Раздел 1. Общие принципы моделирования

Лабораторная № 1

Сравнение аналитического и имитационного моделирования движения частиц в поле потенциальных сил.

Цель: овладение навыками аналитического и имитационного моделирования, исследование устойчивости различных типов моделей.

Краткая теория.

В настоящее время принято разделять моделирование на *аналитическое* и *имитационное*. При аналитическом моделировании изучаются математические (абстрактные) модели реального объекта в виде алгебраических, дифференциальных и других уравнений, при этом предусматривается осуществление однозначной вычислительной процедуры, приводящей к их точному решению. При имитационном моделировании исследуются математические модели в виде алгоритма, воспроизводящего функционирование исследуемой системы путём последовательного выполнения большого количества элементарных операций.

При аналитическом моделировании модели представляются в виде уравнений, имеющих аналитические решения. Известно, что уравнения, лежащие в основе современного естествознания, очень редко и только в простых ситуациях имеют известные аналитические решения. Поэтому для изучения свойств системы методом аналитического моделирования необходимо строить весьма упрощённые модели. Зато, имея аналитическое решение, можно сразу описать состояние системы в любой момент времени или при любых значениях параметров, характеризующих её поведение. Для имитационного моделирования сложность модели (а соответственно и исследуемой системы) не представляет принципиального ограничения, хотя она и может привести к большим затратам вычислительных ресурсов.

Имитационные модели часто представляют в виде компьютерной программы, которая шаг за шагом воспроизводит события, происходящие в реальной системе. Ещё одним преимуществом имитационного моделирования является, то, что с его помощью можно достаточно просто описывать случайные процессы.

Рассмотрим движение материальной точки массы m под действием центральной силы произвольно зависящей только от расстояния между точкой и центром силы. Такая сила потенциальна и стационарна. Предположим, что сила в любой точке направлена к центру. Таковым, в частности, является сила, действующая на заряженное тело, создаваемая полем точечного заряда. В этом случае материальная точка движется в одной плоскости. Помещая начало отсчета в центр силы, и вводя полярные координаты (r,φ), можно получить следующие уравнения движения

$$\begin{cases} m\dfrac{d^2r}{dt^2} = \dfrac{\partial U(r)}{\partial r} \\ \dfrac{d^2\phi}{dt^2} = 0 \end{cases}, \qquad (1.1)$$

где $U(r)$ – потенциал поля силы. Данная система уравнений допускает аналитическое решение. Его также можно решить численными методами, что будет соответствовать имитационному моделированию.

Решение систем, подобных (1.1) может быть как устойчивой, так и неустойчивой. *Устойчивыми* называются модели, в которых малое изменение входных параметров приводит так же к малым изменениям выходных параметров. Если же малое воздействие на вход модели может вызвать сколь угодно большое значение её выходных параметров, то модель называется *неустойчивой*. Неустойчивость модели не обязательно связана с неустойчивостью моделируемой системы. Она может возникнуть на стадии формализации, когда математическая задача, соответствующая модели оказывается *некорректно поставленной*, см. например [1].

Порядок выполнения.

1. Получить аналитическое решение системы (1.1) с потенциалом и начальными условиями, заданными преподавателем.

2. На основе системы (1.1) провести имитационное моделирование задачи с теми же потенциалом и начальными условиям, при которых было получено аналитическое решение. Разработать для этого алгоритм решения (1.1) методом Эйлера.

3. Внести в начальные данные случайные возмущения, согласно указаниям преподавателя. Выполнить пункты 1 и 2, используя возмущённые начальные данные.

4. Провести сравнение результатов аналитического и имитационного моделирования.

Контрольные вопросы.

1. Что такое аналитическое моделирование?
2. Что такое имитационное моделирование?
3. В чем отличие аналитического моделирования от имитационного?
4. Можно ли решать посредством имитационного моделирования задачи, имеющие аналитические модели?
5. Какой тип моделирования целесообразно применять для решения задачи, рассматриваемой в данной работе?
6. Объяснить понятие устойчивости для системы и для её модели?
7. Могут ли быть неустойчивые модели у устойчивых систем, и наоборот, могут ли быть устойчивые модели у неустойчивых систем?
8. Проанализировать устойчивость системы, рассматриваемой в данной работе.

Лабораторная № 2

Оптимизация алгоритмов решения комбинаторных моделей.

Цель: овладение методом решения комбинаторных моделей посредством перебора, знакомство с основными принципами его оптимизации.

Краткая теория.

Важным качеством модели является её *экономичность*, которую следует рассматривать в двух аспектах. Первый аспект можно определить понятием *простота*. Сюда входят минимизация составляющих модель элементов (уравнений, переменных и т.п.), хорошая интерпретируемость, наглядность и ряд других подобных факторов. Вторым аспектом экономичности является максимально возможное сокращение вычислительных ресурсов, требуемых для исполнения модели. Несмотря на то, что современные суперкомпьютеры имеют огромную производительность, многие модели не могут быть решены на них за приемлемое время. Поэтому задача оптимизации моделей с целью сокращения времени компьютерного счёта является очень актуальной. Ниже рассматриваются две модели, имеющие важные приложения в экономике.

Задача коммивояжёра – одна из самых известных задач комбинаторной оптимизации, заключающаяся в поиске самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешёвый, совокупный критерий и тому подобное) и соответствующие матрицы расстояний, стоимости и тому подобного. Как правило, указывается, что маршрут должен проходить через каждый город только один раз. Существует несколько частных случаев общей постановки задачи коммивояжёра, в частности, *геометрическая задача*, когда матрица расстояний отражает расстояния между точками на плоскости, *метрическая задача*, когда на матрице стоимостей выполняется неравенство треугольника, *симметричная* и *асимметричная* задачи. Также существует *обобщённая задача коммивояжёра*.

Чисто теоретически задача коммивояжера может быть решена методом *полного перебора*, т. е. путем перебора всех возможных маршрутов, вычисления их характеристик (длины, стоимости и т. д.) и последующего выбора оптимального (согласно условию задачи) маршрута. Однако уже при относительно небольшом числе городов (66 и более) она не может быть решена методом перебора вариантов никакими мыслимыми компьютерами за время, меньшее нескольких миллиардов лет. Все сокращающие полный перебор методы решения задачи коммивояжёр являются эвристическими. В их большинстве находится не самый эффективный маршрут, а приближённое решение. Далее перечислены некоторые из них.

Метод ветвей и границ позволяют находить оптимальные или приблизительные решения для достаточно больших задач. Рассматривается единичный куб в -мерном пространстве. Каждое ребро куба соответствует маршруту между двумя городами. По определенным правилам строятся гиперплоскости, отсекающие участки куба, содержащие число рёбер, меньшее числа городов. Подробно метод описан в работе [2]. *Метод эластичной сети*, здесь каждый из маршрутов коммивояжёра рассматривался как отображение окружности на плоскость (в каждый город на плоскости отображается некоторая точка этой окружности). Соседние точки на окружности должны отображаться в точки, по возможности ближайшие и на плоскости, подробности см. в [2]. *Алгоритм динамического программирования*, его основная идея заключается в вычислении и запоминании пути от исходного города и до каждого из остальных городов, затем суммирования этой величины с путем из каждого из остальных городов до оставшихся городов и т. д. Для решения задачи коммивояжера используются также *муравьиный алгоритм*, *генетический алгоритм* и другие.

Традиционная постановка *задачи Джонсона* состоит в следующем: требуется выбрать порядок обработки деталей, обеспечивающее минимальное суммарное время выполнения всего задания, а именно за минимальное время осуществить обработку группы из m деталей, каждая из которых должна последовательно пройти обработку на каждом из n станков, образующих технологическую линию. Предполагаются заданными величины t_{ij} – время обработки i-ой детали ($i=1,\dots,m$) на j-ом станке ($j=1,\dots,n$). Также как и задача коммивояжера, задача Джонсона решается методом полного перебора. При этом большинство оптимизирующих перебор эвристических методов, используемых для задачи коммивояжера, могут после соответствующей коррекции использоваться для решения задачи Джонсона. Однако для некоторых частных случаев задачи Джонсона разработаны свои специфические алгоритмы решения, один из которых рассмотрен ниже.

