**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ**

Факультет \_\_\_\_\_\_\_\_Компьютерных технологий и управления\_\_

(название факультета)

Кафедра \_\_\_\_\_\_\_\_\_Вычислительной техники\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(название кафедры)

Направление подготовки (специальность) \_\_\_\_Информатика и вычислительная техника\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**О Т Ч Е Т**

**о практике**

**Тема задания: «»**

**Студент \_\_\_\_Казанцева А.И. \_\_\_\_\_\_\_\_\_\_\_3105\_\_\_\_\_\_\_\_**

(Фамилия И.О.) (номер группы)

**Руководитель практики \_\_\_\_\_\_\_\_Соснин В. В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Фамилия И.О., должность и место работы)

**Оценка, рекомендуемая руководителем \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Практика пройдена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подписи членов комиссии**

**\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)**

(Фамилия И.О.) **\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)**

(Фамилия И.О.)

**\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)**

(Фамилия И.О.)

**Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Санкт-Петербург**

**2014г.**

# Содержание:

[Содержание: 2](#_Toc380520127)

[Система контроля версий Git 3](#_Toc380520128)

[Редкие события. Критические системы. 5](#_Toc380520129)

[Основные сведения о методе Монте-Карло. 5](#_Toc380520130)

[Проблема погрешности при использовании метода Монте- Карло для редких событий. 6](#_Toc380520131)

[Методы 6](#_Toc380520132)

[Метод существенной выборки(выборка по значимости) 6](#_Toc380520133)

# Система контроля версий Git

Существует несколько типов систем контроля версий:

* Локальные – имеется простая база данных, в которой хранятся версии необходимых файлов.
* Централизованные – имеется сервер, на котором хранятся все файлы, клиенты получают копии этих файлов.(прим. CVS)
* Распределенные – в отличии от централизованных, клиенты получают копию всего репозитория, при потере данных, можно восстановить их с помощью клиентской копии. К такому типу СКВ относится Git

Основным отличием Git от других систем контроля версий, таких как CVS, является система хранения данных. Вместо патчей (изменений) используются наборы слепков. При коммите сохраняется слепок того, как выглядят файлы в данный момент( если файл не изменялся, то используется ссылка на его предыдущую версию).

Три состояния файлов: зафиксированный, измененный, подготовленный.

Процесс работы:

1. Внести изменения в файлы
2. Подготовка файлов (индексация)
3. Коммит (подготовленные файлы перемещаются в каталог Git’a на постоянное хранение)

Изученные команды Git:

git init - создание репозитория

git add <filename> - индексация файла (указывается, что файл изменен и должен быть включен в следующий коммит)

git commit – производит коммит проиндексированных файлов

git clone <url>– происходит копирование всех данных с сервера клиенту

git status – определение состояния файлов

git diff – показывает конкретные различия в файлах

git rm <filename>– удалить файл из индекса(при использовании ключа –f принудительное удаление)

git mv <from> <to>- перемещение файла (по сути происходит создание нового файла, в указанном месте, и удаление старого)

git log – просмотр истории, при использовании разных ключей возможны разные форматы получения данных, за разный период, разное количество коммитов и т.д.

git revert HEAD – отмена последнего коммита

git checkout -- <file> - отмена изменений файла (используется до индексации)

git reset HEAD <file> - отмена индексации

Метки(tag) – позволяют помечать определенные моменты в истории, как важные

Бывают легковесные – указатель на коммит(сохраняется только контрольная сумма коммита) и аннотированные(кроме суммы хранятся сведения об авторе, дате и т.д.)

Изученные команды для работы с метками:

git tag – просмотр всех меток

git tag v1.5-lw – создание легковесной метки

git tag –a v1.5 - создание аннотированной метки

Ветка – подвижный указатель на один из коммитов, по умолчанию ветка называется master, указывает на последний сделанный коммити передвигается автоматически

git branch <name> - создание новой ветки

git checkout <name> - переход на другую ветку

git merge <name> - слияние веток

git branch –d <name> -удаление ветки

Конфликты при слиянии веток возможны при изменении одной и той же части кода по-разному( нужно изменить конфликтующие файлы и проиндексировать)

Ссылки на использованные материалы:

1. <http://git-scm.com/book>
2. <http://githowto.com/ru/>
3. <http://pcottle.github.io/learnGitBranching/>

# Редкие события. Критические системы.

Редкие события(медленные процессы) – ситуации, возникающие очень редко, но достаточно важные, чтобы оправдать свое исследование.

Примером редкого события является отказ оборудования.

