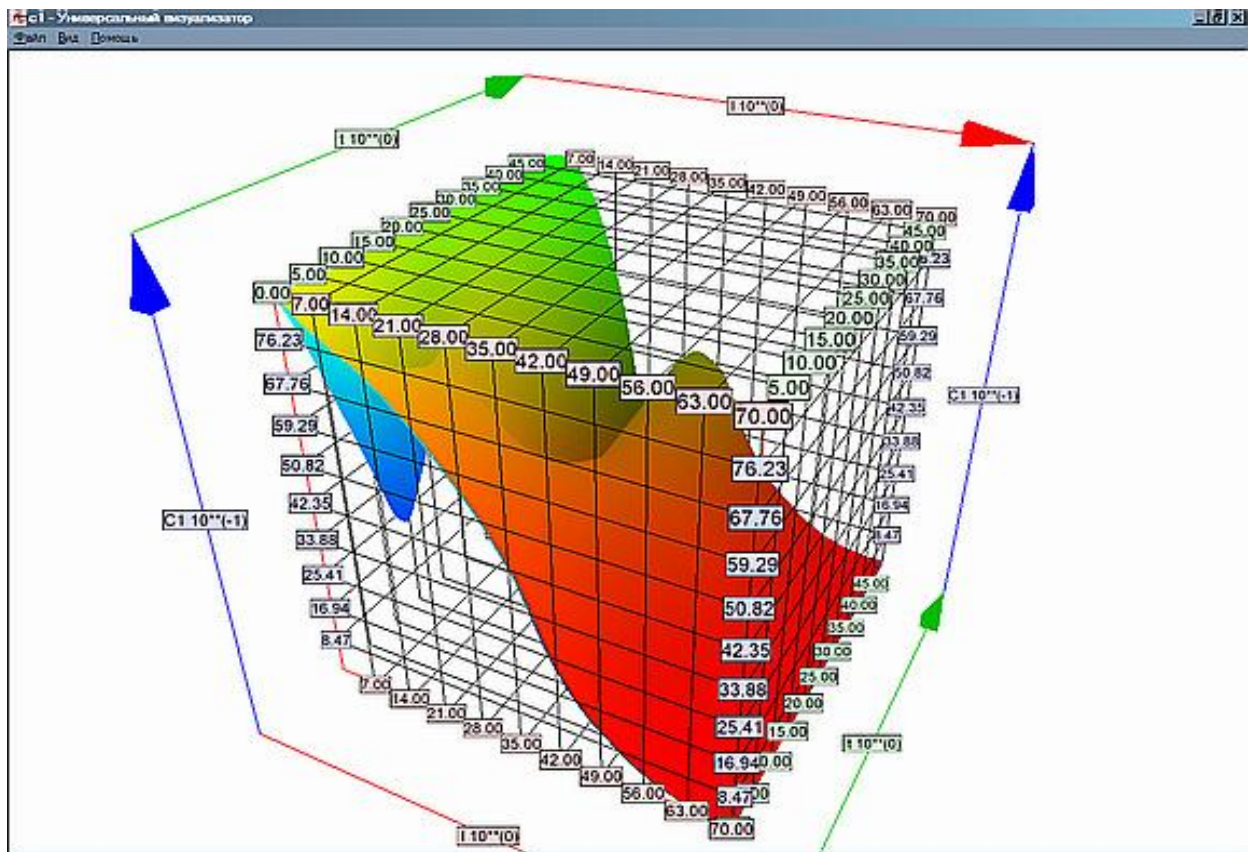


**В.Ф. Барабанов С.Л. Подвальный**

**О.С. Плахотнюк**

**МНОВОВАРИАНТНОЕ МОДЕЛИРОВАНИЕ  
ДИНАМИЧЕСКИХ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА  
ДЛЯ УПРАВЛЕНИЯ В ЭКСТРЕМАЛЬНЫХ СИТУАЦИЯХ**



**Воронеж 2007**

ГОУВПО “Воронежский государственный технический  
университет”

В.Ф. Барабанов С.Л. Подвальный

О.С. Плахотнюк

МНОГОВАРИАНТНОЕ МОДЕЛИРОВАНИЕ  
ДИНАМИЧЕСКИХ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА ДЛЯ  
УПРАВЛЕНИЯ В ЭКСТРЕМАЛЬНЫХ СИТУАЦИЯХ

Воронеж 2007

УДК 681.3

В.Ф. Барабанов. Многовариантное моделирование динамических систем эволюционного типа для управления в экстремальных ситуациях: монография / В.Ф. Барабанов, С.Л. Подвальный, О.С. Плахотнюк. Воронеж: ГОУВПО “Воронежский государственный технический университет”, 2007. 152 с.