Весьма простым алгоритмом нахождения оптимальной последовательности обработки *m* деталей на двух станках является *алгоритм Джонсона*. Он включает следующие шаги:

1) выбирается деталь с наименьшей продолжительностью обработки на одном из станков (таких деталей может быть, в этом случае берётся одна из них);
2) выбранная деталь помещается в начало очереди, если наименьшая длительность обработки соответствует первому станку, или в конец очереди, если соответствует второму станку;
3) строка матрицы времен t_{ij}, соответствующая выбранной детали исключается из дальнейшего рассмотрения (вычеркивается);
4) выбирается деталь среди оставшихся со следующей наименьшей продолжительностью обработки на одном из станков;
5) выбранная деталь помещается ближе к началу или к концу очереди по указанному в шаге 2 правилу;
6) если определена очередность запуска для всех деталей, то решение получено, иначе переходим к шагу 3.

В итоге реализации данного алгоритма можно получить оптимальное расписание работы двух станков

Порядок выполнения.

Вариант 1.

1. Разработать компьютерную программу решения задачи о коммивояжере методом полного перебора.

2. Разработать компьютерную программу решения задачи о коммивояжере оптимизированным методом, указанным преподавателем.

3. Результаты работы представить в виде зависимости от числа пунктов маршрута отношения времени решения задачи методом полного перебора ко времени её решения оптимизированным методом.

Вариант 2.

1. Разработать компьютерную программу решения задачи Джонсона для двух станков методом полного перебора.

2. Разработать компьютерную программу решения задачи Джонсона для двух станков алгоритмом Джонсона.

3. Результаты работы представить в виде зависимости от числа деталей отношения времени решения задачи методом полного перебора ко времени её решения алгоритмом Джонсона.

Контрольные вопросы.

1. Объяснить суть метода полного перебора.
2. В чём состоят преимущества и недостатки полного перебора?
3. Каким образом полный перебор оптимизируется методом ветвей и границ?
4. Объяснить в чём состоит методическая и практическая значимость задачи о коммивояжере.
5. Как формулируется задача Джонсона для числа станков более чем два?
6. Объяснить алгоритм Джонсона решения задачи для двух станков.
7. Как в общем случае оценивается трудоёмкость решения задачи Джонсона методом полного перебора.
8. Как меняется трудоёмкость решения задачи Джонсона методом полного перебора, если изменять порядок подачи деталей на каждый из станков?
9. Объяснить в чём состоит методическая и практическая значимость задачи Джонсона.

Раздел 2. Моделирование управления в распределенных системах

Лабораторная № 3

Моделирование управления параллельными процессами.

Цель: знакомство с основными принципами управления в системах с параллельными процессами и разделяемыми ресурсами.

Краткая теория.

Практически все современные операционные (ОС) работают в *мультизадачном* (*многозадачном*) *режиме*. Под этим подразумевается свойство ОС обеспечивать возможность параллельной (или псевдопараллельной) обработки нескольких задач. Истинная многозадачность ОС возможна только в распределённых вычислительных системах. Существует два типа многозадачности. *Процессная многозадачность*, основанная на процессах – одновременно выполняющихся программах. Здесь программа – наименьший элемент кода, которым может управлять планировщик ОС. *Поточная многозадачность*, основанная на потоках. Наименьший элемент управляемого кода – поток. Потоки возникают в случае, когда одна программа может выполнять две и более задачи одновременно.

Примитивные многозадачные ОС обеспечивают полное разделение ресурсов. В этом случае за каждой задачей закрепляется определённый участок памяти, и она активизируется в строго определённые интервалы времени. Более развитые ОС проводят *распределение ресурсов динамически*. Работа такой многозадачной среды осуществляется следующим образом. Каждая задача имеет свой приоритет, в соответствии с которым получает процессорное время и оперативную память. По окончании положенного времени, отсчитываемого системным таймером, ОС переводит задачу из состояния выполнения в состояние готовности, отдавая ресурсы другим задачам. Причём, при нехватке памяти страницы невыполняющихся задач могут быть вытеснены на диск (такая процедура называется свопингом). Передача управления происходит после обработки прерываний, которые в штатной ситуации поступают, как правило, от таймера.

Для обеспечения надёжной работы распределённой системы ОС должна выполнять следующие функции: защищать адресного пространства своего ядра и выполняющихся задач от несанкционированного вмешательства других программ; решать конфликты доступа к ресурсам и устройствам, не допуская тупиковых ситуаций, вызванных ожиданием заблокированных ресурсов; распознавать сбои отдельных задач и прекращать их. Кроме надёжности, многозадачная среда должна быть эффективной, в частности она должна обеспечивать быструю коммуникацию между процессами. Затраты ресурсов на её поддержание не должны: мешать процессам, замедляя их работу или ограничивая требуемую память.

Далее описаны некоторые способы реализации многозадачности. *Невытесняющая многозадачность* – ОС одновременно загружает в память два или более приложений, но процессорное время предоставляется только основному приложению. Для выполнения фонового приложения оно должно быть активизировано. Подобная многозадачность может быть реализована не только в операционной системе, но и с помощью программ-переключателей задач.

Совместная или корпоративная многозадачность – это тип многозадачности, при котором следующая задача выполняется только после того, как текущая задача явно объявит себя готовой отдать процессорное время другим задачам. Кооперативную многозадачность можно назвать многозадачностью «второй ступени», поскольку она использует более передовые методы, чем простое переключение задач, реализованное многими известными программами. При простом переключении активная программа получает все процессорное время, а фоновые приложения полностью замораживаются. При кооперативной многозадачности приложение может захватить фактически столько процессорного времени, сколько оно считает нужным. Все приложения делят процессорное время, периодически передавая управление следующей задаче. Преимуществом кооперативной многозадачности является отсутствие необходимости защищать все разделяемые структуры данных объектами типа критических секций. В качестве её недостатков можно назвать неспособность всех приложений работать в случае ошибки в одном из них; трудность реализации многозадачной архитектуры ввода-вывода в ядре ОС, которая бы позволяла процессору исполнять одну задачу в то время, как другая задача инициировала операцию ввода-вывода и ждет её завершения.

Вытесняющая, или приоритетная, многозадачность – это вид многозадачности, в котором ОС сама передает управление от одной выполняемой программы другой в случае завершения операций ввода-вывода, возникновения прерываний от таймера и т. д. В этом виде многозадачности процессор может быть переключен с исполнения программы между любыми двумя инструкциями в её коде. Распределение процессорного времени осуществляется планировщиком процессов. Каждой задаче может быть назначен пользователем или самой операционной системой определенный приоритет, что обеспечивает гибкое управление распределением процессорного времени. Преимущества этого вида многозадачности состоят в следующем: возможность полной реализации многозадачного ввода-вывода в ядре ОС, высокая надежность системы в целом, в сочетании с использованием защиты памяти, возможность полного использования многопроцессорных и многоядерных систем. Это наиболее распространенный вид многозадачности, в частности, он применяется в пользовательском режиме (а часто и в режиме ядра) всех UNIX-подобных ОС, во всех поколениях ОС Windows, а также во многих других операционных системах.

Отметим, что многозадачность может быть реализована не только в операционной, но и языковой среде. Например, спецификации языков программирования *Modula-2* и *Ada* допускают поддержку многозадачности вне привязки к какой-либо ОС. В частности, реализация языка программирования TopSpeed Модула-2 позволяет организовывать кооперативную и вытесняющую многозадачности для потоков одной программы в рамках такой принципиально однозадачной операционной системы, как MS-DOS. Это осуществляется путём включения в модуль программы компактного планировщика задач, содержащего обработчик прерываний таймера. Языки программирования, обладающие таким свойством, иногда называют *языками реального времени*.

