**Компьютерная алгебра**

(курс лекций)

**Содержание лекции **

1.Классификация СКА

2.Типовая архитектура

3.Библиотеки алгоритмов и пакеты расширения

4.Пользовательские интерфейсы 

5.Многообразие реализаций 

6.Перспективные направления развития

**1.Классификация СКА**

Системы компьютерной математики: табличные процессоры: Microsoft Excel, GNU Calc и др. системы для статистических расчётов: SPSS, Statistica и др. системы компьютерной алгебры системы для моделирования, анализа и принятия решений (в том числе, интеллектуальные): GPSS, AnyLogic и др. (DSS, NLP и другие AI-системы) универсальные математические системы: Matlab, MathCAD и др.

**Классификационные признаки:**

* функциональное назначение
* тип архитектуры 
* средства реализации
* области применения
* интегральные оценки качества

**Классы СКА по функциональному назначению:**

* СКА общего назначения (Решение задач для большинства основных разделов символьной математики.)

Примеры: Maxima, Axiom, Maple, Mathematica, Sage, Yacas.

* Специализированные СКА (Решение задач для одного или нескольких смежных разделов символьной математики.)

Примеры: теория групп – GAP тензорная алгебра – Cadabra.

**Классы СКА по типу архитектуры:**

* СКА классической архитектуры: системное ядро + прикладные расширения. Примеры: Axiom, Maple, Mathematica.
* Программный пакет для расширения базовой прикладной математической системы. Пример: Maple (ядро) для Matlab и MathCAD.
* Встраиваемое расширение (плагин) для языка и / или системы программи- рования. Пример: MathEclipse / Symja – Java-библиотека.
* Open Source, GNU GPL, мульти- плат- формные СКА. Примеры: Maxima (Lisp), PARI/GP (C).

**Классы СКА по средствам реализации.**

**1.Аппаратно-программные**

Язык АНАЛИТИК: ЭВМ серии «МИР», АНАЛИТИК-2010.

**2. Программные**

Язык LISP: REDUCE, MATHEMATICA, MACSYMA(MAXIMA), muMATH(DERIVE).

Язык C / C++:MAPLE

**Классы СКА по областям применения:**

* Микрокалькуляторы

Примеры: Hewlett-Packard, Texas Instruments, Casio, Citizen

* Спецпроцессоры

Примеры: БПФ- процессоры для систем ЦОС

* Мобильные устройства и web-сервисы Интернет

Примеры: iPhone и Windows эмуляторы научных калькуляторов HP

* Образование и научные исследовани

Примеры: пакеты СКА

**Классы СКА по интегральным оценкам качества:**

* контролируемая достоверность вычислений
* производительность и масштабируемость
* унификация реализации, информационная совместимость
* архитектурная расширяемость

**2.Типовая архитектура**

**Справочная система**

**Функциональное назначение блоков:**

* **Ядро** содержит машинные коды реализаций операторов и встроенных функций СКА, обеспечивающих выполнение аналитических (символьных) преобразований математических выражений на основе системы определённых правил.
* **Библиотеки алгоритмов** содержат каталогизированные (по типам обрабатываемых абстрактных объектов – числа, функции, алгебры и т.п. и/или методам вычислений – аналитические, численные, смешанные) реализации алгоритмов решения типовых математических задач. Они функционально расширяют ядро СКА. 
* **Пакеты расширения** обеспечивают различные формы адаптации СКА к классам математических задач, внешнему ПО (операционным системам, графическим пакетам и т.п.) и целям пользователей.
* **Пользовательские интерфейсы** обеспечивают поддержку всех функций, необходимых для информационных и управляющих взаимодействий между СКА и пользователями (людьми, программами, аппаратными средствами).
* **Справочная система** содержит описание функциональных возможностей и примеров работы в СКА, информационные сообщения о текущем состоянии системы, а также сведения о математических основах алгоритмов СКА.

**Требования к ядру СКА:**

* **Целевой показатель – производительность**.  Машинно-ориентированный язык реализации (С, С++).  Оптимизация машинного кода (в том числе, с помощью частичной реализации функциональности СКА на языке ассемблера или аппаратно).
* **Целевой показатель – достоверность**.  Функциональная полнота операторов описания и преобразования математических объектов.  Ограничение (в идеале – запрет) модификации ядра пользователями СКА.

**Требования к библиотекам алгоритмов:**

* **Целевой показатель – производительность**.  Распределение состава поддерживаемых СКА алгоритмов символьной математики между ядром и библиотеками по принципу баланса производительности и функциональности.  Исключение из библиотек наименее эффективных (сложных) алгоритмов и алгоритмов решения уникальных (специализированных) математических задач.
* **Целевой показатель – функциональность**.  Поддержка в СКА средств модификации существующих и создания новых библиотек (в том числе, непривилегированными пользователями СКА).  Обеспечение возможности программирования алгоритмов не только на языке самой СКА, но и на языке реализации СКА.