ISBN 978-5-7731-0174-1

В монографии рассмотрены вопросы разработки методов многовариантного моделирования динамических режимов систем эволюционного типа. Разработаны алгоритмы численного решения нелинейных систем, представленных уравнениями в частных производных, обеспечивающие повышенную счетную устойчивость и характеризующиеся минимальными вычислительными затратами и требованиями к машинной памяти. Также исследован вопрос обнаружения нештатных ситуаций и формирования управляющих воздействий, выводящих систему из экстремального режима.

Издание предназначено научным сотрудникам и специалистам, занимающихся исследованием динамических систем эволюционного типа.

Табл. 2 . Ил. 26 . Библиогр.: 66 назв.

Научный редактор д-р техн. наук, проф. С.Л. Подвальный

Рецензенты: кафедра математического моделирования информационных и технологических систем Воронежской государственной технологической академии (зав. кафедрой д.т.н., проф.

Г.В. Абрамов);

д-р техн. наук, проф. В.К. Зольников

ISBN 978-5-7731-0174-1 © Барабанов В.Ф., Подвальный С.Л.,  
Плахотнюк О.С., 2007  
© Оформление. ГОУ ВПО  
“Воронежский  
государственный технический  
университет”, 2007

## **ВВЕДЕНИЕ**

Динамические системы эволюционного типа возникают в связи с современными прикладными задачами математической физики при описании поведения сплошной среды, математические модели которых представлены в виде эволюционных дифференциальных уравнений в частных производных, в которых решение зависит от одной «пространственной» и одной «временной» переменной. В ходе эволюции системы могут возникать различные экстремальные (нештатные) ситуации (режимы), приводящие к разрушению объектов технологического процесса (оборудования, сырья, продуктов и т.д.). Важнейшей задачей математического моделирования является обнаружение и предотвращение ситуаций такого рода за счет формирования управляющих воздействий на граничных условиях параметров эволюционных систем.

Проведение вычислительного эксперимента при исследовании сложных систем нелинейных эволюционных уравнений в современных универсальных системах математического моделирования осложняется высокими требованиями к точности начального приближения к решению, низкой скоростью приближения к решению или отсутствием сходимости применяемых итерационных численных методов при резких изменениях краевых условий. Кроме того, низкая производительность при использовании интерпретируемых языков описания

математических моделей (MATLAB, MathCad, Mathematica и т.д.), а также недостаточная гибкость систем с трансляцией в компилируемые языки (XMDS), не позволяют использовать перечисленные программные системы для обнаружения экстремальных ситуаций и формирования управляющих воздействий, выводящих моделируемый объект из экстремального режима в реальном времени.

Решение проблемы снижения требований к точности начального приближения к решению может быть достигнуто с использованием интервальных методов, однако повышенные вычислительные затраты и погрешности при вычислении интервальных отображений функций приводят к необходимости применения многовариантного подхода, заключающегося в совместном использовании интервальных и численных итерационных методов. При этом язык описания математических моделей эволюционных систем должен иметь средства формализации экстремальных ситуаций и задания управляющих воздействий.

Таким образом, актуальность работы определяется необходимостью создания алгоритмов и методов для повышения производительности и счетной устойчивости процессов численного решения при моделировании динамических режимов объектов эволюционного типа, осуществления обнаружения нештатных ситуаций и формирования управляющих воздействий, выводящих систему из экстремального режима.

**В первой главе** с системных позиций рассмотрены особенности моделирования динамических систем эволюционного типа, определены процессы, описываемые эволюционными уравнениями в частных производных, рассмотрены методы численного решения систем нелинейных уравнений, проведен сравнительный анализ

средств математического моделирования динамических эволюционных систем.

**Во второй главе** сформированы требования к основным функциональным блокам математического и программного обеспечения для многовариантного моделирования динамических эволюционных систем и функции подбора квазиоптимальных значений параметров законов управления граничными условиями в экстремальных режимах. На основе этих требований разработана модульная структура программной системы.

**В третьей главе** проведена разработка алгоритмов и методов формирования динамической частично регуляризированной расчетной сетки, использования всей информации о ходе решения для эффективной адаптивной подстройки шага дифференцирования.

**В четвертой главе** осуществляется формирование пользовательских интерфейсов, иерархического языка описания математических моделей, разработка метода повышения производительности процесса моделирования за счет трансляции описания математической модели в машинный код, применение интервальной проекции расчетной сетки на экранную плоскость.

**В пятой главе** проведена апробация системы многовариантного интерактивного моделирования на исследовании модельных задач процесса регенерации неподвижного слоя катализатора с учетом экстремального характера температурного режима.

**В приложении** приводится спецификация структурного языка описания математических моделей динамических режимов систем эволюционного типа.

