# Моделирование систем

## 1. Понятие имитационного моделирования. Формальная модель объекта. Типовые математические схемы моделирования.

**Имитационное моделирование** – процесс конструирования моделей реальной системы и постановки экспериментов на ней с целью либо понять поведение системы, либо оценить в рамках ограничений, накладываемых критерием или совокупностью критериев, различной стратегии, обеспеч-ие функционир-ие данной системы.

Модель объекта моделирования, т. е. системы ***S***, можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

совокупность *входных воздействий* на систему

;

совокупность *воздействий внешней среды*

;

совокупность *внутренних (собственных) параметров* системы

;

совокупность *выходных характеристик* системы

.

При этом в перечисленных подмножествах можно выделить управляемые и неуправляемые переменные. В общем случае  являются элементами непересекающихся подмножеств и содержат как детерминированные, так и стохастические составляющие.

При моделировании системы ***S*** входные воздействия, воздействия внешней среды ***Е*** и внутренние параметры системы являются независимыми переменными, которые в векторной форме имеют соответственно вид



а выходные характеристики системы являются зависимыми переменными и в векторной форме имеют вид .

Процесс функционирования системы ***S*** описывается во времени оператором ***Fs***, который в общем случае преобразует независимые переменные в зависимые в соответствии с соотношениями вида

 (4.1)

Совокупность зависимостей выходных характеристик системы от времени *yj(t)* для всех видов  называется *выходной траекторией* . Зависимость (4.1) называется законом функционирования системы ***S*** и обозначается ***Fs***. В общем случае закон функционирования системы ***Fs*** может быт задан в виде функции, функционала, логических условий, в алгоритмической и табличной формах или в виде словесного правила соответствия.

Весьма важным для описания и исследования системы ***S*** является понятие *алгоритма функционирования* ***As***, под которым понимается метод получения выходных характеристик с учетом входных воздействий , воздействий внешней среды  и собственных параметров системы . Очевидно, что один и тот же закон функционирования ***Fs*** системы ***S*** может быть реализован различными способами, т. е. с помощью множества различных алгоритмов функционирования ***As***.

Соотношения (4.1) являются математическим описанием поведения объекта (системы) моделирования во времени ***t***, т. е. отражают его динамические свойства. Поэтому математические модели такого вида принято называть *динамическими моделями* (системами).

Для *статических моделей* математическая модель (4.1) представляет собой отображение между двумя подмножествами свойств моделируемого объекта *Y* и *{X, V, Н}*, что в векторной форме может быть записано как

 (4.2)

Соотношения (4.1) и (4.2) могут быть заданы различными способами: аналитически (с помощью формул), графически, таблично и т. д.

Такие соотношения в ряде случаев могут быть получены через свойства системы ***S*** в конкретные моменты времени, называемые состояниями/

Состояние системы ***S*** характеризуется векторами



Где  в момент ,  в момент .

Если рассматривать процесс функционирования системы ***S*** как последовательную смену состояний *z1(t)*, *z2(t)*, ..., *zk(t)*, то они могут быть интерпретированы как координаты точки в *k*-мерном фазовом пространстве, причем каждой реализации процесса будет соответствовать некоторая фазовая траектория. Совокупность всех возможных значений состояний  называется пространством состояний объекта моделирования Z, причем zk ∈ Z.

Состояния системы ***S*** в момент времени *t0<t\*≤Т* полностью определяются начальными условиями входными воздействиями , внутренними параметрами  и воздействиями внешней среды , которые имели место за промежуток времени t\* - t0, с помощью двух векторных уравнений

; (4,3)

. (4.4)

Первое уравнение по начальному состоянию  и независимым переменным  определяет вектор-функцию , а второе по полученному значению состояний  - зависимыми переменные на выходе системы . Таким образом, цепочка уравнений объекта «вход — состояния — выход» позволяет определить характеристики системы

 (4.5)

В общем случае время в модели системы ***S*** может рассматриваться на интервале моделирования *(0, Т)* как непрерывное, так и дискретное, т. е. квантованное на отрезки длиной Δt временных единиц каждый.

Таким образом, под *математической моделью* объекта (реальной системы) понимают конечное подмножество переменных  вместе с математическими связями между ними и характеристиками ю.

Если математическое описание объекта моделирования не содержит элементов случайности или они не учитываются, т. е. если можно считать, что в этом случае стохастические воздействия внешней среды  и стохастические внутренние параметры  отсутствуют, то модель называется детерминированной в том смысле, что характеристики однозначно определяются детерминированными входными воздействиями

 (4.6)

Очевидно, что детерминированная модель является частным случаем стохастической модели.

**3. Типовые схемы.**

Приведенные математические соотношения представляют собой математические схемы общего вида и позволяют описать широкий класс систем.

Однако в практике моделирования объектов в области системотехники и системного анализа на первоначальных этапах исследования системы рациональнее использовать типовые математические схемы: дифференциальные уравнения, конечные и вероятностные автоматы, системы массового обслуживания, сети Петри и т. д.

Не обладая такой степенью общности, как рассмотренные модели, типовые математические схемы имеют преимущества простоты и наглядности, но при существенном сужении возможностей применения. В качестве детерминированных моделей, когда при исследовании случайные факторы не учитываются, для представления систем, функционирующих в непрерывном времени, используются дифференциальные, интегральные, интегро-дифференциальные и другие уравнения, а для представления систем, функционирующих в дискретном времени,— конечные автоматы и конечно-разностные схемы.

