

Проблема создания программного обеспечения для реализации интервальных алгоритмов

Юлдашев З.Х., Ибрагимов А.А., Калханов П.Ж.

1. Проблема реализации интервальных алгоритмов

Интервальные методы[1-4] в настоящее время вызывают всё больший интерес исследователей, как способ моделирования в условиях недетерминированности данных, когда амплитуда их колебания имеет ограниченный характер. В ряде случаев интервальные методы рассматриваются в контексте альтернативных методов вероятностно-статистическим методам и методам теории нечётких множеств[5]. Можно отметить, что указанный интерес к интервальным методам в свою очередь инициирует работы по созданию новых интервальных методов и различных вариантов *интервальных арифметик*(ИА)[2-4], которые учитывали бы особенности тех или иных задач и давали бы инструмент для адекватного исследования некоторого процесса. Возникающие из практических потребностей описания новых трактовок интервального числа, правил соответствующих арифметик можно найти в обширной литературе по *интервальному анализу* или *интервальной математике*[4], как характеризуют некоторые авторы это новое направление в прикладной математике. В традиционном интервальном анализе выделяются следующие проблемы:

- исследование множества интервальных чисел, как некоторой структуры и определяемых подходящим образом интервальнозначных функций, как некоторого множества отображений множества интервальных чисел в себя;
- построение и обоснование адекватности интервальных моделей для конкретных задач в рамках соответствующего исчисления интервалов;
- построение, обоснование и исследование свойств интервальных алгоритмов;
- создание программного обеспечения для реализации интервальных алгоритмов.

Данная работа посвящена проблеме создания программного обеспечения для реализации интервальных алгоритмов.

В настоящее время разработаны различные способы организации интервальных вычислений на ЭВМ, созданы конкретные системы и библиотеки.

Однако имеющийся значительный перечень вариантов программного обеспечения, состоящий из по-своему эффективных реализаций применения методов интервальной математики, носит разнопланформенный характер. Это обстоятельство не только требует изучения конкретных подходов в реализации алгоритмов с целью их использования, но в отдельных случаях не

представляет “широкому пользователю” удовлетворительный интерфейс. Прежде всего, это выражается в ограниченной возможности варьирования базовых вариантов интервальных арифметик. Другой важной особенностью совокупности этих реализаций является отсутствие возможности оперативной перегрузки методов в условиях многопланового численного эксперимента и освобождении пользователя от необходимости детального изучения самой интервальной математики. Когда же речь идет об освобождении пользователя от необходимости знакомства с основами интервального анализа и проведения некоторых подготовительных работ путем возложения их на “плечи” ЭВМ, то с точки зрения парадигмы модульного принято было говорить об актуальности или необходимости создания некоторого пакета прикладных программ (ППП) (некоторой алгоритмической системы или генератора программ), производящего автоматическое построение интервальных алгоритмов и состоящего из подпрограмм–модулей, разработанных на модульном принципе [6].

При алгоритмизации методов интервального анализа существенные трудности возникают из-за неприспособленности аппаратных средств ЭВМ к учету специфических требований интервального исчисления. Эти требования обусловлены тем, что в процессе генерации конкретных интервальных алгоритмов на ЭВМ возникает потребность в подпрограммной машинной интервальной арифметике, в вычислении значений интервальнозначных функций и в некоторых аналитических преобразованиях. Следует отметить, что в большинстве предлагаемых интервальных алгоритмов предполагается, что пользователь должен иметь знакомство с основами интервального анализа, уметь проводить некоторые подготовительные работы. Это обстоятельство в определенной мере ограничивает круг пользователей интервальных методов. Под ППП здесь понимается общепринятое понятие пакета, т.е. совокупность программ и системных средств, необходимых для решения определенного класса задач. Если имеется аппаратная реализация интервальных операций и вычисления интервальнозначных функций в виде микропрограмм, которая позволяет уменьшить время выполнения интервально–арифметических операций, то, очевидно, написание и отладка программ в терминах интервального анализа значительно упрощается и ускоряется сам процесс вычислений. Отсутствие аппаратной реализации интервальных операций делает необходимым включение в пакет “библиотеки обслуживающих модулей”, которую условно можно разбить на три группы.

К первой группе относятся модули машинной интервальной арифметики, ко второй же группе можно отнести модули интервального вычисления элементарных функций. Модули специальных и логических интервальных операций составляют третью группу библиотеки. В зависимости от целей пакета список библиотеки обслуживающих модулей может быть расширен либо сжат.

2. Объектно-ориентированный подход при реализации интервальных алгоритмов

В [2] дан, возможно, несколько устаревший, анализ вариантов реализации интервальной арифметики по Муру и некоторых вариантов интервальной арифметики, предлагаемых для различных целей. Он отражает доминировавшую в тот период *парадигму модульного программирования*. Что касается непосредственной реализации конкретного алгоритма, то большинство исследователей решало эту задачу, создавая программы моделируя вычислительный процесс путём расчленения интервальных выражений на обращения к соответствующим подпрограммам [7-8]. В работе [8] задача освобождения пользователя от необходимости подробного владения методами интервального анализа решалась построением соответствующих М-матриц.

Анализируя публикации, появившиеся за период прошедший после выхода часто цитируемой работы [2], можно отметить не только разнообразие исследованных вопросов интервального анализа и разнообразие приложений интервальных методов, но и тенденцию в разработке новых вариантов ИА и сочетании различных ИА при решении конкретной задачи, создании пакетов интервальных алгоритмов. Например, в работе [9] при анализе интервальных вычислительных процессов в пространстве \mathbb{R} , наряду с операцией вычитания по Муру, вводится так называемое *нестандартное вычитание по Маркову*, что позволяет автору учитывать случаи точного выполнения операций и получать улучшенные оценки роста погрешностей вычислений. В работах других авторов выбор конкретно варианта ИА играет существенную роль. Например, в [10] при решении задачи внутреннего оценивания множеств решений систем линейных алгебраических уравнений с интервальными коэффициентами, автор использует *обобщённую интервальную арифметику Каухера*. Это не только является естественным для поставленной задачи, но и приводит к существенным преимуществам разрабатываемых интервальных алгоритмов.

Следовательно, возникает необходимость в выборе конкретного варианта интервальной арифметики и создании соответствующего программного обеспечения, обеспечивающего автоматизацию такого выбора при реализации интервальных алгоритмов. Конкретно была сформулирована следующая задача: *создать пакет интервальных алгоритмов по реализации вычисления значения алгебраически допустимого выражения с выбором подходящего варианта интервальной арифметики*.

Концепция объектно-ориентированного программирования наиболее удобно реализуемая на языке C++, предполагает возможность автоматического переключения процесса вычислений из одного варианта интервальной арифметики в другой при условии сохранения записи самого алгебраически допустимого выражения. Для решения указанной задачи был разработан соответствующий пакет [11], в котором модули, реализующие

алгоритмы операций одной из следующих вариантов интервальных арифметик, а именно, ИА "по Муру", ИА "по Маркову", ИА Кахана, ИА Каухера, ИА круговых интервальных чисел, а также их варианты для комплексных интервальных чисел, реализованы в виде классов. Такой механизм позволяет, не меняя конструкции алгебраически допустимых выражений, *перегрузить* соответствующие алгоритмы выбранной ИА. При этом пользователю нужно лишь знать свойства выбираемого варианта ИА либо экспериментально выбирать наиболее удобный вариант ИА для решения данной задачи. Таким образом, пользователь получает возможность проводить вычислительный эксперимент не только для различных начальных данных, но и устанавливать предпочтительную вычислительную среду, не меняя конструкции, реализующие вычислительный алгоритм. Кроме того, в силу концептуальной открытости пакета, пользователь имеет возможность самостоятельного расширения списка указанных арифметик. Как развитие идеи по автоматизации применения различных вариантов ИА возникла задача создания *пакета интервальных алгоритмов для широкого круга пользователей*[12].

Настоящий пакет интервальных алгоритмов разработан на основе принципов объектно-ориентированного программирования. Поскольку он обладает открытой архитектурой, то в него интегрирован описанный выше *«Комплекс программ для вычисления значений интервальных алгебраически допустимых выражений в рамках различных интервальных арифметик»*. С другой стороны данный вариант пакета интервальных алгоритмов в отличие от ряда других пакетов интервальных алгоритмов не только отличается многообразием базовых интервальных арифметик, но и тем, что при его разработке авторы старались придерживаться *принципа дружелюбности*. В данном случае *принцип дружелюбности* понимается не только в простоте и удобстве интерфейса, но и разработанности достаточного количества программных средств расширяющих, круг пользователей интервальными методами. В частности, в рамках пакета пользователям не знакомым с методами интервального анализа, предоставляется возможность проведения определённого ряда вычислительных экспериментов. При этом достаточно записать задания, используя обычные математические выражения. В данном варианте пакета, пользователь имеет возможность решать интервальными методами ряд задач вычислительной математики, являющихся базовыми для многих алгоритмов решающих конкретные задачи. Кроме того, пользователь имеет возможность настраивать пакет на применение новых алгоритмов, оставаясь в рамках привычных языковых конструкций.

В нашем случае – это класс задач интервального анализа либо класс вещественных задач, которые необходимо решить в рамках интервального анализа. Системными средствами являются штатные средства операционных систем ЭВМ. Под модулем понимается программа, оформленная как функция в C++, реализующая на ЭВМ некоторый законченный фрагмент

вычислительного процесса решения конкретной задачи, который заключается в вычислении выражений как с действительными, так и с интервальными числами.

3. Структура и функциональные возможности пакета интервальных алгоритмов

Как отмечалось выше на языке C++ создан пакет ориентированный на широкий круг пользователей. В данном пакете тематически однотипные алгоритмы оформлены в виде классов. Процесс разработки пакета был основан на принципах объектно-ориентированного программирования, дополнен операторами постановок задач, которые близки к общепринятому языку. Он дает пользователю возможность вести диалог с пакетом. В частности, пользователь имеет возможность создания конкретной интегрированной программы, сочетающую обращения к модулям пакета с исполнением собственных кодов.

Следует отметить, что разработка совокупности программ *«Пакет интервальных алгоритмов для широкого пользователя»* осуществлялась на основе *принципа дружественности*. В данном случае *принцип дружественности* понимается не только в простоте и удобстве интерфейса, но и разработанности достаточного количества программных средств, расширяющих круг пользователей интервальными методами. Кроме наличия достаточного количества алгоритмов, предназначенных для решения задач при условии ограниченного колебания недетерминированных данных, пользователь имеет возможность перегружать интервальные арифметики на основе соответствующего механизма из средств C++[13]. Это обстоятельство позволяет расширить спектр вариантов вычислительного эксперимента, ориентируя выбор, как базовой арифметики, так и подходящего метода на содержательную сторону исходной задачи, имеющей неопределённость различных типов. По существу здесь речь идёт о двух взаимосвязанных пакетах[11,12], набор методов которых открыт для самостоятельного и независимого пополнения по мере необходимости применения новых вариантов интервальных арифметик или решения новых задач с недетерминированными параметрами. Учитывая это обстоятельство, ниже будет идти речь об интегрированном пакете, создаваемом объединением методов из [11] и [12] в рамках одного шаблонного класса[13].

Существенным является то обстоятельство, что пользователю предоставляется возможность проведения определённого ряда вычислительных экспериментов, без требования углубленного знакомства с методами интервального анализа. При этом достаточно записать задания, используя обычные математические выражения. В данном варианте пакета, реализованного средствами C++ на основе принципов объектно-ориентированного программирования[13], пользователь имеет возможность решать интервальными методами ряд задач вычислительной математики и

настраивать пакет на применение новых алгоритмов, оставаясь в рамках привычных языковых конструкций.

Численные эксперименты в интервальной математике играют особую роль, поскольку в рамках исчисления интервалов пользователь при решении своей задачи может не только “редактировать” применяемую математическую модель, выбирать подходящий алгоритм, корректировать машинный алгоритм, то есть саму программу, как это происходит при решении задач вещественными методами, но проследить результаты варьирования отдельных параметров. Ещё раз отметим в процессе вычислительного эксперимента в нашем случае пользователь имеет возможность выбирать подходящую интервальную арифметику, при необходимости без особых усилий редактировать, точнее, изменять набор методов соответствующего класса.

Возможности настоящего пакета, созданного на языке C++ с использованием принципов объектно-ориентированного программирования, позволяют прогнозировать достаточно широкий круг пользователей, поскольку дополнен операторами постановок задач, которые близки к общепринятому языку. Он дает пользователю возможность вести диалог с пакетом. В частности, пользователь имеет возможность указывать на необходимость интервального решения задачи, получать интервальную формулировку задачи удобочитаемого вида. Так как операторы постановки задачи имеют естественный неалгоритмический характер, то программное обеспечение языка запросов должно быть таким, чтобы результатом обработки запросов являлась некоторая программа, полностью удовлетворяющая пользователей для формулировки задачи, что и предлагается потенциальным пользователям данного пакета. Пользователь, взаимодействия с пакетом, формирует исходную информацию о задаче на неограниченном естественном языке, а пакет, анализируя эту информацию, строит интервальную модель задачи по признакам входной информации, решает интервальную задачу.