Критические системы (если редкое событие происходит, то):

* Потери человеческих жизней(метро, самолеты, поезда)
* Значительный материальный ущерб(банковские системы)
* Молекулярные реакции, исследование энергии электрона

Для исследования таких систем, моделирования редких событий применяется метод Монте – Карло(используется в таких областях, как физика, биология, телекоммуникации, анализ риска, транспортные системы)

# Основные сведения о методе Монте-Карло.

Решения научных проблем требуют сложных вычислений сумм, интегралов.

Аналитические методы быстро становятся бесполезными, так как устанавливают жесткие ограничения с точки зрения сложности и допущений(предположений) о модели.

Численный анализ также требует предположений о модели, количество шагов для достижения заданной точности растет экспоненциально с размерностью.

Вместо выше перечисленных методов мы можем использовать методы моделирования Монте-Карло, которые являются методами статистического приближения.

Предположим, что нам нужно оценить вероятность γ некоторого события А.

Модель системы моделируется n раз. На каждой симуляции записываем, происходит событие А или нет.

Если х – случайная величина, то х = 1 событие произошло, х = 0 не произошло. Оценим γ = .

Математическое ожидание E(x) = γ. Дисперсия Var(x) = γ(1-γ).

При использовании предельной теоремы было доказано, что при

Распределение стремится к стандартному нормальному распределению N(0;1) (Распределение Гаусса)

# Проблема погрешности при использовании метода Монте- Карло для редких событий.

Для достижения погрешности в 10% требуется провести несколько сотен миллиардов экспериментов, что является нереальным для достаточно сложных систем. Для обеспечения требуемого ограничения погрешности необходимо увеличить размер выборки. Однако существует проблема в разработке стратегии отбора таким образом, чтобы размер выборки при фиксированной погрешности не увеличивался, если уменьшается вероятность события.

Также проблемой является надежность доверительного интервала. Необходимо проверить охватывает ли интервал теоретические значения при вероятности появления стремящейся к 0.

# Методы

* Существенной выборки
* Расщепления

# Метод существенной выборки(выборка по значимости)

Используется для снижения дисперсии случайной величины. Редкие события являются более важными для оцениваемой модели. Если эти события в нашей выборке будут появляться чаще, то дисперсия уменьшится. Поэтому необходимо выбрать распределение, которое способствует выбору нужных значений случайной величины. Заданная изначально функция при этом изменяется, поэтому конечный результат расчета необходимо перевести с учетом исходных данных. Это делается путем умножения на, так называемое, отношение правдоподобия.

Суть метода:

Начальная функция распределения вероятностей - p

Предположим, что есть какое-то другое распределение q. Оно является более простым, и мы можем сделать выборку.

Тогда алгоритм метода выглядит следующим образом:

1. Взять выборку по q.
2. Подправить полученную выборку так, чтобы получилась выборка p.

Формальный алгоритм:

1. Взять сэмплы(по одной из размерностей)

по распределению q

1. Рассчитать веса(отношение правдоподобия)

1. Оценить функцию по формуле

Однако, используя данный метод, можно столкнуться с некоторыми проблемами:

Нужно заранее выбрать q так, чтобы оно хорошо аппроксимировало функцию p. При неверном выборе функции q можно так и не получить разумную выборку, а только усложнить задачу. Возможна ситуация, при которой веса некоторых сэмплов слишком велики – тогда выборка также не даст адекватных результатов.

Ограниченная относительная погрешность.

Оценку математического ожидания мы можем получить с относительной точностью, только если .

# Эксперимент Монте-Карло в AnyLogic

Возьмем для примера расчет числа π. Теоретические основы:

Площадь круга: , где R – радиус круга

Примем радиус круга равным 1, тогда площадь круга будет равна π.

Для того, чтобы найти площадь круга воспользуемся методом Монте-Карло.

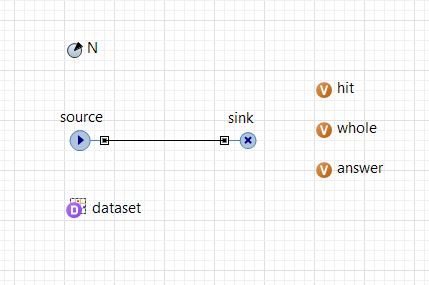
Для определения площади фигуры неопределенной формы, необходимо вписать ее в геометрическую фигуру, площадь которой легко вычисляется (в нашем случае мы впишем фигуру неопределенной формы - круг, в квадрат).