В качестве стохастических моделей (при учете случайных факторов) для представления систем с дискретным временем используются вероятностные автоматы, а для представления системы с непрерывным временем — системы массового обслуживания и т. д.

Перечисленные типовые математические схемы, естественно, не могут претендовать на возможность описания на их базе всех процессов, происходящих в больших информационно-управляющих системах.

Для таких систем в ряде случаев более перспективным является применение агрегативных моделей.

Агрегативные модели (системы) позволяют описать широкий круг объектов исследования с отображением системного характера этих объектов. Именно при агрегативном описании сложный объект (система) расчленяется на конечное число частей (подсистем), сохраняя при этом связи, обеспечивающие взаимодействие частей.

Таким образом, при построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно-детерминированный (например, дифференциальные уравнения); дискретно-детерминированный (конечные автоматы); дискретно-стохастический (вероятностные автоматы); непрерывно-стохастический (системы массового обслуживания); обобщенный, или универсальный (агрегативные системы).

## 2. Непрерывно-стохастические модели (Q-схемы).

При непрерывно-стохастическом подходе в качестве типовых математических схем применяется система массового обслуживания (англ. queueing system), которые будем называть Q-схемами. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д.

При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. Остановимся на основных понятиях массового обслуживания, необходимых для использования Q-схем, как при аналитическом, так и при имитационном.

В любом элементарном акте обслуживания можно выделить две основные составляющие:

1. ожидание обслуживания заявки;
2. собственно обслуживание заявки.

Это можно изобразить в виде некоторого *i*-гo прибора обслуживания *Пi* (рис. 2.), состоящего из накопителя заявок *Нi*, в котором может одновременно находиться  заявок, где  —емкость *i*-го накопителя, и канала обслуживания заявок (или просто канала) *Ki*

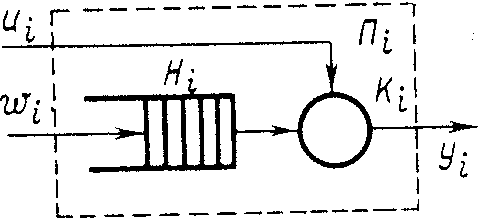


Рис. 2.

На каждый элемент прибора обслуживания *Пi*, поступают потоки событий: в накопитель *Hi* — поток заявок *wi* на канал *Ki* — поток обслуживании *ui*.

*Потоком событий* называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. *Различают* потоки *однородных* и *неоднородных* событий.

*Поток событий называется однородным*, если он характеризуется только моментами поступления этих событий (вызывающими моментами) и задается последовательностью , где *tn* — момент наступления *n*-го события — неотрицательное вещественное число.

Однородный поток событий также может быть задан в виде последовательности промежутков времени между *n*-м и *(n—1)*-м событиями {*τn*}, которая однозначно связана с последовательностью вызывающих моментов {*tn*}, где ,, *to* *= 0*, т. е. *τ1 = t1*

*Потоком неоднородных событий* называется последовательность , где *tn* — вызывающие моменты; *fn* — набор признаков события. Например, применительно к процессу обслуживания для неоднородного потока заявок могут быть заданы принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т. п.

поток, в котором события разделены интервалами времени *τ1*, *τ2,*…, которые вообще являются случайными величинами. Пусть интервалы *τ1*, *τ2,*.., независимы между собой. Тогда поток событий называется *потоком с* *ограниченным последействием*.

Интенсивность потока можно рассчитать экспериментально по формуле



где *N*— число событий, произошедших за время наблюдения *Тн*. Если *Тн =const* или определено какой-либо формулой , то поток называется *детерминированным*. Иначе поток называется случайным.

Случайные потоки бывают:

1. *ординарными* - когда вероятность одновременного появления 2-х и более событий равна нулю. Поток событий называется ординарным, если вероятность того, что на малый интервал времени *Δt*, примыкающий к моменту времени *t*, попадает больше одного события *Р>1 (t, Δt)*, пренебрежительно мала по сравнению с вероятностью того, что на этот же интервал времени *Δt* попадает ровно одно событие *P1 (t, Δt)*, т. е. *P1 (t, Δt)*» *Р>1 (t, Δt)*.
2. *стационарными* - когда частота появления событий постоянная. Стационарным потоком событий называется поток, для которого вероятность появления того или иного числа событий на интервале времени *τ* зависит лишь от длины этого участка и не зависит от того, где на оси времени взят этот участок.
3. *без последействия* - когда вероятность не зависит от момента совершения предыдущих событий.

Обычно при моделировании различных систем применительно к элементарному каналу обслуживания *Кi* можно считать, что поток заявок , т. е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе *Ki* образует подмножество неуправляемых переменных, а поток обслуживания , т. е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных.

Заявки, обслуженные каналом *Кi* и заявки, покинувшие прибор *Пi*, по различным причинам необслуженными (например, из-за переполнения накопителя *Нi*), образуют выходной поток , т. е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а Q-схемы, образуемые композицией многих элементарных приборов обслуживания *Пi* ( (сети массового обслуживания). Если каналы *Кi* различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная Q-схема), а если приборы *Пi* и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная Q-схема). Таким образом, для задания Q-схемы необходимо использовать оператор сопряжения ***R***, отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой. Связи между элементами Q-схемы изображают в виде стрелок (линий потока, отражающих направление движения заявок). Различают разомкнутые и замкнутые Q-схемы.

*В разомкнутой* Q-схеме выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует.