Данный вариант пакета для широкого круга пользователей включает в себя модули для решения следующих задач в интервальной постановке:

- 1) *Выбор и тестирование базовой интервальной арифметики для вычислительного эксперимента.*
- 2) *Задачи интервальной матричной алгебры.*
- 3) *Алгебра интервальных полиномов.*
- 4) *Корни многочленов с интервальными коэффициентами.*
- 5) *Интервальный вариант решения проблемы собственных значений.*
- 6) *Решение интервальных систем линейных алгебраических уравнений (ИСЛАУ).*
- 7) *Задачи линейного программирования в интервальной постановке.*
- 8) *Решение задачи Коши для обыкновенных дифференциальных уравнений.*

В инструкции к пакетам на примерах демонстрируется как удобство структуры интерфейса пакета, так многочисленность возможных применений пакетов, а также приводятся правила и приёмы их использования. При тестировании самих пакетов были выбраны те алгоритмы или методы, которые описаны и обоснованы в многочисленных работах по интервальному анализу. В большинстве случаев, особенно для случаев с небольшими размерностями задачи, когда эффект раскрутки не приводил к потере информативности, удавалось получать более удовлетворительные результаты и содержательные интерпретации в сравнении с использованием известных в литературе пакетов или комплексов программ, в том числе и пакета ИНАН1 [8].

Так, например при расчётах определению эффективности природоохранных мер на динамику распространения однородного экотоксиканта в консервативной среде [14], было показано уменьшение объёма экотоксиканта приходящегося на условную единицу объёма или приближение показателя чистоты единицы объёма к условному показателю чистоты (красная линия). При этом, указанные нижняя и верхняя кривые линии показывают, что насыщенность среды экотоксикантом, при реализации соответствующих природоохранных мер, определяемых уровнем инвестирования, будет приближаться к показателю «катастрофический порог» - *выше красной линии зелёная зона*. Содержательный смысл интервального решения можно сформулировать так: *концентрация экотоксиканта на контролируемом периоде времени $[0, T]$ не опустится ниже порога, определяемого соответствующим интервальным решением, но будет тем ближе к зелёной зоне, чем более интенсивны будут подавляющие загрязнитель факторы.*

Литература

1. Moore R.E. «Interval Analysis».—Englewood Cliffs. N.J.: Prentice–Hall, 1966.
2. Калмыков С.А., Шокин Ю.И., Юлдашев З.Х. Методы интервального анализа. —Новосибирск: Наука, 1986. —224с.
3. Жолен Л., Кифер М., Дидри О., Вальтер Э. Прикладной интервальный анализ.—М.;Ижевск: Институт компьютерных исследований, 2005. — 468 с.
4. Шарый С.П. Конечномерный интервальный анализ. Институт вычислительных технологий СО РАН. Издательство «XYZ». 2010.- 603с.
5. Дилигенский Н.В., Дымова Л.Г., Севастьянов П.В. Нечеткое моделирование и многокритериальная оптимизация производственных систем в условиях неопределенности: технология, экономика, экология. М.: «И з д а т е л ь с т в о М а ш и н о с т р о е н и е – 1», 2004.-397с.
6. Назиров Ш.А., Юлдашев З.Х. Об одном варианте автоматизации применения интервальных методов, основанном на модульном принципе// Модульный анализ. —Новосибирск: ИТПМ СО АН СССР, 1978. —С.55–61.

7. Christ H. Realisierung einer Maschinenintervallarithmetik auf beliebigen ALGOL –60 Compilern. Elektron Rechenanlagen, 10, 217–222, 1968.
8. Шокин Ю.И., Базаров М.Б., Калмыков С.А., Юлдашев З.Х. Пакет прикладных программ «ИНАН–1» для автоматического построения интервальных алгоритмов решения обыкновенных дифференциальных уравнений//Госфонд алгоритмов и программ. СССР, рег №50760000299.– 13с.;–Аннотация в: Алгоритмы и программы. –М.:1987.–вып. №1.–С.21–22.
9. Ибрагимов А.А. Интервально–аналитические методы в математическом моделировании: Дис. ... канд. физ.–мат. наук.– Ташкент: НУУз., 2002. – 141с.
10. Шарый С.П. Интервальные алгебраические задачи и их численное решение: Дисс. ... доктора физ.-мат.наук. – Новосибирск, 2000.
11. Юлдашев З.Х., Ибрагимов А.А., Калханов П.Ж. Комплекс программ для вычисления значений интервальных алгебраически допустимых выражений в рамках различных интервальных арифметик. Зарегистрирован в государственном реестре программ для ЭВМ Республики Узбекистан, Свидетельство об официальной регистрации программ для ЭВМ № DGU 02202, г.Ташкент, 19.05.2011.
12. Юлдашев З.Х., Ибрагимов А.А., Калханов П.Ж. Пакет интервальных алгоритмов для широкого пользователя. Зарегистрирован в государственном реестре программ для ЭВМ Республики Узбекистан, Свидетельство об официальной регистрации программ для ЭВМ № DGU 02201, г.Ташкент, 19.05.2011.
13. Страуструп Б. Язык программирования С++: Спец. изд./ Пер. с англ. – М.: Изд-во БИНОМ; СПб.: Невский диалект, 2001.
14. Юлдашев З.Х., Худайбергенов О.Ж. Калханов П.Ж. Моделирование назначения штрафов за загрязнение окружающей среды в условиях недетерминированности параметров. Труды конференции «Вычислительные и информационные технологии в науке, технике и образовании» г.Павлодар 20-22 сентября 2006г.344-346 стр.

ПРИЛОЖЕНИЯ К ДОКЛАДУ – МАТЕРИАЛ БУДЕТ ИСПОЛЬЗОВАН ПРИ
 ПРЕЗЕНТАЦИИ.

ПРИЛОЖЕНИЕ 1.

**СПИСОК БИБЛИОТЕК ПРОГРАММ ПО ИНТЕРВАЛЬНОМУ
 АНАЛИЗУ**

- [ALIAS](#) – библиотека алгоритмов для систем интервального анализа;
- [CGAL](#) – библиотека вычислительных алгоритмов Геометрии, записанные в C++, использующие интервальные вычисления;
- [CLP\(BNR\)](#) – пакет прикладных программ, предназначенный для логических операций интервальной арифметики;
- [CoStLy](#) – библиотека стандартных функции для комплексных интервалов;
- [COSY](#) – ППП основанный на рядах Тейлора и методах интервального анализа(допускается автономное использование, использование на C++ или на Fortran 90/95);
- [DAEPACK](#) – MIT- библиотека символьных и численных процедур вычислительной математики для интервальных вычислений;
- [filib++](#) (Fast Interval LIbrary) – библиотека C++ для расширенной интервальной арифметики;
- [INTBLAS](#)(INTerval Basic Linear Algebra Software): - интервальный пакет прикладных программ по линейной алгебре реализованный на C++;
- PASCAL-XSC, C-XSC, ACRITH-XSC- языки программирования для интервальных и доказательных вычислений;
- Пакеты в Maple и их аналоги:
 - [Interval Arithmetic PowerTool](#);
 - [intpakX](#) –интервальный пакет компьютерной алгебры;
 - INTAN – интервальный пакет компьютерной алгебры;

- [Mathematica – система компьютерной математики для численных и аналитических вычислений, на основе интервальной арифметики;](#)
- Интервальное программное обеспечение в Matlab:
 - [INTLAB - INTerval LABoratory](#), интервальный toolbox для Matlab'a; (свободно распространяемый продукт);
 - [VERSOFТ](#) – библиотека интервальных алгоритмов для среды MATLAB / INTLAB, предназначенная для решения традиционных и интервальных задач линейной алгебры и оптимизации, в частности, для внешнего оценивания множеств решений интервальных линейных систем и доказательного решения точечных задач линейного и квадратичного программирования(особый интерес представляют реализации интервальных методов для нахождения собственных и сингулярных чисел матриц);

Int4Sci – интервальное расширение Scilab. Особенностью Scilab'a является то, что это свободно распространяемый некоммерческий продукт с открытыми исходными кодами. Существуют его версии для Gnu/Linux, Windows 2000/XP/Vista и других платформ.

ПРИЛОЖЕНИЕ 2.**НЕКОТОРЫЕ РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ**

А). ЭКСПЕРИМЕНТЫ ПО ПРАВИЛЬНОСТИ РАБОТЫ РАЗЛИЧНЫХ
ВАРИАНТОВ ИНТЕРВАЛЬНЫХ АРИФМЕТИК, ПРОВЕДЁННЫЕ
МЕТОДОМ ПЕРЕГРУЗОК САМИХ ОПЕРАЦИЙ.

Операции	по Муру	Кахану	Каухеру		Маркову
A1+B1= [-2;3]+[1;4]	[-1,00; 7,00]	[-1,00; 7,00]	[-1,00; 7,00]	Opp(A1)=[2,00;-3,00] Dual(A1)=[3,00;-2,00]	[-1,00; 7,00]
A2+B2= [-5;-2]+[-4;3]	[-9,00; 1,00]	[-9,00; 1,00]	[-9,00; 1,00]	pro(A1)=[-2,00;3,00] sign(A1)= интервал А не определен	[-9,00; 1,00]
A1-B1=[-2;3]- [1;4]	[-6,00; 2,00]	[-6,00; 2,00]	[-6,00; 2,00]	Opp(A2)=[5,00;2,00]	[-3,00; -1,00]
A2-B2=[-5;-2]-[- 4;3]	[-8,00; 2,00]	[-8,00; 2,00]	[-8,00; 2,00]	Dual(A2)=[-2,00;-5,00] pro(A2)=[-5,00;-2,00]	[-5,00; -1,00]
A1*B1= [-2;3]*[1;4]	[-8,00; 12,00]	[-8,00; 12,00]	[-8,00;12,00]	sign(A2)= интервал А отрицателен	[-8,00; 12,00]
A2*B2= [-5;-2]*[-4;3]	[-15,00; 20,00]	[-15,00; 20,00]	[-15,00; 20,00]		[-15,00; 20,00]
A1/B1= [-2;3]/[1;4]	[-2,00; 3,00]	[-2,00; 3,00]	[-2,00; 3,00]		[-2,00; 3,00]
A1/B2= [-5;-2]/[-4;3]	Деление на интервал содержащий нуль недопустимо	Деление на интервал содержащий нуль недопустимо	[0,00; 0,00]		[-1,67; 5,00]

**Б) РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ ПО ИЗУЧЕНИЮ
ДИНАМИКИ ИЗМЕНЕНИЯ НАСЫЩЕННОСТИ ОДНОРОДНОГО
ЭКОТОКСИКАНТА В КОНСЕРВАТИВНОЙ СРЕДЕ.**

Пример П.1.

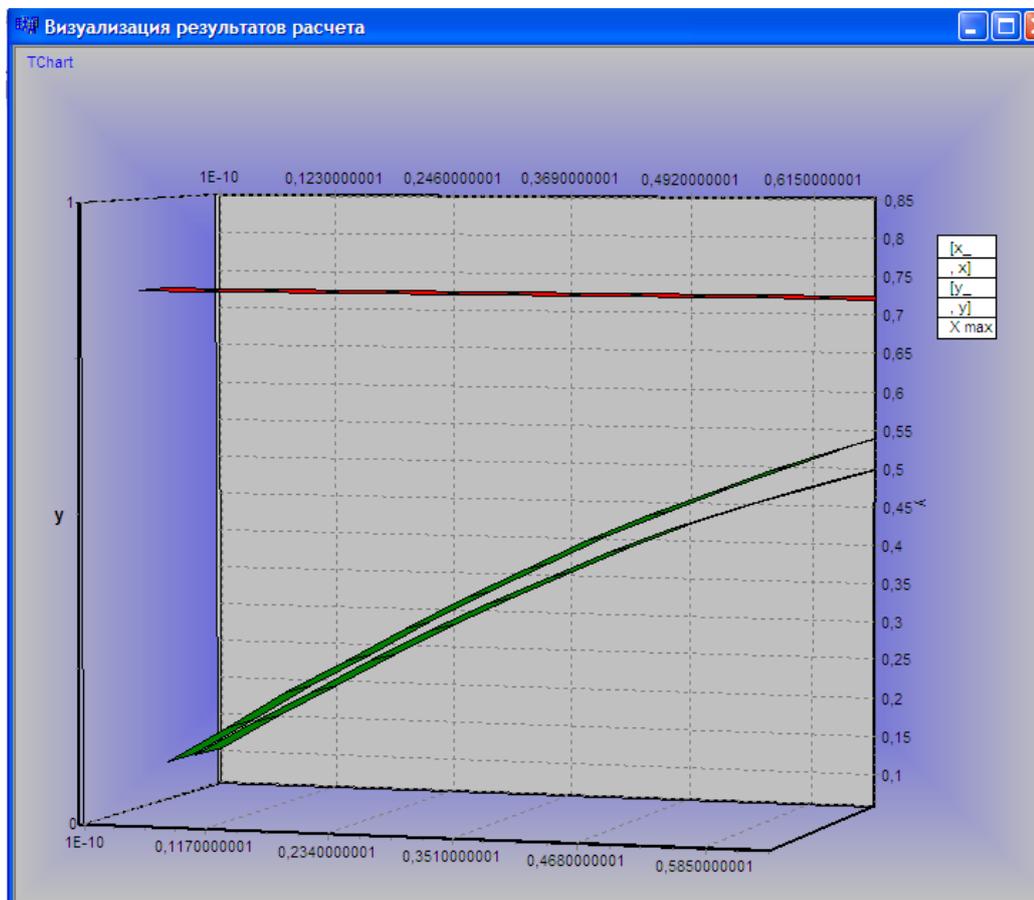
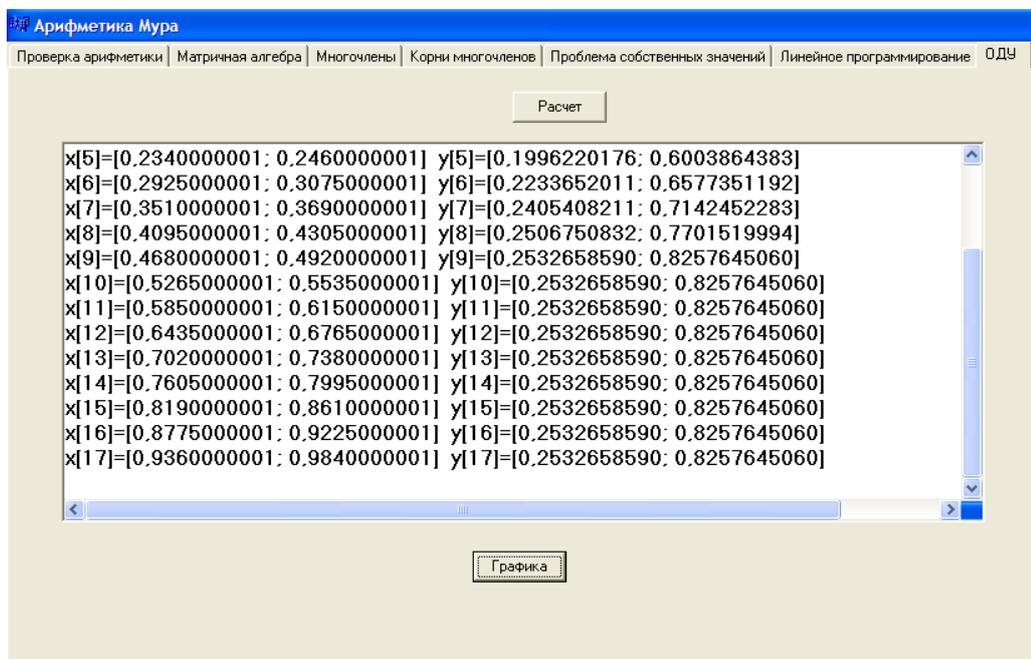
**Решение задачи прогнозирования экологических катастроф
интервальным методом**

