



Dr. Vaganov V.A. was born on February 2, 1946 in Primorye Territory (Russia). Now, Vaganov V.A. is the director of the firm Fortex and owner of the firm Sinfex, engaging of problems of delivery of industrial materials to firms of the Esthonian republic. Simultaneously he is executive director of the Baltic branch of the International Academy of Noosphere (IANSO) whose scientific results have received international recognition, first, in the field of the theory of Cellular Automata. Vaganov V. is known for the investigations on automation of economical and statistical works. Result was a series of the scientific and applied works. Vaganov V. is the honorary member of the IANSO and the author of more than 40 scientific publications, including 7 books.

Professor Aladjev V. was born on June 14, 1942 in the town Grodno (Byelorussia). Now, he is the First vice-president of the International International Academy of Noosphere and academician-secretary of Baltic branch of the Academy whose scientific results have received international recognition, first, in the field of the theory of Cellular Automata. Aladjev V. is known for the works on mathematical packages too. He is full member of a number of Russian and International Academies. Aladjev V. is the author of more than 450 scientific publications, including 75 books, published in many countries. He participates as a member of the organizing committee and/or a guest lecturer in many international scientific forums in mathematics and cybernetics.



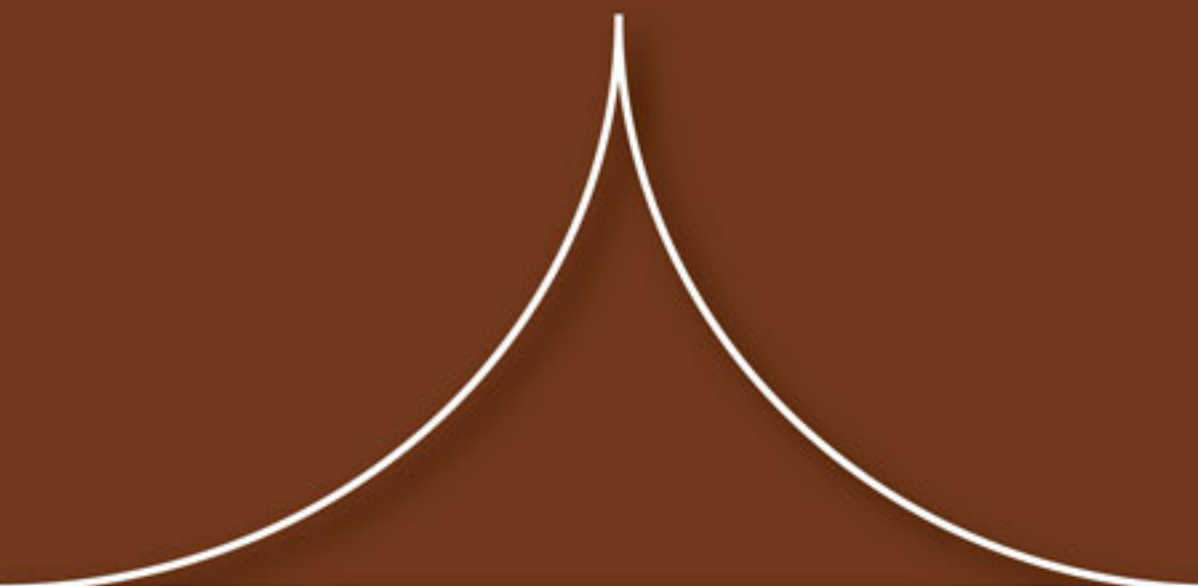
Пакеты Maple и Mathematica являются бесспорными лидерами среди современных систем компьютерной алгебры. В этой связи возникает вполне естественный вопрос о предпочтительности той либо иной из этих систем. Данный вопрос весьма многоаспектен ввиду многофункциональности обоих пакетов. В данной связи в настоящей книге представлен сравнительный анализ обоих пакетов относительно возможностей их программных сред для модульного программирования задач пользователей различной сложности и направленности.

*Published by Fultus Corporation
www.fultus.com*



В.З. Аладьев, В.А. Ваганов * Модульное программирование: Maple vs Mathematica, and vice versa

Модульное программирование: Maple vs Mathematica, and vice versa



```
diff := proc(x::algebraic, y::algebraic) local h, Res;
  if type(x, 'numeric') or type(y, 'numeric') then return x else Res := x end if;
  for h in [args[2 .. -1]] do if type(h, 'symbol') then Res := diff(Res, h)
    else Res := Subs(k = h, diff(Subs(h = k, Res), k)) end if
  end do; simplify(Res)
end proc;
Df[x_, y_] := Module[{a}, If[! HowAct[y], D[x, y], Simplify[Subs[D[Subs[x, y, a], a], a, y]]]
```

В.З. Аладьев, В.А. Ваганов

Модульное программирование Maple vs Mathematica, and vice versa

В.З. Аладьев, В.А. Ваганов



Fultus™ Books



Modular programming: Maple vs Mathematica, and vice versa

by

V.Z. Aladjev and V.A.Vaganov

ISBN-10: 1-59682-268-6

ISBN-13: 978-1-59682-268-9

Copyright © 2011 by V.Z. Aladjev and V.A.Vaganov

All rights reserved.



Published by Fultus Corporation

Corporate Web Site: <http://www.fultus.com>

Fultus eLibrary: <http://elibrary.fultus.com>

Online Book Superstore: <http://store.fultus.com>

Writer Web Site <http://writers.fultus.com/aladjev/>



No part of this book may be used or reproduced in any manner whatsoever without written permission except in the case of brief quotations embodied in critical articles or reviews; violators will be prosecuted to the fullest extent permissible by law.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information.

However, the information contained in this book is offered without warranty, either express or implied.

Neither the author nor the publisher nor any dealer or distributor will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Table of Contents

Предисловие	5
Глава 1. Краткий экскурс в историю компьютерной алгебры	17
Глава 2. Общие сведения по языкам программирования пакетов Maple и Mathematica	27
Глава 3. Базовые управляющие структуры Maple и Mathematica	37
3.1. Управляющие структуры ветвления Maple и Mathematica	40
3.2. Циклические управляющие структуры обоих пакетов	46
3.3. Специальные типы циклических управляющих структур пакетов Maple и Mathematica	53
3.4. Оценка механизмов циклических управляющих структур пакетов Maple и Mathematica	59
Глава 4. Механизм типирования объектов в среде обоих пакетов	61
4.1. Механизм типирования объектов в среде пакета Maple	63
4.2. Механизм типирования объектов в среде пакета Mathematica	73
4.3. Оценка механизмов типирования Maple и Mathematica	79
Глава 5. Организация обработки особых и ошибочных ситуаций	89
5.1. Обработка особых и ошибочных ситуаций в среде Maple	89
5.2. Обработка особых и ошибочных ситуаций в Mathematica	101
5.3. Оценка эффективности пакетов Maple и Mathematica по организации обработки особых и ошибочных ситуаций	106
Глава 6. Организация механизма процедур и модулей в пакетах Maple и Mathematica	107
6.1.1. Определения процедур в пакете Maple и их типы	108
6.1.2. Формальные и фактические аргументы Maple-процедуры	117
6.1.3. Локальные и глобальные переменные процедур в Maple	123
6.1.4. Определяющие параметры и описания Maple-процедур	132
6.1.5. Механизмы возврата процедурой результатов ее вызова	145
6.1.6. Расширенные средства Maple для работы с процедурами	151
6.1.7. Расширение функциональных средств Maple-языка пакета	160
6.1.9. Элементы отладки процедур и функций в среде Maple	188
6.1.10. Организация программных модулей в Maple-языке	194
6.1.11. Сохранение процедур и программных модулей в файлах	208
6.1.12. Создание пользовательских пакетных модулей в Maple	218
6.1.13. Статистический анализ библиотек в среде пакета Maple	226
6.2.1. Определения процедур в пакете Mathematica и их типы	234

Table of Contents

6.2.2. Локальные переменные процедур в пакете Mathematica	237
6.2.3. Глобальные переменные процедур и механизмы возврата результатов процедурами пакета Mathematica.....	247
6.2.4. Некоторые примеры оформления Mathematica-процедур	250
6.2.5. Функциональные конструкции в пакете Mathematica	279
Глава 7. Организация работы с пакетами в среде Mathematica.....	296
Глава 8. Средства ввода/вывода пакетов Maple и Mathematica.....	312
8.1. Средства Maple для работы с внутренними файлами	312
8.2. Средства Mathematica для работы с внутренними файлами.....	317
8.3. Средства Maple для работы с внешними файлами данных.....	321
8.4. Средства Mathematica для работы с внешними файлами.....	336
8.5. Сравнение средств доступа к файлам Maple и Mathematica.....	358
Глава 9. Организация программного обеспечения пользователя	359
9.1. Организация программного обеспечения в среде Maple	361
9.1.1. Классический способ создания Maple-библиотек	364
9.1.2. Специальные способы создания библиотек пользователя.....	372
9.2. Организация программного обеспечения пользователя в среде математического пакета Mathematica	381
9.3. Оценка эффективности пакетов Maple и Mathematica по организации пользовательского программного обеспечения.....	390
Глава 10. Maple или Mathematica – субъективная точка зрения.....	398
Литература	413
Список пользовательских процедур для пакетов Mathematica и Maple, упоминаемых в настоящей книге	417

Предисловие

Системы компьютерной алгебры (СКА) находят все более широкое применение в целом ряде областей как естественных, так и экономико–социальных наук таких как: химия, математика, физика, информатика, техника, технологии, образование и т.д. Системы такие как *Maple, Mathematica, REDUCE, MuPAD, Derive, Magma, Axiom, GAP, Maxima, MathPiper* и др. все более востребованы для преподавания [20,53,60,82] математически ориентированных дисциплин, в научных исследованиях и технологиях. Эти системы являются основными инструментами для ученых, исследователей, преподавателей и инженеров. Исследования на основе СКА–технологии, как правило, хорошо сочетают алгебраические методы с продвинутыми вычислительными методами. В этом смысле СКА – междисциплинарная область между математикой и информатикой, в которой исследования сосредоточиваются как на разработке алгоритмов для алгебраических (символьных) и численных вычислений и обработки данных, так и на создании языков программирования и программной среды для реализации *подобного* типа алгоритмов и базирующихся на них задач различного назначения.

Компьютеризация – составляет одну из наиболее актуальных проблем современного общественного прогресса, технической основой решения которой является наличие **персональных компьютеров (ПК)** и развитых телекоммуникационных средств связи, позволяющих создавать весьма эффективные информационно–вычислительные сети различных уровня и назначения. Второй важнейшей компонентой данного процесса является наличие достаточно развитых программных средств, предназначенных для решения разнообразных задач из различных областей человеческой деятельности, в том числе включая средства поддержки *эксплуатации* самих **ПК** в различных режимах обработки информации и телекоммуникации.

Решение прикладных пользовательских задач в той или другой области приложений поддерживают **пакеты прикладных программ (ППП или просто пакеты) специального, узко–специального либо общего назначения**. Классификация и характеристика данного класса **программных средств (ПС)** может быть найдена в наших предыдущих книгах [1–3,36]. Естественно, квалифицированный пользователь, владеющий в достаточной мере одним из эффективных языков программирования (*например, Basic, C, Fortran, PL/1, Pascal, Lisp, Prolog и др.*), в целом ряде случаев для решения своих задач может самостоятельно написать и отладить отдельную программу или комплекс программ, позволяющих реализовать на **ПК** алгоритм его задачи. Причем, в ряде случаев такой подход может быть более эффективным, чем использование для этих целей готовых **ПС**, т.к. разработчик **ПС одновременно** хорошо владеет спецификой решаемой задачи и условиями ее эксплуатации. Однако, такой подход требует, как правило, серьезных трудозатрат и при нынешнем обилии различного типа и назначения средств для **ПК (особенно широко распространенных IBM–совместимых)** в значительной мере становится нецелесообразным. Вместе с тем, развитые **ППП** снабжаются собственным *встроенным* языком программирования того или иного уровня сложности, позволяющим в среде пакета программировать целые задачи или их отдельные фрагменты, которые может быть нецелесообразно, неэффективно, а в ряде случаев и невозможно, реализовывать

стандартными средствами пакета. Настоящая книга посвящена классу *ПС* (названному *системами компьютерной алгебры*), которые, в первую очередь, предназначены для решения задач математического характера, и, прежде всего, лидерам в данном классе пакетам *Maple* фирмы *MathSoft Inc.* и *Mathematica* фирмы *Wolfram Research Inc.* Книга является вполне естественным продолжением нашей серии книг по вычислительной технике и программному обеспечению: *ЕС ЭВМ (IBM 360/370)*, *СМ ЭВМ (PDP-11)*, *ПК ИСКРА 226 (WANG 2200-MVP)*, *ИСКРА 1030 (IBM PC/XT)*, а также средствам для *IBM-совместимых ПК* [36]. При этом, большое внимание нами уделялось как опыту работы с описываемым средством, так и особенностям его применения, а также вытекающим из этого рекомендациям пользователю. По мере возможности, предлагались наиболее эффективные технологии применения данных средств для решения тех либо других прикладных задач пользователя. Довольно детальная характеристика данной серии книг может быть найдена, например, в [36] и в настоящей книге не рассматривается.