Опишем около круга квадрат. Тогда площадь квадрата будет равна

, т.к. R = 1, .

Далее мы будем бросать в прямоугольник случайные точки, подсчитывая количество тех из них, которые попали внутрь нашего круга. Произведение площади прямоугольника на отношение количества точек, попавших внутрь фигуры (круга), ко всему количеству точек дает приближенное значение искомой площади, т.е. значение числа .

Рассмотрим построенную модель



Элемент source генерирует заявки, у каждой из которой есть переменная x и y. Эти параметры задаются с помощью равномерного распределения (0;1)

Элемент sink уничтожает транзакты, но перед этим проверяет на условие попадания в круг (),и если оно истинно, то увеличивает переменную hit на 1.

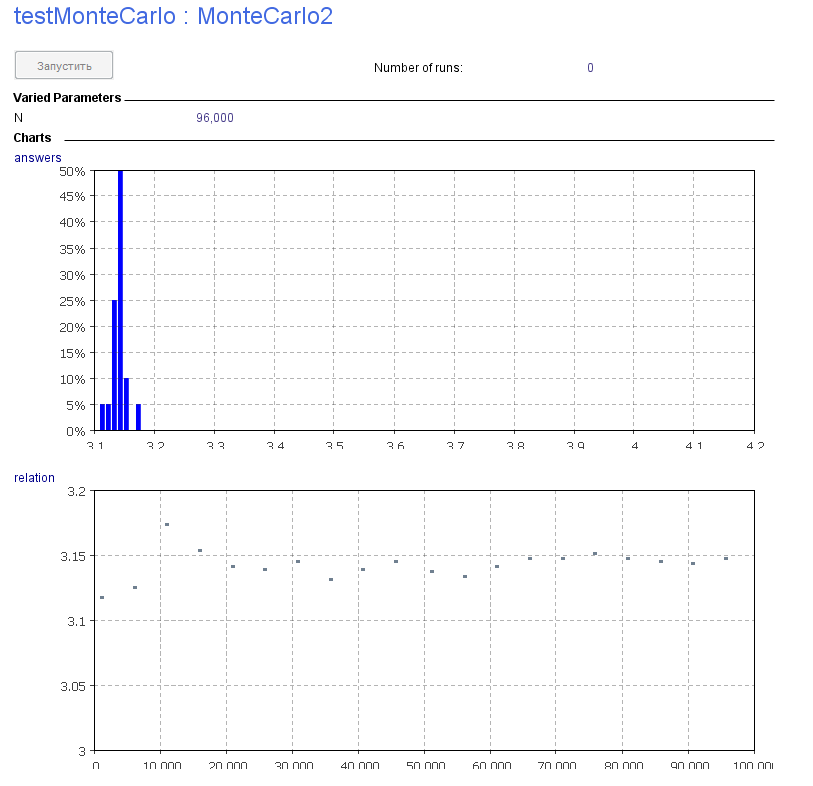
Переменная whole считает общее количество транзактов(используется для контроля правильности работы системы).

Параметр N задает количество транзактов, которые пройдут через систему.

Переменная answer рассчитывает число пи по формуле .

Dataset – используется для сбора статистики и построения графика зависимости точности вычислений от количества заявок(точек).

Далее приведена страница эксперимента Монте-Карло:



На графике relation мы видим зависимость полученного ответа от количества точек. Таким образом, можно сделать вывод, что при увеличении числа использованных точек, точность вычислений возрастает и стабилизируется.

Гистограмма показывает плотность распределения полученных результатов.

50% значений – близки к реальному числу π.

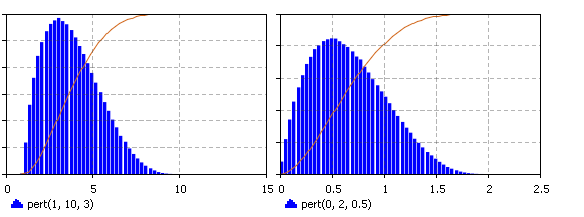
# Моделирование редких событий, путем вычисления площади маленькой фигуры на большом пространстве.

Возьмем за основу предыдущую задачу и уменьшим радиус круга.

Тогда определение площади круга станет невозможным из за слишком редких попаданий в заданную окружностью область.

Для использования метода выборки по важности заменим закон распределения с равномерного на pert.

Pert - это непрерывное распределение, ограниченное с обеих сторон. Оно может использоваться вместо треугольного распределения, имеет те же три исходных параметра: минимальное, наиболее возможное и максимальное значения, но его кривая более гладкая, с меньшим акцентом на экстремумы.