Порядок выполнения.

1. Разработать модель управления доступом нескольких параллельных процессов к одному файлу. Предполагается, что в системе имеются процессы чтения и записи, обращающиеся к рассматриваемому файлу. Одновременно с ним могут работать несколько процессов чтения. Но если работает процесс записи, то никакой другой процесс не может обратиться к файлу, пока запись полностью не завершится. Также считается, что передача управления между процессами происходит случайным образом. Причём вероятность того, что управление будет передано некоторому процессу пропорциональна его приоритету.

2. Построить алгоритм, соответствующий разработанной модели. Функционирование компьютера представляется бесконечным циклом, при входе в очередной период которого происходит передача управления между процессами. За один период может быть считан (записан) один символ из файла. В случае чтения символ помещается в буфер процесса на первое свободное место (заполнение буфера производится последовательно). Когда управление передаётся процессу записи, все процессы чтения прекращаются.

3. Результаты работы представить в виде зависимостей числа считанных символов каждым из процессов чтения от приоритета процесса записи.

Контрольные вопросы.

1. Что такое многозадачный (мультизадачный) режим работы?

2. Как осуществляется мультизадачный режим на однопроцессорном компьютере?
3. Перечислить типы мультизадачности, реализуемые в операционных системах.
4. Что такое приоритет задачи (процесса)?
5. Каким образом устанавливаются приоритеты?
6. Какие задачи имеют наивысший приоритет?
7. В чем отличие корпоративной многозадачности от вытесняющей?
8. Какая мультизадачность реализована в разработанной в работе модели.
9. Что такое язык программирования реального времени?

Лабораторная № 4

Моделирование задач о взаимном исключении при помощи сетей Петри.

Цель: изучение основ моделирования распределённых систем сетями Петри.

Краткая теория.

Сети Петри были предложены Карлом Петри в 60-х годах прошлого столетия. С точки зрения общей теории они представляют собой *двудольный ориентированный мультиграф*. *Двудольный граф* – это граф, имеющий два типа вершин, причём вершины одного типа непосредственно могут соединяться только с вершинами другого типа. Применительно к сетям Петри, вершины одного типа носят название *позиции*, а другого – *переходы*. Обычно при графическом изображении сети Петри позиции обозначаются кружками, а переходы – отрезками. Если дуги, соединяющие вершины графа, имеют направление, то граф называется *ориентированным*. Если допускается соединение вершин не одной, а несколькими дугами одинаковой направленности, то граф является *мультиграфом.*

Для сети Петри вводится понятие *состояния*. Чтобы определить его рассмотрим следующую функцию $\mu : P \rightarrow N$, переводящую множество позиций во множество натуральных чисел N. Функция μ называется *маркировкой сети Петри*. Для наглядности на графе значения маркировки обозначаются точками, называемыми также *фишками*, внутри кружка, изображающего позицию. Распределение фишек по позициям, т.е. маркировка, является состоянием сети.

Как и в любой другой системе, процесс в сети Петри определяется как смена её состояний. Таким образом, процессом в сети Петри является изменение маркировки. Оно происходит в результате запуска переходов, который осуществляется в соответствии со следующими правилами. При запуске перехода из каждой его входной позиции забирается столько фишек, сколько дуг идёт из неё в рассматриваемый переход (эти фишки называются *разрешающими*). В выходные же позиции добавляются фишки в количестве, равном числу дуг, идущих из перехода в позицию. Подробно теория сетей Петри описана в работе [3].

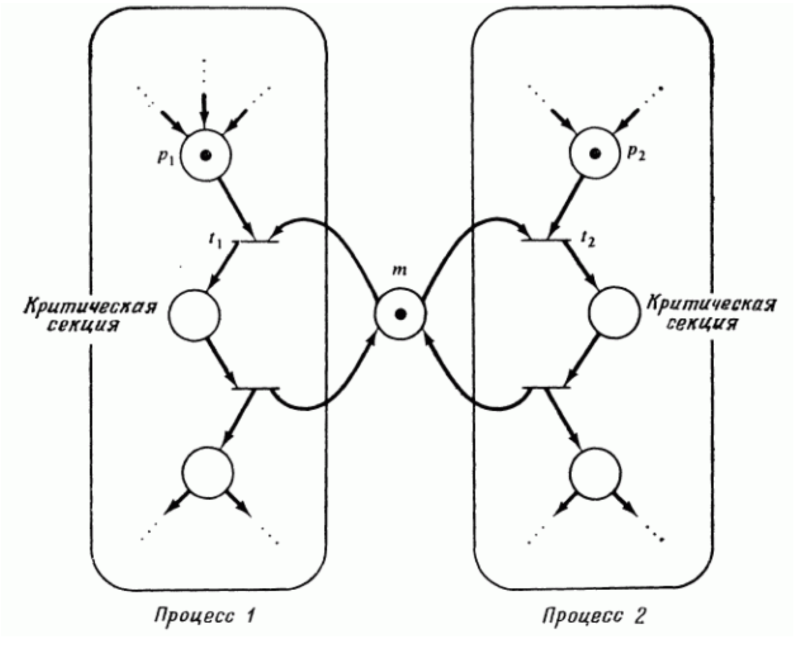


***Рис. 1*: Решение в сети Петри задачи о взаимном исключении.**

Одной из важных задач, решаемых в рамках моделирования сетями Петри, является *синхронизация* – согласование параллельно протекающих процессов. Под *взаимным исключением* понимается метод разработки таких систем управления, в которых одновременно не более чем одному процессу разрешается доступ к разделяемому объекту данных. Часть процесса, где осуществляется доступ к разделяемому объекту, который требует защиты от вмешательства других процессов, называется *критической секцией*. На рис. 1 представлено решение рассматриваемой задачи посредством сети Петри. Предположим, что процесс 1 входит в критическую секцию. Это означает, что запускается переход t_1, при этом забираются фишки из позиций p_1 и m. Для того, чтобы процесс 2 вошёл в критическую секцию, должен запуститься переход t_2. Однако он заблокирован, поскольку отсутствует фишка в позиции m. Появится она там только после того, как процесс 1 покинет критическую секцию

Порядок выполнения.

1. Разработать сеть Петри для следующих вариантов задачи о взаимном исключении: задача о чтении-записи без приоритетов, задача о чтении-записи с приоритетами, задача о производителе-потребителе без буфера, задача о производителе-потребителе с конечным буфером, (вариант даётся преподавателем).

2. Основываясь на построенной сети Петри, разработать компьютерную программу, осуществляющую имитационное моделирование соответствующих процессов.

3. Результаты работы представить в виде зависимостей числа выполненных операций (чтения, записи, производства единицы продукции) от параметров системы (максимального допустимого числа одновременно запущенных процессов чтения, количества мест в буфере и т.п.).

Контрольные вопросы.

1. Что такое сеть Петри?
2. Что такое логическая модель?
3. Иерархические и сетевые модели.
4. Назвать основные принципы моделирования распределённых систем сетями Петри.
5. Каким образом моделируется процесс синхронизации в сетях Петри?

6. Пояснить на примерах суть задачи о взаимном исключении.
7. Что такое критическая секция?
8. Есть ли ограничение на число процессов чтения в сети Петри, моделирующей задачу о чтении-записи?
9. В чём специфика задачи о производителе-потребителе по сравнению с общей постановкой задачи о взаимном исключении?

Лабораторная № 5

Моделирование работы асинхронного конвейера.

Цель: изучение способов обмена данными между параллельными процессами

Краткая теория.