В настоящее время *ПС*, ориентированные на решение *математических* задач (где под *математической* понимается любая задача, чей алгоритм может быть описан в терминах того или иного раздела математики), весьма обширны и в определенной степени могут быть условно дифференцированы на 5 уровней: (1) встроенные средства различной степени развития той либо иной системы программирования; (2) специальные языки программирования; (3) узко-специальные, (4) специальные, (5) общие *ППП*. К *первому* уровню могут быть отнесены такие системы программирования как *Basic*, *C*, *Pascal-XSC*, *PL/1* и др., ко *второму* - *Fortran*, *ISETL*, *Prolog* и др. *Третий* уровень может быть представлен как библиотеками математических подпрограмм (*SSP*, *NAG*, *ППП-БИМ* и др.), так и узко-специальными пакетами *MacMath*, *Phaser*, *VossPlot*, *Eureka* и др. К *четвертому* уровню можно отнести такие пакеты как *S-Plus*, *XploRe*, *SAS*, *Dynamics*, *StatGraf*, *SPSS*, *BMDP*, *PL/1-Formac*, *Systat* и др. Так, в [1-7] дан довольно подробный обзор *ПС*, ориентированных на решение задач в области математической статистики, прикладных статистик и экономического анализа математическими методами.

Пятый уровень ранее представляли *три* основных математических пакета *MathCAD*, *Reduce* и *MatLab*. В указанной литературе приведены довольно подробное описание этих популярных пакетов; особенности их эксплуатации и использования в решении различного типа математических задач. При этом, на основе *всесторонних* апробации и адаптации для отечественные *ПК* пакетов *MathCAD* и *REDUCE* в наших книгах [1-7] были представлены достаточно детальный анализ особых и ошибочных ситуаций, рекомендации по использованию и предложения по дальнейшему развитию данных пакетов. Немало из указанных аспектов, доведенных до сведения разработчиков этих пакетов, было впоследствии учтено при создании последующих версий пакетов. Наш опыт эксплуатации указанных пакетов оказал существенное влияние на апробацию и использование сравниваемых здесь пакетов *Mathematica* и *Maple*. Естественно данная рубрикация программных средств во многом носит субъективный характер, носящий следы опыта *нашей* работы с *ПС*, в той или иной мере ориентированными на вопросы программирования математических задач как в числовом, так и в символьном видах.

Наконец, *современное* развитие компьютерных технологий, ориентированных, прежде всего, на создание интегрированных пакетов *multimedia*-технологии вскоре привело к

появлению нового уровня математических пакетов, из которых наиболее известными являются пакеты *Maple* и *Mathematica* соответственно фирм *MapleSoft Inc.* и *Wolfram Research Inc.* Эти пакеты, превосходя по целому ряду важных показателей упомянутые средства 5-го уровня, вместе с тем, наследуют целый ряд их стандартов, как пионеров-эталонов ПС такого типа, что достаточно легко прослеживается при более детальном их рассмотрении, использовании и апробации.

В среде отечественных пользователей CAS наибольшую популярность завоевал пакет *Maple*; в качестве некоторого косвенного подтверждения степени популярности обоих пакетов в СНГ может служить посещаемость форумов по таким пакетам на наиболее известном русскоязычном образовательном математическом сайте *www.exponenta.ru*. На сегодня на данном сайте форум *Maple* имеет 2066 тем и 8885 сообщений, тогда как форум *Mathematica* лишь 772 тем и 2558 сообщений. Естественно, к такой статистике следует относиться достаточно осторожно, ибо она не совсем верно отражает степень популярности обоих пакетов и вот почему. Прежде всего, данная статистика связана с образовательным процессом в университетах, которые используют именно *Maple* по двум основным причинам, а именно: (1) *Maple* более лоялен к его нелегальному использованию и (2) как для преподавательского состава университетов, так и (прежде всего студентов) он оказался более простым в освоении, прежде всего тем, что его язык синтаксически более близок к известным императивным языкам программирования, в частности, к *Pascal*. А как известно, в общем случае императивные языки несколько проще в освоении, чем функциональные, хотя и здесь не так все однозначно. Может быть, именно по этой причине, несмотря на двадцатилетний возраст *Mathematica*, ее достаточно мощные вычислительные возможности наряду с возможностью установки на такие операционные системы, как *Windows, Linux, Mac OS X* популярность данной CAS на постсоветском пространстве относительно невелика. В беседах, проведенных нами на целом ряде мастер-классов по CAS, многие слушатели считают *Mathematica* слишком сложной, а в функциональном отношении избыточной. И если с первым, в определенной мере можно согласиться, то вторым пакет *Maple* также злоупотребляет, пусть и не в такой степени. Об этом несколько детальнее речь будет идти ниже. Нами специально был проанализирован целый ряд тем и сообщений на указанном сайте в форумах по обоим пакетам и картина сложилась следующая. На обоих этих форумах, в массе своей, обитают студенты, пытающиеся решить свои задачки чужими силами, и если *Maple* им по каким-то причинам не удалось освоить на относительно низком уровне, то более сложный для восприятия неподготовленному пользователю второй пакет для своего освоения требует больших усилий. Тогда как квалифицированные пользователи обоих пакетов, как правило, не посещают форумов такого уровня. Итак, приведенная статистика с определенной долей достоверности может быть отнесена к студенческой среде, хотя и она представляет существенный пласт пользователей CAS.

Кратко охарактеризуем оба рассматриваемых пакета. *Maple* – система компьютерной математики, рассчитанная на весьма широкий круг пользователей. Довольно часто ее называют системой компьютерной алгебры ввиду ее ориентации прежде всего на выполнение символьных вычислений и преобразований. Однако, это не совсем так, т. к. на самом деле система способна довольно эффективно выполнять не только сугубо

символьные, но и численные вычисления в сочетании с довольно развитыми средствами графического отображения и подготовки электронных документов, отражающих ход решения пользовательской задачи в среде пакета. Более того, росту эффективности в направлении *численных* вычислений способствовало не только улучшение численных алгоритмов средств собственно *самого* пакета, но и имплантация в его *вычислительную* среду высоко эффективных матричных вычислений известного пакета *NAG (Numeric Algorithms Group)*. *Maple* – довольно хорошо продуманная система, которая с равным успехом может использоваться как для простых, так и для достаточно сложных задач, связанных как с символьными, так и с численными алгоритмами вычислений. Пакет в настоящее время с успехом используется в самых различных областях.

Исследователи используют пакет *Maple* как важный инструмент для решения разных проблем, связанных с их исследованиями. Пакет идеален (*по нынешним понятиям*) для формулировки, решения и исследования *различных* математических моделей. Так, его алгебраические средства существенно расширяют диапазон проблем, которые могут быть достаточно эффективно решены на качественном уровне. Тогда как педагоги в средних школах, колледжах и университетах обновляют свои традиционные учебные планы, вводя в них задачи и упражнения, которые используют *диалоговую* математику и физику *Maple*. Студенты могут сконцентрироваться на важных концепциях, а не на утомительных алгебраических вычислениях и преобразованиях. Наконец, инженеры и специалисты в промышленности используют пакет *Maple* как весьма эффективный инструмент, заменяющий много традиционных ресурсов, в частности, справочников, калькуляторов, крупноформатных таблиц, редакторов и языков программирования. Эти пользователи достаточно *легко* решают весьма широкий диапазон математически ориентированных задач, разрабатывая проекты и объединяя результаты вычислений (*как числовые, алгебраические, так и графические*) в профессиональные отчеты довольно высокого качества. Отмеченные возможности системы *Maple* в значительной степени обусловлены ее интегрированностью, включающей такие основные компоненты, как:

- мощный встроенный императивный язык программирования процедурного типа;
- редактор для подготовки и редактирования документов и программ;
- современный развитый графический пользовательский интерфейс;
- развитая справочная система по всем средствам с множеством примеров;
- эффективные алгоритмы и правила преобразования математических выражений;
- символьный и численный процессоры;
- довольно развитая система диагностики;
- библиотеки встроенных и дополнительных функций;
- большой набор пакетных модулей, ориентированных на различные приложения;
- большой набор пакетных модулей, разработанных широким кругом пользователей, ориентированных на различные приложения и свободно распространяемых, и т.д.

Система *Maple* воплощает новейшую технологию символьных вычислений, числовых вычислений с любой точностью, расширяемой технологии интерфейса пользователя (*Maplets*), наличие инновационных *Web*-компонентов, довольно развитых алгоритмов для решения сложных математических задач и др. В настоящее время у пакета *Maple*

более, чем 3 миллиона пользователей: ученые, студенты, специалисты, исследователи из различных областей. Многие ведущие университеты и научно-исследовательские институты в мире, включая такие как *MIT, Cambridge, Stanford, Oxford, Waterloo* и целый ряд др., используют пакет для учебных и исследовательских целей. В промышленных целях пакет используется и такими ведущими корпорациями как *Boeing, Bosch, Canon, NASA, Toyota, Sun Microsystems, Hewlett Packard, Motorola, General Electric, Daimler-Chrysler, Ford* и др. История пакета **Maple** занимает более 20 лет развития и апробации; за этот период он развился в весьма сложный многофункциональный продукт. Пакет **Maple** реализован на больших ЭВМ, рабочих станциях, ПК, работающих с операционными системами **Unix, Windows** и др. Такой обширный набор операционных платформ и довольно широкий круг пользователей весьма положительно сказались на отработке пакета. Более того, упрощенная версия **Maple** для операционной системы *Windows CE* используется, в частности, и в миниатюрных компьютерах фирмы *Casio*. Ряд других математических пакетов поддерживают интерфейс с пакетом **Maple** для обеспечения символьных вычислений и преобразований в своей среде.

Вместе с тем современное развитие пакета **Maple** вызывает и ряд серьезных опасений, которые в общих чертах можно определить следующим образом. Качество хорошего программного обеспечения *определяется* целым рядом характеристик, среди которых можно отметить такие важные как: совместимость снизу-вверх, устойчивость работы относительно операционных платформ, качественные сопровождение и поддержка, и др. По данным показателям релизы пакета **Maple**, начиная с 7-го, оставляют желать лучшего, а именно. Из релиза в релиз переходят довольно существенные ошибки (*из которых многие неоднократно отражались как в наших книгах и статьях, так и во многих других источниках, включая многочисленные форумы по Maple*). Отсутствует приемлемая совместимость релизов снизу-вверх. О несовместимости релизов **Maple** мы отмечали неоднократно в своих книгах и статьях. Кое-что для улучшения совместимости нами было сделано (*в частности, посредством упоминаемой в книге нашей библиотеки [45]*), но, однако, не все. Между тем, для релиза 9 была обнаружена несовместимость уже среди его клонов. Как известно, **Maple 9** поддерживает два режима работы – *стандартный и классический (ядра maplew9.exe и cswmaple9.exe соответственно)*. Оказывается, эти клоны несовместимы даже на уровне своих встроенных функций. Так, если в классическом режиме функция ``system`` выполняется корректно, то в стандартном режиме, выдавая код возврата 0, она некорректно выполняет некоторые команды/программы **MS DOS**. Именно поэтому, процедуры, использующие эту функцию и отлаженные в релизах 8 и ниже, а также в классическом **Maple 9**, в стандартном режиме **Maple 9** выполняются некорректно (*часто иницилируя непредсказуемые ошибочные ситуации*). Естественно, что такого типа нарушения требований к качественному программному обеспечению не допустимы для программных средств подобного типа и вполне могут инициировать нежелательные для пакета **Maple** последствия. Причиной же подобных ситуаций, как нам кажется, является то, что разработчики развивают пакет, как говорится, без учета того, что *новшества* могут несогласовываться со средствами предыдущих релизов, что в результате и приводит к отмеченным ситуациям по несовместимости релизов. Нам не хотелось бы думать, что несовместимость релизов обуславливается коммерческими соображениями, но и другие соображения не делают чести разработчикам пакета.