*в замкнутых* Q-схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

Для задания Q-схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе *Нi*, и обслуживания заявок каналом *Кi* каждого элементарного обслуживающего прибора *Пi* Q-схемы. Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

В зависимости от динамики приоритетов в Q-схемах различают *статические* и *динамические* *приоритеты*.

Статические приоритеты назначаются заранее и не зависят от состояний Q-схемы, т. е. они являются фиксированными в пределах решения конкретной задачи моделирования.

Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций.

Исходя из правил выбора заявок из накопителя *Hi* на обслуживание каналом *Кi* можно выделить *относительные* и *абсолютные приоритеты*.

*Относительный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель *Нi* ожидает окончания обслуживания предшествующей заявки каналом *Кi* и только после этого занимает канал.

*Абсолютный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель *Нi* прерывает обслуживание каналом *Кi* заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из *Кi* заявка может либо покинуть систему, либо может быть снова записана на какое-то место в *Нi*).

## 3. Сущность метода статистического моделирования. Примеры использования.

Метод статистического моделирования на ЭВМ - основной метод получения результатов с помощью имитационных моделей стохастических систем, использующий в качестве теоретической базы предельные теоремы теории вероятностей.

Основа - метод статистических испытаний Монте-Карло, который базируется на использовании случайных чисел, то есть возможных значений некоторой случайной величины с заданным распределением вероятностей. Статистическое моделирование представляет собой метод получения с помощью ЭВМ статистических данных о процессах, происходящих в моделируемой системе.

Сущность метода: построение для процесса функционирования исследуемой системы S некоторого моделирующего алгоритма, имитирующего поведение и взаимодействие элементов системы с учетом случайных входных воздействий и воздействий внешней среды и реализации этого алгоритма с использованием программно-технических средств ЭВМ.

2 области применения: 1) изучение стохастических систем; 2) решение детерминированных задач.

Приведем примеры применения методов статистического моделирования.

Пример 1. Входное воздействие - , воздействие внешней среды - - случайные величины с известными функциями распределения.

 Необходимо определить математическое ожидание величины y:



Схема алгоритма (ген. - генерация):

i=1..N

ген. λ



ген. ϕ









Пример 2. Необходимо найти площадь фигуры:





I=1..N

ген. *xi*, *yi*



S=S+h

S=S/N

Пример 3. Проводится s=10 независимых выстрелов по мишени, причем вероятность попадания при одном выстреле задана и равна p. Требуется оценить вероятность того, что число попаданий в мишень будет четным.

Аналитическое решение этой задачи:



Схема алгоритма (статистическое моделирование):

j=1..N

hj=0

i=1..10

ген. *xi*

*xi* < p

hj= hj +1

hj - четн

S=S+hj

P=S/N

## 4. Генерация случайных чисел. Генерация последовательностей псевдослучайных чисел. Требования к генератору псевдослучайных чисел. Улучшение качества последовательностей.

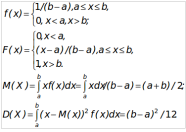
##### Генерация последовательности случайных чисел

Существует **три** способа генерации случайных чисел:

* **Аппаратный** - в основе лежит какой-либо физический эффект (например, шумы в электронных устройствах, случайные числа вырабатываются с помощью специального датчика.  
  Этот способ не гарантирует качество последовательности случайных чисел непосредственно во время моделирования. С помощью этого способа нельзя получать одинаковые последовательности. Используется редко.
* **Табличные** - случайные числа оформлены в виде таблицы в оперативной памяти или на внешнем носителе. При этом способе запас чисел ограничен, вычислительные ресурсы используются неэффективно. Используется редко.
* **Программный** (алгоритмический) - случайные числа формируются с помощью специальных программ. Каждое случайное число вычисляется с помощью соответствующей программы по мере возникновения потребностей при моделировании системы на ЭВМ. Этот способ наиболее распространен.

Программная имитация случайных воздействий сводится к генерированию некоторых стандартных (базовых) процессов и к их последующему функциональному преобразованию. Чаще всего в качестве базовой последовательности используют независимые случайные величины, равномерно распределенные на интервале (0,1).

Непрерывная случайная величина имеет равномерное распределение в интервале (a,b), если ее функции плотности и распределения соответственно примут вид:



Для получения случайных чисел на ЭВМ используются алгоритмы, поэтому такие последовательности, являющиеся по сути детерминированными, называются псевдослучайными. ЭВМ оперирует n-разрядными числами, поэтому на ЭВМ вместо непрерывной совокупности равномерных случайных чисел интервала (0,1) используют дискретную последовательность 2n случайных чисел того же интервала - закон распределения такой дискретной последовательности называется квазиравномерным распределением.

Требования к идеальному генератору случайных чисел:

1. Последовательность должна состоять из квазиравномерно распределенных чисел.
2. Числа должны быть независимыми.
3. Последовательности случайных чисел должны быть воспроизводимыми.
4. Последовательности должны иметь неповторяющиеся числа.
5. Последовательности должны получаться с минимальными затратами вычислительных ресурсов.

Методы улучшения качества последовательностей случайных чисел:

1. Использование рекуррентных формул порядка r:  
   http://dl.dropbox.com/u/1606474/gos/math/1_2.png  
   Но применение этого способа приводит к увеличению затрат вычислительных ресурсов на получение чисел.
2. Метод возмущений:

http://dl.dropbox.com/u/1606474/gos/math/1_3.png

## 5. Моделирование случайных воздействий (моделирование случайных событий). Моделирование дискретной случайной величины.