Конвейер – это устройство, обеспечивающее конвейерное производство какой-либо продукции или конвейерную обработку данных. Поясним это на примере технологии, широко используемой в промышленности. Исходный материал автоматически подаётся по ленте конвейера к рабочему, который осуществляет с ним необходимые действия, следующий за ним рабочий выполняет свои функции над получившейся заготовкой, следующий делает дальнейшую обработку и т. д. К концу конвейера цепочка рабочих полностью выполняет все поставленные задачи. Каждый участок, на котором выполняется определённая технологическая операция, называется *ступенью* конвейера.

Ниже для определённости будет говориться о конвейерах, применяемых в компьютерных процессорах для повышения эффективности их работы, в частности, с большими массивами данных. Однако всё сказанное может быть также отнесено (с соответствующими поправками) к любому другому конвейеру. В процессорах роль рабочих исполняют функциональные модули, входящие в состав процессора.

На конвейере параллельно выполняются несколько инструкций процессора. Сложные инструкции представляются в виде последовательности более простых стадий. Вместо выполнения инструкций последовательно (ожидания завершения конца одной инструкции и перехода к следующей), следующая инструкция может выполняться через несколько стадий выполнения первой инструкции. В *синхронном конвейере* передача операндов между всеми ступенями происходит одновременно (т. е. запускается одним тактом таймера). Это позволяет управляющим цепям процессора получать инструкции со скоростью самой медленной стадии обработки, однако при этом намного быстрее, чем при выполнении эксклюзивной полной обработки каждой инструкции от начала до конца.

Для большего ускорения работы процессора используется *асинхронный конвейер*, для которого отсутствует единый такт работы блоков. При этом информация с одного блока на другой передается, когда предыдущий закончил свою работу, а следующий готов принять данные. Чем больше зависимость продолжительности выполнения отдельных этапов конвейера от типа, формата команд и других условий, тем эффективнее использование асинхронного конвейера. Рассмотрим его работу более подробно.

Предположим, что на очередном такте таймера заканчивается обработка на k-ой ступени конвейера. Об этом выставляется соответствующий сигнал. На последующем такте он принимается системой, управляющей работой конвейера, которая проверяет готовность k+1-ой ступени. Если последняя готова, то на следующих тактах производится пересылка данных с k-ой на k+1-ую ступень. Работа может быть существенно ускорена посредством применения пересылочных буферов. Каждая ступень использует два буфера: входной и выходной. Обычно буферы являются частью конвейера, однако в качестве них могут использоваться регистры общего назначения. Часто выходной буфер k-ой ступени является входным для k+1-ой. Но иногда они бывают раздельными. При наличии буферов обработанный операнд сначала помещается в выходной буфер k-ой ступени, затем во входной буфер k+1-ой ступени и только после этого непосредственно на k+1-ую ступень. Отметим, что на практике используются исключительно одноместные буферы, но теоретически возможны и многоместные.

Порядок выполнения.

1. Разработать сеть Петри для процесса работы однобуферного асинхронного конвейера с заданным количеством ступеней.

2. Основываясь на построенной сети Петри, разработать имитационную модель работы асинхронного конвейера. Предполагается, что в вычислительной системе обрабатывается большой массив данных, число элементов которого много больше количества ступеней конвейера. Число этапов обработки каждого элемента массива равно числу ступеней конвейера. На каждой ступени элемент обрабатывается случайное число тактов, которое имеет нормальное распределение с заданными математическим ожиданием и дисперсией.

3. Разработать компьютерную программу, реализующую модель. Функционирование конвейера представляется бесконечным циклом, при входе в который происходит разыгрывание количества тактов обработки элемента массива на каждой ступени, в случае если обработка на ней закончилась. Один период цикла соответствует одному такту работы процессора. В цикле производится проверка окончания обработки на каждой ступени и проверка наличия данных в пересылочных буферах. Выход из цикла осуществляется, когда будут обработаны все элементы массива.

4. Результаты работы представить в виде зависимости отношения времён обработки одного и того же массива на синхронном и асинхронном конвейере от числа ступеней.

Контрольные вопросы.

1. В чём состоит принцип конвейерной обработки данных?
2. Может ли конвейер работать в режиме обработки двух различных потоков?
3. Как регулируется обмен данными между ступенями синхронного конвейера?

4. Нужна ли в синхронном конвейере проверка готовности ступеней принять очередной операнд?
5. Чем отличается асинхронный конвейер от синхронного?
6. За счёт чего увеличивается скорость обработки данных асинхронным конвейером по сравнению с синхронным?
7. Какова роль пересылочных буферов в асинхронном конвейере?
8. Назвать условия, при которых применение асинхронного конвейера вместо синхронного ускорит обработку массива данных?

Раздел 3. Моделирование стохастических процессов

Лабораторная № 6

Решение задачи Буффона в рамках имитационной модели.

Цель: изучение геометрического и статистического подходов к решению вероятностных задач.

Краткая теория.

Классическое определение вероятности [20] имеет ограничение: пространство элементарных случайных событий может иметь лишь конечную размерность. Поэтому оно не может применяться для решения задач, описываемых моделями, оперирующими с бесконечными величинами. Для их решения был разработан особый подход, получивший название геометрический. В его основе лежит следующее рассуждение. Пусть на плоскости имеется некоторая область G, включающая другую область g. В область G наудачу бросается точка. Это означает, что попадание брошенной точки в любое место области G равновозможно. Ставится вопрос, чему равна вероятность попадания точки в область g. Учитывая равновозможность результатов бросания, полагают, что вероятность попасть в какую-либо часть области G пропорциональна площади этой части и не зависит от ее расположения и формы. Поскольку вероятность попадания в саму область G

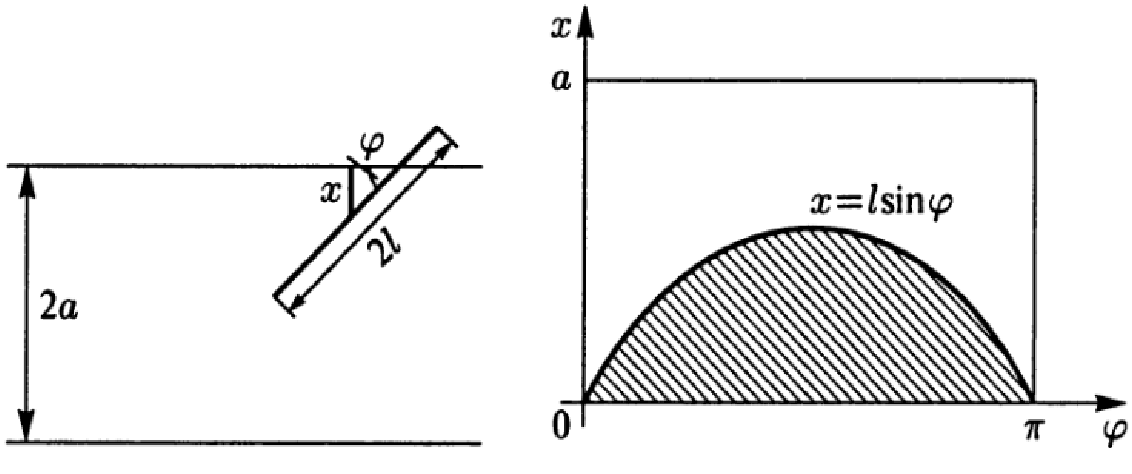
равна единице, то коэффициент пропорциональности должен быть равен её площади S_G. Таким образом, вероятность попадания точки в область g равна S_g / S_G, где через S_g обозначена площадь области g. В зависимости от условий задачи вместо плоских фигур могут рассматриваться отрезки, объемные тела, области пространств более высокой размерности. Тогда вероятность будет определяться через отношение длин, объёмов и т.д.

В качестве примера, иллюстрирующего возможности геометрического подхода рассмотрим <u>*задачу Бюффона*</u>. Плоскость разграфлена параллельными прямыми, расстояние между которыми равно $2a$. На плоскость наудачу, бросается игла длины $2l$ ($l < a$). Найти вероятность того, что игла пересечет какую-нибудь прямую (из условия очевидно, что такая прямая может быть только одна). Под словом «наудачу» здесь подразумевается что все положения упавшей на плоскость иглы равновозможны.