Тем временем, наш эксплуатационный опыт в течение 1997 – 2010 г.г. с пакетом *Maple* релизов 4 – 11 позволил нам не только оценить его преимущества относительно ряда других подобных пакетов, но также выявил ряд ошибок и недостатков, устраненных нами. При этом, пакет *Maple* не поддерживал целый ряд довольно важных процедур обработки информации, алгебраических и численных вычислений, включая также и средства доступа к файлам данных. В свете сказанного, в процессе работы с пакетом *Maple* мы разработали достаточно много эффективного программного обеспечения различного назначения (*процедуры, программные модули*), целым рядом характеристик расширяющих базовые и по выбору возможности пакета. Программное обеспечение было организовано в виде Библиотеки [45], структурно подобной *главной* библиотеке *Maple* и обеспеченной развитой справочной системой, аналогичной подобной *Help*-системе пакета *Maple* и органично с ней связанной. Достаточно детальное описание данной Библиотеки может быть найдено в наших работах [9-14,28-37,42-45].

Программные средства, составляющие Библиотеку, в большинстве своем имеют дело именно с *базовой* средой пакета *Maple*, что пролонгирует их актуальность на текущие релизы, начиная с шестого, и на последующие релизы пакета. В данной связи вполне уместно обратить внимание на один *довольно* существенный момент. При достаточно частом объявлении о новой продукции *MapleSoft*, тем временем, уделяет *недостаточно* внимания устранению имеющихся ошибок и недостатков, переходящих от релиза к релизу. Некоторые из которых являются достаточно существенными. В своих книгах мы неоднократно отмечали данное обстоятельство, этому вопросу посвящено немало замечаний и членов *MUG*. При этом, расширению инструментальных средств *базовой* среды пакета также уделяется явно недостаточное внимание, что особенно заметно в режиме продвинутого программирования в его среде. Объявленная нами библиотека содержит расширение инструментальных средств, прежде всего, *базовой* среды *Maple*, что пролонгирует их актуальность также на последующие релизы пакета, достаточно существенно упрощая программирование целого ряда задач в его *среде* и обеспечивая более высокий уровень совместимости релизов 6 – 11.

Хочется надеяться, что наши издания по пакету *Maple*, содержащие многочисленные замечания, рекомендации и описания процедур, как расширяющих его возможности, так и устраняющих замеченные недостатки пакета, наряду с нашей Библиотекой [45], расширяющей функциональные возможности пакета и устраняющей целый ряд его существенных недоработок, сыграли вполне определенную роль в деле дальнейшего развития и усовершенствования *Maple*. По меньшей мере косвенные данные говорят в пользу данного предположения. В частности, особенно это заметно в плане средств доступа к файлам данных, хотя и здесь отсутствуют подвижки по целому ряду весьма важных позиций, эффективно разрешенных средствами нашей Библиотеки [45], и в определенной мере отмеченных здесь в контексте сравнения *Maple* и *Mathematica*.

Пакет *Mathematica* наряду с уже упомянутым пакетом *Maple* является одной из *CAS*, наиболее известных и пользующихся популярностью систем. *Mathematica* содержит большое число функций для обеспечения как символьных преобразований, так и для численных вычислений. Наряду с этим, пакет поддерживает работу с графикой 2 и 3 измерений, импорт и экспорт изображений и звука и целый ряд других. *Mathematica*

обеспечивает возможность решения следующих основных классов задач, а именно:

– интегрирование и дифференцирование функций, решение систем *полиномиальных* и тригонометрических уравнений и неравенств, рекуррентных уравнений, решение дифференциальных уравнений и уравнений в частных производных, ряды Тейлора, упрощение выражений, вычисление пределов, нахождение конечных и бесконечных сумм и произведений, а также целый ряд других задач в символьном виде;

– решение систем уравнений, вычисление пределов, полиномиальная интерполяция функций, нахождение сумм и произведений, вычисление значений элементарных и специальных функций с произвольной степенью точности, преобразование Лапласа, интегрирование и дифференцирование, решение дифференциальных уравнений и уравнений в частных производных, а также целый ряд других в численном виде;

– решение задач из линейной алгебры, теории чисел и других разделов математики;

– задачи графического представления информации (*построение графиков функций по их формулам, параметрических кривых и поверхностей, построение геометрических фигур, импорт и экспорт графики во многих растровых и векторных форматах*), включая также анимацию графических объектов;

– начиная с последних версий, пакет поддерживает распределенные вычисления, для чего используется пакет *Parallel Computing Toolkit* (*пакет поддерживает довольно широкий спектр многопроцессорных систем, кластеров на базе CCS и grid-сетей*); планировщик для оптимизации решения конкретных задач, автоматическое восстановление после сбоев и перезапуск остановленных процессов, средства *распараллеливания* встроенного языка пакета, тестирование и эмуляция *параллельных* приложений и др. При этом, отдельно разработчики поставляют интегрированную систему *Mathematica Personal Grid Edition*, сочетающую в себе систему *Mathematica* и пакет *Parallel Computing Toolkit*;

– пакет поддерживает встроенный процедурно-функциональный язык, обеспечивая пользователя довольно эффективным средством программирования своих задач.

При этом, *Mathematica* обладает даже большей функциональной *избыточностью*, чем *Maple*. Средства поддержки работы со звуком и простейшими *клеточными* автоматами в *Mathematica*, *Socket* в *Maple* и ряд других вполне можно отнести к «архитектурным» излишествам пакетов. Между тем, пакет *Mathematica* обеспечивает высокую скорость численных вычислений. А вот к недостаткам *Mathematica* довольно часто относится несколько необычный язык программирования, что на наш взгляд не совсем верно. И именно в этом отношении настоящая книга попытается прояснить ситуацию. Также отметим, что оба пакета не являются универсальными с общепринятой точки зрения программирования, давая возможность пользователю создания средств, выполнимых только в среде самого пакета (*т.е. в полной мере невозможно создание exe- и com-файлов*), что существенно ограничивает переносимость созданных таким образом средств.

Оба пакета имеют массу весьма существенных как ошибок, так и недоработок, немало таких моментов отражено на форумах по данным средствам. При этом, имеющиеся у них ошибки, как правило, не пересекаются, т.е. некоторое вычисление в среде пакета *Mathematica* возвращает корректный результат, тогда как эквивалентное вычисление в среде пакета *Maple* возвращает некорректный результат либо не вычисляется вовсе,

и наоборот. С другой стороны, одна и та же задача не решаема ни одним из пакетов, как это иллюстрирует следующий достаточно простой фрагмент, а именно:

```
> 0^0; ⇒ 1
In[521]:= 0^0
Power::indet: Indeterminate expression 0^0 encountered. >>
Out[521]= Indeterminate
> evalf(int(cos(x)*sin(x)/x^2, x=10^(-128) .. 10000)); ⇒ 294.4605291
In[522]:= NIntegrate[Cos[x]*Sin[x]/x^2, {x, 10^(-128), 10000}]
NIntegrate::ncvb: NIntegrate failed to converge to prescribed accuracy after 9 ... .>>
Out[522]= 3.23881
In[611]:= Integrate[1 + 1/(x^2 + 1/(x^2 + 1/x^2)), x^2]
Integrate::ilim: Invalid integration variable or limit(s) in x^2. >>
Out[611]= Integrate[1 + 1/(x^2 + 1/(1/x^2 + x^2)), x^2]
In[612]:= Int[1 + 1/(x^2 + 1/(x^2 + 1/x^2)), x^2]
Out[612]= 1/4 (4 x^2 + 2 Log[x^2] + Log[2 + x^4])
> int(1 + 1/(x^2 + 1/(x^2 + 1/x^2)), x^2);
Error, (in int) wrong number (or type) of arguments
> inttt(1 + 1/(x^2 + 1/(x^2 + 1/x^2)), x^2); ⇒ x^2 + 1/2*ln(x^2) + 1/4*ln(x^4 + 2)
```

В частности, последние два примера фрагмента наглядно иллюстрируют *идентичный* для обоих пакетов недостаток, а именно невозможность интегрирования выражения по переменной *интегрирования*, в качестве которой выступает не простая переменная, а подвыражение интегранда. Для устранения данного недостатка в обоих пакетах мы создали довольно простые процедуры **Df** и **inttt** соответственно, успешно решающие в достаточно широких пределах задачи данного типа. Не лишены серьезных ошибок и неработок и вновь включаемые в пакеты средства. В частности, введенные в версию *7 Mathematica* средства *распараллеливания* в полной мере не выполняют свои функции, о чем свидетельствует тот факт, что если аргумент функции *ParallelTable* определяет довольно сложное выражение, то многоядерность не используется. Подобное можно в полной мере отнести и к средства *распараллеливания*, включенным в *Maple 15*. Еще немало недостатков обоих систем можно было бы отметить, которые с ростом версий пакетов только количественно растут, вызывая растущее недовольство пользователей обоих систем, что весьма четко прослеживается по целому ряду тем и сообщений на форумах по обоим пакетам. Однако, не взирая на довольно большое число ошибок в обоих пакетах, в целом ряде случаев *недопустимых* для средств подобного класса, они на сегодня *реально* являются жестко конкурирующими лидерами среди средств этого класса, что, возможно, в значительной мере определяет и многие из их недостатков.

По обоим пакетам можно было бы еще немало представить их свойств, однако не это входит в задачу настоящей книги. Все эти вопросы можно найти в соответствующих изданиях, на сайтах разработчиков и в интернете. Отметим только, каждая из данных двух систем имеет свои особенности, однако по большому счету обе эти лидирующие системы практически равноценны. Постоянно конкурируя друг с другом, оба пакета взаимно развиваются и совершенствуются. Например, появившиеся в *Mathematica 7*

средства *распараллеливания*, были подхвачены и пакетом *Maple 15*. В то же самое время данная конкуренция наряду с положительными моментами имеет и свои негативные стороны, отмеченные выше. В целом, с полным основанием оба пакета можно считать лидерами среди класса универсальных *CAS*.

В заключение настоящей преамбулы весьма кратко изложим наше личное мнение по сравнительной оценке пакетов *Maple* и *Mathematica*. Оба данные пакета несомненно являются на сегодня лидерами среди *CAS*, однако они изобилуют многочисленными ошибками (*в целом ряде случаев недопустимыми для систем подобного рода*), устранению которых разработчиками как *MapleSoft*, так и *Wolfram Research* уделяется сравнительно небольшое внимание. Вероятно, исходя из *коммерческих* соображений, разработчики часто весьма необоснованно выпускают новые релизы, сохраняющие старые ошибки и привнося в ряде случаев как новые ошибки, так и различного рода «*архитектурные*» излишества. Данный вопрос неоднократно поднимался как в наших изданиях, так и непосредственно перед разработчиками. Однако, если разработчики *Maple* в режиме открытого диалога с пользователями в какой-то мере пытаются решить эту проблему, то *Wolfram Research* весьма болезненно воспринимает любую (*совершенно обоснованную в подавляющем большинстве случаев*) критику в свой адрес. Между тем, *Wolfram Research*, с другой стороны, ведет весьма агрессивную маркетинговую политику, совершенно не адекватную качеству ее продукции. Именно это, прежде всего, объясняет ее *временные* количественные преимущества в среде пользователей *CAS*, которые с определенного момента вполне могут уменьшиться. Сравнивая отклики пользователей этих пакетов наряду с нашим опытом работы с ними, можно вполне однозначно констатировать, что вторые при использовании пакета имеют несколько больше проблем.