Теперь будем изменять значения R и, зная искомую площадь, вычислим весовой коэффициент для коррекции результатов.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| радиус | 0,50 | 0,25 | 0,15 | 0,01 | 0,001 | 0,005 | 0,025 | 0,050 | 0,10 |
| попадание pert |  |  |  | 308 | 2 | 67,00 | 1787,00 | 6958,00 | 7918 |
|  |  | | | | | | | | |
| площадь искомая | 0,7850 | 0,1963 | 0,0707 | 0,0003140 | 0,0000031 | 0,0000785 | 0,0019625 | 0,00785 | 0,0314000 |
| полученная | 0,787 | 0,194 | 0,071 | 0,00111 | 0,000008 | 0,000276 | 0,007 | 0,028 | 0,109 |
|  |  | | | | | | | | |
| коэффициент |  |  |  | 3,535032 | 2,547770701 | 3,515923567 | 3,566878981 | 3,566878981 | 3,47133758 |

Зеленые ячейки – равномерный закон, Голубые ячейка - pert

Начиная с радиуса равного 0.1 , использование равномерного распределения не дает должной точности, поэтому необходимо использовать другую функцию распределения.

Путем расчетов было доказано, что при отношение правдоподобия не изменяется и остается равным 3.5669. Таким образом можно вычислить любую площадь при радиусе, находящемся в указанном диапазоне без потери точности. Следовательно, для каждого отдельного случая можно высчитать данный коэффициент и производить последующее изучение модели.

При подборе функции h(x) для метода выборки по важности нужно руководствоваться следующими пунктами:

1. h(x) > 0

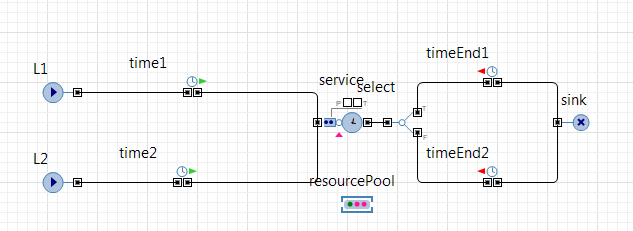
2. h(x) должна быть как можно более пропорциональна исходной функции

3. функцию должно быть легко симулировать

4. плотность функции должна легко вычисляться

# Редкие события в СМО

Построим модель в AnyLogic - СМО с неоднородным потоком заявок.



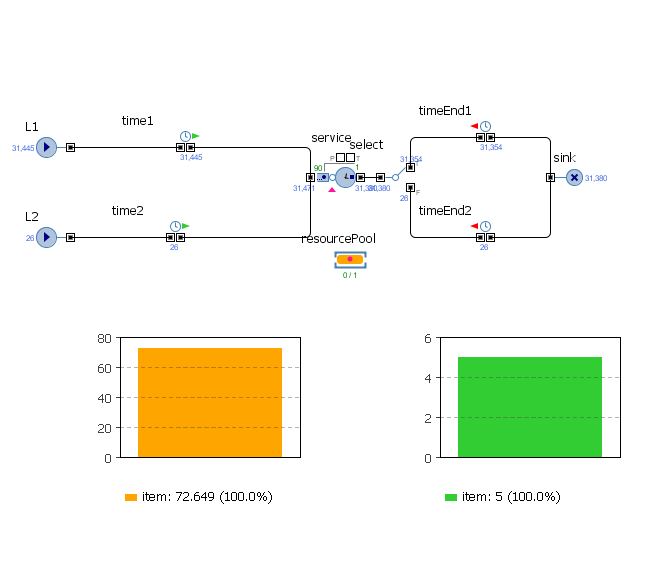
Источники заявок L1 и L2, λ2, но заявки второго класса имеют относительный приоритет над заявками первого класса. Также различается время обслуживания: у заявки с большим приоритетом оно больше.

Пары : time1 - timeEnd1, time2 - timeEnd2 данные элементы предназначены для сбора статистики по времени.

Service – представляет собой очередь и обслуживающий прибор(связан с resourcePool).

Sink –уничтожение заявки.

Запуск модели:



Оранжевая гистограмма показывает время пребывания заявок 1го класса, зеленая – заявок 2го класса.

Анализ гистограмм и модели показывает, что собирается большая очередь, время пребывания заявок 1го класса только увеличивается, а заявка 2го класса прошла всего 1 раз (редкое событие) и результат этого прохода, естественно является не показательным.