1. Необходимо реализовать случайное событие А, наступающее с заданной вероятностью p. Определим А как событие, состоящее в том, что выбранное значение xi равномерно распределенной на интервале (0,1) случайной величины удовлетворяет неравенству:

xi =<p.

Тогда вероятность события А будет  Противоположное событие состоит в том, что xi >p, его вероятность равна 1-р.

2. Рассмотрим группу событий. Пусть А1, А2 ,..., Аs - полная группа событий, наступающих с вероятностями p1, p2 ,..., ps соответственно. Определим событие Аm как событие, состоящее в том, что выбранное значение xi случайной величины удовлетворяет неравенству

,

где 

Процедура моделирования испытаний в этом случае состоит в последовательном сравнении случайных чисел xi со значениями *lr*. Если условие выполняется, исходом испытания оказывается событие Аm .

3. Рассмотрим независимые события А и В с вероятностями наступления рА и рВ. Возможными исходами совместных испытаний в этом случае будут события АВ,  с вероятностями рАрВ, (1-рА)рВ, рА(1-рВ), (1-рА)(1-рВ). Для моделирования совместных испытаний можно использовать два варианта процедуры:

* Последовательное выполнение процедуры, рассмотренной в п.1.
* Определение одного из исходов АВ,  по жребию с соответствующими вероятностями, т.е. процедура, рассмотренная в п.2.

Первый вариант потребует двух чисел xi и двух сравнений. При втором варианте можно обойтись одним числом xi , но сравнений может потребоваться больше. С точки зрения удобства построения моделирующего алгоритма и экономии количества операций и памяти ЭВМ более предпочтителен первый вариант.

4. События А и В являются зависимыми и наступают с вероятностями pА и pВ . Обозначим через pА(В) условную вероятность наступления события В при условии, что событие А произошло. Алгоритм модели подобного случая может быть следующим:

ген. *xi*

*xi* < pA

КА=КА+1

ген. *xi+1*

*xi*<pA(В)

КАВ=КАВ+1

КАNВ=КАNВ+1

КNА=КNА+1

ген. *xi+1*

*xi*<pNA(В)

КNАВ=КNАВ+1

КNАNВ=КNАNВ+1

## 6. Моделирование непрерывных случайных величин (метод обратных преобразований, показательный закон, треугольный закон распределения).

Моделирование непрерывной случайной величины

Теорема. Для заданной функции распределения F(y) случайной величины y, -∞<y<∞, случайная величина z=F(y), 0≤z≤1, имеет плотность вероятности f(z)=1, 0≤z≤1, т.е. является случайной величиной, равномерно распределенной на интервале [0,1].

Доказательство:

Случайная величина z является равномерно распределенной на интервале [0,1] тогда итлько тогда, когда P{z≤Z}=Z, 0≤Z≤1. Это соотношение непосредственно следует из следующих равенств:

P{z≤Z}=P{F(y) ≤Z}=P{y≤F-1(Z)}=F(F-1(Z))=Z, 0≤Z≤1, т.к. 0≤P{z≤Z}≤1.

Для получения непрерывных случайных величин с заданным законом распределения, как и для дискретных величин, можно воспользоваться методом обратной функции (обратного преобразования). Если случайная величина Y имеет плотность распределения f(y), то распределение случайной величины



является равномерным на интервале (0,1). Чтобы получить число, принадлежащее последовательности случайных чисел {yi}, имеющих функцию плотности f(y), необходимо разрешить относительно yi уравнение



где xi - число, принадлежащее последовательности случайных чисел равномерно распределенных на интервале от (0,1).

Пример 1. Необходимо получить случайные числа с показательным законом распределения (например, интервалов времени между поступлениями заявок на обслуживание):

.





 - случайное число, имеющее равномерное распределение на интервале (0,1). Тогда



Пример 2. Треугольное распределение. Треугольное распределение применяется в тех случаях, когда о случайной величине ничего неизвестно, кроме наиболее вероятного значения и диапазона возможных значений этой случайной величины (рис. 3).

*f(y)*

*f(С)*

0 А С В y

Рис. 3. Общий случай треугольного распределения вероятностей

Формула для получения последовательностей случайных чисел, подчиненных треугольному распределению, выводится методом обратных функций (обратного проебразования):

, где *x* - случайное число, имеющее равномерное распределение на интервале (0,1).

## 7. Моделирование непрерывных случайных величин (универсальный метод (кусочная аппроксимация), нормальный закон распределения).

На практике часто пользуются приближенными способами преобразования случайных чисел, которые можно классифицировать следующим образом:

а) универсальные способы, с помощью которых можно получать случайные числа с законом распределения любого вида;

б) неуниверсальные способы, пригодные для получения случайных чисел с конкретным законом распределения.

К приближенным универсальным способам получения случайных чисел относятся методы кусочной аппроксимации и усечения.

Метод кусочной аппроксимации. Пусть требуется получить последовательность случайных чисел {*xj*} с функцией плотности *fη(x)*, значения которой лежат в интервале (a,b). Дискретизируем заданную функцию - разобъем интервал (a,b) на k интервалов, и будем считать *fη(y)* на каждом интервале постоянной. hi - высота i-ого столбца показывает насколько вероятно выпадение значения в соответствующем интервале. Далее воспользуемся методом моделирования дискретной случайной величины.



Фрагмент алгоритма





Метод усечения. Функцию заключают в прямоугольник. На ось X подают случайное равномерно распределенное число r1. На ось Y подают случайное равномерно распределенное число r2. Если точка в пересечении этих двух координат лежит ниже кривой плотности вероятности, то событие X произошло, иначе нет.