Из нашего опыта довольно глубокого использования и апробирования обоих пакетов отметим, что *Maple* – существенно более дружелюбная и открытая система, которая в качестве программной среды предоставляет достаточно развитый встроенный *Pascal*-подобный *императивный* язык процедурного типа, что весьма существенно упрощает освоение пакета пользователю, имеющему опыт современного программирования в среде одного из процедурных языков. Тогда как *Mathematica* имеет в определенной мере «*архаичный*» (*точнее, скорее непривычный*) и не столь изящный язык, в целом ряде отношений, отличный от популярных языков программирования. Наконец, *Maple* по ряду показателей имеет более развитые средства (*например, для решения диффуравнений в частных производных, предоставления пользователю используемого алгоритма для решения задачи, настройки графического интерфейса пользователя на конкретные приложения и др.*), а также довольно широкий спектр бесплатных приложений во многих областях. Оба эти пакета не являются универсальными с точки зрения систем программирования, не давая возможности пользователю создавать средства, выполнимые *вне* среде самого пакета (*т.е. в полной мере их программная среда не позволяет создавать {exe | com}-файлы с программными средствами, созданными в ней*), что довольно существенно ограничивает мобильность созданных таким образом средств. Однако, здесь необходимо отметить, что далеко не все так *однозначно*, прежде всего, относительно встроенного языка обоих пакетов. Именно этому аспекту применения пакетов *Maple* и *Mathematica* в качестве программной среды разработки и посвящается настоящая книга.

Оба пакета на сегодня являются многофункциональными средствами, включающими большое число возможностей для решения широкого круга задач. Естественно, дать *полный* анализ данным средствам в отведенных книгой рамках просто невозможно. И здесь акцентируется внимание на одном их аспекте – *модульном программировании*, поддерживаемом обоими пакетами. Этот аспект имеет особую значимость не только для решения прикладных задач, но, прежде всего, он достаточно важен при создании средств, расширяющих часто используемые средства пакета и/или устраняющих их недостатки, либо дополняющих пакет новыми средствами. Именно в этом контексте и представляют интерес *возможности* пакетов по созданию такого типа процедур или функций. Итак, программирование в среде обоих пакетов – многоаспектная тема и в ней мы акцентируем внимание только на вопросах разработки процедур/функций, которые составляют *основные* программные объекты для наиболее частоиспользуемых средств пользователя, так и для средств, расширяющих и улучшающих стандартные средства пакетов, в программной среде самих пакетов, т.е. реализованные средствами встроенных языков пакетов. Именно в данном контексте и можно в довольно полной мере оценить программные среды обоих пакетов, не взирая на *субъективные* моменты, прежде всего, пользовательские предпочтения и привычки. Естественно, эти моменты играют довольно существенную роль для пользователя, имеющего *определенный* опыт работы с языками процедурного типа, тогда как для начинающего это не столь остро.

Книга состоит из **10** глав и затрагивает довольно обширный материал по *программной* среде пакетов *Maple* и *Mathematica* в контексте *сравнения* их возможностей *модульного* программирования. Между тем, ограниченный объем и сроки подготовки настоящей книги не позволили нам изложить рассматриваемые вопросы с одинаковой степенью полноты. Поэтому, порой, имеет место освещение лишь сути дела без обстоятельного обсуждения отдельных нюансов и рассмотрения смежных (*часто интересных и весьма важных самих по себе*) вопросов. Основные положения данной книги сопровождаются соответствующими иллюстративными примерами, проработку которых мы считаем необходимой для лучшего усвоения предлагаемого материала. Более того, примеры, приведенные в книге, могут быть использованы в качестве довольно полезных средств при разработке собственных приложений в среде пакетов *Maple* и/или *Mathematica*. Для удобства их использования именно в этом качестве предоставляется возможность загрузки содержащих их библиотеки для *Maple* [45] и пакета для *Mathematica* [90]. В целом, содержание глав книги может быть охарактеризовано следующим образом.

Система компьютерной алгебры (Computer Algebra System – CAS) представляет собой тип программного средства, предназначенного для манипулирования математическими формулами. Основной задачей *CAS* является автоматизация достаточно громоздких и сложных символьных преобразований. При этом, основное различие между *CAS* и традиционными компьютерными системами состоит в том, что первая имеет дело, в основном, с уравнениями в символьной, а не в числовой форме. И если как цели, так и возможности этих систем изменяются в весьма широком диапазоне, все же основная их цель остается неизменной, а именно: обработка символьных выражений. И *первая* глава книги представляет краткий экскурс в историю систем компьютерной алгебры. Тогда как *вторая* глава на содержательном уровне представляет общие сведения по

языкам программирования пакетов *Maple* и *Mathematica*.

В 3-й главе в сравнительном плане рассматриваются базовые управляющие структуры пакетов *Mathematica* и *Maple* в разрезе управляющих структур ветвления, циклических и специальных циклических. Завершается глава 3 оценкой механизмов управляющих структур пакетов *Maple* и *Mathematica*. Управляющие структуры данного типа играют одну из важнейших ролей в программировании различного рода алгоритмов, посему они играют далеко не последнюю роль и при сравнении программных сред пакетов. Концепция типа данных в языках программирования высокого уровня появилась как совершенно естественное отражение факта, что обрабатываемые программой данные и выражения могут иметь различные множества допустимых значений, сохраняться в оперативной памяти компьютера различным способом, обрабатываться различными командами процессора и т.д. Эта концепция является одной из основополагающих в современном программировании. Поэтому, учитывая важность типизации языковых объектов, следует этот аспект рассмотреть и в контексте сравнения рассматриваемых пакетов *Maple* и *Mathematica*. И именно с точки зрения развитости поддерживаемых ими механизмов типизации объектов оба пакета рассматриваются в четвертой главе, как наиболее развитые и популярные на сегодня универсальные CAS.

В процессе выполнения достаточно сложных алгоритмов возникают различного рода особые и ошибочные ситуации, которые для повышения робастности алгоритмов мы должны программно обрабатывать. От успешного решения данной задачи с учетом того, насколько мы сумели предусмотреть программную обработку всех возможных ситуаций, могущих возникать в процессе работы алгоритма, в значительной степени зависит его надежность, устойчивость, эффективность и конкурентоспособность. И в первую очередь это относится к средствам широкого (массового) использования, где робастность играет весьма существенную роль. Именно в контексте сравнения обоих пакетов в пятой главе рассматриваются средства, предоставляемые пакетами *Maple* и *Mathematica* для организации обработки особых и ошибочных ситуаций, которые в значительной степени можно отнести к вопросам робастности программных средств.

Так как основной сравнительный анализ обоих пакетов лежит в области процедурного программирования, вопросы поддержки данного типа программирования пакетами представляют первостепенный интерес. Именно поэтому шестая глава данной книги посвящена вопросам организации механизма процедур и модулей в пакетах *Maple* и *Mathematica*. Рассматриваются такие вопросы, как: определения процедур и их типы, формальные и фактические аргументы, локальные и глобальные переменные наряду с механизмами возврата процедурой результатов ее вызова и ряд других. Для пакета *Maple* рассматриваются также расширенные средства языка для работы с процедурами и организация программных и пакетных модулей, а также создание пользовательских пакетных модулей и сохранение процедур и программных модулей в файлах. Глава рассматривает также целый ряд примеров оформления процедур и функций в среде обоих пакетов, а также элементы функционального программирования *Mathematica*.

Подобно пакету *Maple*, *Mathematica* поддерживает достаточно развитые средства для расширения своих функциональных возможностей. Действительно, для очень многих видов вычислений стандартный набор средств, поставляемых с *Mathematica*, является

вполне достаточным. Однако, приложения в целом ряде различных областей вполне могут выявить недостаточность стандартных средств, требуя создания новых средств для своих нужд. В этом случае *Mathematica* предоставляет возможность написания на ее языке специального пакета, содержащего определения требуемых функций. Такие пакеты написаны на *Math*-языке и содержат наборы определений функций, которые не входят в набор стандартных средств *Mathematica*, либо программы решения задач пользователя; эти пакеты ориентированы на конкретные прикладные области. Глава 7 рассматривает основные вопросы организации пакетов, их создания, сохранения в файлах специального формата и использования в программной среде *Mathematica*.

Языки программирования *Maple* и *Mathematica* ориентированы, в первую очередь, на символьные вычисления, располагая относительно ограниченными возможностями в работе с данными, находящимися во *внешней* памяти ПК. Вместе с тем, ориентируясь, прежде всего, на решение задач в *символьном* виде, такие языки предоставляют набор средств для доступа к файлам данных, который может удовлетворить широкий круг пользователей пакета, существенно не использующих работы с файлами. Между тем, данный вопрос весьма актуален для задач, довольно активно использующих работу с файлами различных форматов и организации. Поэтому в *восьмой* главе рассмотрены средства обоих пакетов для работы как с внутренними, так и с внешними файлами. В завершении главы проводится сравнение средств доступа пакетов к файлам данных.

Учитывая важность не только разработки программных средств, но и их сохранения в файлах специального формата для возможности последующего их использования, *девятая* глава рассматривает вопросы организации пользовательского программного обеспечения в среде *Maple* и *Mathematica*, специальные способы создания библиотек пользователя и представляет оценку эффективности пакетов *Mathematica* и *Maple* по организации пользовательского программного обеспечения в их среде.

Наконец, *десятая* глава представляет нашу, до определенной степени субъективную, точку зрения на предпочтительность каждого из рассматриваемых пакетов в той или иной области приложений. Не зирая на определенную субъективность, заключение по пакетам лежит в русле соображений достаточно широкого круга пользователей. В приложенном списке литературы нашли свое отражение работы наиболее известных отечественных авторов по системам компьютерной алгебры *Maple* и *Mathematica*.

В процессе подготовки наших изданий по пакетам *Maple* [8–44] и *Mathematica* [1–3,6, 7,36] наряду с библиотекой для *Maple* [45] и пакетом для *Mathematica* [90] нами были использованы лицензионные версии 4 – 11 и 2.2, 7 (*License # L4704–9074*), 8 для пакетов соответственно, предоставленные разработчиками MapleSoft Inc. и Wolfram Research Inc.

В заключение авторы книги за спонсорское содействие с большой признательностью выражают благодарность руководителю Эстонского отделения Hydroscaud AS K.L. Solba, что позволило не только в кратчайшие сроки выполнить работу над книгой, но и весьма существенно способствовало ее реальному выходу в свет.

*Maple V, Maple 6, Maple 7, Maple 8, Maple 10, Maple 11 – торговые марки MapleSoft Inc.
Mathematica 2, Mathematica 7, Mathematica 8 – торговые марки Wolfram Research Inc.*

Глава 1. Краткий экскурс в историю компьютерной алгебры

Система компьютерной алгебры (*Computer Algebra System – CAS*) представляет собой тип программного средства, предназначенного для манипулирования математическими формулами. Основная цель *CAS* состоит в автоматизации зачастую утомительных и в целом ряде случаев трудных алгебраических преобразований. Между тем, основное различие между *CAS* и традиционным вычислителем состоит в том, что первая имеет дело в основном с уравнениями в символьной, а не в числовой форме. И если целевые назначения и возможности этих систем изменяются в весьма широком диапазоне, все же основная их цель остается одинаковой, а именно: манипулирование символьными выражениями. При этом, *CAS* обеспечивают того либо иного уровня развитости язык программирования, позволяя пользователю создавать собственные процедуры. Более того, *CAS* не только существенно повлияли на методику преподавания математики и других математически-ориентированных дисциплин, но и обеспечили математиков и специалистов из ряда других областей гибким инструментальным средством. *CAS*, например, могут использоваться для факторизации полиномов, решения различных систем уравнений, включая дифференциальные, нахождения пределов, символьного дифференцирования и интегрирования выражений и так далее. Итак, *CAS* – система обеспечения символьной (*алгебраической*) математики, а ядро ее функциональности – манипулирование математическими выражениями в символьном виде.

История *CAS* восходит к началу 1970-х прошлого века и является непосредственным следствием прогресса исследований по *искусственному* интеллекту, однако на сегодня оба направления в значительной степени являются самостоятельными. На начальном этапе *CAS* были довольно простыми, обладая очень ограниченными возможностями, по символьному решению относительно узких классов задач. И лишь с таких *CAS* как *Reduce*, *Derive* и *Macsyma* начинается серьезный этап становления систем этого типа; более того, данные системы и сегодня все еще коммерчески доступны, тогда как *CAS Macsyma (Maxima)* все еще поддерживается и доступна бесплатно. Текущие лидеры в данной области – *Maple*, *Mathematica*, и в определенной мере также *MuPAD*. Данные системы, как правило, используются математиками, исследователями и инженерами, активно использующими математические методы, наряду с использованием данных систем в совершенствовании преподавания математически-ориентированных курсов. Наконец, целый ряд *CAS* сосредоточивается на определенной прикладной области и, как правило, они разрабатываются и поддерживаются в академической среде, являясь свободно распространяемыми. Такие *CAS* могут быть достаточно неэффективны для числовых вычислений по сравнению с системами, ориентированными на численные вычисления. Далее мы несколько детализируем историю становления *CAS*.

