

Костюкова Нина Ивановна

**СОЗДАНИЕ НОВОЙ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ В СРЕДЕС++, JAVA НА БАЗЕ
ВЫЧИСЛЕНИЯ ГРУППЫ, ДОПУСКАЕМОЙ ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ**

Адрес статьи: www.gramota.net/materials/1/2010/7/16.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2010. № 7 (38). С. 59-61. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2010/7/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

Мы считаем, что Всероссийская сеть типовых мини-площадок «Атлетическая физкультурно-оздоровительная беседка» может стать основной платформой для формирования в стране «общества активного досуга», культивирующей новую концепцию человеческого общения и доступного улучшения здоровья общества. Функционирование развёрнутой сети «атлетических беседок» не только создаст предпосылки для формирования и возрождения российских традиций разумного оздоровительного досуга, но и станет социальной базой поддержания норм физического статуса населения. Вовлекая в занятия горожан, принадлежащих к различным, в том числе слабо защищённым в социальном отношении возрастным и статусно-имущественным слоям, такая сеть сможет ощутимо повысить качество их жизни и улучшить социальную атмосферу в стране.

Список литературы

1. Варзиев С. Х. Новые технологии совершенствования оздоравливающей физкультурно-атлетической индустрии // Вестник спортивной науки. 2010. № 1. С. 39-43.
2. Варзиев С. Х. Физкультурно-оздоровительный комплекс «Дворовая атлетическая беседка» как эффективная форма реализации современных здоровьесберегающих и здоровьесозидающих технологий по месту жительства и учебы // Современные проблемы науки и образования. 2008. № 5. С. 67-70.

УДК 681.3+518.9

Нина Ивановна Костюкова

Институт вычислительной математики и математической геофизики СО РАН

**СОЗДАНИЕ НОВОЙ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ В СРЕДЕ C++, JAVA
НА БАЗЕ ВЫЧИСЛЕНИЯ ГРУППЫ, ДОПУСКАЕМОЙ ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ[©]**

Как и всякая индустрия, индустрия информатики основана на определенной технологии (ее часто называют новой информационной технологией), а в основе любой технологии лежит наука - в данном случае информатика. Предлагаем частный подход автоматизации научно-исследовательской работы.

В качестве базы рассмотрим проблемы решения нелинейных систем дифференциальных уравнений с частными производными, каким образом новые информационные технологии могут облегчить труд исследователя при нахождении решений этих систем.

В монографии Л. В. Овсянникова [2] дан метод для вычисления группы, допускаемой системой дифференциальных уравнений. Знание группы позволяет строить целый класс точных решений, так называемые инвариантно-групповые решения, отыскание которых сводится к интегрированию более простых систем уравнений. Эти, так называемые определяющие уравнения, представляют собой систему однородных дифференциальных уравнений с частными производными. Определяющие уравнения обладают двумя важными свойствами, облегчающими нахождение их решений: во-первых, они линейны, и, во-вторых, система переопределена (существование решения гарантировано соответствующей теоремой).

Алгоритм построения определяющих уравнений содержит трудоемкие аналитические выкладки, к тому же выражения, с которыми приходится оперировать, весьма длинны. Например, в случае уравнений газовой динамики для 5 независимых и 4 независимых переменных одних только определяющих уравнений будет 680. Естественно, что при выполнении аналитических манипуляций ослабевает внимание и появляется много ошибок, сводящих на нет все исследование. Очевидно, что такую работу следует автоматизировать.

Философские замечания по поводу выбора языков реализации

Для того чтобы автоматизировать процесс получения решений нелинейных дифференциальных уравнений в частных производных, формализуем алгоритмы (1), (2). В качестве инструмента реализации рассмотрим языки программирования C++, Java. Связь между языком, на котором мы думаем (программируем), и задачами, а так же решениями, которые мы можем представлять в своем воображении, очень близка. По этой причине ограничивать свойства языка только целями исключения ошибок программиста неверно. Как и в случае с естественными языками, есть огромная польза владеть, по крайней мере, двумя языками. Язык представляет программисту набор концептуальных инструментов, а если они не отвечают задаче, то их просто игнорируют. Система типов должна быть особенно полезна в нетривиальных задачах. Когда мы программируем, мы создаем конкретное представление идей нашего решения задачи. Пусть структура отражает эти идеи: если это отражение считаем отдельным понятием, сделаем его классом; если считаем его отдельным объектом, сделаем его объектом некоторого класса; если два класса имеют общим нечто существенное, сделаем его базовым классом, если все классы в нашей программе будут иметь нечто общее - введем универсальный базовый класс.