Фрагмент алгоритма

Рассмотрим пример применения способа преобразования последовательности равномерно распределенных случайных чисел {xi} в последовательность с заданным законом распределения {yi} на основе предельных теорем теории вероятностей. Такие способы ориентированы на получение последовательностей чисел с конкретным законом распределения, т.е. не являются универсальными.

Моделирование нормально распределенных случайных величин. Нормальный закон распределения встречается в природе весьма часто, поэтому для него разработаны отдельные эффективные методы моделирования. Формула распределения вероятности по нормальному закону имеет вид:



,где mx - математическое ожидание

x - среднеквадратичное отклонение



Нормализованным нормальным распределением называется такое, у которого: x=1 и mx = 0.

Функция нормального распределения имеет вид колокольчика (см. график)



(mx  x) - 68% точек

(mx  2x) - 95% точек

(mx  3x) - 99.7% точек

Чтобы получить нормальное число существуют следующие методы.

1. Табличный.

Для этого нормальное число можно взять из справочника в таблице функции Лапласа и получить случайное число по методу взятия обратной функции.

|  |  |  |
| --- | --- | --- |
| X | F |  |
| 0,00 | 0,5 |  |
| ... | ... |  |
|  |  | r |

F - интегральная функция Лапласа.

2) ЦПТ (используя центральную предельную теорему).

Теорема: “для большого числа случайных чисел с любым законом распределения, их сумма является случайным числом с нормальным законом распределения”. То есть требуется сложить случайные числа с любым законом их распределения, нормализовать их и перевести в нужный диапазон нормального распределения.

1. *Ri* – случайное число, равномерно распределенное на интервале [0,1].

1. получение нормализованного нормального числа

 *mz = 0 z =1*

где z - нормализованное число (*m* = 0, **= 1)

3. получение нормального числа с заданными *(mx , x )*

*x = z  x + mx*

Отметим, что математическое ожидание смещает случайные числа, среднеквадратичное отклонение масштабирует закон распределения вероятности.



3) Метод Муллера



В современных системах имитационного моделирования обычно используются метод Муллера.

## 8. Управление модельным временем (принцип t и принцип z).

Одной из наиболее важных задач при создании модели является реализация механизма корректировки временной координаты состояния системы и обеспечение согласованности различных блоков и событий в системе (синхронизация во времени, координация с другими блоками).

Принципы построения моделирующих алгоритмов

Состояние системы характеризуется вектором состояний Z(t). Организуем счетчик системного времени, который в начальный момент показывает время t0. Прибавим интервал времени Δt, тогда счетчик будет показывать t1= t0 +Δt. Вычислим значения Z(t0 +Δt), затем перейдем к моменту времени t2= t1 +Δt и т.д. Если шаг Δt достаточно мал, то таким путем можно получать приближенные значения Z.

Рассмотренный принцип построения моделирующих алгоритмов называется *принципом* Δt. Это наиболее универсальный принцип, позволяющий определить последовательные состояния процесса функционирования системы через заданные интервалы времени Δt. Но с точки зрения затрат машинного времени он иногда оказы­вается неэкономичным.

При рассмотрении процессов функционирования некоторых си­стем можно обнаружить, что для них характерны два типа состоя­ний:

1) особые, присущие процессу функционирования системы то­лько в некоторые моменты времени (моменты поступления входных или управляющих воздействий, возмущений внешней среды и т. п.);

2) неособые, в которых процесс находится все остальное время.

Особые состояния характерны еще и тем обстоятельством, что функции состояний Z(t) в эти моменты времени изменяются скач­ком, а между особыми состояниями изменение координат Z(t) происходит плавно и непрерывно или не происходит совсем. Таким образом, следя при моделировании системы только за ее особыми состояниями в те моменты времени, когда эти состояния имеют место, можно получить информацию, необходимую для построения функций Z(t)*.* Очевидно, для описанного типа систем могут быть построены моделирующие алгоритмы по «принципу особых состо­яний». Обозначим скачкообразное (релейное) изменение состояния z как δz*,* а «принцип особых состояний» — как *принцип δz.*

Отметим, что характеристики процесса функционирования таких систем с особыми состояниями оцениваются по информации об особых состояниях, а неособые состояния при моделировании не рассматриваются. «Принцип δz*»* дает возможность для ряда систем существенно уменьшить затраты машинного времени на реализа­цию моделирующих алгоритмов по сравнению с «принципом Δt». Логика построения моделирующего алгоритма, реализующего «принцип δz», отличается от рассмотренной для «принципа Δt» только тем, что включает в себя процедуру определения момента времени tδ, соответствующего следующему особому состоянию си­стемы*.* Для исследования процесса функционирования больших систем рационально использование комбинированного принципа построения моделирующих алгоритмов, сочетающего в себе пре­имущества каждого из рассмотренных принципов.

У каждого из этих методов есть свои преимущества с точки зрения адекватного отражения реальных событий в системе и за­трат машинных ресурсов на моделирование. При использовании «принципа δz» события обрабатываются последовательно и время смещается каждый раз вперед до начала следующего события. В модели, построенной по «принципу Δt», обработка событий про­исходит по группам, пакетам или множествам событий. При этом выбор Δt оказывает существенное влияние на ход процесса и резуль­таты моделирования, и если Δt задана неправильно, то результаты могут получиться недостоверными, так как все события появляются в точке, соответствующей верхней границе каждого интервала мо­делирования. При применении «принципа δz» одновременная об­работка событий в модели имеет место только тогда, когда эти события появляются одновременно и в реальной системе. Это позволяет избежать необходимости искусственного введения ран­жирования событий при их обработке в конце интервала Δt.