Обозначим через x расстояние от центра иглы до ближайшей прямой и через φ – угол, составленный иглой с этой прямой, см. рис. 2,*а*. Величины x и φ полностью определяют положение иглы. Таким образом, область G для этой задачи представляет собой прямоугольник со сторонами a и π. Из рис. 2,*а* видно, что для пересечения иглы с прямой необходимо и достаточно, чтобы $x \le l\sin\varphi$. На рис. 2,*б* область g, определяемая этим условием, заштрихована. Искомая вероятность равна

$$P = \frac{1}{a\pi}\int_0^{\pi} l\sin\varphi d\varphi = \frac{2l}{a\pi}. \qquad (6.1)$$

Отметим, что, зная левую часть уравнения (6.1), из него можно определить число π.



а) *б*)

***Рис. 2*: К решению задачи Буффона**

Порядок выполнения.

1. Разработать компьютерную программу, осуществляющую моделирование процесса бросания иглы на разграфлённую плоскость (отношение l/a задаётся преподавателем). Для моделирования одного броска надо дважды запустить встроенный в программное обеспечение генератор псевдослучайных случайных чисел. Из первого числа определить расстояние x, из второго – угол φ. Следующим шагом нужно проверить принадлежность получившейся точки к заштрихованной области на рис. 2,*б*. В случае если точка принадлежит области, бросок считается удачным. Экспериментальная вероятность пересечения иглой линии $P_{экс}$ определяется как отношение числа удачных бросков к их полному числу N. Далее вычисляется нормированное отклонение между вероятностью, полученной по формуле (6.1) и экспериментальной вероятностью: $\Delta = |P - P_{экс}|/P$.

2. Результаты работы представить в виде зависимости Δ от полного числа бросков N.

Контрольные вопросы.

1. В чём состоит классическое определение вероятности?
2. Как определяется вероятность в геометрическом подходе?
3. Чем псевдослучайные числа отличаются от случайных?

4. Назовите причины, по которым в компьютерном моделировании используются псевдослучайные числа.
5. Возможен ли генератор истинных случайных чисел?
6. Возможно ли исполнение имитационной модели задачи Буффона без помощи компьютера?
7. Может ли ошибка Δ, полученная при имитации 1000000 бросков оказаться больше, чем ошибка полученная при имитации 1000 бросков?
8. Каково наиболее вероятное поведение зависимости $\Delta(N)$?
9. Объяснить поведение зависимости $\Delta(N)$ при $N < 100$.

Лабораторная № 7

Моделирование прохождения нейтронов сквозь пластинку.

Цель: овладение методом Монте-Карло для моделирования стохастических процессов.

Краткая теория.

Пусть на однородную бесконечную пластинку перпендикулярно ей падает поток нейтронов с энергией E_0. Требуется, используя *метод Монте-Карло* [4], определить вероятности прохождения, отражения и поглощения нейтронов.

Согласно современной теории, при столкновении с атомами вещества, из которого состоит пластинка, нейтроны могут рассеиваться или поглощаться. Предположим, что происходит упругое рассеяния, т.е. энергия нейтрона в этом процессе не меняется. Введём допущение о том, что любое направление рассеяния нейтрона одинаково вероятно (вообще говоря, это допущение выполняется лишь приближённо). Взаимодействие нейтронов с веществом характеризуется *сечением поглощения* Σ_a и *сечением рассеяния* Σ_s. Значения этих параметров зависят от энергии нейтронов, но поскольку по условию задачи все нейтроны имеют одинаковую энергию E_0, их можно считать постоянными. Сумма $\Sigma = \Sigma_a + \Sigma_s$ называется полным сечением рассеяния.

Примем следующую гипотезу перемещения нейтрона внутри пластинки. При столкновении с атомами нейтрон мгновенно меняет направление скорости. Между столкновениями с атомами нейтрон движется равномерно и прямолинейно, проходя при этом путь λ, называемый *длиной свободного пробега*. Длина свободного пробега нейтрона – это случайная величина. Обычно предполагается, что она подчиняется экспоненциальному закону распределения: $f_\lambda(x) = \Sigma e^{-\Sigma x}$. Процесс задания значения какой-либо случайной величины в методе Монте-Карло называется *разыгрыванием*. Предположим, что с помощью генератора получается последовательность псевдослучайных чисел γ, каждое из которых лежит в интервале от 0 до 1. Для разыгрывания λ имеем следующую формулу, [4]:

$$\lambda = -\frac{1}{\Sigma}\ln(1-\gamma). \tag{7.1}$$

Построим формулу для разыгрывания направление нейтрона после рассеяния. Предположим, что пластинка перпендикулярна оси X и занимает область $0 \le x \le h$. Так как задача симметрична относительно оси X, то направление вполне определяется одним углом φ между направлением скорости нейтрона и осью X. Более того, поскольку в данной задаче нас интересует проекция пути пробега на эту ось, удобнее разыгрывать не сам угол φ, а величину $\cos\varphi$, которая очевидно меняется в интервале [–1; 1]

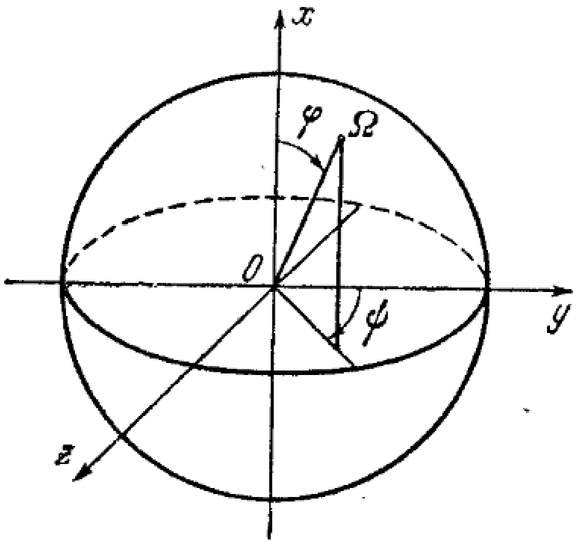


***Рис. 3*: К определению формулы для разыгрывания направления скорости нейтрона**

Пусть скорость нейтрона направлена вдоль вектора $\Omega \in S^2$, где S^2 – единичная сфера в трёхмерном пространстве, см. рис. 3. Если направление вектора Ω распределено равномерно, то вероятность того, что оно окажется в любом элементе поверхности *dS*, равна $dS/4\pi$. Выберем на поверхности сферы сферические координаты (*φ*, *ψ*) с полярной осью *X*, см. рис. 3. Тогда $dS = \sin\phi \, d\phi \, d\psi$, причём $0 \le \phi \le \pi, \ 0 \le \psi < 2\pi$. Так как координаты *φ* и *ψ* независимы, то функция плотности вероятности для точки (*φ*, *ψ*) распадается на произведение $f(\phi,\psi) = f_\phi(\phi) f_\psi(\psi)$. Из этого равенства, выражения для *dS* и соотношения $f(\phi,\psi)\, d\phi \, d\psi = dS/4\pi$ вытекает, что

$$f_\phi(\phi) f_\psi(\psi) = \frac{\sin\phi}{4\pi}. \tag{7.2}$$

Проинтегрируем выражение (7.2) по *ψ* от 0 до 2π. Принимая во внимание условие нормировки функции плотности вероятности $f_\psi(x)$ и формулу для получения псевдовыборки случайной величины с функцией плотности распределения $f_\varphi(x)$ по последовательности псевдослучайных чисел *γ*, получим формулу для разыгрывания *φ*

$$\gamma = \frac{1}{2}\int_0^{\phi} \sin x dx = \frac{1}{2}\left(1-\cos\phi\right). \qquad (7.3)$$

Из (7.3) получаем формулу для разыгрывания косинуса: $\cos\phi = 1-2\gamma$ Отметим, что $\cos\varphi$ имеет равномерное распределение в интервале [–1; 1].