# ГЛАВА 1. ПРОБЛЕМАТИКА МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ РЕЖИМОВ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА

## 1.1. Особенности моделирования динамических режимов эволюционных процессов

Динамические системы эволюционного типа возникают в связи с современными прикладными задачами математической физики при описании поведения сплошной среды (газ, жидкость, твердое тело). Под системами эволюционного типа, в данной работе, понимаются системы, математические модели которых представлены в виде эволюционных дифференциальных уравнений в частных производных, в которых решение зависит от одной «пространственной» переменной и одной «временной». Эти системы могут быть различными по своему физическому смыслу, но имеют общий вид по математическому описанию, что позволяет единообразно формулировать краевые задачи и затем по единым принципам исследовать их [1].

Как частный случай таких задач могут рассматриваться «стационарные» задачи, то есть краевые задачи для систем дифференциальных уравнений в частных производных.

К динамическим системам эволюционного типа можно отнести систему, описывающую неустановившееся течение газа (жидкости) в трубопроводах [2]; систему, описывающую неустановившееся движение газа в трубе, в области с подвижной границей, которая является траекторией движения твердого тела (контейнера) [3]; систем, описывающих движение крови по сосудам [4]; систему, описывающую движение автомобилей по скоростной автостраде [5] и др.

Математические эволюционные системы могут описываться различными способами [6]. Наиболее сложный и получивший широкое распространение класс эволюционных систем – системы с распределенными параметрами, описываемые эволюционными уравнениями в частных производных.

Математическое моделирование применяют, как правило, при исследовании сложных систем, когда невозможно заранее оценить последствия решений, направленных на совершенствование системы и управления ею, из-за действия внешних или внутренних случайных факторов [7]. Математическая модель дает возможность как бы сжимать или растягивать реальное время. С ее помощью можно проводить обучение персонала управлению системой [8].

Современный подход к математическому моделированию состоит в активном привлечении методов вычислительного эксперимента [9, 5], заключающихся в оптимальном сочетании вычислительных процедур и натурального эксперимента. Вычислительный эксперимент в настоящее время является основным методом реализации математического моделирования сложных динамических процессов.

Детерминированные математические модели отражают закономерности процессов, протекающих в динамических системах. Детерминированные модели, построенные с использованием теоретического подхода, имеют ряд существенных преимуществ: их можно разрабатывать даже при отсутствии действующего объекта, как это часто бывает при проектировании; они качественно более правильно характеризуют процессы, протекающие в объекте, даже при наличии недостаточно точных в количественном отношении параметров модели; пригодны для обобщений, связанных с изучением общих свойств



объектов определенного класса и для прогнозирования поведения объекта [10]. Наиболее полное представление о поведении объекта дают динамические модели.

Динамические математические модели описывают как стационарные, так и переходные, неустановившиеся режимы функционирования объектов. Причем статический режим можно рассматривать как результат завершения переходного процесса.

В ходе эволюции моделируемого динамического процесса во времени, могут наступать различного рода экстремальные ситуации, такие как авария, выход из нормы экономических показателей и т.д. Целесообразно в автоматическом или автоматизированном режиме осуществлять формирование управляющих воздействий, выводящих систему из подобного рода ситуаций.

По структуре математические модели эволюционных динамических объектов, как правило, нелинейны. Структура и степень детализации математического описания зависит от целей исследования, объема информации, положенного в основу модели, и других факторов. Ввиду сложности динамических процессов при разработке их математических моделей обычно вводят ряд упрощающих допущений. Например, для описания структуры потока в аппарате часто используют два предельных режима: режим идеального вытеснения и полного смешения.

Применение процедуры линеаризации, традиционно используемой для упрощения математических моделей, при рассмотрении эволюционных технологических процессов во многих случаях неприемлемо, т.к. характерные особенности протекания многих технологических процессов обусловлены именно действием нелинейных физико-химических механизмов, присутствием обратных связей между ними.

## 1.2. Аппарат динамических систем эволюционного типа

### 1.2.1. Математические модели динамических систем эволюционного типа

Математическая модель динамической системы эволюционного типа в общем случае представляется системой нелинейных уравнений в частных производных, имеющей вид:

$$\mathbf{F}\left(\mathbf{x}, \frac{\partial^m \mathbf{x}}{\partial t^m}, \frac{\partial^m \mathbf{x}}{\partial l^m}, t, l\right) = 0, \quad n = 1 \dots m - 1,$$
$$\mathbf{x}(0, l) = \mathbf{f}_{n_0}(l), \quad \frac{\partial^n \mathbf{x}(0, l)}{\partial t^n} = \mathbf{f}_{n_n}(l), \quad \mathbf{x}(t, 0) = \mathbf{f}_{z_0}(t), \quad \frac{\partial^n \mathbf{x}(t, 0)}{\partial l^n} = \mathbf{f}_{z_n}(t),$$