На этапе объектно-ориентированной разработки определяются все необходимые классы и располагаются в порядке иерархии. Рассматриваются две версии реализации. В некотором смысле язык C++ - самый универсальный, так как кроме набора средств, присущих современным языкам программирования высокого уровня (структурность, модульность, определяемые типы данных), в него включены средства для программирования почти на уровне ассемблера (использование указателей, побитовые операции, операции сдвига). Большой набор операторов и операций позволяет писать компактные и эффективные программы.

Java представляет собой современный объектно-ориентированный язык программирования, поддерживающий технологию клиент/сервер. На этапе выполнения объектно-ориентированного анализа делается попытка представить приложение в виде набора взаимосвязанных классов. Даже если единственным достоинством Java была бы возможность создания машинно-независимых приложений, то и этого было бы вполне достаточно, чтобы завоевать расположение широкого круга программистов.

Авторы языка Java воспользовались синтаксисом языка - C++ и позаимствовали лучшие его принципы, в том числе позволяющие легко превращать объектно-ориентированную модель в программу. Одной из целей, которые ставились при разработке языка C++ и Java, было обеспечение возможности хранения машинно-независимых особенностей отдельно от языка программирования, в библиотеках. В языке Java стандартные библиотеки (пакеты) значительно расширены по сравнению с библиотеками языка C++ и включают функции поддержки сетевых соединений. Например, в Java мы можем открыть соединение с Web-страницей или приложением Internet и читать или записывать данные подобно тому, как мы на C++ работаем с локальными терминалами. Создавая программы на Java, мы не задумываемся над тем, в какой операционной системе они будут работать.

Алгоритм вычисления группы

Пусть $x^i (i = 1, 2, \dots, n)$ - независимые переменные, $u^k (k = 1, 2, \dots, m)$ - зависимые переменные и P_α^k - производные:

$$P_\alpha^k = \frac{\partial^{|\alpha|} u^k}{(\partial x^1)^{\alpha_1} \dots (\partial x^n)^{\alpha_n}},$$

где α_i - неотрицательные целые числа. Системы уравнений, для которых отыскивается группа, должны иметь вид:

$$(P_\alpha)_\mu = F_\mu(x, u, c, P), \quad \mu = 1, 2, \dots, M \quad (1)$$

Здесь $F_\mu(x, u, c, P)$ - полиномы от P , коэффициенты которых, в общем случае - рациональные функции x, u, c , а c - скаляры или функции от x, u .

Предполагается, что уравнения системы разрешены относительно старших производных $(P_\alpha)_\mu$, т.е. относительно M производных максимального порядка. Эти производные всюду ниже будем называть главными производными, А остальные - параметрическими производными.

Группа, допускаемая системой (1), вполне характеризуется своим инфинитезимальным оператором

$$X = \xi^i(x, u) \frac{\partial}{\partial x^i} + \eta^k(x, u) \frac{\partial}{\partial u^k}$$

Здесь ξ^i, η^k - координаты оператора; по индексу встречающемуся дважды, предполагается суммирование (тензорная форма записи).

Алгоритм вычисления группы состоит из следующих частей:

- продолжения инфинитезимального оператора на производные максимального порядка, встречающегося в уравнениях;
- взятие производной от исходных уравнений (1) и исключение главных производных;
- получение определяющих уравнений;
- приведение определяющих уравнений к пассивной форме.

Алгоритм приведения определяющих уравнений к пассивной форме

Как правило, система определяющих уравнений получается переопределенной [1]. Это означает, что к системе необходимо добавить условия совместности, получающиеся в результате перекрестного дифференцирования, если только они не обращаются в тождество. Система, в которой автоматически выполняются все условия совместности, называется пассивной. Обычно определяющие уравнения в пассивной форме имеют простой вид и интегрируются элементарно.

Заключение

Разработанные алгоритмы вполне достаточны, для организации связанных между собой понятий в древовидную структуру, что позволяет организовать программу как множество деревьев, т.е. базовых классов, каждый из которых имеет свое собственное множество производных классов.

