید 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 برنامه ای که نوشته اید 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.

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 می شود ولی برای همراهی با این کتاب دانش برنامه نویسی ابتدایی با یک زبان ساده و میانی مانند Python، 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. بنابراین در این کتاب برای تمرینها نمونه حل های ساده به زبان Python ارائه شده است.

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