(1.1)

где  $\mathbf{F} = (F_1, \dots, F_n)$  – вектор-функция состояний, характеризующие закон эволюции системы;  $\mathbf{x} = (x_1, \dots, x_n)$  – переменные, характеризующие состояние системы,  $t$  – временная координата,  $l$  – пространственная координата,  $m$  – любое натуральное число,  $\mathbf{f}_n$  – начальные условия,  $\mathbf{f}_z$  – граничные условия. Функции  $\mathbf{F}$  в общем случае гладкие (дифференцируемые), нелинейные относительно указанных переменных. В некоторых случаях для адекватного

описания динамической системы требуется учет постоянного или переменного запаздывания, наличия разрывов характеристик, зависимости от пространственных производных и других особенностей [11-14]. При дальнейшем анализе свойств эволюционных динамических систем считаем, что размерность системы на два больше числа переменных, т.к. система (1.1) может в этом случае рассматриваться как система  $N+2$  переменных, у которой  $x_{N+1} = t$ ,  $x_{N+2} = l$ , а  $\frac{\partial x_{N+1}}{\partial t} = F_{N+1} = 1$  и  $\frac{\partial x_{N+2}}{\partial l} = F_{N+2} = 1$ .

### **1.2.2. Методы исследования динамических режимов систем эволюционного типа**

Методы исследования эволюционных динамических систем можно разделить на аналитические (символьные) и неаналитические (численные, графические). Достоинством аналитических методов (точных и приближенных) является возможность представления результатов в общем виде, охватывающим различные случаи и удобном для выполнения анализа. К тому же точные аналитические методы принципиально нужны для глубокого анализа нелинейных механизмов, получения точных решений, на которые можно ориентироваться при использовании численных методов. Неаналитические методы (всегда приближенные) дают окончательный результат лишь для конкретных исходных данных, полученные результаты нельзя экстраполировать на систему с другими параметрами. Кроме того, для многих задач представляет интерес не аналитический вид решения и не приближенное представление решений, а, например, ответ на следующие вопросы: каково число состояний равновесия у данной динамической системы, и устойчивы они или нет; существуют ли замкнутые траектории, сколько их и как они расположены? Эти задачи определили развитие

методов и приемов качественного исследования динамических систем.

Аналитические и качественные методы исследования динамических систем достаточно развиты и позволяют решать сложные задачи. Трудности исследования значительно возрастают при переходе от линейных систем к нелинейным. Основными методами исследования нелинейных систем являются [15-22]: качественные методы Ляпунова и Пуанкаре и другие методы качественной теории дифференциальных уравнений; методы, основанные на рассмотрении фазового пространства (особое развитие получил метод точечных отображений); рассмотрение протекания процесса во времени; частотный метод Попова; перевод рассмотрения поведения системы из временной области путем интегральных преобразований в какую-либо другую область; разделение сложной системы на простую; метод сечения пространства параметров и некоторые другие.

*Качественные методы Ляпунова и Пуанкаре и другие методы качественной теории дифференциальных уравнений.* [15,17,18] Эти методы, не будучи количественными, являются строгими и в прикладных задачах выполняют те же функции, что и точные аналитические методы. Теоретической основой линеаризованной теории автоматического регулирования являются теоремы Ляпунова об устойчивости по первому приближению. Благодаря этим теоремам, проблема устойчивости нелинейной системы в малом (гарантирующей осуществимость исследуемого движения или состояния равновесия) сводится к задаче об устойчивости линеаризованной системы и поэтому разрешима. При рассмотрении устойчивости не только в малом, но и в большом (при конечных отклонениях) задача

об устойчивости нелинейных систем не поддается разрешению.

Исследовать математически строго устойчивость в большом и в целом нелинейных систем со многими степенями свободы позволяет прямой метод Ляпунова [17]. Необходимо отметить, что прямой метод Ляпунова не предназначен для отыскания точных количественных характеристик, в частности точных границ областей устойчивости в пространстве параметров и точных границ областей приближения в фазовом пространстве исследуемых на устойчивость невозмущенных движений. Задача нахождения с помощью данного метода точной границы области абсолютной устойчивости нелинейной системы наталкивается на значительные трудности.

Качественная теория дифференциальных уравнений для автономных систем второго порядка достаточно полно развита в работах Пуанкаре и Бендиксона и многих последующих авторов. Для систем третьего порядка и выше качественная теория дифференциальных уравнений имеет ряд принципиально важных нерешенных вопросов.