Корни возникновения *CAS*, как отдельного направления программного обеспечения, восходят к 60-м годам прошлого века и лежат в *двух* совершенно различных областях: теоретической физике и искусственном интеллекте. Действительно, первой работой в этом направлении можно считать разработку в 1963 г. Мартином Вельтманом (*Martin Veltman*) программы *Schoonschip*, ориентированной на символьное решение проблем и, прежде всего, в области физики высоких энергий. В свою очередь в *MITRE* в рамках

исследований по искусственному интеллекту на базе языка программирования *Lisp* в 1964 г. Карлом Энгельманом (*Carl Engelman*) была создана система *MATHLAB*, которая в дальнейшем использовалась на компьютерах *PDP-6* и *PDP-10*. И даже до настоящего времени ее можно использовать на *SIMH*-эмуляторах *PDP-10*. При этом, не следует ее отождествлять с одноименным пакетом *MATHLAB*, ориентированном на численные вычисления и созданном лишь в 1979 г. в университете Нью Мехико (*New Mexico*). Из первых же наиболее известных *CAS* общего назначения необходимо особо отметить систему *REDUCE*, во многом ориентированную на физическую область приложений.

Создание *REDUCE* относится к началу 60-х прошлого века и было инициировано Э.С. Хиерном (*A. Hearn*). С тех пор много специалистов из многих стран внесли свой вклад в ее развитие под его руководством. *CAS REDUCE* целиком написана на собственном диалекте языка *Lisp*, называемом стандартным *Lisp*; чей синтаксис алголо-подобен и назван *RLisp*. Именно он использован в качестве основы для пользовательского языка программирования в *CAS REDUCE*. Реализации *REDUCE* доступны на большинстве релизов *Unix*, *Linux*, *Windows Microsoft* или *Apple Macintosh*, используя стандартный мобильный *Lisp*. Уже первые версии *REDUCE* ориентированы на решение довольно несложных задач скалярной и векторно-матричной алгебры, при этом одновременно поддерживая произвольную точность числовых вычислений и базовые графические функции. Она допускает интерактивный режим для простых вычислений, но также предоставляет пользователю язык программирования, синтаксис которого довольно подобен другим современным языкам программирования. В целом, *REDUCE* можно рассматривать в качестве интерактивной программной системы, предназначенной, в первую очередь, для алгебраических вычислений общего характера, интересную для математиков, физиков, а также ученых и специалистов из других областей. В качестве ее основных возможностей следует отметить следующие, а именно:

- разложение и упорядочивание многочленов и рациональных функций;
- подстановки и сравнения шаблонов в весьма широком разнообразии форм;
- автоматическое и управляемое пользователем упрощение выражений;
- вычисления с символьными матрицами;
- поддержка произвольной точности арифметики целых и действительных чисел;
- возможности определения новых функций и расширения программного синтаксиса;
- символьные дифференцирование и интегрирование выражений;
- факторизация полиномов;
- решение разнообразных алгебраических уравнений;
- вывод выражений в разнообразных форматах;
- генерация оптимизированных числовых программ на основе символьного входа;
- вычисления с широким набором специальных функций и др.

Между тем, эта система подобно другим современным *CAS* весьма часто используется как алгебраический калькулятор для задач, достаточно легко выполнимых вручную. Однако, главная цель *REDUCE* состоит в решении задач, не выполнимых или трудно выполнимых вручную. Многие из подобных задач требуют достаточно существенных временных затрат (часы, а то и дни) на самых мощных компьютерах. Обеспечивают эту цель следующие важные характеристики *REDUCE* такие, как:

Стабильность исходного кода. Различные версии *REDUCE* использовались в течение более сорока лет. За этот период большим сообществом пользователей, прежде всего, из академической среды (*главным образом физиков и математиков*) была произведена большая работа по расширению и совершенствованию *CAS*, что обеспечило развитие ее в мощную систему, основные компоненты которой весьма надежны, устойчивы, а также достаточно эффективны во временном отношении.

Обширная пользовательская среда. Поддерживается как собственно возможностями *CAS*, так и большим числом специальных пакетов, ориентированных на различные, в первую очередь, *физические* приложения. Многие из них созданы многочисленными пользователями системы из различных областей вычислительной науки и техники.

Полная доступность исходного кода. Начиная с *первых* версий системы, было можно получить исходный код *REDUCE*, включая ее ядро. По этой причине *CAS REDUCE* не только прекрасный образовательный ресурс, но и хорошая основа для экспериментов в области компьютерной алгебры. Именно данные предпосылки и позволили многим пользователям эффективно изменять исходный код для своих собственных нужд. Так, в нашей книге [5] представлены некоторые расширения данной *CAS*. Тут же отметим, что наши книги [4-8] явились одними из самых первых отечественных книг по таким математическим пакетам как *MatCAD*, *REDUCE*, *Mathematica* и *Maple*; причем книги [5-6] вообще были *первыми* массовыми книгами по *MatCAD*, *REDUCE* и *Mathematica*.

Гибкое обновление. Важнейшим преимуществом открытости исходного кода является то, что относительно легко модифицировать *CAS* как с целью устранения возможных ошибок, так и ее развития с целью расширения применимости существующего кода к новым прикладным областям. Электронная почта и Интернет дают пользователям возможность оперативно получать такие модернизации, не дожидаясь формального выпуска очередного релиза системы. Именно данная открытость весьма импонирует многим серьезным специалистам и разработчикам в области *CAS* и выгодно отличает *REDUCE* от таких *CAS*, как *Maple* и в значительно большей степени, как *Mathematica*. Именно закрытость *Maple*, и в большей степени *Mathematica* не позволяют устранять силами большого пользовательского сообщества имеющиеся многочисленные огрехи в обоих пакетах, весьма часто недопустимые для такого уровня средств, тем более так агрессивно себя рекламирующих (*в первую очередь это относится к Mathematica*).

Современные алгоритмы. Еще одно важное преимущество открытых систем состоит в том, что возможно совместное участие разработчиков и пользователей по улучшению системы и используемых ею алгоритмов различного назначения. В результате, более просто поддерживать систему в актуальном состоянии при *лучших*, существующих на текущий момент алгоритмах, используемых вскоре после их разработки. В этой связи *REDUCE* отличается в значительно лучшую сторону от пакетов *Maple* и *Mathematica*.

Алгебраический акцент. *CAS REDUCE* позиционируется в качестве *специальной* части научной вычислительной среды, а не все охватывающей вычислительной средой, как это, например, пытается сделать *Mathematica*. Данная концепция предполагает, что *CAS REDUCE* ориентируется на эффективное символьное решение пользовательских задач, тогда как пользователь по мере своих нужд может дополнительно использовать и другие современные системы, специализирующиеся на численных, графических и

других вычислениях, а не зависеть лишь от одной системы, да и вряд ли одна даже и очень большая (*зачастую громоздкая и требующая больших ресурсов*) система обладает в равной степени эффективными алгоритмами. Немало «архитектурных» излишеств в *CAS Maple* и *Mathematica* с большой натяжкой можно отнести к *CAS*-проблематике. И здесь вполне уместно напомнить изречение К. Прутковы – «не объять необъятное».

Мобильность. Успешность *REDUCE* объясняется также доступностью на новых либо специальных компьютерах вскоре после их появления. В настоящее время *REDUCE* в полной мере доступен по существу на всех вычислительных платформах.

Единообразие. Даже при том условии, что *CAS REDUCE* поддерживается различными версиями языка *Lisp* на различных вычислительных платформах, большое внимание было уделено такому аспекту, как *единообразие* выполнения системы безотносительно различных ее реализаций и релизов.

Гибкие предложения. Для обеспечения различных требований пользователя *REDUCE* на сегодня доступна в широком диапазоне различных конфигураций.

Стоимость. Первая цена полных версий *REDUCE* для пользователя была умеренной и существенно не изменялась от платформы к платформе. Кроме того, поскольку все версии получены из одного и того же источника, они совместимы безотносительно от используемой платформы (*от ПК до суперкомпьютера Cray*). В настоящее время версия *Open Source CAS REDUCE* зарегистрирована на сайте *www.SourceForge.net*, с которого ее можно загружать для бесплатного использования и дальнейшего развития. Между тем, ранее стоимость *CAS REDUCE* составляла \$ 695.

Повышенное внимание, уделенное *REDUCE*, обуславливается тем обстоятельством, что она занимает выдающееся место в *истории* систем компьютерных алгебр. Она оказала большое влияние на дальнейшее развитие исследований в этом направлении так, что основные ее черты четко прослеживаются в последующих системах. И до настоящего времени *REDUCE* достаточно широко используется англоязычными пользователями, прежде всего из-за ее бесплатности и возможностей для такого доступного средства. В результате работы со многими достаточно серьезными *CAS (Axiom, Maxima, Maple и Mathematica)* именно *REDUCE* выделяется своей полезностью, надежностью наряду с универсальностью как для исследовательских целей, так и для обучения. Она весьма хорошо документирована, имеет открытый код, а ее программная среда достаточно удобна как для программирования задач пользователя, так и для расширения *CAS*. В среде отечественных пользователей, имеющих дело с исследованиями в физических областях, она до сих пор пользуется значительной популярностью. И действительно, в виду истории своего появления она ориентирована на физику и ее приложения, и в 80-е годы прошлого века система получила весьма широкое распространение в СССР. По нашему мнению система *REDUCE* со временем была значительно подзабыта, но даже относительно современных популярных *CAS* она все еще продолжает обладать рядом преимуществ. Так, в задачах, имеющих дело с *внешними* дифференциальными системами (*дифференциальные формы, производные Ли и т.д.*) пакеты *EXCALC, CRACK* и *EDS* из *REDUCE* превосходят соответствующие аналоги в *CAS Maple* и *Mathematica*. Несколько детальнее с системой можно ознакомиться, например, в книгах [5,63-72], а более детальную информацию можно получить в интернет по фразе «*CAS Reduce*».

Довольно важное место среди систем *компьютерной* алгебры занимает система *Axiom*. В отличие от остальных систем, представляющих собой пакеты программ, общение с которыми осуществляется на некотором входном языке, система *Axiom*, развившаяся из *SCRATCHPAD-II*, имеет дело с более привычными для математиков объектами. В частности, в ней ключевым понятием является понятие *категории*; по этой причине в ней можно рассматривать, например, категории *множеств*, *полугрупп*, *левых модулей*, *дифференциальных колец* и т.д. Система имеет высокий уровень универсальности, но требует для своей реализации мощных компьютеров, и раньше распространялась на коммерческой основе, поэтому использовалась лишь в ограниченном числе довольно мощных университетских и научных центров. К началу 21 века продажи резко упали и проект стал коммерчески нерентабельным, посему система *Axiom* была обеспечена лицензией *GPL*. Эта система с полным основанием может рассматриваться в качестве лидера среди *CAS* на европейском уровне подобных средств.

Система *SCRATCHPAD*, лежащая у истоков *Axiom*, была разработана для обеспечения диалогового символьного средства, ориентированного на пользователя-математика. Система обеспечивает пользователя языком общения, учитывающим лаконичность и стиль математической нотации наряду с предоставлением возможности расширения языка новыми понятиями. Более того, развитая библиотека системы включает также символьные средства, поддерживаемые такими системами как *REDUCE* и *MATHLAB*.

Axiom для математика имеет целый ряд весьма привлекательных черт. Язык системы обладает весьма мощной и простой системой типов, ею поддерживаются *абстрактные* операции из области абстрактной алгебры и теории чисел. Более того, система *Axiom* имеет весьма *эффективный* алгоритм вычисления интегралов. Так, большинство *CAS*, получив на вход слишком сложный для них *интеграл*, или ничего не возвращают, или возвращают его *невычисленным*. Тогда как *Axiom* возвращает интеграл *невычисленным* лишь в случае, когда может доказать, что он не берется в элементарных функциях. К тому же, подобно *Maxima*, *Axiom* обладает достаточно высоким уровнем абстракции, предоставляемым пользователю. Обе системы поддерживают т.н. *Rule-преобразования*, позволяя выполнять преобразования не только над математическими выражениями, но и над любыми формульными выражениями.