При моделировании по «принципу Δt» можно добиться хорошей аппроксимации: для этого Δt должно быть малым, чтобы два неодновременных события не попали в один и тот же временной интервал. Но уменьшение Δt приводит к увеличению затрат машин­ного времени на моделирование, так как значительная часть тратит­ся на корректировку «часов» и отслеживание событий, которых в большинстве интервалов может и не быть. При этом даже при сильном «сжатии» Δt два неодновременных события могут попасть в один и тот же временной интервал Δt, что создает ложное представление об их одновременности.

Для выбора принципа построения машинной модели необходимо знать: цель и назначение модели; требуемую точность результатов моделирования; затраты машин­ного времени при использовании того или иного принципа; не­обходимый объем машинной памяти для реализации модели, построенной по принципу Δt и δz; трудоемкость программирования и отладки.

## 9. Механизм протяжки модельного времени.

Обычно в моделирующих системах выделяют два основных списка событий: cписок текущих событий (СТС) и список будущих событий (СБС). Каждое событие ассоциируется с динамическим объектом. В список текущих событий входят все события, запланированные на текущий момент модельного времени. Программа управления моделированием просматривает в первую очередь этот список и пытается переместить по модели те динамические объекты, для которых выполнены условия. Если в этом списке таких динамических объектов нет, процесс управления моделированием обращается к другому списку - списку будущих событий. Она переносит все события, которые запланированы на ближайший момент модельного времени, из этого списка в список текущих событий и повторяет его просмотр.

При моделировании модель­ное время может меняться быстрее или медленнее, чем в реальной сис­теме. Это зависит от степени детализации модели и сложности описа­ния изучаемого процесса. В любом случае модельное время изменяется при выполнении некоторых событий, а события в системе моделирова­ния возникают в результате перемещения динамических объектов. Причиной изме­нения модельного времени может послужить явная задержка динамического объекта на некоторый отрезок модельного времени.

Если в системе имеются динамические объекты, способные к продвиже­нию в данный момент времени, то они находятся в списке текущих событий; если этот список не пуст, то из него извлекается первый находящийся в нем объект, и он ста­новится активным, по­сле чего продвигается по модели. Наконец наступит такой мо­мент, что ни одного динамического объекта в списке текущих событий не останет­ся; тогда и просматри­вается список будущих событий*.* Из него вы­бирается динамический объект с минимальным "буду­щим временем про­движения", модельное время устанавливается равным времени продвижения этого объекта, а сам объект перемещается в спи­сок текущих событий*.* Вместе с ним в список текущих событий перемещаются все динамические объекты с таким же временем продвижения. После этого первый объект из списка текущих событий становится активным, и процесс моделирования продолжа­ется. Эта схема протяжки модельного времени далеко не полна, но тем не менее она отражает принцип изменения модельного времени.

СТС

Пуст?

Продвижение активного объекта в модели

Начало моделирования

Нормальное

завершение моделирования

Изменение текущего модельного времени

Поместить все объекты со временем выхода, равным тек. модельному времени, из СБС в СТС

Движение активного объекта прекращено

Извлечение динамического объекта из СБС с ближайшим временем активизации

Конец

моделиро-вания?

## 10. Объектно-ориентированная моделирующая система.

В большинстве случаев с помощью имитационных моделей исследуются характеристики и поведение системы на определенном отрезке времени, поэтому для моделирующей системы необходимым является реализация **двух функций**:

1. предоставление средств для формализованного описания системных компонентов, дисциплин выполнения различных работ, для задания структуры модели, привязки объектов модели к временной и пространственной координате;
2. осуществление координации событий, определение путей прохождения динамических объектов модели, изменение состояний узлов и передачу управления между блоками модели.

Существует несколько основных понятий, являющихся общими для большинства современных моделирующих систем:

1. **Граф модели.** Все процессы независимо от количества уровней структурного анализа, объединяются в виде направленного графа (многослойный иерархический).
2. **Динамические объекты.** Это объекты, моделирующие формальный запрос на какое-либо обслуживание.

Динамический объект может выполнять следующие действия:

* Порождать группы (семейства) других динамических объектов (GPSS, Split);
* Поглощать другие динамические объекты конкретного семейства;
* Захватывать ресурсы и использовать их некоторое время, а затем освобождать;
* Определять времена обслуживания, накапливать информацию о пройденном пути и иметь информацию о своем дальнейшем пути и о путях других динамических объектов.

Основные параметры динамических объектов:

* Уникальный идентификатор объекта;
* Идентификатор (номер) семейства, к которому принадлежит объект;
* Наборы различных ресурсов, которые динамический объект может захватывать и использовать какое-то время;
* Время жизни динамического объекта;
* Приоритет – неотрицательное число;
* Параметры обслуживания в каком-либо обслуживающем устройстве.

1. **Узлы графа модели** представляют собой центры обслуживания динамических объектов. В узлах динамические объекты могут задерживаться, обслуживаться, порождать семейства новых динамических объектов, уничтожать другие динамические объекты. Динамический объект всегда принадлежит одному из узлов графа модели.
2. **Событие** – факт входа или выходы из узла одного динамического объекта. Функция управления событиями отдана специально управляющей программе – координатору, автоматически внедряемому в состав модели.
3. **Ресурс** независимо от его природы в процессе моделирования может характеризоваться тремя общими параметрами: мощностью, остатком и дефицитом. Мощность – это макс. число ресурсных единиц. Остаток ресурса – число незанятых на данный момент единиц. Дефицит ресурса – количество единиц ресурса в суммарном запросе динамических объектов, стоящих в очереди к данному ресурсу.
4. **Пространство** – географическое, декартова плоскость. Узлы, динамические объекты и ресурсы могут быть привязаны к точкам пространства и мигрировать в нем.