*Методы, основанные на рассмотрении фазового пространства.* [16,17,19,20] Среди методов, основанных на непосредственном изучении фазового пространства, наибольшее развитие получил метод точечных отображений. Метод точечных отображений является аппаратом для решения ряда вопросов теории колебаний, в частности для нахождения предельных циклов в фазовых пространствах нелинейных систем и объяснения механизма мягкого и жесткого возбуждения и устойчивости автоколебаний, для получения результатов по теории бифуркаций, по динамике нелинейных неавтономных систем. Изучение фазовых портретов нелинейных систем с помощью метода точечных отображений позволило решить ряд важных прикладных задач.

Аналитические методы, связанные с изучением фазового пространства, в том числе метод точечных отображений [16], наиболее эффективны, если задача исследуется на фазовой плоскости или на многолистной фазовой плоскости. Методы фазовой плоскости позволили изучить оптимальные, самонастраивающиеся, цифровые и другие автоматические системы, осуществлять выбор корректирующих элементов. Понятие фазового пространства является фундаментальным в теории нелинейных систем и плодотворным даже при большом порядке системы. В частности, метод точечных отображений позволил рассмотреть ряд задач, в которых фазовое пространство имеет три и больше измерений. Однако, как и в качественной теории дифференциальных уравнений, при увеличении размерности возникают серьезные проблемы, преодолевая которые приходится многое терять. Если при размерности фазового пространства, равной двум, метод точечных отображений позволяет исследовать фазовую картину в большом и целом (например, с помощью диаграмм Ламерея), то при большей размерности приходится прибегать к локальному рассмотрению фазового пространства, линеаризуя функцию соответствия и исследуя малые окрестности неподвижных точек точечного отображения. Основная сложность состоит в том, что при размерности фазового пространства, большей двух, для отыскания неподвижных точек, соответствующих периодическим движениям, нет регулярных аналитических способов. Существенную помощь для решения такого рода задач может оказать теория бифуркации.

Практические трудности исследования методом точечных отображений многомерных фазовых пространств могут быть преодолены с помощью численных

вычислений, однако при этом метод перестает быть аналитическим.

*Рассмотрение протекания процесса во времени.* [17,18,20,22] Основным представителем этого подхода является метод припасовывания и другие родственные ему методы. Данные методы результативны для исследования периодических решений в релейных системах и системах  $n$ -порядка, имеющих кусочно-линейные характеристики.

При применении данных методов, обладающих большой значимостью, приходится сталкиваться с некоторыми затруднениями. Эти методы основаны на локальном подходе к анализу фазового пространства. Они не приспособлены для отыскания всех возможных периодических решений, ибо тип искомого периодического решения первоначально задается, а затем проверяется возможность его существования. Далее, эти методы в применении к системам более сложным, чем релейные, оказываются слишком громоздкими; трансцендентные «уравнения периодов», получаемые в ходе их применения к кусочно-линейным системам, не могут быть решены в общем виде и требуют приближенного численного решения.

*Частотный метод Попова.* [20] Данный метод широко применяется при рассмотрении задач абсолютной устойчивости нелинейных систем автоматического регулирования. Частотный метод обобщен на случай рассмотрения устойчивости не состояния равновесия, а устойчивости процесса, в частности устойчивости вынужденных колебаний. В ряде работ этот метод применен для решения задачи абсолютной устойчивости систем с гистерезисом.

Как и прямой метод Ляпунова, частотный метод Попова позволяет получать достаточные условия абсолютной устойчивости, вопрос о близости которых к

достаточным и необходимым в общем случае остается нерешенным.

*Перевод рассмотрения поведения системы из временной области путем интегральных преобразований в какую-либо другую область* составляет специфическую группу методов. Так, обычное преобразование Лапласа, эффективно применимое к линейным системам и переводящее дифференциальную задачу во временной области в алгебраическую задачу в некоторой комплексной области, может быть использовано и в случаях кусочно-линейных систем [20, 21]. Однако более целесообразно для исследования нелинейной системы применять другие преобразования. Примером может служить преобразование Коши-Тейлора и многократное преобразование Лапласа.

*Разделение сложной системы на простую* позволяет преодолеть трудности, испытываемые при исследовании нелинейных систем высокого порядка. Еще в классической механике для интегрирования уравнений движения механической системы с  $n$  степенями свободы был предложен метод разделения переменных уравнения Гамильтона-Якоби [22], осуществляемый путем удачного выбора координат и позволяющий в частных, но практически важных случаях найти полный интеграл этого уравнения.