Задача решалась на интервале $[0, T]$, $\tau = 0.1 * T$

Таблица П.1

i	X[i].inf	X[i].sup	y[i].inf	Midrad(Y)	y[i].sup
1.	0,0000000001	0,0000000001	0,0938640856	0,25814915	0,4224342223
2.	0,1170000001	0,1230000001	0,1697875810	0,35592131	0,5420550431
3.	0,2340000001	0,2460000001	0,2233650583	0,44054816	0,6577312706
4.	0,3510000001	0,3690000001	0,2506752492	0,5104106	0,7701459602
5.	0,4680000001	0,4920000001	0,2527870125	0,56696686	0,8811467010
6.	0,5850000001	0,6150000001	0,2527870125	0,56696686	0,8811467010
7.	0,7020000001	0,7380000001	0,2527870125	0,56696686	0,8811467010
8.	0,8190000001	0,8610000001	0,2527870125	0,56696686	0,8811467010
9.	0,9360000001	0,9840000001	0,2527870125	0,56696686	0,8811467010

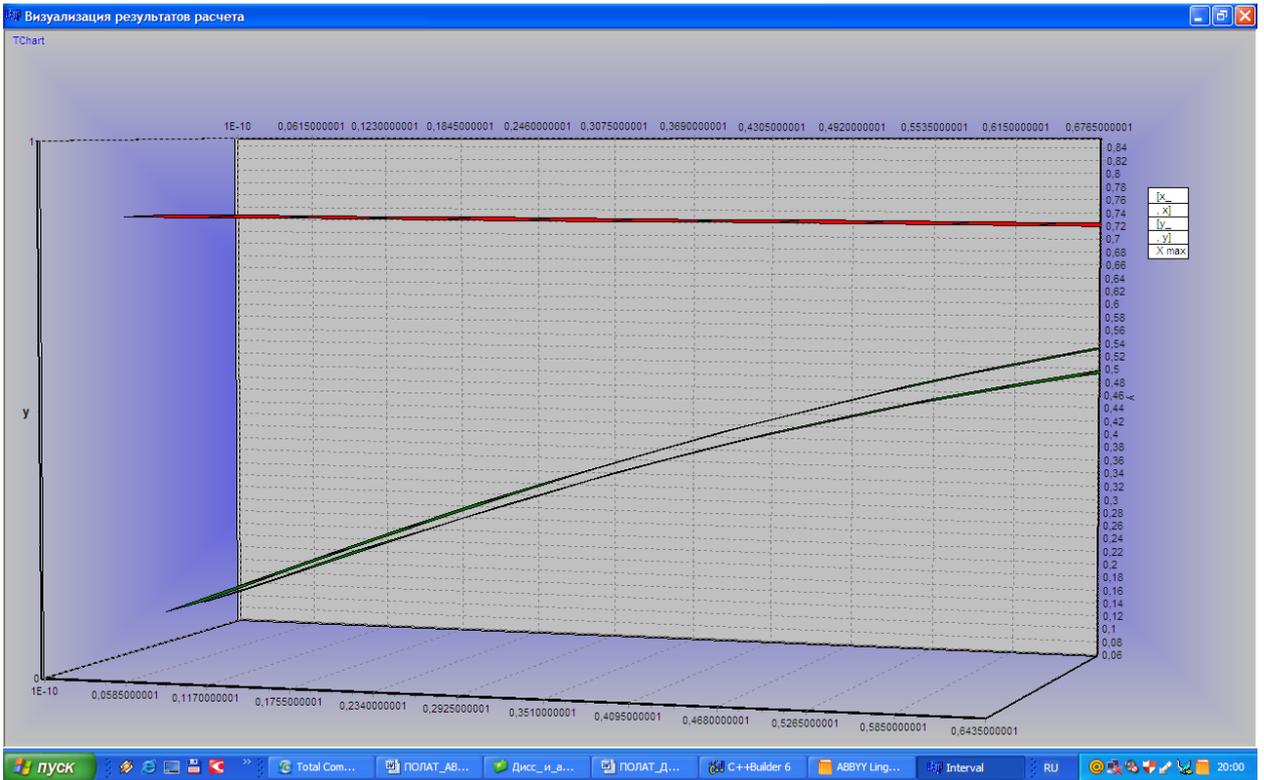
Пример П.2.



Задача решалась на интервале $[0, T]$, $\tau = 0.05 * T$.

Таблица П.2

i	X[i].inf	X[i].sup	y[i].inf	Midrad(Y)	y[i].sup
1.	0,0000000001	0,0000000001	0,0488740361	0,180712582	0,3614251642
2.	0,0585000001	0,0615000001	0,0938641917	0,211217532	0,4224350641
3.	0,1170000001	0,1230000001	0,1343587232	0,241356383	0,4827127663
4.	0,1755000001	0,1845000001	0,1697877520	0,271028582	0,5420571648
5.	0,2340000001	0,2460000001	0,1996220176	0,300193219	0,6003864383
6.	0,2925000001	0,3075000001	0,2233652011	0,32886756	0,6577351192
7.	0,3510000001	0,3690000001	0,2405408211	0,357122614	0,7142452283
8.	0,4095000001	0,4305000001	0,2506750832	0,385076	0,7701519994
9.	0,4680000001	0,4920000001	0,2532658590	0,412882253	0,8257645060
10.	0,5265000001	0,5535000001	0,2532658590	0,412882253	0,8257645060
11.	0,5850000001	0,6150000001	0,2532658590	0,412882253	0,8257645060
12.	0,6435000001	0,6765000001	0,2532658590	0,412882253	0,8257645060
13.	0,7020000001	0,7380000001	0,2532658590	0,412882253	0,8257645060
14.	0,7605000001	0,7995000001	0,2532658590	0,412882253	0,8257645060
15.	0,8190000001	0,8610000001	0,2532658590	0,412882253	0,8257645060
16.	0,8775000001	0,9225000001	0,2532658590	0,412882253	0,8257645060
17.	0,9360000001	0,9840000001	0,2532658590	0,41288253	0,8257645060

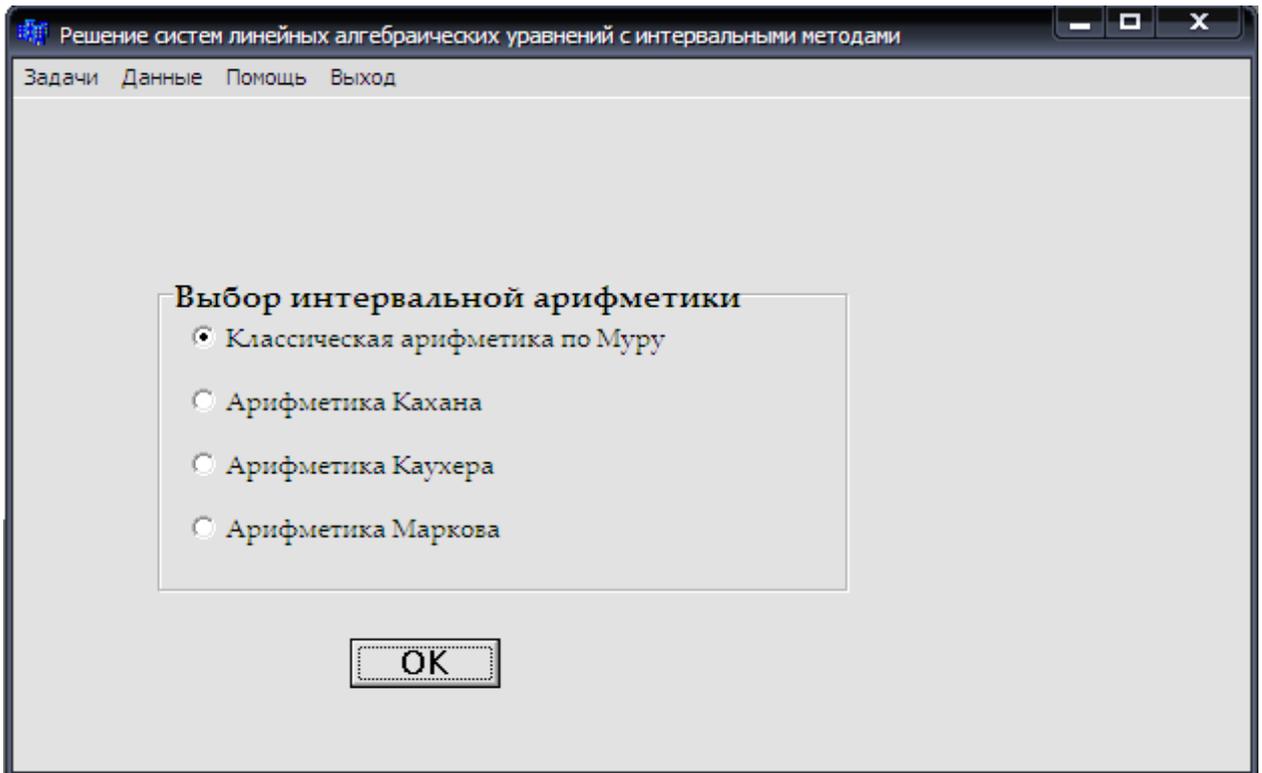


+++++

Приложение 3.

Другие примеры реализации пакета

Пример 3.1., демонстрирующий возможность выбора базового варианта интервальной арифметики при решении ИСЛАУ.



Пример 3.2., демонстрирующий варианты выполнения операций матричной алгебры.

Ввод матрицы A

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0,5	0,5	0,33333	0,33333	0	0	0	0
1	0,33333	0,33333	0,25	0,25	0,2	0,2	0	0
2	0	0	0,2	0,2	0,16666	0,16666	0,14286	0,14286
3	0	0	0	0	0,14286	0,14286	0,125	0,125
4	0	0	0	0	0	0	0,11111	0,11111
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0

Ввод матрицы B

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0,5	0,5	0,33333	0,33333	0,25	0,25	0,2	0,2
1	0,33333	0,33333	0,25	0,25	0,2	0,2	0,16666	0,16666
2	0,25	0,25	0,2	0,2	0,16666	0,16666	0,14286	0,14286
3	0,2	0,2	0,16666	0,16666	0,14286	0,14286	0,125	0,125
4	0,16666	0,16666	0,14286	0,14286	0,125	0,125	0,11111	0,11111
5	0,14286	0,14286	0,125	0,125	0,11111	0,11111	0,1	0,1
6	0	0	0	0	0	0	0	0

Детерминант матрицы A

Результат операции для введенных матриц

Массив C	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	1	1	0,666666666	0,666666666	0,25	0,25
1	0,666666666	0,666666666	0,5	0,5	0,4	0,4
2	0,25	0,25	0,4	0,4	0,333333333	0,333333333
3	0,2	0,2	0,166666666	0,166666666	0,285714286	0,285714286
4	0,166666666	0,166666666	0,142857142	0,142857142	0,125	0,125
5	0,142857142	0,142857142	0,125	0,125	0,111111111	0,111111111
6	0,125	0,125	0,111111111	0,111111111	0,1	0,1

Пример 3.3., демонстрирующий вариант решения частичной проблемы собственных значений для интервальной матрицы (данные и результаты).