Из некоторых других популярных в данном направлении *CAS* можно отметить такие, как *muMATH*, *Derive* (базируется на *CAS muMATH*) и *Macsyma*; при этом, популярная бесплатная версия *Macsyma*, именуемая *Maxima*, довольно активно поддерживается и до настоящего времени. Вкратце остановимся и на этих популярных системах.

В 1980 г. *Soft Warehouse* - один из пионеров в разработке символьного математического программного обеспечения выпустил первый официальный релиз системы *muMATH* (последний релиз *MuMATH-83*), которая предназначалась для символьного решения на небольших компьютерах и простых калькуляторах относительно несложных задач. В дальнейшем *muMATH* была заменена более развитой системой *Derive*, написанной на языке программирования *muLisp* - одном из наиболее удачных диалектов языка *Lisp* [72-74]. Существует несколько реализаций: *muLisp81*, *muLisp83*, *muLisp85*, *muLisp87*. В то же время система *Derive* не получила сколько-нибудь широкого признания в среде специалистов, но относительно широко применяется в учебном процессе в школах.

В отличие от системы *REDUCE*, *Macysma* разрабатывалась с самого начала в качестве коммерческого продукта в МПТ в 1968 – 1982 годах как часть известного проекта *MAC*. В системе более тщательно проработаны алгоритмические вопросы, эффективность ее существенно выше; в то же время, по причинам реализации на малом количестве вычислительных платформ и коммерческому распространению она довольно долго имела ограниченное распространение. Система *Maxima* выросла из *CAS Macysma* и подобно большинству систем компьютерной алгебры реализована на языке *Lisp*. Она включает в себя *Algol*-подобный язык программирования и лучше всего подходит для обучения концепции программирования в системах компьютерной алгебры. *Maxima* доступна на бесплатной основе и имеет открытый исходный код. В настоящее время последней версией *Maxima* является 5.22 (2010). Несмотря на это, распространенность системы на отечественном рынке весьма ограничена и причина этого, на наш взгляд, состоит в том, что до появления *Maple* и *Mathematica* отечественные пользователи в массе своей применяли *REDUCE*, а с появлением двух последних предпочтение было отдано именно им, правда, с различной степенью популярности.

Тут же вполне уместно отметить и отечественные работы по системам компьютерной алгебры, осуществленные в Институте кибернетики АН УССР под руководством акад. В.М. Глушкова. В рамках этих работ в 1968 г. был реализован язык программирования *Аналитик*, являющийся развитием языка *АЛМИР-65* с сохранением совместимости с ним. Отличительными чертами языка *Аналитик* являются абстрактные типы данных, вычисления в произвольных алгебрах, аналитические преобразования; он реализован на компьютерах *МИР-2*. Затем была разработана версия *Аналитик-74*, реализованная на компьютерах *МИР-3*. В настоящий момент язык *Аналитик* входит в состав системы компьютерной алгебры *АНАЛИТИК-2010*, совместно разрабатываемой и Институтом проблем математических машин и систем НАН Украины, и Полтавским национальным техническим университетом.

Отдельно здесь уместно также упомянуть известный математический пакет *MathCAD* [4,78-80], довольно широко используемый отечественными учеными, специалистами из различных областей и студентами естественно-научного профиля. Пакет ранних версий нельзя было относить к классу *CAS*, но, начиная уже с версии 3.0, этот пакет в определенной мере наследует черты современных *CAS*, чему успешно способствовал альянс *MathSoft Inc.* – разработчика пакета – с *Waterloo Maple Software* – разработчиком пакета *Maple*. Результатом явилась имплантация в пакет *MathCAD* ядра *Maple V* при довольно существенных ограничениях возможностей последнего. С другой стороны, появилась возможность использовать функции *Maple V* с вызовом их из *MathCAD*. По этой причине говорить о *MathCAD* как о полноценной *CAS* пока преждевременно. В этой связи хотелось бы также отметить, что разработчики *MathCAD 14* отказались от использования ядра *Maple V* и для обеспечения поддержки символьных вычислений обратились к более дешевому, но менее мощному пакету *MuPAD* фирмы *Mathworks*.

На сегодня самыми популярными коммерческими *CAS* общего назначения являются *Mathematica* и *Maple*, достаточно широко используемые в различных теоретических и прикладных областях [1-3,6-62]. Детальнее эти системы на сравнительном уровне рассматриваются в настоящей книге. При этом, бесплатно доступной альтернативой

для указанных двух CAS выступает система SAGE (как внешний интерфейс к некоторым бесплатным CAS). Эта система уникальна тем, что может функционировать в качестве интегрирующей среды для целого ряда других компьютерных алгебраических систем, позволяя пользователю усиливать индивидуальные возможности различных пакетов программ. Система SAGE реализована на языке программирования Python, используя интерактивную Python-оболочку в качестве пользовательского интерфейса. Первый релиз системы появился в феврале 2005, текущим является релиз 4.6 (октябрь 2010). В качестве основной задачи разработчики CAS SAGE полагают создание эффективной бесплатной с открытым исходным кодом системы, альтернативной к Magma, Matlab, Maple и Mathematica [76,77]. Система SAGE все чаще именуется как sagemath.

Выше рассматривались CAS т.н. *общего* назначения (т.е. универсальные по назначению), тогда как наряду с ними существует немало систем специального назначения и даже CAS для калькуляторов. Специализированные системы отличаются несколько более высокой эффективностью, но область их применения более ограничена. Например, в качестве специальных можно отметить такие, как GAP, CALEY, Magma (алгебра, теория групп, теория чисел), Fermat, CoCoA (полиномиальные, матричные вычисления), MathPiper, Algebrator (ориентированы на задачи обучения), SCHOONSHIP (система для вычислений в физике высоких энергий), Macaulay2 (алгебраическая геометрия, коммутативная алгебра), TRIP (механика небесных тел), SINGULAR (полиномы, кольца, идеалы), VEGA (операции с дискретными математическими структурами), Cadabra (тензорная алгебра, теория поля), Mathomatic (элементарная алгебра, комплексные числа, операции с полиномами), PARI/GP (теория чисел, эллиптические кривые, арифметика с произвольной точностью), GiNaC (б-ка языка C++), Meditor (б-ка символьных вычислений языка Java совместно с математическим редактором) и др., включая немногочисленные отечественные CAS. Особый интерес с точки зрения математика здесь имеет CAS GAP (Groups, Algorithms and Programming). GAP распространяется бесплатно и ориентирована на задачи абстрактной алгебры. На сегодня GAP наиболее мощная из систем такого рода, включая и коммерческие. В ней, например, в качестве типов данных выступают такие математические структуры, как «поле», «кольцо», «группа», а вычисление идеала банаховой алгебры поддерживается стандартной функцией системы. С более детальными обзорами современных систем компьютерной алгебры можно ознакомиться в целом ряде изданий и в интернете.

Наконец, имеется также целый ряд CAS, ориентированных на ресурсы портативных калькуляторов, среди которых можно отметить такие популярные, как ZoomMath300, TI-Nspire CAS, TI-89 Titanium, Voyage 200 (для графических калькуляторов фирмы Texas Instruments), Casio CFX-9970G, Casio ClassPad 330 (для калькуляторов фирмы Casio), iCAS, i41CX+ (системы на основе REDUCE для iPhone и iPad), HP 49 series (для калькуляторов фирмы Hewlett-Packard) и др. Имеются отечественные разработки в этом направлении.

Более детальный статистический сравнительный анализ современных различных как по назначению, так и по поддерживаемым вычислительным платформам CAS можно найти в [81], однако некоторые его аспекты представляются нам довольно спорными. Между тем, анализ в целом представляет общую картину в этой области. Рассмотрим несколько более детально историю развития рассматриваемых здесь пакетов Maple и Mathematica, которая сама по себе достаточно поучительна.

Истоки создания *CAS Maple* восходят к концу 1980, когда исследователи из *Waterloo University* (Канада) вместо приобретения мощного компьютера под систему *Macsyma* решили разработать свою собственную *CAS*, способную функционировать на менее мощных компьютерах. Эта идея получила весьма быстрое развитие и уже в декабре 1980 появилась первая пробная версия системы. В основу подхода к созданию проекта была положена концепция разработки развивающейся и достаточно открытой *CAS*. Получив название *Maple* (в соответствии с кленовым листом – символом Канады), пакет, начиная с 1982, вполне успешно демонстрировался на различных конференциях, что принесло свои результаты – уже к концу 1983 г. более 50 университетов приступили к его использованию, а в 1984 г. разработчики запатентовали и начали распространять свой пакет. В 1988 разработчики пакета основали компанию *Waterloo Maple Inc.*, чьи функции не ограничились распространением пакета, но и проводились работы по его дальнейшему развитию. Между тем, хотя основная доля разработок по пакету *Maple* и принадлежит именно этой фирме, достаточно существенное развитие пакета было проведено и проводится по сей день в целом ряде других исследовательских центров и университетах, например: *Ontario Research Centre for Computer Algebra at the University of Western Ontario*, *Symbolic Computation Laboratory at the University of Waterloo*, и во многих других университетах и научно-исследовательских организациях, а также обширным кругом пользователей пакета. В рамках тематики по Системам компьютерной алгебры в *Baltic Branch of International Academy of Noosphere* (Таллинн, Эстония) проводились работы по апробации и применению пакета *Maple* версий 5–11, результатом чего явилось не только проведение целого ряда курсов и мастер-классов по пакету в разных странах, подготовка и издание в США, Белоруссии, Литве, России и Эстонии ряда монографий и книг по пакету, статей в журналах и докладов на международных конференциях, но и разработка обширной Библиотеки [45] новых программных средств для *Maple*, как расширяющих, дополняющих, так и улучшающих стандартные средства пакета.

Если до версии 4.3 *Maple* для интерфейса с пользователем использовал только режим командной строки, то уже с этой версией для *Macintosh* в 1989 был использован первый графический пользовательский интерфейс, тогда как улучшенная версия интерфейса под *Windows* была представлена в 1990 с *Maple V*. В 1999 пакет *Maple 6* включал в себя часть широко известной библиотеки *NAG* (*The Numerical Algorithms Group Ltd., Oxford, UK*) для задач линейной алгебры и был довольно существенно улучшен для работы с арифметикой произвольной точности. В версию *Maple 9* дополнительно к прежнему, т.н. классическому интерфейсу, который допустим на платформах *Windows* и *UNIX*, и требует меньше памяти, был добавлен более развитый и современный стандартный интерфейс. При этом, в среде *Windows* пользователь имеет возможность выбора типа интерфейса простым переключением. Текущий вид «стандартного» интерфейса был введен в 2003, интерфейс написан на *Java* и частично на встроенном *Maple*-языке, но на первых порах вызвал немало критики из-за весьма низкой реактивности на малой памяти, что требовало от пользователя вновь обращаться к классическому интерфейсу, который разработчиками пакета более не поддерживается.

Несмотря на существенное улучшение интерфейса, в период 1995 – 2005 пакет *Maple* потерял в пользу конкурентов определенную часть своего рынка из-за более слабого

пользовательского интерфейса. По этой причине в версию *Maple 10* был введен *новый «режим документа»* как часть *стандартного* интерфейса. Главная особенность данного режима состоит в том, что математические выражения вводятся в обычном 2-мерном формате, подобно используемому в книгах. Более того, в интерфейс версии *Maple 12* (2008) был добавлен ряд дополнительных возможностей, позаимствованных из пакета *Mathematica*. При этом, *классический* интерфейс написан на *C*, тогда как интерфейсы *стандартный* и с калькулятором написаны на языке *Java*. Более того, пакет допускает создание *интерфейсов*, определяемых пользователем. По-видимому, дополнительные возможности добавлялись (*в том числе*) для более приемлемого использования пакета с пакетом *MATLAB*. Пакет *Maple* располагает процедурным *Pascal*-подобным языком программирования *императивного* типа, поддерживая интерфейсы с такими языками программирования, как *Java*, *Fortran*, *C* и *Visual Basic*, а также с пакетами *MATLAB* и *Excel*. В архитектурном контексте *Maple* базируется на относительно небольшом *ядре*, написанном на языке *C*, которое поддерживает *встроенный Maple-язык*. Большинство функциональных возможностей пакета обеспечивается библиотеками, созданными в различных организациях и *отдельными* пользователями пакета. В частности, большое число процедур для численных вычислений поддерживаются такими библиотеками, как *ATLAS*, *NAG* и *GMP*. Тогда как большинство библиотек собственно самого пакета написано на *Maple-языке* с открытым для просмотра исходным кодом, что довольно важно со многих точек зрения, включая возможность редактирования средств *Maple*, создания пользовательских на их основе, а также использования их в качестве весьма прекрасных примеров для осваивающих программирование в среде *Maple*. При этом, за период 1982 – 2011 было создано 37 версий пакета, из которых *первой* доступной для массового пользователя явилась версия *Maple 3.3* (1985), тогда как на сегодня текущей является версия *Maple 15* (2011). Отметим, что в сентябре 2009 пакет *Maple* совместно с фирмой *MapleSoft Inc.* были приобретены фирмой *Cybernet Systems Ltd.* (Япония).