**Требования к пакетам расширения:**

* **Целевой показатель – функциональность**.  Наличие у СКА собственного языка программирования приложений, обеспечивающего одновременно средства абстракции данных для описания произвольных математических объектов и управляющие конструкции достаточные для описания произвольных алгоритмов.  Наличие в СКА средств динамической (в идеале – без рестарта СКА) конфигурации состава и приоритетности выбора алгоритмов (как из базовых библиотек, так и из пакетов расширения) решения (математических и нематематических) задач любыми категориями пользователей СКА.
* **Целевой показатель – совместимость**.  Унификация интерфейсов сопряжения пакетов расширения с СКА, а также (в случае необходимости) непосредственно между собой.  Открытость исходного кода реализации СКА для нейтрализации противоречивых ситуаций взаимодействия с пакетами расширения.

**Требования к пользовательским интерфейсам:**

**Целевой показатель – функциональность.** 

* Унификация элементов наблюдения и управления СКА в соответствии с принципами обслуживания типовых прикладных программ в среде операционной системы (перемещаемые и масштабируемые окна документов, диалоговые и информационные окна, кнопки управления, доступ к периферийным устройствам – принтерам, сетевым адаптерам и т.п.). 
* Возможность настройки архитектуры пользовательского интерфейса в зависимости от квалификации пользователя (начинающий, профессионал, разработчик СКА), специфики решаемых задач (обучение, расчёты, исследования, сопряжение с внешним ПО), вычислительных и графических мощностей инструментальных платформ (суперкомпьютеры, персональные компьютеры, многофункциональные мобильные устройства и т.п.).  Наличие встроенного языка программирования высокого (или сверхвысокого) уровня с расширяемой функциональностью.

**Требования к справочной системе (1):**

**Целевой показатель – информативность. **

* Локализация всех справочных материалов: 
* Одноязычная (национальная), выбираемая пользователем при инсталляции СКА. 
* Двуязычная:
* по умолчанию – интернациональная (англоязычная);
* альтернативная – национальная (по выбору пользователя). 
* Обеспечение системы получения справок с вложенными уровнями абстракции и/или конкретизации информации: 
* Краткая контекстная справка о функциональном назначении выбранного элемента СКА. 
* Справка о синтаксисе и семантике операторов и функций языка СКА (с поясняющими примерами). 
* Справка о схемах решения типовых задач компьютерной алгебры в среде СКА. 
* Справка о математических основах алгоритмов компьютерной алгебры.

**Требования к справочной системе (2):**

**Целевой показатель – информативность. **

* **Обеспечение полноты справочной системы: описание всех, без исключений, функций СКА и эффективности поиска сведений об объекте СКА:** 
* в основном справочнике – по имени, по тематическому разделу и по ключевым словам;
*  в кольцевом списке-журнале предыдущих запросов – последовательно и по тэгам. 
* **Наличие обучающих материалов для различных категорий пользователей:**
* рекламные видеоматериалы для демонстрации основных потребительских характеристик СКА;
* интерактивный учебный курс решения математических задач в среде СКА;  консультант-репетитор, выполняющий пошаговое решение примеров с поясняющими комментариями.

**3.Библиотеки алгоритмов и пакеты расширения: алгоритмы **

* Арифметика произвольной точности 
* Упрощение выражений различной сложности
* Математический анализ (вычисление пределов, производных, интегралов)
* Поиск решений (корней) в математических структурах (уравнения и неравенства; диофантовы и дифференциальные уравнения; рекуррентные отношения) 
* Линейная алгебра (операции над матрицами) 
* Теория графов 
* Теория чисел 
* Булева алгебра 
* Тензорная алгебра

**Библиотеки алгоритмов и пакеты расширения : сервисы **

* Редактор математических формул 
* Арифметика с плавающей точкой 
* 2D- и 3D- графика (статичная и анимационная)
* Интерактивная справочная система 
* Динамическая оптимизация вычислений (баланс интерпретации и компиляции)  Генерация кодов на языках программирования 
* Генерация документов для печатных изданий
* Сопряжение с внешней средой (периферийные устройства, файловый ввод / вывод данных, диспетчеризация виртуальной памяти, прикладные программные оболочки)