Частичная проблема собственных значений

Ввод данных **Расчет**

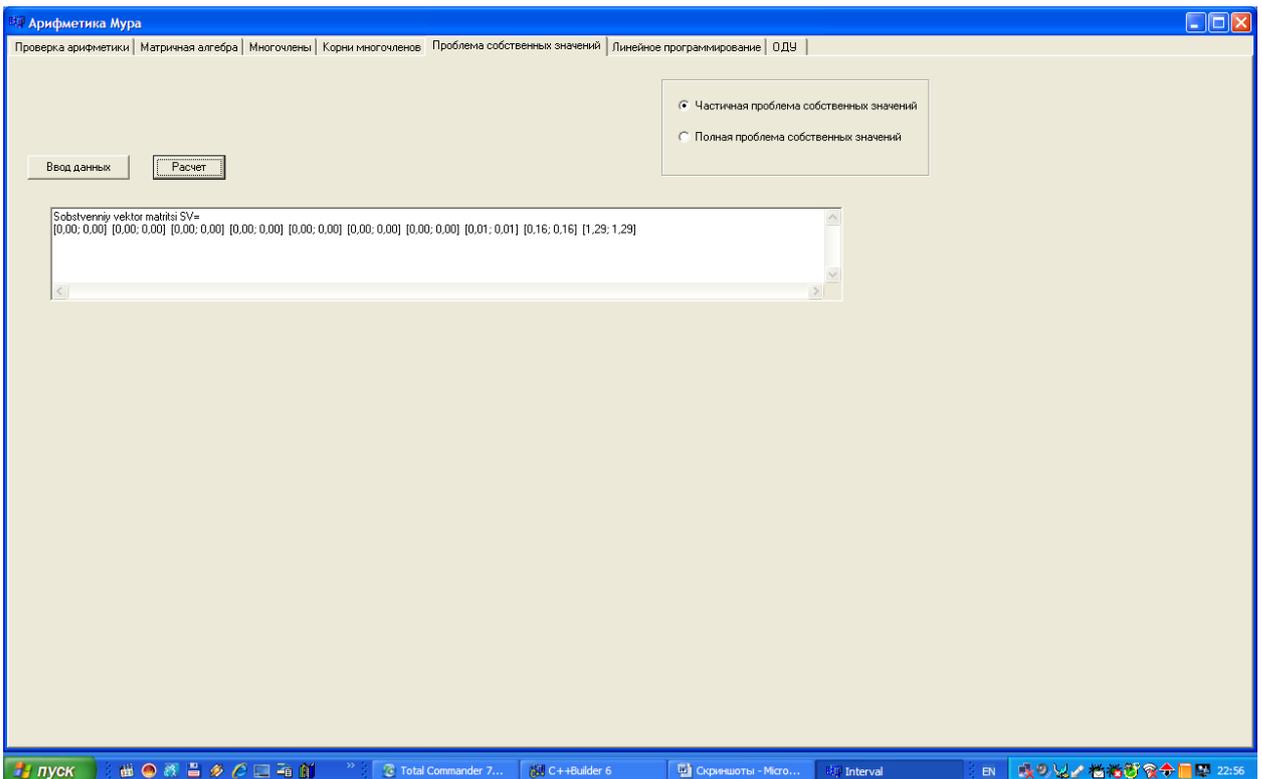
Мемо_Собст_знач

Данные (Арифметика Мура)

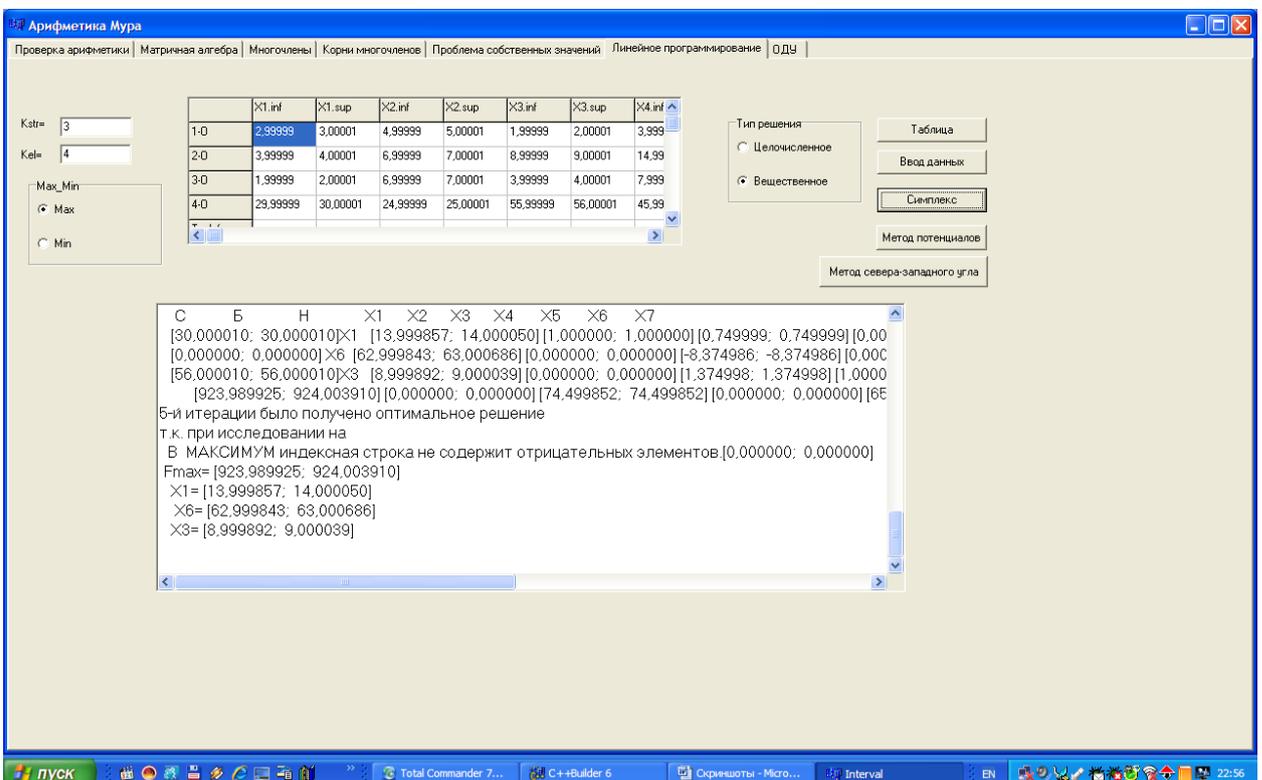
N= Интервальная матрица

	[inf]	sup]								
0	0,5	0,5	0,33333	0,33333	0	0	0	0	0	0
1	0,33333	0,33333	0,25	0,25	0,2	0,2	0	0	0	0
2	0	0	0,2	0,2	0,16666	0,16666	0,14286	0,14286	0	0
3	0	0	0	0	0,14286	0,14286	0,125	0,125	0,11111	0,11111
4	0	0	0	0	0	0	0,11111	0,11111	0,1	0,1
5	0	0	0	0	0	0	0	0	0,09090	0,09090
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0

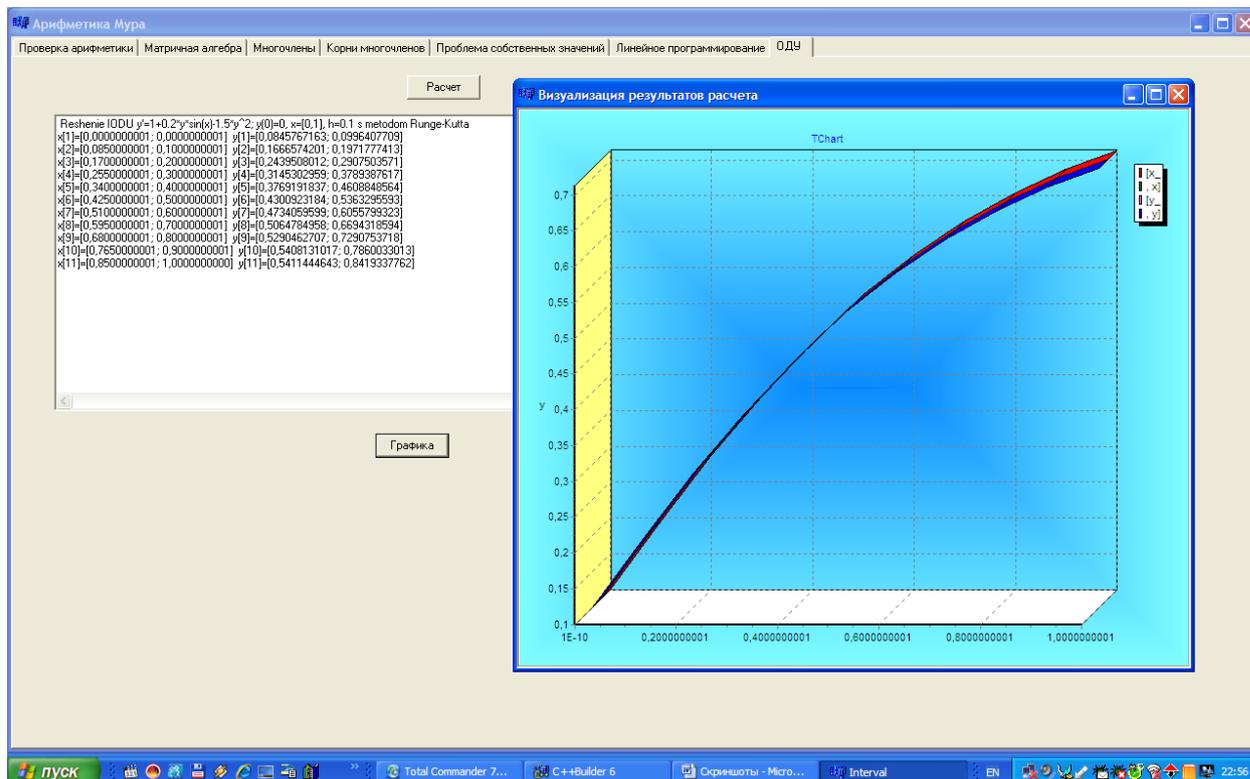
Открыть Новый Сохранить Закрыть



Пример 3.4., демонстрирующий вариант решения интервального варианта канонической задачи линейного программирования.



Пример 3.5., демонстрирующий вариант решения задачи Коши для ОДУ интервальным методом Рунге-Кутты.



3.3. Вычислительный эксперимент по исследованию модели межотраслевого баланса с недетерминированными данными

Статическая модель Леонтьева

В первой главе (п.1.3.1.) была рассмотрена *вещественная межотраслевая модель Леонтьева* и на основе принципа рациональной интервализации параметров был выведен её интервальный вариант, где недетерминированные данные с ограниченной амплитудой колебания рассматриваются как интервальные величины. Здесь она называется *статическая модель Леонтьева*, так как в данном параграфе, также рассмотрена и *динамическая модель Леонтьева*. Поскольку интервальный вариант динамической модели может быть построен аналогичным образом, для краткости он не был включён в первую главу. Однако вариант

динамической модели в ряде случаев более адекватно описывает макроэкономические пропорции, которые являются по существу целевыми для экономических систем, как с точки зрения обыкновенного потребителя, так и с точки зрения управляющих структур. С целью демонстрации возможностей, возникающих вследствие рассмотрения задачи в рамках интервального анализа, ниже на примере статической и динамической модели Леонтьева приводятся результаты численных экспериментов. Эксперименты были проведены на базе пакетов, описанных в первом и втором параграфах данной главы. Для большей ясности, кратко напомним известные из экономической литературы факты по проблеме макроэкономических пропорций, несколько изменяя обозначения и их смысл, не играющие принципиального значения, поскольку для проведения численных экспериментов мы выберем гипотезу линейности. Кроме того в экспериментах мы будем искать не только вектор валового выпуска при фиксированном векторе конечного потребления, но и наоборот, будем определять вектор конечного потребления, считая заданным вектор валового выпуска. Отметим, что в отличие от вещественного случая интервальный вариант даёт нижнюю и верхнюю границы искомых величин, что во многих случаях имеет весьма содержательные интерпретации. Однако интервальную матрицу в наших экспериментах мы будем вводить “искусственно”, возмущая коэффициенты технологической матрицы, тогда как интервальность в реальной ситуации вносится либо на основе экспертных оценок, либо отражает насущную степень недетерминированности данных.

Пример 3.6. Таблица 2 содержит данные баланса трех отраслей промышленности за некоторый период. Требуется найти объем валового выпуска продукции, если конечное потребление по отраслям увеличить соответственно до 60, 70 и 30.

Таблица 2.