В работах Нелепина [19, 21] предложен *метод сечения пространства параметров*, суть которого заключается в том, что нелинейная автоматическая система высокого порядка изучается в условиях специально выбираемых сечений пространства ее коэффициентов. Для этих сечений исходная система неособым линейным преобразованием переменных преобразуется к такой системе, исследование которой сводится к последовательному изучению ряда подсистем более низких порядков. Метод позволяет исследовать аналитически многомерные фазовые



пространства и выполнять разбиение бифуркационными кривыми плоскостей изучаемых сечений. Ограниченность метода состоит в том, что он позволяет исследовать исходную систему не при любых, а лишь при вполне определенных значениях параметров (коэффициентов). Однако, точные решения, получаемые методом сечений параметров, служат эталоном, на который можно ориентироваться при использовании в пространствах между сечениями приближенных методов. Кроме того, эти точные решения дополняют результаты приближенных исследований данными, которые приближенными методами не могут быть получены по причине их грубости. В некоторых задачах сечения удается построить так, что их анализ дает достаточное представление о структуре пространства параметров.

Для уравнений, описывающих поведение динамической системы, не существует обобщенного аналитического метода решения. Поэтому большое значение имеют численные методы. Численные методы позволяют получить частные решения достаточно общих задач. Полученные решения могут быть достаточно точными. Однако численные решения с перебором начальных значений и возможных значений параметров не дают общую характеристику поведения системы и не гарантируют от пропуска интересных решений.

### **1.2.3. Методы численного решения систем нелинейных уравнений**

Существующие методы численного решения систем нелинейных уравнений делятся на два больших класса: локальные и глобальные методы. Локальные методы (метод Ньютона, метод Бroyдена, полюсный метод Ньютона и т.д. [23-27]) обладают быстрой сходимостью, хорошей точностью найденного решения, но при этом требуют

задания хорошего начального приближения к решению. Глобальные методы решения (поисковые алгоритмы – метод градиентного спуска, метод покоординатного спуска, метод Хука и Дживса, и т.д. [23, 28-31]) обладают более медленной сходимостью, но при этом менее чувствительны к начальному приближению: при любой начальной точке глобальный метод решения найдет какой-нибудь ближайший экстремум.

Оба названных класса методов численного решения систем нелинейных уравнений обладают общим недостатком: сугубо локальный характер найденного решения. Кроме того, поиск других локальных решений затруднен тем, что в общем случае, не учитывая специфику решаемой задачи, невозможно с достаточной степенью достоверности выбрать начальную точку, поиск решения из которой приведет к следующему локальному экстремуму. Остается только выбор большого количества начальных точек, распределенных по случайному или по какому-то регулярному закону, что ведет к неоправданным вычислительным расходам и опять же не гарантирует отыскания всех локальных решений, пусть и в заданном ограниченном пространстве поиска.

Перечисленными недостатками не обладают пространственные, или интервальные методы численного решения систем уравнений [32]. Интервальные методы гарантированно находят все локальные решения в ограниченном пространстве поиска с заданной точностью, либо определяют отсутствие решений в заданном пространстве. Недостатком таких методов является плохая масштабируемость, т.е. быстрый рост вычислительной сложности с увеличением размерности решаемой задачи.

Возникает задача модификации пространственного метода решения систем уравнений с целью снижения

темпов роста вычислительных затрат при увеличении количества уравнений в системе.

### **1.3. Сравнительный обзор распространенных средств математического моделирования динамических режимов систем эволюционного типа**

В настоящее время существует большое количество программных пакетов, предназначенных для решения задач математического моделирования, каждый из которых имеет свои сильные и слабые стороны. Наиболее заметны эти различия становятся при решении задач класса нелинейных уравнений, в частности, систем нелинейных эволюционных уравнений в частных производных. Некоторые программные продукты предоставляют достаточную гибкость в настройке процесса численного решения, в других же большинство настроек жестко зашито в коде программы. Также во всех рассмотренных далее программных продуктах очень мало внимания уделено управлению в экстремальных ситуациях.

Почти все средства моделирования представляют стандартный набор средств визуализации результатов моделирования, однако без внимания оставлены способы визуализации объема данных массивом точек с различными оптическими свойствами, а также массивом точек, изменяющим оптические свойства во времени.

В программном пакете Math Cad основной упор сделан на возможность одновременного проведения расчета и оформления инженерной работы для печати. Math Cad отлично справляется с формулами любой степени сложности за счет наличия декларативной записи и декомпозиции с помощью нее сложных выражений на более простые.

В составе программного пакета имеется специализированный императивный интерпретируемый язык программирования, значительно расширяющий функциональность программного пакета. Интерфейс программирования встроенного языка предоставляет доступ к вычислительным механизмам среды Math Cad, однако не позволяет расширять средства визуализации, пользовательского интерфейса и численного счета, что делает невозможным осуществление автоматического обнаружения нештатных ситуаций и формирования управляющих воздействий, выводящих систему из экстремального режима.