В свою очередь, *история* создания пакета *Mathematica* восходит к достаточно простой *программе* символьного манипулирования (*SMP*), являющейся довольно простой *CAS*, созданной *C. Cole* и *S. Wolfram* в 1979 и которая первоначально развивалась большой группой исследователей под руководством *S. Wolfram*. Программа *SMP* возникла под влиянием ранних *CAS Schoonschip* и *Macsyma*; так, *S. Wolfram* исследовал исходный код первой и являлся пользователем второй системы. Коммерческое *распространение* данной программы было начато в 1981 *Computer Mathematics Corp.* (США), вошедшей в последующем в состав *Inference Corp.*, в рамках которой и проводились последующие развитие и распространение программы *SMP* в период с 1983 по 1988. Есть основание рассматривать программу *SMP* в качестве точки отсчета в разработке *Mathematica*.

Начало интенсивного развития пакета *Mathematica* началось в рамках основанной в 1987 *Wolfram Research Inc.*, первая коммерческая версия которого увидела свет в 1988. В архитектурном отношении *Mathematica* по большому счету состоит из *двух* основных компонент, а именно – *ядра* и *интерфейса* с пользователем. *Ядро* пакета обеспечивает все вычисления, оно реализовано на *Mathematica-языке* и языке программирования *C*, тогда как *интерфейс*, разработанный *T. Gray*, обеспечивает развитый графический *пользовательский интерфейс (GUI)*, поддерживающий создание и достаточно удобное

редактирование текущего документа (*notebook*), содержащего как программный текст, так и обычный текст, графические объекты и др. При этом, и содержимое документа, и его форматирование можно выполнять как в программном, так и в интерактивном режимах. Интерфейс поддерживает наиболее принятые функции обработки текстов, однако он поддерживает лишь *один* уровень восстановления предыдущего состояния (*undo*) текущего документа, что вряд ли вызывает определенные затруднения. Интерфейс включает средства разработки типа отладчика и автоматической окраски синтаксиса вводимых выражений, что существенно упрощает ввод, позволяя по ходу ввода выражений корректировать их корректность. Пакет *Mathematica* использует по умолчанию стандартный интерфейс, допуская, при этом, другие интерфейсы такие, как *Wolfram Workbench* и *Eclipse* наряду с интерфейсом командной строки, который на компьютерах небольшой мощности является наиболее приемлемым.

С целью повышения производительности пакета версия *Mathematica 5.2 (2005)* была дополнена режимом автоматической мультиобработки для обеспечения вычислений на мульти-ядерных процессорах. Наряду с этим, эта версия содержала специальные библиотеки, оптимизирующие работу процессора, наряду с поддержкой *ClearSpeed* – специальным процессором, обеспечивающим весьма высокую производительность в вычислениях с плавающей точкой. Так, *CSX600* на сегодня является самым быстрым в мире 64-битным процессором с производительностью 25 гигафлоп, что почти в 10 раз быстрее стандартного персонального компьютера. При этом, такой процессор может работать и с традиционными 32-битными системами. С подобной целью в 2002 была создана надстройка *gridMathematica*, обеспечивающая *параллельное* программирование на однородных кластерах и мультипроцессорных системах, тогда как в 2008 в пакете была обеспечена технология *параллельных* вычислений, включая поддержку сеточной технологии типа *Windows HPC Server 2008*, *Microsoft Compute Cluster Server* и *Sun Grid*. С 2010 пакетом обеспечивается поддержка для *CUDA* и *OpenCL GPU*. Более того, версия *Mathematica 8.0* может генерировать C-код, который автоматически компилируется *Intel C++ Compiler* и *Visual Studio 2010*, обеспечивая скорость выполнения *Mathematica*-кода аналогично коду, написанному на языке программирования C.

Протокол *MathLink* обеспечивает интерфейс не только между ядром и GUI, но также с рядом других приложений, расширяя функциональные возможности *Mathematica*. Свободно распространяются средства для обеспечения интерфейса написанных на C приложений с *Mathematica* через *MathLink*. Можно отметить интерфейсы с такими приложениями, как *Visual Basic*, *Python*, *Ms Excel*, *MATLAB*, *SINGULAR*, *MathModelica*, *R*, *OpenOffice.org Calc* и с рядом других. Некоторые из приложений позволяют проверять документы, подготовленные в среде *Mathematica*; так, бесплатная версия *Wolfram CDF Player* служит для выполнения *Mathematica*-программ, сохраненных в CDF-формате, а также для просмотра стандартных *Mathematica*-файлов без их выполнения.

Более того, за период 1988 – 2011 было создано 20 версий пакета, из которых *первой* из доступных для массового пользователя выступает версия *Mathematica 1.0 (1988)*, тогда как текущей на сегодня является версия *Mathematica 8.0.1 (2011)*. Настоящая книга при подготовке использовала, в основном, версии *Mathematica 8.0.0/8.0.1* пакета.

Литература

1. Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. Основы информатики.- Гомель: Изд-во Salcombe Eesti, 1997, 396 с., ISBN 5-14-064254-5.
2. Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. Основы информатики: Учебное пособие.- Москва: Изд-во ФилинЪ, 1998, ISBN 5-89568-068-2.
3. Аладьев В.З., Хунт Ю.Я., Шишаков М.Л. Основы информатики: Учебное пособие. 2-е изд.- Москва: Изд-во ФилинЪ, 1999, 520 с.
4. Аладьев В.З., Гершигорн Н. Вычислительные задачи на персональном компьютере.- Киев: Изд-во Тэхника, 1991, 248 с.
5. Аладьев В.З., Тупало В.Г. Алгебраические вычисления на компьютере.- Москва: Изд-во Минтопэнерго, 1993, 251 с., ISBN 5-942-00456-8.
6. Аладьев В.З., Хунт Ю., Шишаков М.Л. Математика на персональном компьютере.- Гомель: Изд-во ФОРТ, 1996, 498 с.
7. Аладьев В.З., Шишаков М.Л. Введение в среду пакета *Mathematica 2.2*.- Москва: Изд-во ФилинЪ, 1997, ISBN 5-89568-004-6.
8. Аладьев В.З., Ваганов В.А. и др. Введение в среду математического пакета *Maple V*.- Минск: Изд-во IAN Press, 1998, 452 с., ISBN 14-064256-98.
9. Аладьев В.З. др. Программирование в среде математического пакета *Maple V*.- Гомель: TRG & Salcombe Press, 1999, 470 с.
10. Аладьев В., Ваганов В., Хунт Ю., Шишаков М. Рабочее место для математика.- Гомель-Таллинн: International Academy of Noosphere, 1999, 605 с.
11. Аладьев В.З., Богдявичус М.А. Решение математических и физико-технических задач с пакетом *Maple V*.- Вильнюс: Technics Press, 1999, 686 с., ISBN 9986-05-398-6.
12. Аладьев В.З., Шишаков М.Л. АРМ математика.- М.: Изд-во БИНОМ, 2000 + CD.
13. Аладьев В., Богдявичус М. *Maple 6*: Решение математических, статистических и инженерно-физических задач.- М.: Изд-во БИНОМ, 2001, ISBN 5-93308-085-X.
14. Aladjev V.Z., Bogdevicius M.A. *Interactive Maple: Solution of Mathematical, Engineering, Statistical and Physical Problems*.- Tallinn-Vilnius: Academy of Noosphere, 2002, CD.
15. Aladjev V.Z., Bogdevicius M.A. Use of package *Maple V* for solution of physical and engineering problems // Int. Conf. TRANSBALTICA- 99.- Vilnius: Technics Press, 1999.
16. Aladjev V.Z., Hunt U.J. Workstation for mathematicians // Int. Conf. TRANSBALTICA- 99.- Vilnius: Technics Press, April 1999.
17. Aladjev V.Z., Hunt U.J. Workstation for mathematicians // Internat. Conf. «Perfection of Mechanisms of Management», Institute of Modern Knowledge, 1999, Grodno, Byelorussia.
18. Aladjev V.Z., Shishakov M.L. Programming in Package *Maple V* // 2nd Internat. Conf. «Computer Algebra in Fundamental and Applied Researches and Education».- Minsk, 1999.
19. Aladjev V.Z., Shishakov M.L. A Workstation for mathematicians // 2nd Internat. Conf. «Computer Algebra in Fundamental and Applied Researches and Education».- Minsk, 1999.
20. Aladjev V.Z., Shishakov M.L., Trokhova T.A. Educational computer laboratory of the engineer // Proc. 8th Byelorussia Mathemat. Conf., vol. 3, Minsk, Byelorussia, 2000.
21. Aladjev V.Z., Shishakov M.L., Trokhova T. Modelling in program environment of the

- mathematical package *Maple* // Int. Conf. on Math. Mod. MKMM-2000.- Herson, 2000.
22. *Aladjev V.Z., Shishakov M.L., Trokhova T.A.* A workstation for solution of systems of differential equations // 3rd Internat. Conf. «*Differential Equations and Applications*».- Sant-Petersburg, Russia, 2000.
 23. *Aladjev V.Z., Shishakov M.L., Trokhova T.A.* Computer laboratory for engineering researches // Internat. Conference ACA-2000.- Saint-Petersburg, Russia, 2000.
 24. *Aladjev V.Z., Bogdevicius M., Hunt U.J.* A Workstation for mathematicians / Lithuanian Conf. TRANSPORT-2000.- Vilnius: Technics Press, April 2000, Lithuania.
 25. *Аладьев В.З.* Компьютерная алгебра // Альфа, № 1.- Гродно: Изд-во Гродненский госуниверситет, Беларусь, 2001.
 26. *Aladjev V.Z.* Modern computer algebra for modeling of the transport systems // Intern. Conf. TRANSBALTICA-2001.- Vilnius: Technics Press, April 2001, Lithuania.
 27. *Aladjev V.Z., Shishakov M.L., Trokhova T.* Workstation for the engineer-mathematician // Proc. of the GSTU, № 3, 2000.- Gomel: State University, Gomel, Byelorussia.
 28. *Aladjev V.Z., Bogdevicius M.A.* *Special Questions of Operation in Environment of the Mathematical Maple Package*.- Vilnius: Vilnius Gediminas Technical University, 2001.
 29. *Aladjev V.Z., Vaganov V.A., Grishin E.P.* *Additional Functional Tools of Mathematical Package Maple 6/7*.- Tallinn: International Academy of Noosphere, 2002.
 30. *Аладьев В.З.* Эффективная работа с *Maple 6/7*.- М.: Изд-во БИНОМ, 2002 + CD.
 31. *Аладьев В.З., Лиопо В., Никитин А.В.* Математический пакет *Maple* в физическом моделировании.- Гродно: Гродненский госуниверситет, 2002, ISBN 3-093-31831-3.
 32. *Aladjev V.Z., Vaganov V.A.* *Computer Algebra System Maple: A New Software Library*.- Tallinn: Изд-во International Academy of Noosphere, 2002, 420 p. + CD.
 33. *Aladjev V.Z., Bogdevicius M.A., Prentkovskis O.* *New Software for Mathematical Package Maple of Releases 6, 7 and 8*.- Vilnius: Vilnius Gediminas Technical University, 2002.
 34. *Aladjev V.Z.* *Computer Algebra Systems: A New Software Toolbox for Maple*.- Palo Alto: CA: Fultus Publishing, 2004, ISBN 1-59682-000-4.
 35. *Aladjev V.Z.* *Computer Algebra Systems: A New Software Toolbox for Maple*.- Palo Alto: CA: Fultus Publishing, 2004, Adobe Acrobat eBook.
 36. *Aladjev V. et al.* *Electronic Library of Books and Software for Experts, Scientists, Teachers and Students in Natural and Social Sciences*.- Palo Alto: CA: Fultus Publishing Co., 2005, CD.
 37. *Aladjev V.Z., Vaganov V.A.* *Systems of Computer Algebra: A New Software Toolbox for Maple*.- Tallinn: International Academy of Noosphere, 2003, 270 p., ISBN 9985-9277-6-1.
 38. *Aladjev V.Z., Bogdevicius M.A., Vaganov V.A.* *Systems of Computer Algebra: A New Software Toolbox for Maple. 2nd ed.*- Tallinn: International Academy of Noosphere, 2004.
 39. *Aladjev V.Z., Bogdevicius M.A.* Computer algebra system *Maple*: A new software toolbox // 4th Intern. Conf. TRANSBALTICA-03.- Vilnius: Technics Press, April 2003.
 40. *Aladjev V.Z.* Computer Algebra System *Maple*: A New Software Library // Intern. Conf. «*Computer Algebra Systems and Their Applications*», CASA-2003, Saint-Petersburg, 2003.
 41. *Aladjev V., Bogdevicius M., Vaganov V.* Systems of Computer Algebra: A New Software Toolbox for package *Maple* // Int. Conf. on Soft. Engin. Res. and Practice, 2004, Las Vegas.
 42. *Аладьев В.З.* Системы компьютерной алгебры. *Maple: Искусство программирования*,

Москва: Изд-во БИНОМ, 2006, 792 с.