Порядок выполнения работы.

Вариант 1.

Разработать компьютерную программу, реализующую алгоритм *моделирования истинных траекторий*. Предположим, что нейтрон испытал k-е рассеяние внутри пластинки в точке с абсциссой x_k и после этого начал двигаться под углом к оси X, косинус которого равен $\left(\cos\phi\right)_k$. По формуле (7.1) Разыграем длину свободного пробега λ_k, и вычислим абсциссу следующего столкновения

$$x_{k+1} = x_k + \lambda_k \left(\cos\phi\right)_k. \qquad (7.4)$$

Проверим, пройдёт ли при этом нейтрон сквозь пластинку. Это означает, что имеет место $x_{k+1} > h$. Если это условие выполнено, то расчёт траектории нейтрона заканчивается, и добавляется единица к счетчику прошедших частиц. В противном случае проверяем условие отражения: $x_{k+1} < 0$. Если оно выполнено, то расчёт траектории также заканчивается, а единица добавляется к счетчику отраженных частиц. Если же нейтрон остался внутри пластинки, т.е. оказалось, что $0 \le x_{k+1} \le h$, то это означает что, он испытал (k+1)-е столкновение, и надо продолжить моделирование траектории.

Сгенерируем очередное значение случайной величины γ и проверим условие поглощения: $\gamma \le p_a = \Sigma_a / \Sigma$. Если это неравенство выполнено, то счёт траектории заканчивается и добавляется единица к счётчику поглощённых частиц. В противном случае мы считаем, что нейтрон испытал рассеяние в точке с абсциссой x_{k+1}. Тогда разыгрывается новое направление скорости нейтрона, и затем повторяется весь цикл снова. После того как будут сосчитаны N траекторий, окажется, что N^+ нейтронов прошли сквозь пластинку, N^- нейтронов отразились от нее, а N_0 нейтронов были поглощены. Тогда оценки искомых вероятностей будут отношения чисел N^+, N^-, N_0 к N.

Вариант 2.

Разработать компьютерную программу, реализующую алгоритм *с использованием весов, заменяющих поглощение*. Предположим, что вдоль одной и той же траектории движется *пакет*, состоящий из большого числа w_0 одинаковых нейтронов. При первом столкновении пакета, как единого целого, количества испытавших рассеяние и поглощенных из него нейтронов в среднем равны соответственно $\left(\Sigma_s/\Sigma\right)w_0$ и $\left(\Sigma_a/\Sigma\right)w_0$. Тогда число оставшихся в пакете нейтронов будет $w_1=\left(\Sigma_s/\Sigma\right)w_0$, при этом считается, что все они полностью рассеивается в одном направлении. Последующие столкновения представляются таким же образом. Всё приведённые выше сказанное выше для варианта 1 остается в силе. Отличие состоит в том, что теперь траектория не может закончиться поглощением.

Контрольные вопросы.

1. В чём состоит суть метода Монте-Карло?
2. Можно ли разыгрывать дискретные случайные величины?
3. Выборки из каких распределений можно получать по выборке из равномерного распределения на интервале от 0 до 1?
4. Чем алгоритм моделирования истинных траекторий отличается от алгоритма с использованием весов?
5. Какой из алгоритмов, упомянутых в предыдущем вопросе, обеспечивает более точную оценку вероятностей?
6. Каковы недостатки модели, использованной в данной лабораторной работе?

Лабораторная № 8

Моделирование броуновского движения частицы.

Цель: изучение случайных процессов.

Краткая теория.

Броуновское движение – это беспорядочное движение микроскопических частиц, взвешенных в жидкости или газе, вызываемое тепловым движением молекул жидкости или газа. Если промежуток наблюдения гораздо больше, чем характерное время изменения силы, действующей на частицу со стороны молекул среды, и прочие внешние силы отсутствуют, то средний квадрат проекции смещения частицы на какую-либо ось пропорционален времени. Это положение иногда называют *законом Эйнштейна*. Кроме поступательного броуновского движения, существует вращательное броуновское движение – беспорядочное вращение броуновской частицы под влиянием ударов молекул среды. Для вращательного броуновского движения среднее квадратичное угловое смещение частицы также пропорционально времени наблюдения.

Современная модель броуновского движения на качественном уровне выглядит следующим образом. Макроскопическое тело, погруженное в среду, испытывает множество взаимодействий с молекулами. В силу того, что масса любого макроскопического тела (в том числе броуновской частицы) несравненно больше массы молекулы, последняя при взаимодействии отражается от тела, которое практически остается неподвижным. До тех пор, пока средняя кинетическая энергия молекул много меньше энергии их колебательных переходов (в частности, это имеет место для температур, которые встречаются в близи земной поверхности) взаимодействие с телом можно считать упругим. Следствием этого является то, что молекула передаёт телу практически удвоенный импульс, который она имела до столкновения: $2m\boldsymbol{v}$, где m – масса молекулы, $\boldsymbol{v}$ – вектор её скорости до столкновения. При этом после столкновения вектор скорости молекулы имеет тот же модуль, что и до столкновения, но противоположное направление, т. е. он равен $-\boldsymbol{v}$.

Импульсы, передаваемые телу молекулами, усредняются, давая в результате практически постоянные силы, действующие на тело во всех направлениях. Отнесённые к единице площади, они формируют постоянное давление. Если тело достаточно крупное и окружено средой со всех сторон, то давление на него почти полностью уравновешивается. практически уравновешивается, остаётся только подъёмная сила Архимеда. В зависимости от своего веса такое тело всплывает, тонет либо находится остаётся в безразличном равновесии в любом месте жидкости. Если же тело мало то становятся заметны флуктуации давления, которые создают заметную случайно изменяющуюся силу, приводящую к его движению. Экспериментальным путём было установлено, что крупные частицы с размерами более 5 мкм в броуновском движении практически не участвуют (они неподвижны), более мелкие частицы (менее 3 мкм) двигаются поступательно по весьма сложным траекториям или вращаются.

Порядок выполнения работы.

1. Алгоритмизировать следующую двумерную модель движения броуновской частицы. Броуновская частица моделируется кругом радиуса r, обладающим массой M. Молекулы моделируются точками, не имеющими размера, но имеющими массу m. Количество молекул равно n. По модулю скорости они имеют распределение Максвелла. Начальное направление модуля скорости каждой молекулы имеет равномерное распределение в промежутке от 0 до 2π. «Молекулы» и «броуновская частица» заключены в квадрат со стороной a. «Молекулы» не взаимодействуют между собой, взаимодействие же с «частицей» и сторонами ограничивающего квадрата происходит как мгновенное абсолютно упругое соударение, т.е. модуль скорости не меняется, а её направление меняется по закону «угол падения равен углу отражения».

2. Разработанный на предыдущем этапе алгоритм реализовать в виде компьютерной программы. Модельное время представить в виде N промежутков длительности Δt. Программа должна вычислять положение всех «молекул» и центра «частицы» на момент окончания каждого промежутка, т. е. в моменты времени Δt, $2\Delta t$, $3\Delta t$ и так далее до $N\Delta t$. В момент времени $t = 0$ центр «частицы» находится в центре квадрата, «молекулы» располагаются внутри квадрата случайно с равномерным распределением. (Все значения параметров задаются преподавателем.)

3. Результаты работы представить в виде изображения траектории центра «частицы».

Контрольные вопросы.

1. Что является причиной броуновского движения?
2. Можно ли заметить движение броуновской частицы, вызванное взаимодействием с одной молекулой?
3. Описать концептуальную модель броуновского движения, реализованную в работе. В чём заключаются её недостатки?
4. Может ли броуновская частица по истечении длительного промежутка времени вернуться в исходную точку?
5. Подтверждают ли результаты моделирования закон Эйнштейна о смещении броуновской частицы?
6. Каковы недостатки модели, использованной в данной лабораторной работе?