№ п/п	Отрасль	Потребление			Конечный продукт	Валовой выпуск
		1	2	3		
1	Добыча и переработка углеводородов	5	35	20	40	100
2	Энергетика	10	10	20	60	100
3	Машиностроение	20	10	10	10	50

Решение. Выпишем векторы валового выпуска и конечного потребления и матрицу коэффициентов прямых затрат. Согласно формулам (3.4) и (3.6),

$$\bar{x} = \begin{pmatrix} 100 \\ 100 \\ 50 \end{pmatrix}, \bar{y} = \begin{pmatrix} 40 \\ 60 \\ 10 \end{pmatrix}, A = \begin{pmatrix} 0.05 & 0.35 & 0.40 \\ 0.10 & 0.10 & 0.40 \\ 0.20 & 0.10 & 0.20 \end{pmatrix}.$$

Матрица A удовлетворяет обоим критериям продуктивности. В случае заданного увеличения конечного потребления новый вектор конечного продукта будет иметь вид

$$\bar{y}_* = \begin{pmatrix} 60 \\ 70 \\ 30 \end{pmatrix}.$$

Требуется найти новый вектор валового выпуска \bar{x}_* , удовлетворяющий соотношениям баланса в предположении, что матрица A не изменяется. В таком случае компоненты x_1, x_2, x_3 неизвестного вектора \bar{x}_* находятся из системы уравнений, которая, согласно (4.3), имеет в данном случае вид

$$\begin{cases} x_1 = 0.05x_1 + 0.35x_2 + 0.4x_3 + 60, \\ x_2 = 0.1x_1 + 0.1x_2 + 0.4x_3 + 70, \\ x_3 = 0.2x_1 + 0.1x_2 + 0.2x_3 + 30. \end{cases}$$

В матричной форме эта система выглядит следующим образом:

$\bar{x}_* = A\bar{x}_* + \bar{y}_*$, или $(E - A)\bar{x}_* = \bar{y}_*$, где матрица $(E - A)$ имеет вид

$$(E - A) = \begin{pmatrix} 0.95 & -0.35 & -0.40 \\ -0.10 & 0.90 & -0.40 \\ -0.20 & 0.10 & 0.80 \end{pmatrix}.$$

Отсюда рассчитывается новый вектор \bar{x}^* как решение этого уравнения баланса:

$$\bar{x}^* = (E - A)^{-1} \bar{y}^*.$$

Найдем обратную матрицу (матрицу полных затрат) $(E - A)^{-1}$, с использованием формулы

$$A^{-1} = \begin{pmatrix} \frac{A_{11}}{\Delta} & \frac{A_{21}}{\Delta} & \dots & \frac{A_{n1}}{\Delta} \\ \frac{A_{12}}{\Delta} & \frac{A_{22}}{\Delta} & \dots & \frac{A_{n2}}{\Delta} \\ \dots & \dots & \dots & \dots \\ \frac{A_{1n}}{\Delta} & \frac{A_{2n}}{\Delta} & \dots & \frac{A_{nn}}{\Delta} \end{pmatrix}. \quad (3.8)$$

Определитель матрицы $(E - A)$, $\det(E - A) = 0.514 \neq 0$, так что обратная матрица и решение указанной системы уравнений существуют. Вычисление обратной матрицы дается с точностью до третьего знака:

$$(E - A)^{-1} = \frac{1}{0.514} \begin{pmatrix} 0.68 & 0.32 & 0.50 \\ 0.16 & 0.68 & 0.42 \\ 0.19 & 0.165 & 0.82 \end{pmatrix} = \begin{pmatrix} 1.323 & 0.623 & 0.973 \\ 0.311 & 1.323 & 0.817 \\ 0.370 & 0.321 & 1.595 \end{pmatrix}.$$

Заметим, что найденная обратная матрица удовлетворяет первому критерию продуктивности матрицы A .

Теперь можно вычислить вектор валового выпуска \bar{x}^* :

$$\bar{x}^* = \begin{pmatrix} 1.323 & 0.623 & 0.973 \\ 0.311 & 1.323 & 0.817 \\ 0.370 & 0.321 & 1.595 \end{pmatrix} \begin{pmatrix} 60 \\ 70 \\ 30 \end{pmatrix} = \begin{pmatrix} 152.2 \\ 135.8 \\ 92.5 \end{pmatrix}.$$

Таким образом, для того чтобы обеспечить заданное увеличение компонент вектора конечного продукта, необходимо увеличить соответствующие валовые выпуски: добычу и переработку углеводов на 52.2%, уровень энергетики – на 35.8% и выпуск машиностроения – на 85% по сравнению с исходными величинами, указанными в табл. 2.

Теперь предположим, что элементы матрицы A и вектора конечного потребления y заданы неточно, а их амплитуда колебания равна $\delta=10^{-5}$. Тогда соответствующая интервальная матрица будет иметь вид:

$$\hat{A} = \begin{bmatrix} [0.0499, 0.0501] [0.3499, 0.3501] [0.3999, 0.4001] \\ [0.0999, 0.1001] [0.0999, 0.1001] [0.3999, 0.4001] \\ [0.1999, 0.2001] [0.0999, 0.1001] [0.1999, 0.2001] \end{bmatrix}.$$

Обозначая через $K=E-A$, получим

$$\hat{E} = \begin{bmatrix} [0.9499, 0.9501] [-0.3501, -0.3499] [-0.4001, -0.3999] \\ [-0.1001, -0.0999] [0.8999, 0.9001] [-0.4001, -0.3999] \\ [-0.2001, -0.1999] [-0.1001, -0.0999] [0.7999, 0.8001] \end{bmatrix}.$$

Поскольку в интервальном анализе операции сложения и вычитания, а также умножения и вычитания не являются взаимно обратными, то задача обращения интервальных матриц ставится следующим образом [115]:

Для некоторой интервальной матрицы A необходимо определить матрицу A^{-1} , такую, что

$$\hat{A}^{-1} \supseteq B = [b_{ij}, \bar{b}_{ij}],$$

$$\underline{b}_{ij} = \inf \{ \Pr_{ij} B \mid B \in \tilde{B} = \{ B \mid \forall A \in A, \exists \hat{A}: BA = AB = E \} \}, \quad (3.9)$$

$$\bar{b}_{ij} = \sup \{ \Pr_{ij} B \mid B \in \tilde{B} = \{ B \mid \forall A \in A, \exists \hat{A}: BA = AB = E \} \}.$$

При этом $\hat{A} \hat{A}^{-1} \neq \hat{A}$, то есть обозначение A^{-1} следует понимать в смысле соотношения (3.9).

Применяя, соответствующий алгоритм пакета, вначале определим

$$\hat{E}^{-1} = \begin{bmatrix} [1.3228, 1.3231] [0.6225, 0.6227] [0.9726, 0.9729] \\ [0.3112, 0.3114] [1.3229, 1.3231] [0.8170, 0.8173] \\ [0.3696, 0.3697] [0.3209, 0.3211] [1.5952, 1.5955] \end{bmatrix},$$

и затем интервальный вектор валового продукта

$$\tilde{\delta} = \begin{bmatrix} [152.1289, 152.1512] \\ [135.7883, 135.8070] \\ [92.5010, 92.5185] \end{bmatrix}.$$

Для данного вектора \mathbf{x} имеем $\|\tilde{\delta}\| = \max \omega(\mathbf{x}_i) = \bar{x}_i - \underline{x}_i = 0.0223$.

Простейшие расчеты показывают - для того чтобы обеспечить заданное увеличение компонент вектора конечного продукта в указанном интервале, необходимо увеличить соответствующие валовые выпуски: добычу и переработку углеводородов в интервале [49.3%, 54.5%], уровень энергетики в интервале [36.8%, 39.1%], выпуск машиностроения в интервале [83.6%, 87.7%] по сравнению с исходными величинами, указанными в табл. 1.

Задача определения вектора конечного продукта \mathbf{y} может быть решена выполнением операции матричной алгебры пакета: $\mathbf{y} = \mathbf{K}\mathbf{x}$. Для выбранных данных примера 3.1 получены следующие результаты:

$$\mathbf{y} = \begin{bmatrix} [194.5353, 194.6045] \\ [259.9429, 260.0018] \\ [196.7454, 196.8006] \end{bmatrix}$$

В случае большего количества отраслей производства обычно наблюдается “эффект раскрутки”, и на практике применяют интервальные итерационные методы при определении вектора валового выпуска \mathbf{x} [7,115].

Следующие скриншоты демонстрируют процесс получения приведённых результатов в рамках вышеописанного пакета.

Пример 3.7., решения статистической модели Леонтьева(пояснения вынесены в заголовки скриншотов):

Арифметика Мура

Проверка арифметики | Матричная алгебра | Многочлены | Корни многочленов | Проблема собственных значений | Линейное программирование | ОДУ

Интервальный вариант статической модели Леонтьева

Ввод единичной матрицы I

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	1	1	0	0	0	0
1	0	0	1	1	0	0
2	0	0	0	0	1	1

Ввод матрицы A | Детерминант матрицы A | Метод Гауса

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0.0499	0.0501	0.3499	0.3501	0.3999	0.4001
1	0.0499	0.1001	0.0999	0.1001	0.3999	0.4001
2	0.1999	0.2001	0.0999	0.1001	0.1999	0.2001

Кнопки: A+B, I-A, A*B, K=inv(I-A), K*D'

Вычисление вектора валового выпуска x

Массив C	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0.9499	0.9501	-0.3501	-0.3499	-0.4001	-0.3999
1	-0.1001	-0.0499	0.8999	0.9001	-0.4001	-0.3999
2	-0.2001	-0.1999	-0.1001	-0.0999	0.7999	0.8001

0:20

Арифметика Мура

Проверка арифметики | Матричная алгебра | Многочлены | Корни многочленов | Проблема собственных значений | Линейное программирование | ОДУ

Интервальный вариант статической модели Леонтьева

Ввод единичной матрицы I

	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	1	1	0	0	0	0
1	0	0	1	1	0	0
2	0	0	0	0	1	1

Ввод матрицы A | Детерминант матрицы A | Метод Гауса

	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0.0499	0.0501	0.3499	0.3501	0.3999	0.4001
1	0.0499	0.1001	0.0999	0.1001	0.3999	0.4001
2	0.1999	0.2001	0.0999	0.1001	0.1999	0.2001

Кнопки: A+B, I-A, A*B, K=inv(I-A), K*D'

Вычисление вектора валового выпуска x

	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	1.3228	0.3231	0.6225	0.6227	0.9726	0.9729
1	0.3112	0.3114	1.3229	1.3231	0.817	0.8173
2	0.3696	0.3697	0.3209	0.3211	1.5952	1.5955

0:38

Арифметика Мура

Проверка арифметики | Матричная алгебра | Многочлены | Корни многочленов | Проблема собственных значений | Линейное программирование | ОДУ

Интервальный вариант статической модели Леонтьева

Ввод единичной матрицы I

Массы	[inf]	[sup]	[inf]	[sup]	[inf]	[sup]
0	1,3228	0,3231	0,6225	0,6227	0,9726	0,9729
1	0,3112	0,3114	1,3229	1,3231	0,817	0,8173
2	0,3696	0,3697	0,3209	0,3211	1,5952	1,5955

Ввод матрицы A | Детерминант матрицы A | Метод Гауса

	[inf]	[sup]	[inf]	[sup]	[inf]	[sup]
0	59,9995	60,0001	0	0	0	0
1	69,9995	70,0001	0	0	0	0
2	29,9995	30,0001	0	0	0	0

Вычисление вектора валового выпуска x

	[inf]	[sup]	[inf]	[sup]	[inf]	[sup]
0	152,1289	152,1512	0	0	0	0
1	135,7883	135,807	0	0	0	0
2	92,501	92,5185	0	0	0	0

Windows taskbar: ПУСК, C++Builder 6, ПРИМЕР_Модель Ле..., Interval, EN, 0:48

Арифметика Мура

Проверка арифметики | Матричная алгебра | Многочлены | Корни многочленов | Проблема собственных значений | Линейное программирование | ОДУ

Интервальный вариант статической модели Леонтьева

Введение исходных параметров

Ввод единичной матрицы I

	[a ₁₁]	a ₁₁	[a ₁₂]	a ₁₂	[a ₁₃]	a ₁₃
0	1,3228	0,3231	0,6225	0,6227	0,9726	0,9729
1	0,3112	0,3114	1,3229	1,3231	0,817	0,8173
2	0,3696	0,3697	0,3209	0,3211	1,5952	1,5955

Ввод матрицы A

	[a ₁₁]	a ₁₁	[a ₁₂]	a ₁₂	[a ₁₃]	a ₁₃
0	152,1289	152,1512	0	0	0	0
1	135,7883	135,807	0	0	0	0
2	92,501	92,5185	0	0	0	0

Детерминант матрицы A

Метод Гаусса

Вычисление вектора валового выпуска x

	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	194,5353	194,6045	0	0	0	0
1	259,9429	260,0018	0	0	0	0
2	196,7454	196,8006	0	0	0	0

пуск

ГЛАВА 3 ...

ГЛАВА 3 П...

ABBY Lin...

MATLAB

Total Com...

C++Builder 6

Interval

EN

14:35

Арифметика Мура

Проверка арифметики | Матричная алгебра | Многочлены | Корни многочленов | Проблема собственных значений | Линейное программирование | ОДУ

Интервальный вариант статической модели Леонтьева

Введение исходных параметров

Ввод единичной матрицы I

	[a ₁₁]	a ₁₁	[a ₁₂]	a ₁₂	[a ₁₃]	a ₁₃
0	1,3228	0,3231	0,6225	0,6227	0,9726	0,9729
1	0,3112	0,3114	1,3229	1,3231	0,817	0,8173
2	0,3696	0,3697	0,3209	0,3211	1,5952	1,5955

Ввод матрицы A

	[a ₁₁]	a ₁₁	[a ₁₂]	a ₁₂	[a ₁₃]	a ₁₃
0	152,1289	152,1512	0	0	0	0
1	135,7883	135,807	0	0	0	0
2	92,501	92,5185	0	0	0	0

Детерминант матрицы A

Метод Гаусса

Определения вектора конечного продукта y

	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	194,5353	194,6045	0	0	0	0
1	259,9429	260,0018	0	0	0	0
2	196,7454	196,8006	0	0	0	0

пуск

ГЛАВА 3 ...

ГЛАВА 3 П...

ABBY Lin...

MATLAB

Total Com...