Программная реализация моделирующих систем использует объектно-ориентированный способ представления процессов. Динамические объекты (транзакты), узлы, события и ресурсы являются основными объектами имитационной модели.

## 11. Событийный и процессно-ориентированный подход к построению моделей.

Существуют два различных подхода к построению имитационных моделей, они связаны с определенными элементами абстракций, важнейшими из которых являются понятия события и процесса.

Событием называется последовательность логически связанных действий, происходящих в некоторый фиксированный момент модельного времени. Появление события связано с изменением состояния модельной среды. Примером события может служить генерация сообщений, передача сообщения по каналу связи, обработка сообщения узлом и т.п.

Событийный подход основан на формировании потока событий (рис. 5). Такой поток образует сгруппированные последовательности действий:



Группы событий последовательно, происходящих в один и тот же момент времени, условно заключены в скобки. События  образуют первую группу, события  - вторую, а  - группу *N*. Обозначим  модельное время осуществления событий тогда для последовательности времен справедливо упорядочение



Если выполняемые внутри группы действия, связанные с одиночным событием, назвать процессом, то можно сказать, что часть процессов, определенных в группе, выполняются квазипараллельно. Соответствующее этим определениям понятие события было раскрыто выше. Собственно события являются элементами, с которыми оперирует управляющий алгоритм системы моделирования. С одной стороны, появление событий связано с изменениями в модельной среде. С другой, исполнение действий, связанных с событием, приводит к изменениям модельной среды. Действия, связанные с каждым событием, должны быть описаны в модели, а также должны быть составлены условия возникновения события.

Как правило, условия описываются отношениями связанных с событием переменных. В результате выполнения события значения некоторых модельных переменных могут измениться; это приводит к тому, что возникают условия, необходимые для выполнения другого события. При выполнении последовательности действий, связанной с выполнением некоторого события, можно изменить условия, связанные с его последующим вызовом. Таким образом, в результате выполнения потока событий изменяется среда моделирования; это приводит к возникновению новых условий, новые условия приводят к выполнению новых событий. Моделирование развивается по такой схеме сколь угодно долго.

Модельная среда

Действия 1

Действия 2

Действия К

Алгоритм управления

Событие К

Событие 2

Событие 1

Цикл моделирования

Несколько другая картина характерна для процессно-ориентированного подхода. Языки, реализующие процессно-ориентированный подход, имеют в своем составе блоки или операторы, позволяющие описать процесс продвижения компонентов через систему. В моделях, использующих подобную схему, описываются не события и условия их возникновения, а процесс, развивающийся в ней.

Для этого используются средства языка системы моделирования. Естественно, эти средства должны быть достаточно представительными для того, чтобы описывать широкий круг процессов. В частности, для процессного подхода необходимыми являются средства явного (или неявного) определения участков модели, на которых необходимо представление квазипараллельных процессов.

Оба подхода имеют как достоинства, так и недостатки. К достоинствам процессно-ориентированного представления моделей следует отнести компактность и наглядность. Здесь стрелками показано направление развития процессов. Событийные модели обладают большей гибкостью, но они уступают процессно-ориентированным системам в простоте и наглядности составления моделей.

## 12. Инструментальные средства моделирования.

Универс. Инструм.ср-вом создания моделей яв-ся языки программир. общего пользования (Pascal, C/C++ и др.). Кроме того, сущ-ет множ-во специализир.средств моделир-я, позволяющих быстрее и с меньшими затратами создавать и исследовать модели. В развитии специализир. Ср-в можно выделить **два направления**: **1.** Ср-ва моделир-я для анализа достаточно широкого класса систем: языки имитац.моделир-я (GPSS, SIMSCRIPT и др.), пакеты приклад. программ, использующих для моделир-я аналитич. методы, такие как MathCad, MathLab, SAS и др. Основной недостаток: их применение требует от исследователя спец.подготовки.**2.** Програм. комплексы, специализирующиеся на моделир-нии узкого круга систем одной конкретной предметной области. Недостаток: ограниченность применения таких программ одной предметной областью.Преимущ-ва: легкость их освоения специалистами в данной предмет.области, и эффективность применения вследствие узкой специализации.

В кажд. цикле создания программ. модели можно выделить **этапы**:**1. Формулирование проблемы**: описание исслед. проблемы, установление границ и ограничений моделируемой системы, определение целей исследования.**2. Разработка модели**: переход от реальной системы к некоторой логической схеме (абстрагирование).**3. Подготовка данных**: отбор данных, необх. для построения модели, и представление их в соответств. форме.**4. Трансляция модели**: описание модели на языке имитац. Моделир-я.**5. Оценка адекватности**: повышение до приемлемого уровня степени уверенности, с кот. можно судить относительно корректности выводов о реальной системе, получ. на основании обращения к модели.**6. Планирование**: определение условий проведения машинного эксперимента с имитац.моделью.**7. Экспериментирование**: многократн. прогон имит. модели на компьютере для получения требуемой инф-ии.**8. Анализ результатов**: изучение рез-тов для подготовки выводов и рекомендаций по решению проблемы.**9. Реализация и документирование**: реализация рекомендаций, получ.на основе имитации, и составление документации по модели и ее использованию.