Раздел 4. Моделирование систем массового обслуживания

Лабораторная № 9

Изучение систем массового обслуживания с различным числом приборов.

Цель: Знакомство с принципами функционирования систем массового обслуживания.

Краткая теория.

Система массового обслуживания (СМО) – это система способная совершать однородные операции по выполнению однородных требований, а также организовывать определённый порядок обслуживания поступающих требований. *Теория массового обслуживания* – область прикладной математики, занимающаяся анализом процессов в системах производства, обслуживания, управления, в которых однородные события повторяются многократно, см., например [5]. На первичное развитие теории массового обслуживания особое влияние оказали работы А. К. Эрланга. В частности, им была решена задача обслуживания звонков, поступающих на телефонную станцию.

Система массового обслуживания считается заданной, если известны:

1: поток требований, его характер;

2: множество обслуживающих приборов;

3: дисциплина обслуживания (совокупность правил, задающих процесс обслуживания).

Ниже перечислены некоторые признаки, по которым классифицируются системы массового обслуживания. К *замкнутым* относятся системы, в которых поступающий поток требований ограничен. Например, мастер, задачей которого является поддержание в рабочем состоянии станков в цехе, должен периодически их обслуживать. При этом каждый отремонтированный станок становится потенциальным требованием на починку в будущем. В подобных системах общее число циркулирующих требований конечно и чаще всего постоянно. Если источник требований может порождать их без ограничений, т.е. очень большое, в пределе бесконечное число, то такие СМО называются *разомкнутыми*. Примерами подобных систем могут служить супермаркеты, кассы вокзалов, портов и др. Для этих систем поступающий поток требований можно считать неограниченным.

Если СМО имеет один прибор, обслуживания, то она называется *одноканальной*, в противном случае – *многоканальной*. В СМО *с отказами* поступившее требование теряется, если на момент его прибытия все каналы обслуживания оказываются занятыми. В СМО *с ожиданием* требование становится в очередь. Если очередь имеет ограниченное число мест, и все они оказываются занятыми, то требование также теряется. Если на длину очереди и время пребывания в ней нет ограничений, то требование находится в очереди до тех пор, пока не будет обслужено. Говорят, что СМО находится в *N-ом состоянии*, если в её каналах обслуживания и в очереди находятся N требований.

Под *взаимопомощью* понимают правила, по которым обслуживающие приборы подключаются к обслуживанию поступающих требований, при этом все требования считаются равноценными. С этой точки зрения различают две разновидности СМО: *с взаимопомощью*, в которых для ускорения процесса обслуживания допускается подключение нескольких каналов к работе над одним требованием, и системы *без взаимопомощи*, где каждое требование обслуживается только одним прибором.

Требования, поступающие в СМО, могут обладать приоритетами. В частности, наличие у требования *абсолютного приоритета* означает, что оно начинает обслуживаться сразу же по поступлению в СМО. При этом если все каналы обслуживания заняты, один из них освобождается: находящееся в нём требование вытесняется.

Среднее число требований, поступающих в систему обслуживания за единицу времени, называется *интенсивностью поступления требований* или *интенсивностью входного потока*. Оно определяется следующим соотношением: $\lambda = 1/T$, где T – среднее значение временного интервала между поступлением очередных требований. *Интенсивность обслуживания* – это среднее число требований, обслуженных системой за единицу времени $\mu = 1/t_{обс}$, $t_{обс}$ – среднее время обслуживания. *Коэффициент загруженности* – это отношение интенсивности входного потока требований к интенсивности обслуживания: $\rho = \lambda/\mu = \lambda t_{обс}$.

Время обслуживания одного требования τ, также как и промежуток времени между приходом требований в СМО, является случайной величиной. Оно зависит как от стабильности работы обслуживающих устройств, так и от различия параметров, характеризующих поступающие в систему требования. Последние, хотя и являются однотипными, но отнюдь не идентичны. На практике чаще всего принимают гипотезу об экспоненциальном законе распределения времени обслуживания. На практике эта гипотеза в большинстве случаев подтверждается путём статистических исследований.

Пусть СМО имеет m устройств одинаковой мощности, для каждого из которых имеет место экспоненциальное распределение времени обслуживания: $F_\tau(t) = 1 - \exp(-\mu t)$, где μ – интенсивность обслуживания одного требования одним устройством. Тогда функция распределения времени обслуживания всей системой в целом будет также распределено экспоненциально:

$$F_\tau(t) = 1 - \exp(-m\mu t). \qquad (9.1)$$

В случае экспоненциального закона распределение длительности оставшейся части работы по обслуживанию не зависит от того, сколько она уже продолжалась.

Порядок выполнения работы.

1. Разработать компьютерную программу, моделирующую работу двух систем массового обслуживания. Обе принимают одинаковые пуассоновские потоки требований с интенсивностью 1.0. Первая система имеет один прибор обслуживания с интенсивностью 0.9, вторая – три прибора обслуживания с интенсивностью 0.3 каждый. Распределение времени обслуживания для всех приборов экспоненциальное. Функционирование системы представляется циклом, при входе в который происходит разыгрывание факта прихода требования, а также длительности его обработки. В случае системы с тремя приборами дополнительно разыгрывается прибор, на который попадает требование. В цикле производится проверка окончания обработки требования на каждом из устройств. Выход из цикла осуществляется после определённого количества проходов.

2. Результаты работы представить в виде зависимостей длины очереди ожидающих требований для каждой из систем от времени.

Контрольные вопросы.

1. Что такое система массового обслуживания?
2. Привести примеры объектов, моделируемых системами массового обслуживания?
3. Чем характеризуется состояние системы массового обслуживания?
4. Чему равна вероятность того, что требование будет обслужено, если количество мест в очереди СМО бесконечно?
5. Возможны ли значения коэффициента загруженности СМО больше, чем единица?
6. Пусть интенсивность обслуживания составляет 3 требования в минуту. Какой интервал времени обслуживания требования более вероятен от 10 до 15 секунд, или от 25 до 30 секунд?

Лабораторная № 10

Изучение потоков требований.

Цель: изучение моделей потоков событий.

Краткая теория.

Потоком событий называется последовательность однородных событий, следующих одно за другим в некоторые моменты времени. Поток событий называется *регулярным*, если события следуют одно за другим через одинаковые, строго фиксированные промежутки времени. Очевидно, что регулярные потоки подходят для описания реальных систем лишь в исключительных случаях. Для многих реальных процессов существует более адекватная и, в то же время, относительно простая модель потока требований – закон распределения Пуассона, согласно которому вероятность $P_k(\Delta t)$ того, что в систему за промежуток время Δt поступит ровно k требований, равна

$$P_k(\Delta t) = \frac{(\lambda \Delta t)^k}{k!} \exp(-\lambda \Delta t), \quad (10.1)$$

где λ – определённая в предыдущем разделе интенсивность поступления требований. Такой поток является *простейшим*, часто его также называют *пуассоновским*.

Простейший поток обладает тремя важными свойствами.

1. *Свойство стационарности*, означающее, что число требований, поступающих в систему в равные промежутки времени, в среднем должно быть постоянным.
2. *Отсутствие последействия*. Это свойство обуславливает взаимную независимость числа требований, поступающих на обслуживание в непересекающиеся интервалы времени.
3. *Свойство ординарности*, согласно которому одновременное поступление двух или более требований практически невозможно. Математически это выражается в том, что предел отношения $P(dt)/dt$ стремится к нулю, при стремлении к нулю промежутка времени *dt* (здесь $P(dt)$ вероятность того, что два или более требований поступят в систему за время *dt*).

Следующая ниже теорема позволяет определить, является ли поток простейшим.

Теорема Хинчина. Если входящий поток требований представляет собой сумму большого числа независимых между собой стационарных и ординарных потоков, каждый из которых вносит малый вклад в общую сумму, то такой поток близок к простейшему.