C++Builder 6

Interval

EN

14:40

Динамическая модель Леонтьева

Так как мы рассмотрели продуктивную модель межотраслевого баланса безотносительно ко времени, т.е. все ее компоненты полагались

осредненными за некоторый временной интервал и «одномоментными». В реальности продукт, предназначенный для внутреннего и конечного потребления в период t , определяется не текущим выпуском, а выпуском в последующий период $t+1$. Эта задержка производства обусловлена многими факторами, в частности инерцией планирования и перенастройки, мобилизацией внутренних ресурсов и изменением транспорта сырья и т.д.

С учетом этого система уравнений баланса, в предположении о постоянстве доли внутреннего потребления каждой отраслью, будет иметь следующий вид

$$\begin{cases} x_1(t+1) = a_{11}x_1(t) + \dots + a_{1n}x_n(t) + y_1(t), \\ x_2(t+1) = a_{21}x_1(t) + \dots + a_{2n}x_n(t) + y_2(t), \\ \dots \\ x_n(t+1) = a_{n1}x_1(t) + \dots + a_{nn}x_n(t) + y_n(t). \end{cases} \quad (3.10)$$

Соотношения (3.10) составляют систему линейных разностных уравнений первого порядка с постоянными коэффициентами a_{ij} .

Как и прежде, введем вектор валового выпуска $\bar{x}(t)$, матрицу прямых затрат $A = \|a_{ij}\|$ и вектор конечного потребления $\bar{y}(t)$. Тогда систему (3.10) можно переписать в матричной форме:

$$\bar{x}(t+1) = A\bar{x}(t) + \bar{y}(t). \quad (3.11)$$

Теперь задача формулируется следующим образом: при заданном векторе конечного потребления $\bar{y}(t)$ и матрице A определить динамику (изменение во времени) вектора валового выпуска $\bar{x}(t)$.

Одна из основных задач прогноза с использованием динамической модели Леонтьева такова: *известны динамика вектора конечного потребления $\bar{y}(t)$ и вектор валового выпуска \bar{x}_0 в начальный момент времени $t=0$; требуется рассчитать вектор валового выпуска на момент времени $t \geq 1$. Эту задачу можно решить при помощи формулы:*

$$\bar{x}(t) = A^t \bar{x}(0) + \sum_{k=0}^{t-1} A^{t-k-1} \bar{y}(k). \quad (3.12)$$

Пример 3.8. Обратимся к данным таблице П.1.

Пусть известны продуктивная матрица A , а также заданные в момент времени $t=0$ векторы валового выпуска $\bar{x}(0)$ и $\bar{y}(0)$, указанные в этой таблице:

$$\bar{x}(0) = \begin{pmatrix} 100 \\ 100 \\ 50 \end{pmatrix}, \bar{y}(0) = \begin{pmatrix} 40 \\ 60 \\ 10 \end{pmatrix}, A = \begin{pmatrix} 0.05 & 0.35 & 0.40 \\ 0.10 & 0.10 & 0.40 \\ 0.20 & 0.10 & 0.20 \end{pmatrix}.$$

Требуется рассчитать вектор валового выпуска на момент времени $t=2$, если все компоненты вектора конечного потребления \bar{y} увеличиваются на 30% за каждый период.

Решение. Вектор конечного потребления, согласно условию задачи, имеет вид

$$\bar{y}(t) = \bar{y}(0)(1.3)^t.$$

Применяя формулу (4.11-33), получаем $\bar{x}(2) = A^2 \bar{x}(0) + A \bar{y}(0) + \bar{y}(1)$.

Теперь нужно вычислить матрицу A^2 изменяя состояния и вектор $\bar{y}(1)$ и подставить их в это уравнение. Выполняя указанные действия, получим решение поставленной задачи:

$$\bar{x}(2) = \begin{pmatrix} 0.1175 & 0.0925 & 0.24 \\ 0.095 & 0.085 & 0.16 \\ 0.06 & 0.1 & 0.16 \end{pmatrix} \begin{pmatrix} 100 \\ 100 \\ 50 \end{pmatrix} + \begin{pmatrix} 0.05 & 0.35 & 0.4 \\ 0.1 & 0.1 & 0.4 \\ 0.2 & 0.1 & 0.2 \end{pmatrix} \begin{pmatrix} 40 \\ 60 \\ 10 \end{pmatrix} + \begin{pmatrix} 52 \\ 78 \\ 13 \end{pmatrix} = \begin{pmatrix} 33 \\ 26 \\ 24 \end{pmatrix} + \begin{pmatrix} 27 \\ 14 \\ 16 \end{pmatrix} + \begin{pmatrix} 52 \\ 78 \\ 13 \end{pmatrix} = \begin{pmatrix} 112 \\ 118 \\ 53 \end{pmatrix}.$$

Таким образом, при указанном темпе роста продукта конечного потребления необходимо через два временных цикла увеличить компоненты валового выпуска соответственно на 12,18 и 6% по сравнению с исходными величинами на начальный момент времени.

Аналогично случаю со статической моделью рассматривались данные, соответствующие динамической модели, когда параметры матрицы A и

векторов x , y возмущались “искусственно”. Для краткости эти результаты приведены на скриншотах.

Пример 3.9., решения динамической модели Леонтьева(пояснения вынесены в заголовки скриншотов):

Интервальный вариант динамической модели Леонтьева

Ввод единичной матрицы A

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0,0499	0,0501	0,3499	0,3501	0,3999	0,4001
1	0,0499	0,1001	0,0999	0,1001	0,3999	0,4001
2	0,1999	0,2001	0,0999	0,1001	0,1999	0,2001

Ввод матрицы A Детерминант матрицы A

Массив A

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0,0499	0,0501	0,3499	0,3501	0,3999	0,4001
1	0,0499	0,1001	0,0999	0,1001	0,3999	0,4001
2	0,1999	0,2001	0,0999	0,1001	0,1999	0,2001

Массив C

Массив C	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	0,09389003	0,11761503	0,09236503	0,09263503	0,23982003	0,24018003
1	0,08741503	0,09509503	0,06739003	0,08511503	0,13984503	0,16016003
2	0,05492003	0,06008503	0,09989503	0,10010503	0,15985003	0,16015003

Вычисление вектора валового выпуска x при $t=2$

Интервальный вариант динамической модели Леонтьева

Ввод единичной матрицы A

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	99,9999	100,0000	0	0	0	0
1	99,9999	100,0000	0	0	0	0
2	49,9999	50,0001	0	0	0	0

Ввод матрицы A Детерминант матрицы A

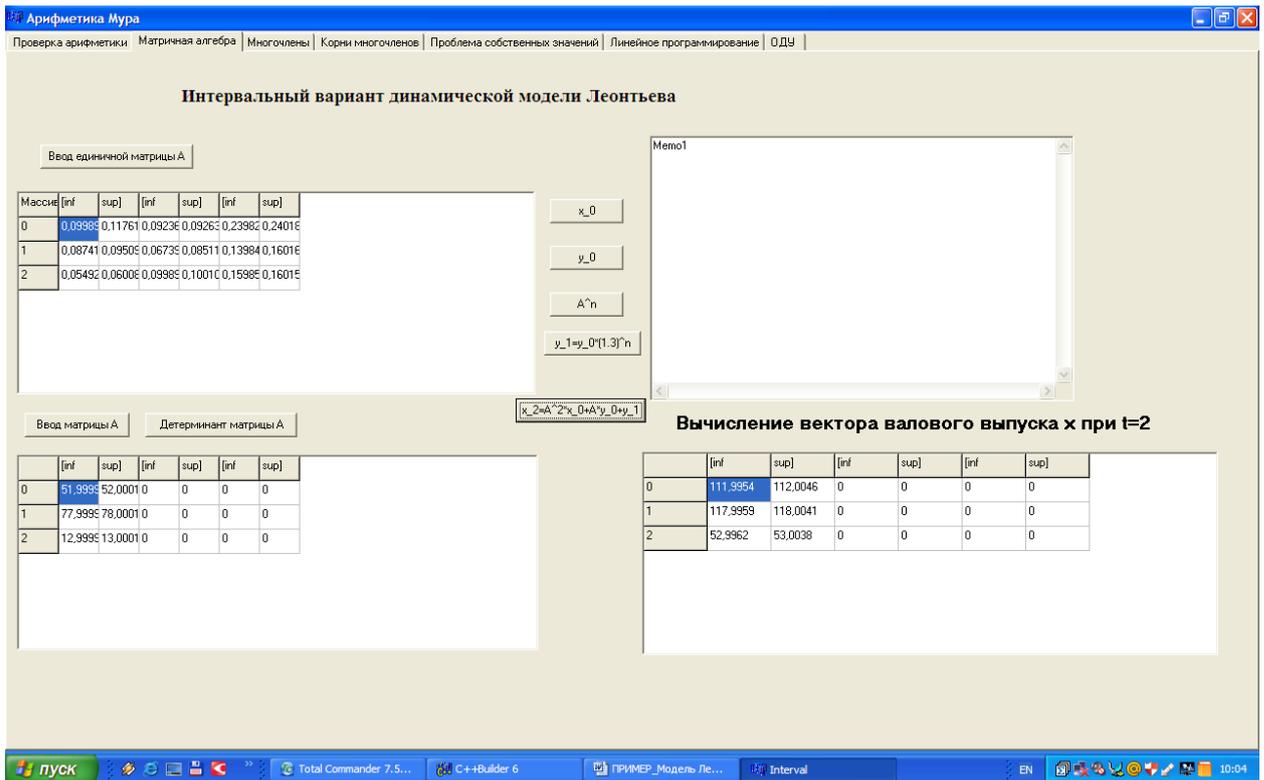
Массив A

Массив	[inf]	sup]	[inf]	sup]	[inf]	sup]
0	99,9999	40,0001	0	0	0	0
1	99,9999	60,0001	0	0	0	0
2	9,9999	10,0001	0	0	0	0

Массив C

Массив C	[inf]	sup]	[inf]	sup]	[inf]	sup]
0						
1						
2						

Вычисление вектора валового выпуска x



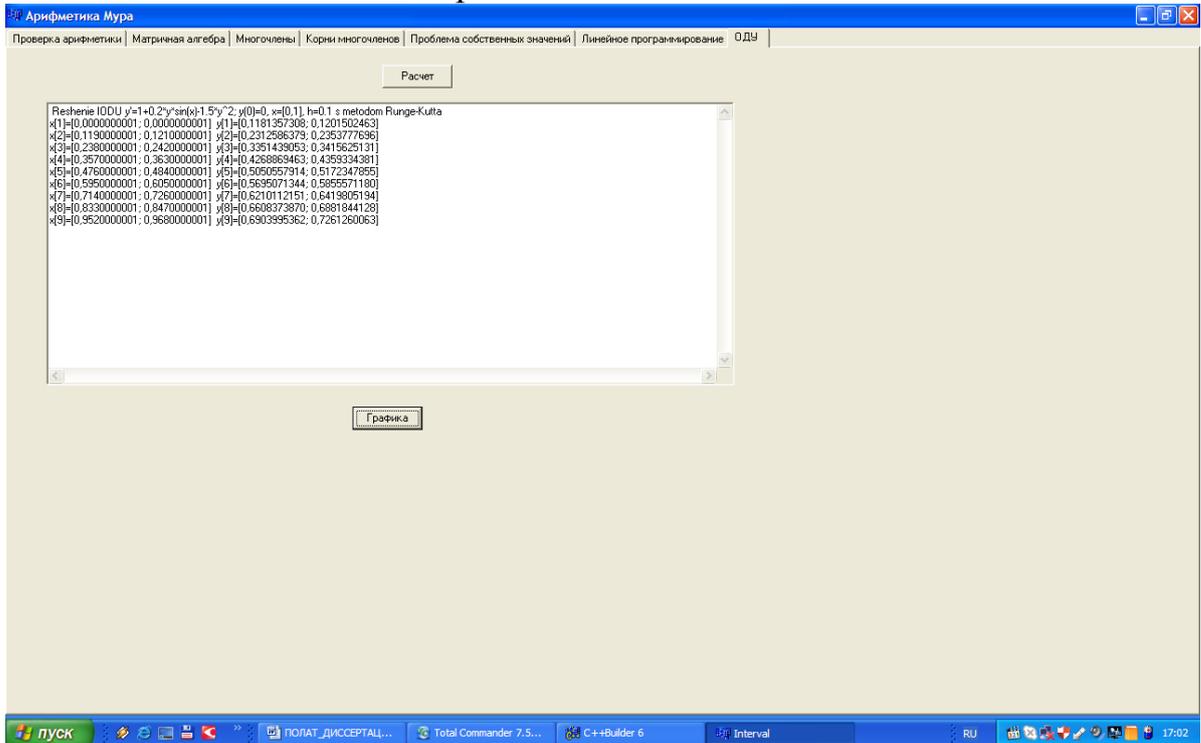
В указанных расчётах по моделям Леонтьева величина δ вводилась искусственно. В реальных задачах при большем количестве отраслей производства n , интервалы a_{ij}, x_i, y_i возникают как следствие учёта недетерминированных данных определяемых как спецификой взаимосвязей в процессе производства и потребления, так и желанием исследователя, который желает определить интервалы колебания компонент матрицы A , а также векторов x и y .