43. **Аладьев В.З.** Основы программирования в *Maple*.- www.aladjev-maple.narod.ru, Таллинн: Международная Академия Ноосферы, 2006, ISBN 9985-9508-1-X.
44. **Аладьев В.З., Бойко В.К., Ровба Е.** Программирование и разработка приложений в *Maple*.- Гродно: Изд-во ГрГУ, <http://www.aladjev-maple-book.narod.ru/>, 2007, 456 с.
45. **Аладьев В.З.** Библиотека версии 2.2214 для пакета *Maple*.- www.aladjev.narod.ru, www.aladjev-maple-book.narod.ru, 2010; <http://depositfiles.com/files/9uuxaburm>.
46. **Голоскоков Д.** Уравнения математической физики. Решение задач в системе *Maple*.- Санкт-Петербург: Изд-во Питер, 2004.
47. **Васильев А.Н.** *Maple 8*. Самоучитель.- Москва: Изд-во Диалектика, 2003.
48. **Кирсанов М.** Задачи по теоретической механике с решениями в *Maple 11*.- Москва: Изд-во Физматлит, 2010, 264 с.
49. **Коробов В.И., Очков В.Ф.** Химическая кинетика: Введение с *Mathcad/Maple/MCS*.- Москва: Изд-во Горячая линия-Телеком, 2009.
50. **Кирсанов М.Н.** Графы в *Maple*.- Москва: Изд-во Физматлит, 2007.
51. **Эдвардс Ч.Г., Пенни Д.Э.** Дифференциальные уравнения и краевые задачи: Моделирование и вычисление с помощью *Mathematica, Maple* и *MATLAB*.- Киев: Изд-во Диалектика, 2007.
52. **Очков В.** Физические и экономические величины в *Mathcad* и *Maple*.- Москва: Изд-во Финансы и Статистика, 2002.
53. **Матросов А.** *Maple 6*: Решение задач высшей математики и механики.- Санкт-Петербург: Изд-во БХВ-Петербург, 2001.
54. **Манзон Б.** *Maple V Power Edition*.- Москва: Изд-во ФилинЪ, 1998.
55. **Проخورов Г., Леденев М., Колбеев В.** Пакет символьных вычислений *Maple*.- Москва: Изд-во Петит, 1997.
56. **Говорухин В., Цибулин В.** Введение в *Maple*. Математический пакет для всех.- Москва: Изд-во Мир, 1997.
57. **Шмидский Я. К.** *Mathematica 5*.- Москва: Изд-во Диалектика, 2004.
58. **Капустина Т.** Компьютерная система *Mathematica 3.0* для пользователя.- Москва: Изд-во Солон, 1999.
59. **Воробьев Е.** Введение в систему *Mathematica*.- Москва: Изд-во ФиС, 1998.
60. **Кулешов А.А. и др.** Электронное учебное пособие по высшей математике на базе системы *Mathematica*.- Минск: Электронная книга БГУ, 2001.
61. www.ecsocman.edu.ru/text/16089443, www.aladjev.narod.ru, www.aladjev-maple.narod.ru
62. **Nelson H. F. Beebe.** A Bibliography of Publications about the *Maple* Symbolic Algebra Language.- Salt Lake City: Univ. of Utah, Department of Mathematics, 119 с., 2010.
63. <http://reduce-algebra.sourceforge.net/bibl/bib.html> - библиография публикаций по физике с использованием системы компьютерной алгебры *REDUCE*.
64. **Еднерал В.Ф., Крюков А.П., Родионов А.Я.** Язык аналитических вычислений *REDUCE*.- Москва: Изд-во МГУ, 1989. 176 с.
65. **Климов Д.М., Руденко В.М.** Методы компьютерной алгебры в задачах механики.- Москва: Изд-во Наука, 1989, 215 с.

66. Теоретическая механика. Вывод и анализ уравнений движения на ЭВМ / Под ред. **В.Г. Веретенникова**.– Москва: Изд-во Высшая школа, 1990, 174 с.
67. Методы аналитических преобразований в системе виртуальных машин ЕС ЭВМ для задач динамики ЛА / Под ред. проф. **А.Г. Сокольского**.– Москва: МАИ, 1988.
68. Программирование на языке *R-ЛИСП*.– Москва: Изд-во Радио и связь, 1991, 192 с.
69. **Hearn A.C. REDUCE User's Manual: Version 3.3**, RAND Publication *CP78*, The Rand Corporation, Santa Barbara, California, 4/1987.
70. **Hearn A.C. REDUCE: The First Forty Years** // A3L Conference in Honor of the 60th Birthday of Volker Weispfenning, Santa Monica, April 2005.
71. **Дэвенпорт Дж. и др.** Компьютерная алгебра.– Москва: Изд-во Мир, 1991, 352 с.
72. **Дэвенпорт Дж.** Интегрирование алгебраических функций / Пер с англ.– Москва: Изд-во Мир, 1985, 192 с.
73. http://saither.ucoz.ru/news/symbolic_integration/2010-08-28-49
74. Компьютерная алгебра: Символьные и алгебраические вычисления / Пер. с англ.– Москва: Изд-во Мир, 1986, 392 с.
75. Reference Manual *muLisp-85*.– Honolulu: Hawaii, Software House, Inc., 1985, 137 p.
76. **Stein W. et. al.** Sage Tutorial.– Charleston: CreateSpace Press, 2009.
77. **Joyner D.** Differential Calculus and Sage.– Charleston: CreateSpace Press, 2009.
78. **Очков В.Ф.** *Mathcad 14* для студентов и инженеров: Русская версия.– Санкт-Петербург: Изд-во ВHV–Петербург, 2009.
79. **Коробов В.И., Очков В.Ф.** Химическая кинетика: Введение с *Mathcad/Maple/MCS*.– Москва: Горячая линия–Телеком, 2009.
80. **Очков В.Ф.** *Mathcad 14* для студентов и инженеров.– Санкт-Петербург: Изд-во ВHV–Петербург, 2007.
81. http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems – Сравнение современных систем компьютерной алгебры различного назначения.
82. **Шишаков М.Л., Трохова Т.А., Стрижак И.В.** Обучение алгоритмизации и программированию в системах компьютерной математики // Информатизация образования, № 2, с. 59–62, 2005.
83. **Jenks R.D., Sutor R.S.** *AXIOM: The Scientific Computation System*.– N.Y.: NAG and Springer Verlag, 1992.
84. **Кулябов Д.С., Кокотчикова М.Г.** Аналитический обзор систем символьных вычислений // Вестник РУДН, № 1–2, с. 38–45, 2007.
85. **Aladjev V.Z.** Classical Cellular Automata: Homogeneous Structures.– CA: PaloAlto: Fultus Corp., 2010, 478 p., ISBN 1-59682-222-8.
86. <http://www.haskell.org> – Сайт по вопросам функционального программирования
87. **Wadler P.** Why no one uses functional languages // ACM SIGPLAN Notices, 1998.
88. **Dominus M.J.** Higher-Order *Perl*.– Massachusetts: Morgan Kaufmann Press, 2005.
89. **Arantes R.D.** *A Computational Reference Guide on Experimental Mathematics, Algorithmic Number Theory and Symbolic Computing*.– Rio de Janeiro: Federal University, 2004.
90. **Aladjev V.Z.** A Package of Procedures and Functions for *Mathematica*.– Tallinn, 2011; The package can be freely downloaded from <http://depositfiles.com/files/5c5nmuulx>.

Список пользовательских процедур для пакетов Mathematica и Maple, упоминаемых в настоящей книге

Пакет Maple:

&Shift, &ma, ArtMod, AtrRW, CALL, ChkPnt, Currentdir, Daclose, Daopen, Daread, DecLevLS, DefOpt, DoF, E_mail, FOR_DO, Fend, FilePal, Find, FoldList, Remove, Globals, Hasfun, ListPack, Lprot, M_Type, MapIndexed, MapleLib, MapleLib, MkDir, MkDir, MkDir, MkDir1, MkDir4, MkDir4, NF, Nest, NestList, NestWhile, Nresults, OpenLN, ParProc, ParProc1, Plib, Read1, Red_n, wsread, Remember_T, RevPath, Save2, SaveMP, SoftTab, Sproc, StatLib, StatLib, Subs, Timetest, UpLib, UpLib, UpLib, WARNING, With, With, With6, _SL, assign67, belong, belong, cmf, com_exe1, com_exe2, `convert/proc`, dcmmod, deltab, diff, elib, email, extexp, extrcalls, helpman, holdof, intproc, isDir, isFile, islabel, lop3, member1, member2, mnames, mod21, mwsname, nvalue, open2, pusers, readdata1, readm1, save1, saveall, savem, savem1, simple1, staff, swmpat, swmpat1, tabar, tpacmod, `type/Type`, `type/boolproc`, `type/color`, inttt, `type/complex1`, `type/dir`, `type/file`, `type/file`, `type/file1`, `type/fpath`, `type/parse`, `type/path`, `type/path`, uglobal, uplib, usertype, utfile, writedata1

Пакет Mathematica:

\$ProcName, AcNb, ActiveProcess, Adrive, Aobj, Args, Args0, Arity, AssignToList, Attrib, AutoLoadPack, BinaryListQ, Bits, BlockQ, CALL, CDir, CallsInProc, CharacterQ, CloseAll, Closes, Contexts1, CopyDir, DeCod, Defaults, DefFunc, DelEl, Df, Df1, Df2, DirName, DirQ, Email, Email1, EmptyFileQ, ExtrPackName, FileFormat1, FileOpenQ, FunCompose, FunctionQ, GC, GV, Gather1, HowAct, Int, Int1, IsFileOpen, ListListQ, ListToString, LoadFile, Locals, Map1, Map2, Map3, Map4, Map5, Map6, Mapp, MaxNestLevel, MemberLN, MemberQ1, MemberQ2, MemberT, MixCaseQ, ModLibraryPath, NamesCS, NamesNbPackage, NamesNbPackage1, NamesProc, NamesProc1, NbName, Need, NestQL, Nobj, Npackage, Nproc, Nvalue, OpenFiles, PackNames, ProcQ, PureFuncQ, Range1, Range2, Range3, ReadFullFile, Rename, ReplaceAll1, RETURN, SearchDir, SearchFile, SeqUnion, SequenceQ, SetDir, SortNL, SortNL1, SortString, Spos, StrDelEnds, StrStr, StreamsU, StringEnd, StringReplace1, SubDelStr, SubStr, Subs, SuffPref, SymbolQ, Tbl, TestArgsTypes, TestProcCalls, ToString1, Type, TypeActObj, UpdateContextPaths, UpdatePackages, UpdatePath, UprocQ, Uprocs, UserLib, VarExch, VarExch1, Ver, WhatObj

Детальное описание как процедур представленного списка, так и целого ряда других пользовательских процедур для пакетов *Maple* и *Mathematica* могут быть найдены в библиотеке процедур для *Maple* [45] и в небольшом пакете для *Mathematica* [90].