Вычислительный блок Math Cad позволяет решать системы нелинейных уравнений, задание приближения к решению осуществляется пользователем при помощи удачно спроектированного интерфейса, где оператор на графике имеет возможность указать точку, находящуюся визуально близко к решению. Такой подход отлично работает при решении задач с небольшим числом неизвестных, однако для многомерных задачах этот способ мало подходит и начальное приближение приходится находить аналитически.

Math Cad обладает широкими возможностями по визуализации результатов вычислений, построению графиков и отображению другой численной информации. В наличии имеются такие способы визуализации, как:

- график вида  $y(x)$ ;
- поверхность вида  $y(x,z)$ ;
- векторное поле;
- траектория вида  $x(t), y(t), z(t)$ ;
- гистограмма;
- представление кластеризованной информации столбцами и т.д.

Таким образом, Math Cad представляет программный продукт для осуществления цикла разработки от вычислений до оформления полученных результатов и подготовки их к печати. Он подходит для обучения и осуществления инженерных работ средней степени сложности. Однако низкая скорость работы за счет использования интерпретируемых языковых конструкций, а также ограниченная расширяемость, не позволяющая реализовывать алгоритмы визуализации и взаимодействия с пользователем, не позволяет называть Math Cad инструментом, удовлетворяющим потребности современной инженерной деятельности.

MATLAB является мультиплатформенной интегрированной вычислительной средой. Вся функциональность MATLAB написана на встроенном языке, от вычислительных алгоритмов до модулей визуализации и сложных схем взаимодействия с пользователем. В настоящее время под MATLAB написано множество программных модулей, от сред моделирования систем управления, построения и изучения нейронных сетей, до систем цифровой обработки сигналов и модулей работы с внешними устройствами через порты ввода-вывода.

Программная система MATLAB имеет расширяемую архитектуру, гибкости его функционального языка достаточно для решения большинства задач моделирования, управления и визуализации. Однако скорость работы интерпретируемого языка, а также специфика внутренней типизации данных не позволяет в достаточной степени эффективно реализовать алгоритмы, работающие с интрузивными типами данных, такими как динамическая нерегулярная расчетная сетка, при решении систем нелинейных эволюционных уравнений. Скорость работы интерпретируемого языка MATLAB не позволяет

реализовать такие ресурсоемкие алгоритмы, как обнаружение нештатных ситуаций и формирование управляющих воздействий, выводящих систему из экстремального режима, в реальном времени.

MATLAB является очень гибким инструментом, позволяющим решать широкий круг задач, однако использование матриц в качестве основного типа данных накладывает некоторые ограничения на реализацию алгоритмов, работающих с нерегулярными структурами. В частности, отсутствие интрузивных типов данных затрудняет создание динамической нерегулярной расчетной сетки для решения систем нелинейных дифференциальных уравнений в частных производных. Также низкая скорость работы встроенного языка не позволяет осуществлять в реальном времени управление в экстремальных режимах.

Extensible Multi-Dimension Simulator является платформонезависимой средой моделирования. В арсенале данной системы имеется множество методов численного решения систем, как поисковых, так градиентных и смешанных. Имеется возможность выбирать между несколькими типами аппроксимации производных, а также настраивать конфигурацию расчетной сетки. Модели описываются на языке XML при помощи специальных тегов системы XMDS, после чего они транслируются в C-код и компилируются в динамически подгружаемую библиотеку, что обеспечивает наивысшую скорость вычислений по сравнению с рассмотренными ранее средами Math Cad и MATLAB.

Достаточно бедные возможности визуализации в системе XMDS, отсутствие возможности сохранения промежуточного результата моделирования и отсутствие механизмов расширения функциональности системы, не позволяют назвать XMDS оптимальным выбором для

исследования объектов, описываемых системами нелинейных эволюционных уравнений в частных производных, и осуществления управления в экстремальных ситуациях.

Обзор наиболее популярных программных пакетов, предназначенных для решения задач математического моделирования, выявил ряд общих недостатков, затрудняющих эффективное применение перечисленных программных продуктов для исследования динамических эволюционных систем и управления в нештатных режимах.

Применение интерпретируемых языков повышает гибкость и портируемость программных пакетов математического моделирования, однако накладывает ограничения на скорость выполнения пользовательских модулей. Это делает невозможным написание собственных модулей обнаружения и управления в экстремальных ситуациях, а в случае реализации внутренних вычислительных алгоритмов системы на таком языке, как в случае MATLAB, приводит к низкой скорости работы вычислительного блока. Программа XMDS лишена недостатка низкой производительности вычислительного блока, однако для эффективного использования этой программе недостает гибкости, из-за отсутствия возможности написания пользовательских модулей.