3.4. Вычислительный эксперимент по решению эколого-экономических задач

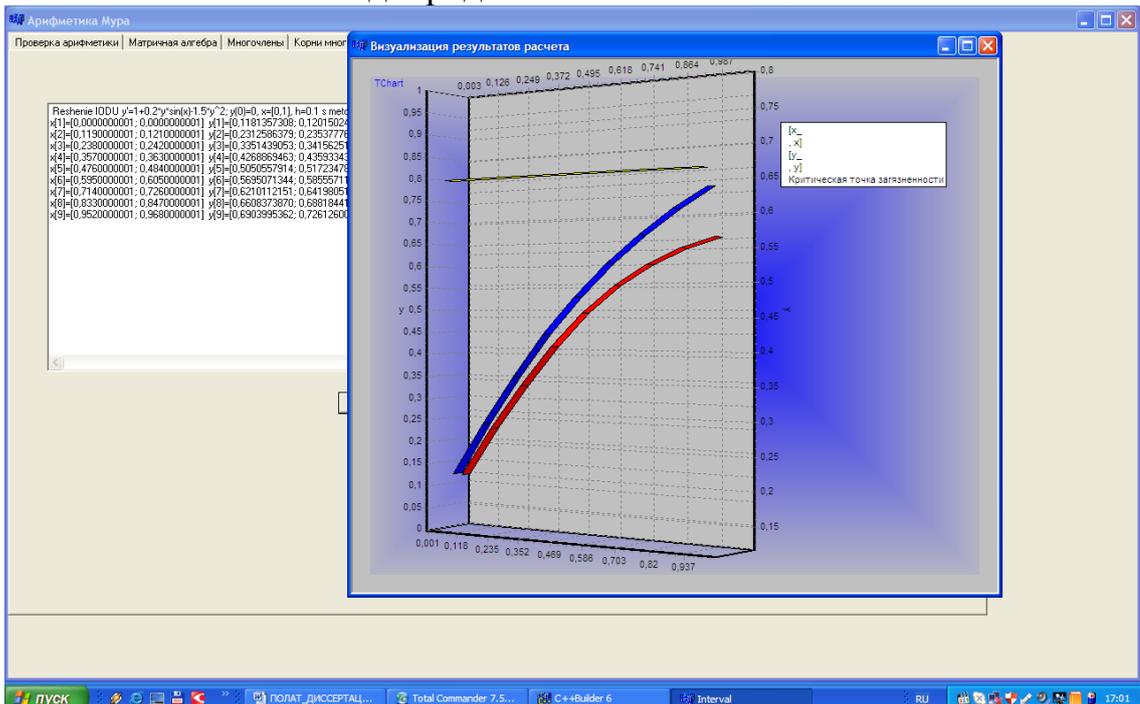
Наряду с указанными, были проведены численные эксперименты по применению интервального варианта модели назначения штрафов за загрязнение окружающей среды и исследование задачи распределения инвестиций по отраслям хозяйствования интервальными методами в рамках пакета. При этом ограниченное колебание параметров, точнее искусственное введение интервальности, потенциально недетерминированных величин достигалось использованием метода ограниченных возмущений на основе

введения параметра типа δ . В каждом случае полученное интервальное решение содержит соответствующее множество решений вещественных задач, что подтверждает теоретические результаты.

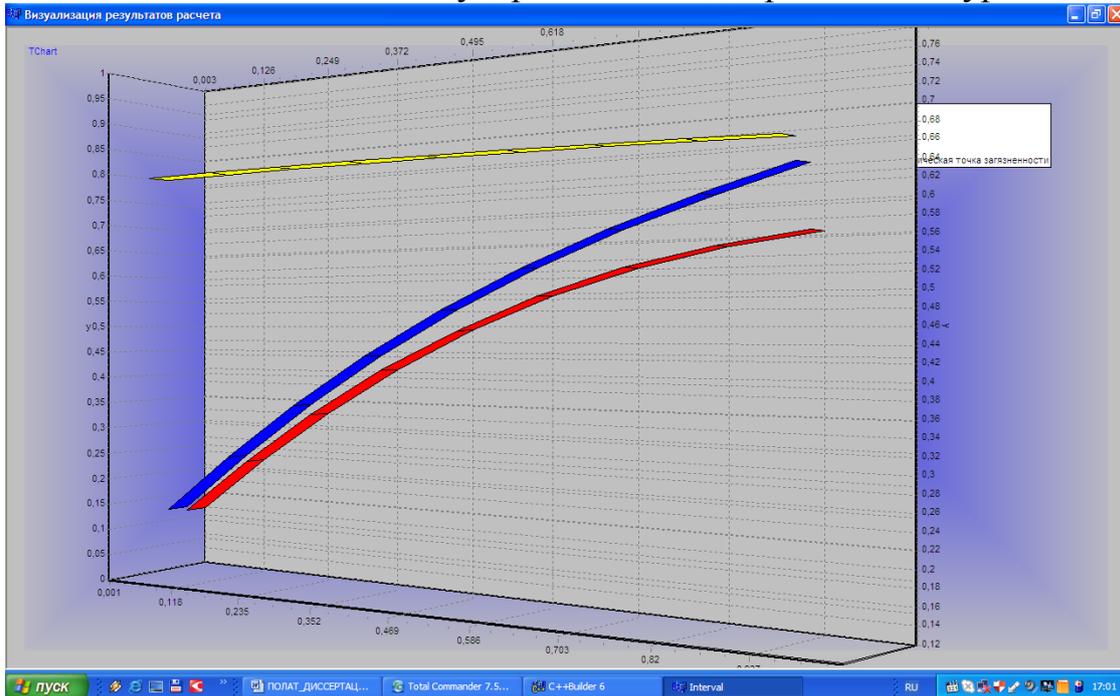
Пример 3.10. Интервальные данные по задаче из 2.1 распространения однородного экотоксиканта



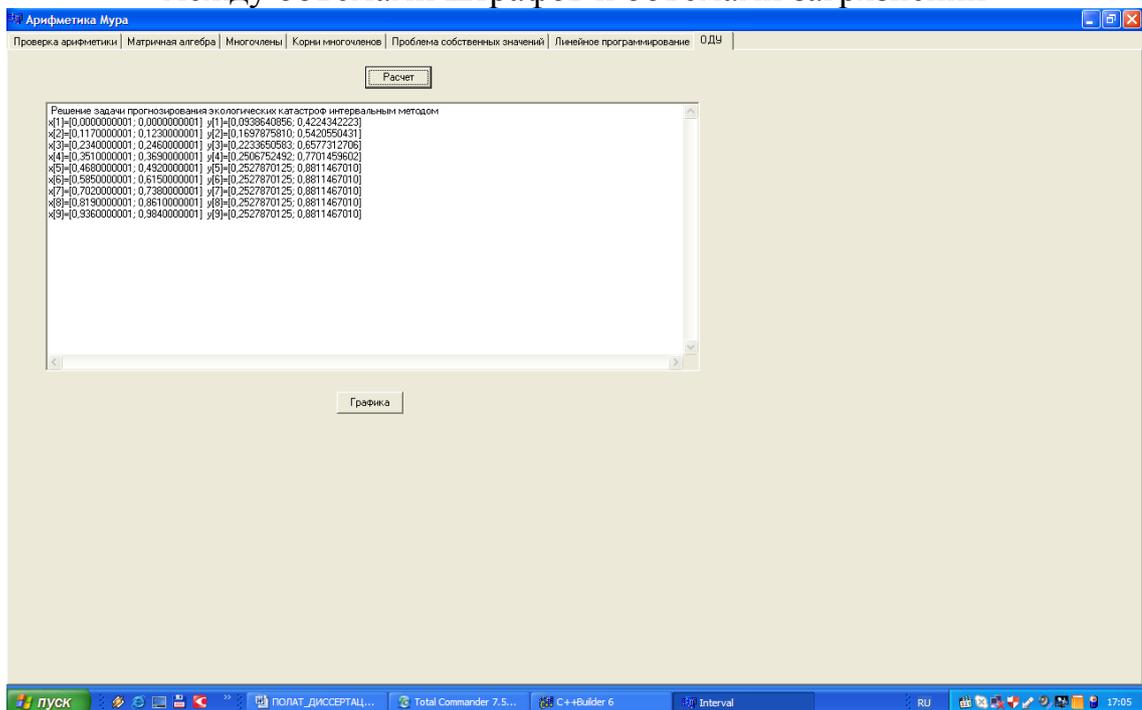
Пример 3.11. Графическая интерпретация динамики распространения однородного экотоксиканта



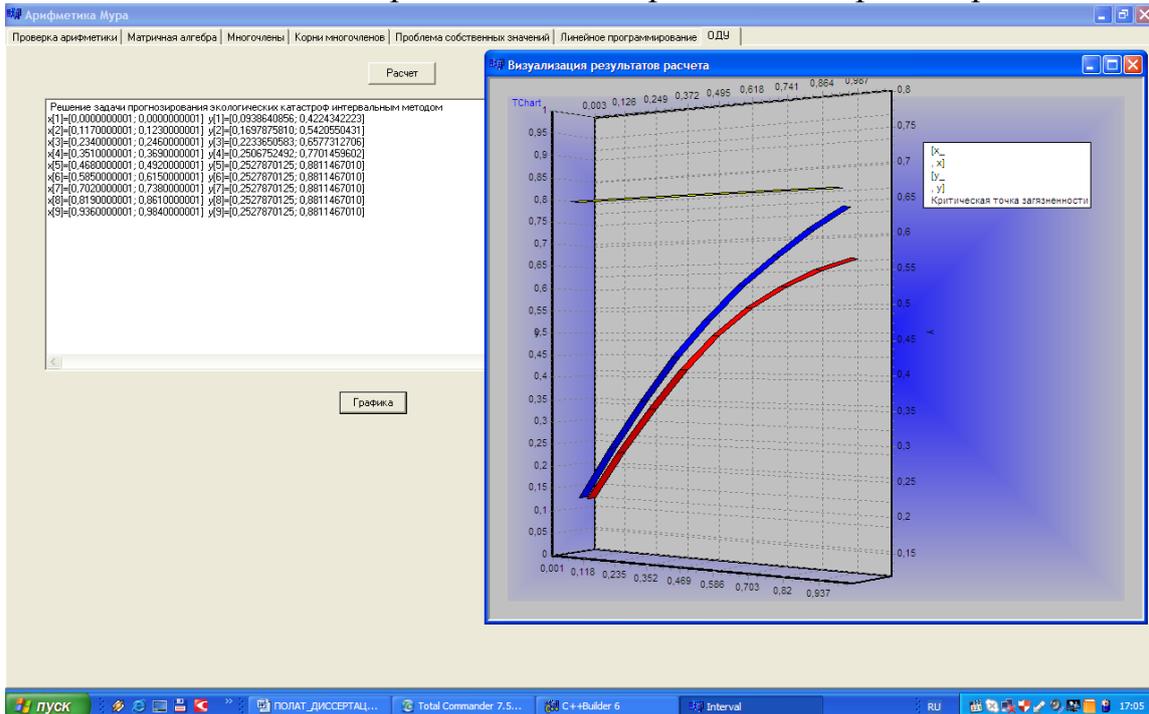
Пример 3.12. Графическая интерпретация стабилизации объёма однородного экотоксиканта с удержанием на докритическом уровне



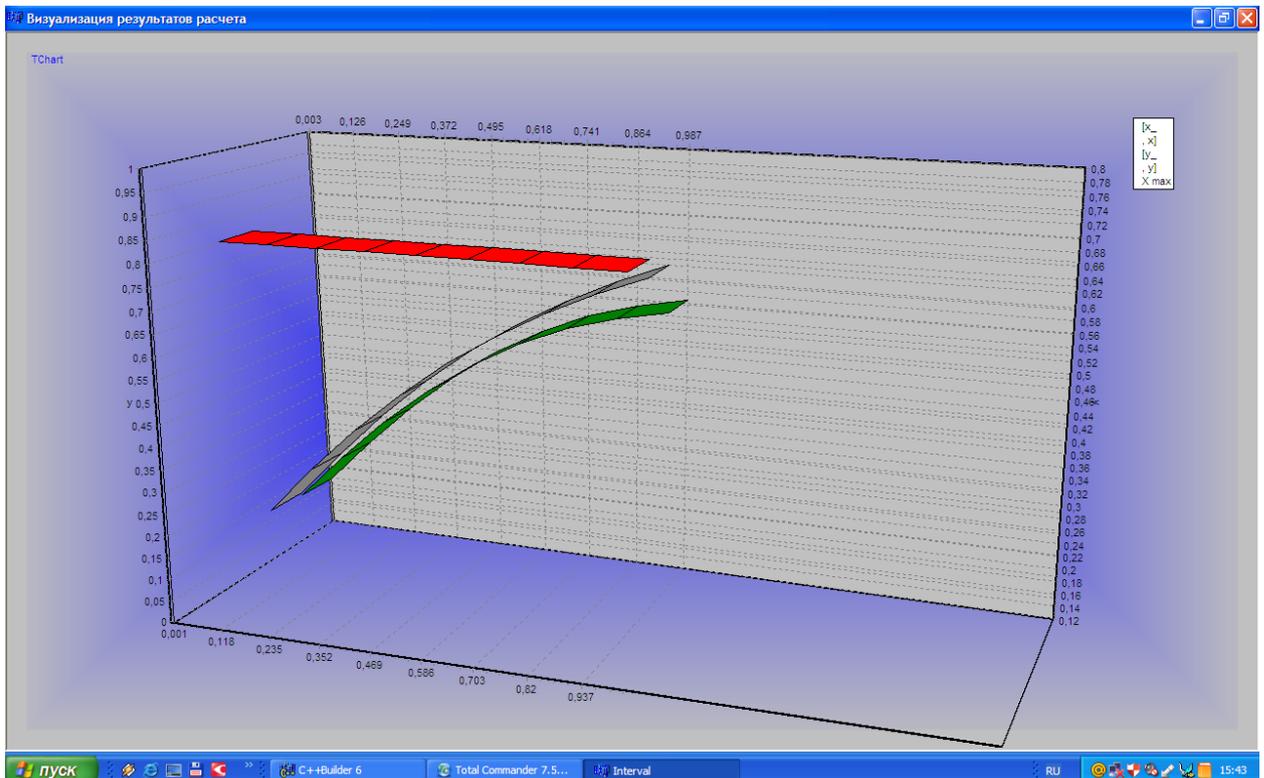
Пример 3.13. Интервальные данные по задаче из 2.2 удержания равновесия между объёмами штрафов и объёмами загрязнений