## 13. Система имитационного моделирования AnyLogic.

**AnyLogic** – уникальный инструмент имит.моделир-я, поддерживающ. на единой платформе абсолютно все существующ. подходы дискретно-событийного и непрерывного моделир-я. AnyLogic имеет развитый базовый язык дискретного и смешан. дискретно-непрерывного моделир-я.

Библиотека AnyLogic Enterprise Library предоставляет высокоуровнев. интерфейс для быстрого создания дискретно-событийных моделей с пом. блок-схем. Графич. представление систем с пом. блок-схем широко исп-ся во многих важных сферах деятельности: произв-ве, логистике, системах обслуживания, бизнес-процессах, моделир-ии компьют. и телекоммуникацион. сетей.

Класс Entity яв-ся базовым классом для всех сообщений, кот. посылаются между активными объектами библиотеки Enterprise Library.

Под заявкой в библиотеке Enterprise Library может пониматься:

* заявка в ее обычном понимании (продукт, потребитель, пакет данных, документ),
* ресурс (оператор, машина, критическая секция),
* транспортер (поезд, автобус, корабль, автопогрузчик).

Заявки в их традиц. понимании генерир-ся объектами Source, проходят через смоделир. систему, где они обрабатыв-ся, обслужив-ся, транспортир-ся, конкурируют за право обладания ресурсами и эту систему покидают.

Ресурсы, созданные объектами Resource, м б.заняты заявками для выполнения каких-то задач, после чего они освобожд-ся и возвращ-ся в объект Resource.

Транспортеры, так же как и обычные заявки, создаются объектами Source, затем передаются в объекты Node и исп-ся для транспортировки других заявок между узлами и вдоль сегментов сети.

Объект класса Entity может исп-ся в любой из этих ролей, и при необходимости даже менять свою роль во время работы модели.

* Source - Источник заявок. Обычно исп-ся в кач.начальной точки потока заявок, или как генератор ресурсов, транспортеров.
* Sink - Уничтожает поступивш. заявки. Обычно исп-ся в кач.конечной точки потока заявок.
* Enter - Пересылает заявки, переданные этому объекту либо “явно” через входной порт inputExternal, либо с помощью функции объекта take(), дальше по блок-схеме.
* SelectOutput - Принимает заявку, и затем, в зависимости от заданного условия, передает ее на один из двух выходных портов.
* Queue - моделирует очередь, он хранит поступающ. заявки в опред.порядке: FIFO (заявки помещ-ся в очередь в порядке поступления), LIFO (заявки помещ-ся в порядке, обратном поступлению), RANDOM (заявки помещ-ся в произвольные места очереди) или PRIORITY (заявки помещ-ся в очередь в соответствии со значением своих полей priority).

## 14. Общие понятия сетей Петри.

Сетевые модели (сети Петри) используется для анализа причинно-следственных связей в сложных системах. Аппарат теории сетей Петри позволяет описывать структуру и взаимодействие параллельных систем и процессов. Сеть Петри (N-схема) задается четырьмя элементами:

*N = <B,D,I,O>,*

где *B* - конечное множество позиций; *D -* конечное множество переходов; *I* - входная функция (прямая функция инцидентности), *I:BXD→{0,1}*; *O* - выходная функция (обратная функция инцидентности), *O:DXB→{0,1}*.

*I* отображает переход *dj* в множество входных позиций , а выходная функция *О* отображает переход *dj* в множество выходных позиций . Для каждого  можно определить множество входных позиций перехода  и выходных позиций перехода  как





Аналогично, для каждого  можно определить множество входных переходов позиции  и выходных переходов позиции :





Графически сеть Петри изображается в виде двудольного ориентированного мультиграфа, представляющего собой совокупность позиций и переходов (рис. 1).

 приход заявки

 заявка ждет обслуживания



Канал обслуживания  заявка обрабатывается

свободен  (канал занят)

 конец обслуживания

 заявка ждет вывода

 выход заявки

Рис. 1. Структура N-схема

Для представления динамических свойств объекта вводится функция маркировки (разметки) . Маркировка - присвоение абстрактных объектов, называемых метками (фишками), позициям:

*NM = <B,D,I,O,M>.*

Функционирование сети Петри отражается путем перехода от разметки к разметке. *M0* - начальная разметка. Смена разметок - срабатывание одного из переходов сети. Необходимое условие срабатывания перехода :

,

где  - разметка позиции .

Срабатывание перехода  изменяет разметку сети  на по следующему правилу:

,

т.е. переход  изымает по одной метке из каждой своей входной позиции и добавляет по одной метке в каждую из выходных позиций. Для отражения временных параметров процесса функционирования моделируемой системы на базе N-схем используется расширение аппарата сетей Петри - временные сети.

Сети Петри представляют удобный математический аппарат для моделирования параллельных технологических процессов с разделяемыми ресурсами. Преимуществом сетей Петри также является легкость построения иерархических конструкций, что позволяет сначала исследовать отдельные подсистемы, а затем, объединяя уже созданные модели, всю систему в целом. Модели, построенные на основе сетей Петри, предназначены для анализа с помощью имитации на компьютере. Такие модели довольно легко реализуются программно даже с помощью универсальных языков программирования. Необходимо также отметить, что на сегодняшний день практически все компиляторы и операционные системы оптимизируются с помощью методов анализа сетей Петри.