**Пользовательские интерфейсы**

**Текстовые:**

• Поле ввода символьных строк

• Поле вывода символьных строк 

**Графические:**

• Ячейки ввода данных / вывода результатов

• Окна отображения графиков 

**Командные:**

• Меню и кнопки управления СКА

• Панели библиотек функций

• Индикаторы состояний СКА

**Многообразие реализаций : универсальные СКА**

** I поколение**

**•** REDUCE (LISP), Anthony C. Hearn

• MACSYMA (LISP), Joel Moses, MTI Æ MAXIMA

• SCRATCHPAD (LISP), Richard D. Jenks, IBM Æ AXIOM

• SAC-1 (FORTRAN), George E. Collins Æ SACLIB

• muMATH (LISP), David Stoutemyer Æ DERIVE

** II поколение**

**•** MATHEMATICA (LISP), Stephen Wolfram, Wolfram Research

• MAPLE (C/C++), Keith Qeddes и Gaston Gonnet, Waterloo University

** III поколение**

• AXIOM (ALDOR, A#), Numeric Algorithm Group (NAG), Great Britain

• MAGMA, University of Sydney, Australia

• MuPAD (C/C++), University of Paderborn, Germany

**Многообразие реализаций : специализированные СКА**

* АНАЛИТИК (Институт кибернетики АН Украины) – системы алгоритмических алгебр В.М. Глушкова
* GAP (GAP group) – теория групп, теория характеров, дискретная математика
* CoCoA (The CoCoa Team) – коммутативная алгебра
* Macaulay2 (Daniel Grayson, Michael Stillman) – алгебраическая геометрия, коммутативная алгебра
* Cadabra (Kasper Peeters) – тензорная алгебра
* KANT / KASH (KANT Group) – теория алгебраических чисел
* TRIP (Jacques Laskar, Michael Gastineau) – небесная механика
* Symbolic C++ [C++]; Meditor [Java]; MathPiper [Java]; DoCon [Haskel] – библиотеки аналитических вычислений для языков программирования

Многообразие реализаций : канонические СКА AXIOM – конструктор алгебраических объектов – 1

AXIOM Tim Daly, 2001:

* FriCAS Waldek Hebisch, 2007
* OpenAXIOM Gabriel Dos Reis, 2007

Строгая типизация всех объектов

Иерархия алгебраических структур: группы, кольца, поля, полиномы и т.п.

Математический язык (A# → Aldor → SPAD) и встроенный компилятор

Стиль грамотного (literate) программирования: программный код, псевдокод и

программная документация (Дональд Кнут, 1981)

Доказательные вычисления

**Многообразие реализаций : канонические СКА**

AXIOM – конструктор алгебраических объектов

**Источники дополнительной информации:**

**Официальные сайты проектов:**

AXIOM - http://axiom-developer.org/

FriCAS - http://fricas.sourceforge.net/

OpenAXIOM - http://www.open-axiom.org/

Видео-навигатор по AXIOM-документации на YouTube:

<http://www.youtube.com/watch?v=CV8y3UrpadY>

**Многообразие реализаций : веб-сервисы**

**Аналитик-С – среда аналитических вычислений**

Официальный сайт проекта:

http://www.sgau.ru/analitik\_c/

**Основные характеристики:**

*  Язык реализации – PHP
*  Платформа реализации – сервер Apache
*  Открытый исходный код
*  Синтаксис и функциональность языка Аналитик:
* Поддержка символов латиницы, кириллицы и греческих
* Двуязычность ключевых слов: пусть (let), вычислить (calculate) и т.п.
* Базовые функции (комбинаторика, полиномы, матрицы, дифференциалы-интегралы, тригонометрия, графика) специальные расширения
* (управление в технических системах)
*  Коллектив разработчиков: аспиранты и студенты СГТУ и СГАУ
*  Формы участия в проекте: производственная практика, ВКР

**Многообразие реализаций : Java-мидлеты**

**Jasymca (Java Symbolic Calculator)**

Официальный сайт проекта:

http://webuser.hs-furtwangen.de/~dersch/jasymca2/indexEN.html

**Основные характеристики:**

*  Язык реализации – Java
*  Открытый исходный код
*  Maxima-подобные синтаксис (ASCII-режим) и функциональность
*  Минимальные требования к ресурсам (объём кода менее 200 Kb)
*  Аппаратные платформы: смартфоны, КПК, ПК
*  Программные платформы: Windows / Linux / MacOS; J2SE / J2ME

Научные микрокалькуляторы HP-15C (HP-48S, HP-95)

http://www.shopping.hp.com/product/calculator/Scientific/1/storefronts/NW250AA%2523ABA

**Функциональные возможности:**

*  Сложные интегралы и производные функций
*  Матрицы с действительными и комплексными аргументами
*  Системы линейных и нелинейных уравнений
*  Численные методы