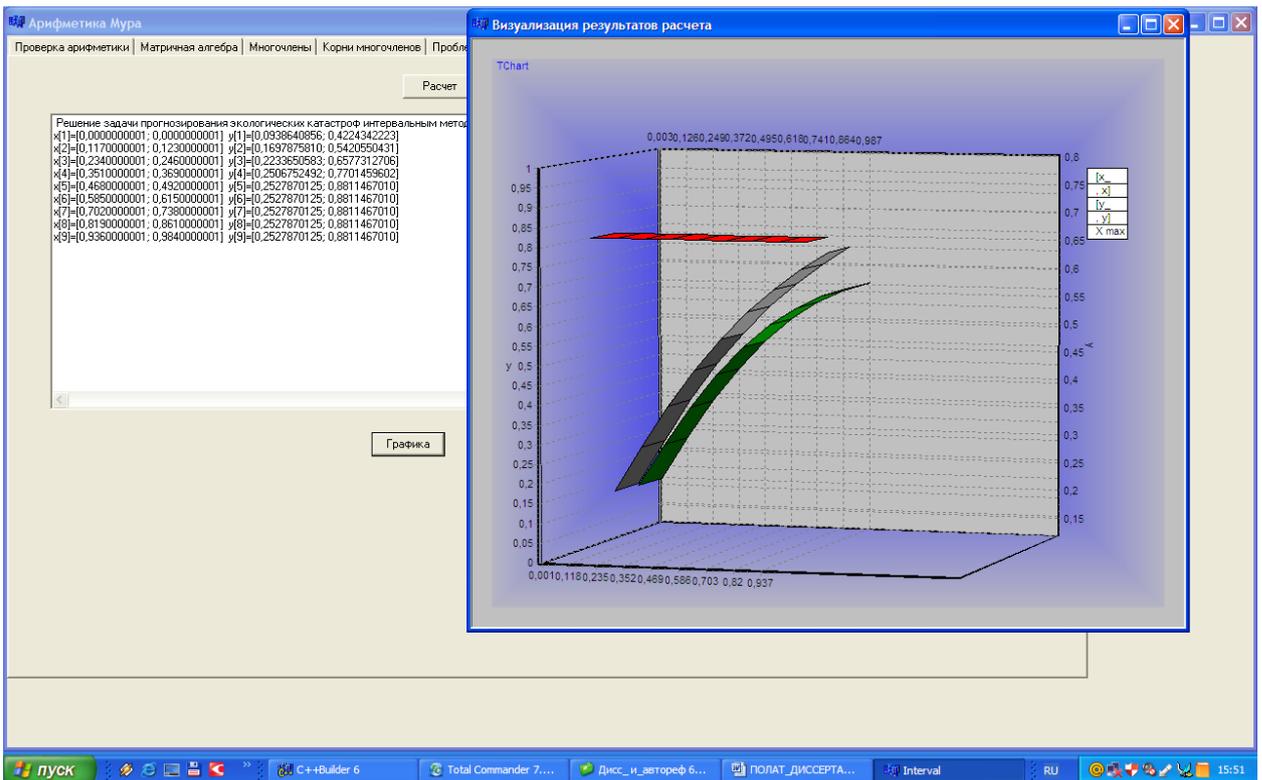


Пример 3.14. Динамика удержания паритета между объемами штрафов и объемами загрязнений для короткого интервала времени

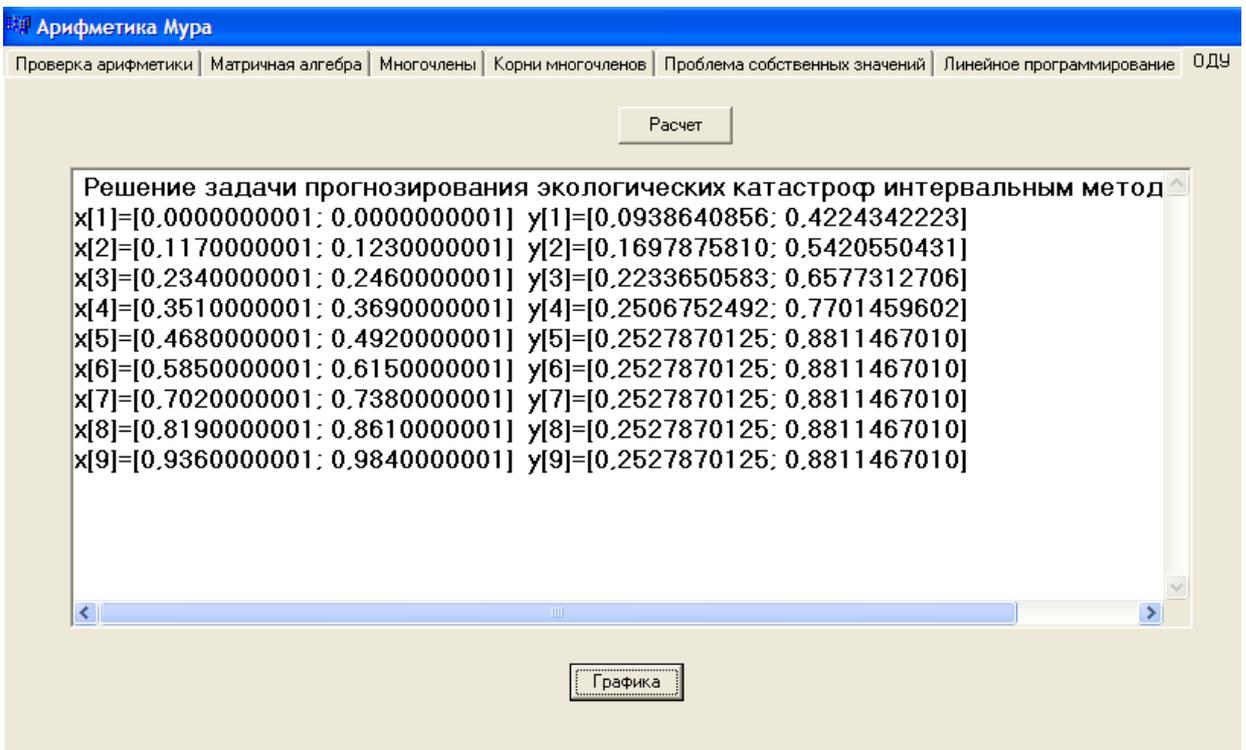


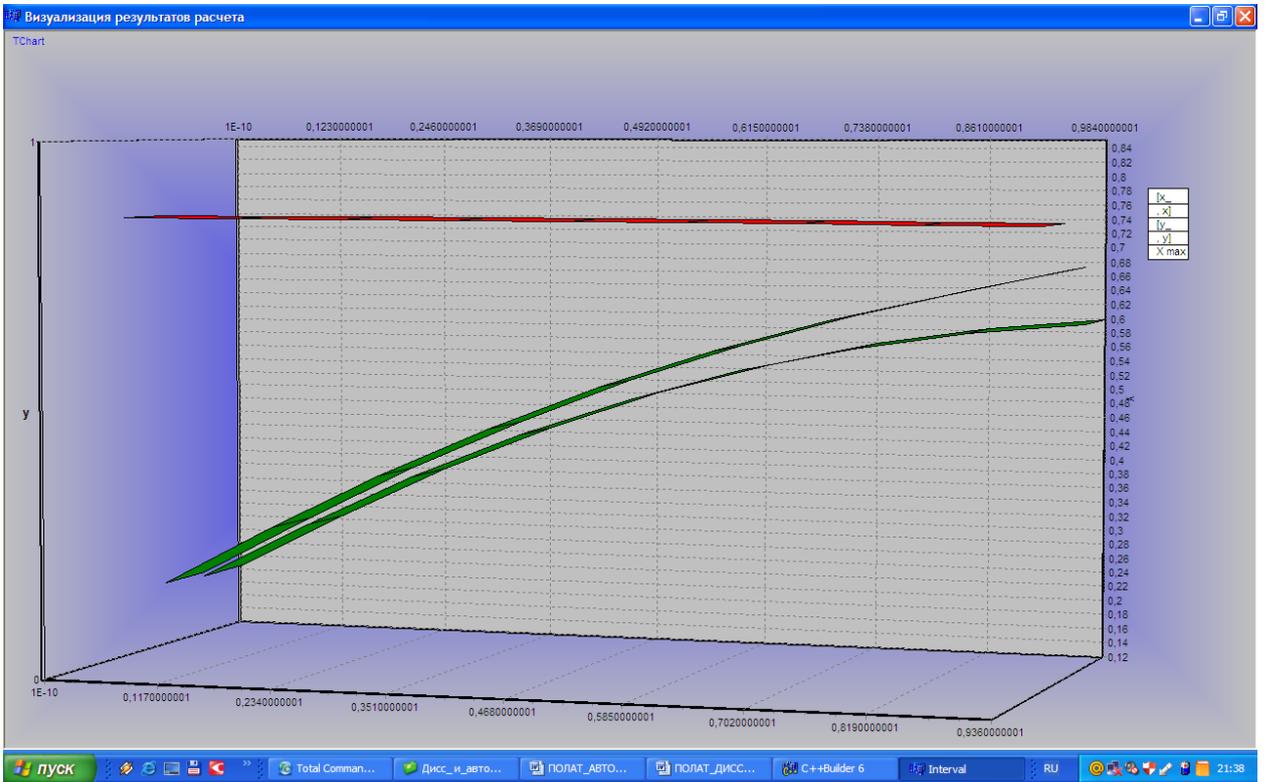
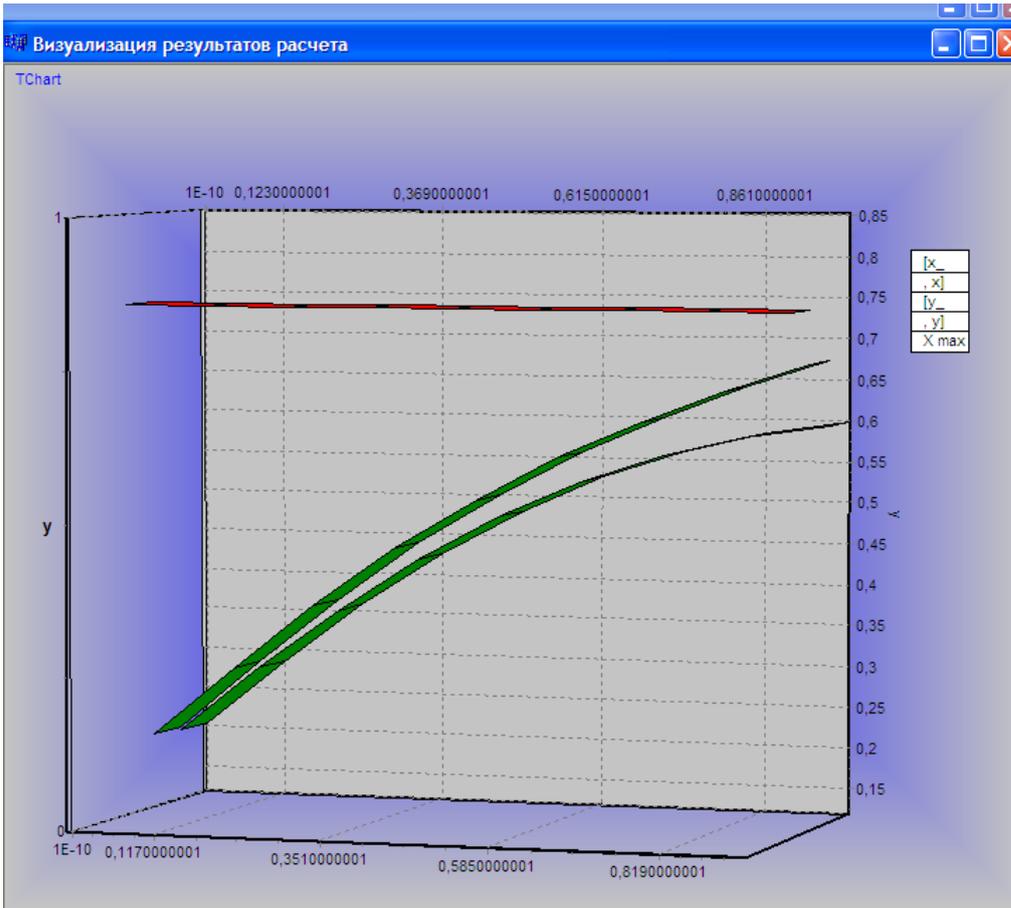
Пример 3.15. Динамика удержания паритета между объемами штрафов и объемами загрязнений для длительного интервала времени





Пример 3.16. Решение задачи прогнозирования экологических катастроф интервальным методом





Выводы по численным экспериментам:

Во всех случаях интервальное решение на модельных задачах содержит искомое точное решение. Соответствующие управления позволяют выводить уровень загрязненности среды однородным экотоксикантом до приемлемого уровня(задача о распространении однородного экотоксиканта) либо сохранять паритет в хозяйственной инфраструктуре региона(задача о назначении штрафов).