Известно, что для пуассоновского потока распределение длительности промежутков времени между соседними событиями подчиняется экспоненциальному закону с функцией плотности вероятности $f(t) = \lambda \exp(-\lambda t)$. Помимо закона Пуассона (10.1) в теории массового обслуживания применяются и другие, более общие, модели входящих потоков требований. Среди них важную роль играют *потоки Пальма* (они же *потоки с ограниченным последействием*). *Потоком Пальма* называется поток, в котором промежутки времени между двумя соседними событиями представляют собой независимые случайные величины, распределенные по одному и тому же закону. Таким образом, простейшие потоки, в которых интервалы между соседними событиями распределены по экспоненциальному закону, являются частным случаем потоков Пальма. Если интервалы между событиями подчиняются гауссовскому распределению, то такие потоки называются *нормальными*. Очевидно, что для регулярного потока закон распределения выражается δ-функцией.

В теории массового обслуживания также широко используются *потоки Эрланга*. Они образуются из простейших потоков путем применения к ним операции “просеивания”, которая заключается в том, что из простейшего потока удаляется некоторое число точек по определенному правилу. Если удаляются точки через одну, т.е. остается каждая 2-я точка, то поток Эрланга называется потоком 2-го порядка ($Э_2$). Обобщаем это на произвольное число точек: если удаляются (k-1) точек подряд, а остается каждая k-я точка, то получаем поток Эрланга k-го порядка ($Э_k$). Можно показать, что для него функция плотности распределения длительности промежутков времени между событиями есть

$$f_k(t) = \lambda \frac{(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \qquad \mathrm{t} > 0. \tag{10.2}$$

Это так называемый *закон Эрланга*. Очевидно, что простейший поток является потоком Эрланга 1-го порядка, для него (11.2) переходит в показательное распределение.

Порядок выполнения работы.

1. Разработать компьютерную программу, моделирующую работу системы массового обслуживания при различных потоках требований одинаковой интенсивности, равной 1.0. Нужно смоделировать три потока: регулярный, пуассоновский и поток Эрланга порядка указанного преподавателем.

Система имеет один прибор обслуживания с интенсивностью 0.95. Распределение времени обслуживания экспоненциальное. Функционирование системы представляется циклом, при входе в который происходит разыгрывание факта прихода требования, а также длительности его обработки. В цикле производится проверка окончания обработки требования. Выход из цикла осуществляется после определённого количества проходов.

2. Результаты работы представить в виде зависимостей от времени длины очереди ожидающих требований для каждого из рассматриваемых потоков.

Контрольные вопросы.

1. Что такое поток требований?
2. Какими свойствами обладает простейший (пуассоновский) поток требований? Объяснить эти свойства на примерах.
3. Что утверждает теорема Хинчина?
4. Описать каким образом получаются потоки Эрланга.
5. Является ли регулярный поток потоком Эрланга.
6. Как связаны потоки Эрланга и Пальма?
7. Чему равно математическое ожидание длительности промежутков времени между событиями для потока Эрланга третьего порядка.
8. Простейший поток определяется формулой (11.1). Обозначим случайный промежуток времени между последовательными событиями через τ. Для какого из потоков неравенство $\tau > 1/\lambda$ более вероятно, для рассматриваемого простейшего потока, или полученного из него потока Эрланга второго порядка?

Лабораторная № 11

Максимизация доходов от эксплуатации системы массового обслуживания.

Цель: изучение методов исследования и оптимизации экономических систем

Краткая теория.

Помимо интенсивности входного потока λ и интенсивности обслуживания μ для оценки экономической эффективности СМО рассматриваются следующие параметры. <u>*Вероятность отказа*</u> $P_{отк}$. – это вероятность того, что поступающее в систему требование не будет присоединено к очереди и потеряется. Пусть СМО имеет m каналов обслуживания. Тогда для системы с отказами $P_{отк}$. равна вероятности того, что в системе находится m требований: $P_{отк} = P_m$. Для системы с ограниченной длиной очереди $P_{отк}$ равна вероятности того, что в системе находится $m + l$ требований: $P_{отк} = P_{m+l}$, где l – допустимая длина очереди. Противоположным показателем является <u>*вероятность обслуживания требования*</u> $P_{обсл} = 1 - P_{отк}$.

<u>*Средняя длина очереди*</u> $M_{ож}$ – это среднее количество требований, ожидающих начала обслуживания. Она определяется как математическое ожидание числа требований, находящихся в очереди: $M_{ож} = \sum_{n=m+1}^{m+l} (n-m) P_n$. Здесь, как и ранее, m – число каналов обслуживания; l – количество мест в очереди; P_n – вероятность того, что в системе находятся n требований.

<u>*Среднее число занятых обслуживанием приборов*</u> $m_з$ для системы массового обслуживания с отказами определяется как математическое ожидание: $m_з = \sum_{n=1}^{m} nP_n$. <u>*Общее количество требований, находящихся в системе*</u> M определяется следующим образом: $M = m_з$ для СМО с отказами, и $M = m_з + M_{ож}$ для СМО с ожиданием в очереди ограниченной длины.

Показатели, характеризующие экономические особенности СМО, формируют обычно в соответствии с конкретным видом системы и ее назначением. Одним из общих экономических показателей является *экономическая эффективность*: $E = P_{обсл} \lambda c t - G_{п}$. Здесь c – средний экономический эффект, полученный при обслуживании одного требования, t – рассматриваемый интервал времени, $G_{п}$ – величина потерь в системе.

Отметим, что способ вычисления величины потерь зависит от типа СМО, при этом в расчётах используются определённые выше характеристики.

Общим подходом к исследованию экономических показателей СМО является анализ определённой целевой функции. Ею, например, может служить доход, получаемый при эксплуатации СМО. После того, как целевая функция записана в виде зависимости от параметров, характеризующих систему, некоторые из этих параметров фиксируются. Остальные же считаются переменными, по ним проводится поиск экстремума (минимума или максимума) целевой функции.

Порядок выполнения.

1. Построить целевую функцию, описывающую доход от эксплуатации системы массового обслуживания (необходимые параметры задаются преподавателем).

2. Разработать компьютерную программу, моделирующую работу заданной системы массового обслуживания для пуассоновского потока требований интенсивности равной 1.0. При этом в расчётах должна варьироваться интенсивность потока требований. Причём затраты на обработку требования меняются при изменении интенсивности обслуживания в соответствии с законом, указанным преподавателем. Нужно рассчитать ряд значений целевой функции и найти её минимум в зависимости от интенсивности обслуживания.

3. Результаты работы представить в виде зависимости дохода, получаемого при эксплуатации системы массового обслуживания, от интенсивности обслуживания.

Контрольные вопросы.

1. Какими параметрами характеризуется система массового обслуживания?
2. Из чего складываются доходы и затраты в системе массового обслуживания?
3. Что такое целевая функция?

4. Как в общем случае, рассчитывается рентабельность системы массового обслуживания?
5. Всегда ли экономически выгодно повышать коэффициент загруженности СМО?

Список литературы

1. *Тихонов А. Н.* Методы решения некорректных задач. / А. Н. Тихонов, В. Я. Арсенин — М.: Наука, 1979. — 142 с.

2. *Костевич Л. С.* Математическое программирование: Информ. технологии оптимальных решений: Учеб. пособие / Л. С. Костевич. — Мн.: Новое знание, 2003. — 150 с.

3. *Питерсон Дж.* Теория сетей Петри и моделирование систем. / Дж. Питерсон. — М: Мир, 1984. — 623 с.

4. Лихачев А. В. Методы математического моделирования процессов и систем. Учебное пособие. / А. В. Лихачев. — Новосибирск: Изд-во НГТУ, 2015 — 96 с.

5. *Бочаров П. П.* Теория массового обслуживания. / П. П. Бочаров, А. В. Печинкин. — М.: РУДН, 1995. — 529 с.