**Графические микрокалькуляторы** TI-89, TI-92 (TI-92 Plus)

http://www.exponenta.ru/soft/others/ti/ti.asp

**Функциональные возможности:** система компьютерной алгебры DERIVE

**Графический микрокалькулятор HP-50G**

(семейство микрокалькуляторов - HP-49G)

http://ru.wikipedia.org/wiki/HP\_50g

**Функциональные возможности:**

* Язык программирования RPL (Reverse Polish Lisp):LISP-совместимый язык для создания СКА-ориентированных приложений
* Язык программирования HP BASIC:для создания приложений структурного программирования
* Огромная библиотека встроенных функций (> 2300) для решения уравнений, выполнения алгебраических и численных вычислений

**Сравнительный анализ различных СКА:**

**Сводный перечень СКА (Википедия):**

http://en.wikipedia.org/wiki/Comparison\_of\_computer\_algebra\_systems

**Обзорные статьи:**

*  Я. ван Хюльзен, Ж. Калме. Системы компьютерной алгебры // В кн.: Компьютерная алгебра: Символьные и алгебраические вычисления. – М.: Мир,1986.
* Мысовских В.И. Системы компьютерной алгебры и символьные вычисления // Записки научных семинаров ПОМИ РАН, 2001, том 281.
* Кулябов Д.С., Кокотчикова М.Г. Аналитический обзор систем символьных вычислений // Вестник РУДН, серия «Математика. Информатика. Физика», 2007.

**Перспективные направления развития**

(1) Расширение состава встроенных и программируемых типов математических объектов;

(2) Интеграция СКА с другими компьютерными системами;

(3) Унификация и объектная ориентация интерфейса пользователя;

(4) Программирование символьных вычислений произвольной сложности;

(5) Ускорение работы СКА.

**Направление (1) :**

• Принадлежность математического объекта СКА к встроенным

должна определяться не случайной практической

необходимостью (СКА ранних поколений), а ролью в

иерархической системе математических абстракций (СКА Axiom).

• Способность к созданию расширений (объектных, структурных,

функциональных и т.п.) СКА должна поддерживаться

интерфейсом (желательно с помощью объектно-

ориентированного, специализированного языка

программирования) (СКА Maple, СКА Mathematica).

• Интеграция ядра и расширений СКА должна быть прозрачной

для любого пользователя СКА (цель пользователя – решение

прикладной задачи, а не организация взаимодействия

вычислительных модулей).

**Направление (2) :**

• Связь с программами числовой обработки

* Связь типа «СКА(СВМ)» или типа СВМ(СКА): вставка «машинных» кодов программ на процедурных языках в тело программ аналитических вычислений или наоборот(язык С и СКА Mathematica).
* Связь типа «СКА+СВМ»: обмен результатами вычислений с помощью файлов.
* Связь нецелесообразна: разработка специализированной системы смешанных (численно-аналитических) вычислений.

• Генерация текста программ вычислений

* Поддержка не одного (СКА Reduce – язык Fortran), а нескольких (СКА Maple, СКА Mathematica – языки Fortran и C) целевых языков программирования.
* Сложно-структурированные математические выражения наглядно создаются в СКА, а затем без ошибок (!) транслируются в строковую форму операторов присваивания.

• Связь с текстовыми процессорами

* Поддержка в СКА общепринятых форматов нетекстовых объектов – формул, графиков, рисунков – обеспечивает либо полную, либо частичную вёрстку научных документов (TEX-формат).

**Направление (3) :**

• Для унификации пользовательский интерфейс СКА должен

иметь те же функциональные возможности, что и интерфейсы

других сред программирования и проектирования

(настройка параметров, редактирование объектов, отладка

проектов и т.п.).

• Для объектной ориентации необходима реализация специальных

классов объектов, представляющих алгебраические и другие

абстрактные математические категории

(тождества, многообразия, исчисления и т.п.).

• Для образовательных и рекламных целей требуется наличие

инструментальных средств создания интерактивных документов

(анимационная графика, панели управления и т.п.).

**Направление (4) :**

• Увеличение количества встроенных в СКА библиотек шаблонов пользовательских приложений для различных предметных областей (СКА Maple).

• Использование в качестве языка реализации СКА – функционально

расширяемого языка программирования (LISP), обеспечивающее не только неограниченный рост сложности вновь создаваемых приложений, но и совершенствование базовых объектов и алгоритмов аналитических вычислений (СКА Mathematica).

**Направление (5) :**

• Постоянное совершенствование способов символьного представления математических объектов и алгоритмов выполнения аналитических преобразований.

• Применение технологии JIT-компиляции машинных кодов для реализации наиболее трудоёмких операций компьютерной алгебры (решение дифференциальных уравнений, статистическое моделирование и т.п.).