Список литературы

1. **Костюкова Н. И.** Процессор ПАССИВ // Вычислительная математика и вычислительная техника: сборник / ФТИНТ АН УССР. Харьков, 1972. Вып. III. С. 38-42.
2. **Овсянников Л. В.** Групповые свойства дифференциальных уравнений: монография. Новосибирск: Наука, 1962.
3. **Фиников С. П.** Методы внешних форм Картана. М.-Л., 1948.

УДК 519.6

Нина Ивановна Костюкова, Евгений Витальевич Родин

Институт вычислительной математики и математической геофизики СО РАН

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ ОТРАСЛЕЙ, СВЯЗАННЫХ С РИСКОМ[©]

Последствия решений, принятых нами, проявляются в будущем. А будущее неизвестно. Мы обречены принимать решения в условиях неопределенности. Мы всегда рискуем, поскольку нельзя исключить возможности нежелательных событий. Но можно сократить вероятность их появления. Для этого необходимо спрогнозировать дальнейшее развитие событий, в частности, последствия принимаемых решений. Для этого мы создаем систему, позволяющую дать характеристику любой ситуации и оценить шансы на успех.

Рассмотрим несколько известных классификаций методов Data Mining по различным признакам.

Область применения этой программы очень широка: от решения бытовых вопросов до оценки бизнес проектов.

Проект реализован в программной среде Visual Studio 2008 Express Edition, язык реализации C#.

Постановка задачи

Мы работаем над созданием системы поддержки принятия решений (СППР) для отраслей, связанных с риском. То есть, система должна не только определить вероятность успеха, но и дать соответствующие советы для снижения риска и увеличения шансов на успех.

Мы используем теорию хаоса и теорию риска.

Теория хаоса гласит, что сложные системы чрезвычайно зависимы от первоначальных условий, и даже небольшие изменения в окружающей среде ведут к непредсказуемым последствиям. Следует отметить, что хаос не случаен, несмотря на свойство непредсказуемости. Более того, хаос динамически детерминирован. На первый взгляд непредсказуемость граничит со случайностью - ведь мы, как правило, не можем предсказать как раз случайные явления. И если относиться к изменениям событий как к случайным блужданиям, то это как раз тот самый случай. Однако хаос не случаен, он подчиняется своим закономерностям. Согласно теории хаоса, если вы говорите о хаотичных изменениях, то вы должны иметь в виду не случайный ход событий, а другое. Особенно упорядоченное движение. Если ход событий хаотичен, то он не случаен, хотя и по-прежнему непредсказуем.

Непредсказуемость хаоса объясняется в основном существенной зависимостью от начальных условий. Такая зависимость указывает на то, что даже самые малые ошибки при измерении параметров исследуемого объекта могут привести к абсолютно неверным предсказаниям. Эти ошибки могут возникать вследствие элементарного незнания всех начальных условий. Что-то обязательно ускользнет от нашего внимания, а значит, уже в самой постановке задачи будет заложена внутренняя ошибка, которая приведет к существенным погрешностям в предсказаниях. Применительно к невозможности делать долгосрочные прогнозы погоды существенную зависимость от начальных условий иногда называют «эффектом бабочки». «Эффект бабочки» указывает на существование вероятности того, что взмах крыла бабочки в Бразилии приведет к появлению торнадо в Техасе.

Это обозначает, что для выбора оптимального пути реализации какого-либо проекта мы должны четко представлять структуру событий, из которых этот проект состоит, и которые мы можем осуществить, либо которые являются следствием других событий.

Исходя из теории хаоса, мы также должны определить события, которые могут развиваться хаотично, то есть оценить возможность риска. Для реализации наиболее подходит модель антагонистической игры (игра с нулевой суммой).

Формально антагонистическая игра может быть представлена в виде $\langle X, Y, F \rangle$, где X и Y - множества стратегий первого и второго игроков, соответственно; F - функция выигрыша первого игрока, ставящая в соответствие каждой паре стратегий (ситуации) (x, y) действительное число, соответствующее полезности первого игрока при реализации данной ситуации. Так как интересы игроков противоположны, функция F одновременно представляет и проигрыш второго игрока.