В рассмотренных программных пакетах вычислительный процесс является жестко зашитым в коде системы, с возможностью конфигурирования параметров вычисления только перед началом вычисления, что нельзя назвать удобным при осуществлении длительных расчетов. Кроме того, в Math Cad и XMDS отсутствует возможность сохранения промежуточных результатов вычисления, что делает их вовсе непригодными для продолжительных численных экспериментов. Исключением является MATLAB, где за счет открытости всех вычислительных

алгоритмов имеется возможность как сохранять, так и изменять параметры в ходе вычисления.

Важной составляющей вычислительного эксперимента является обнаружение экстремальных ситуаций с последующим формированием управляющего воздействия, выводящего систему из нештатного режима. Полученную информацию можно использовать как пассивно, в качестве совета оператору, так и активно для автоматической корректировки хода моделирования.

В настоящее время задачи моделирования постоянно усложняются, решением эволюционных модельных задач является распределение параметров в пространстве и во времени. Наглядности стандартных средств визуализации для отображения многомерной информации становится недостаточно. Наибольшее количество информации можно отобразить средствами пространственной визуализации, где каждой точке в заданном ограниченном пространстве ставятся оптические свойства в соответствии со значениями моделируемых параметров. У каждой пространственно точки можно менять цвет (3 компоненты – RGB или HSB), прозрачность, коэффициент преломления и коэффициент отражения, что дает 5 одновременно отображаемых пространственно распределенных параметров. Если помимо трех пространственных координат, параметры моделируемой системы распределены также по времени, визуализация должна осуществляться динамически.

Ни одна из рассмотренных систем моделирования не позволяет осуществить многомерную пространственную визуализацию описанного типа, что затрудняет анализ результатов моделирования с использованием данных систем.

Ни одно из рассмотренных средств моделирования не осуществляет визуализацию работы блоков численного



счета и оптимизации, без чего невозможна интерактивность процесса моделирования и изменение оператором настроек расчета при моделировании длительных процессов. За исключением среды MatLab, в остальных рассматриваемых программных продуктах отсутствует возможность вмешательства в алгоритмы численного счета.

## ГЛАВА 2. ФОРМИРОВАНИЕ СТРУКТУРЫ СИСТЕМЫ МНОГОВАРИАНТНОГО ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА

### 2.1. Формирование требований к функциональности системы многовариантного интерактивного моделирования

#### 2.1.1. Формирование вычислительных требований к моделированию динамических систем эволюционного типа

Основным методом численного исследования динамических эволюционных систем является рассмотрение протекания процесса во времени. Данное исследование осуществляется в несколько последовательных этапов:

- Задание математического описания динамической системы в виде системы эволюционных уравнений в частных производных:

$$\mathbf{F}\left(\mathbf{x}, \frac{\partial^m \mathbf{x}}{\partial t^m}, \frac{\partial^m \mathbf{x}}{\partial l^m}, t, l\right) = 0.$$

(2.1)

- Задание временных и пространственных границ расчетной сетки:

$$t = t_{\min} \dots t_{\max}, \quad l = l_{\min} \dots l_{\max}.$$

(2.2)

- Задание начальных и граничных условий динамической системы:

$$\begin{aligned}
 \mathbf{x}(t_0, l) &= \mathbf{f}_n(t_0, l), \quad t_0 = t_{\min} \dots t_{\max}^0; \\
 \mathbf{x}(t, l_0) &= \mathbf{f}_z(t, l_0), \quad l_0 = l_{\min} \dots l_{\max}^0;
 \end{aligned}
 \tag{2.3}$$

где  $t_{\max}^0$  и  $l_{\max}^0$  - границы краевых областей.

- Переход от аналитической записи системы уравнений к разностной аппроксимации:

$$\mathbf{F}_j^i(\mathbf{x}_{j-n}^{i-m}, \dots, \mathbf{x}_j^i, t_i, l_j) = 0,
 \tag{2.4}$$

где  $i, j$  - индексы ячейки в расчетной сетке;  $m, n$  - количество ячеек, участвующих в аппроксимации производных;  $t_i, l_j$  - значения временной и пространственной координат в ячейке.

- Определение значений переменных модели в краевых ячейках расчетной сетки:

$$\begin{aligned}
 \mathbf{x}_j^{i_0} &= \mathbf{f}_n(t_{i_0}, l_j), \quad j = 0 \dots j_{\max}^{i_0}, \quad i_0 = 0 \dots m - 1, \\
 \mathbf{x}_{j_0}^i &= \mathbf{f}_r(t_i, l_{j_0}), \quad i = 0 \dots i_{\max}, \quad j_0 = 0 \dots n - 1,
 \end{aligned}
 \tag{2.5}$$

где  $j_{\max}^{i_0}$  - количество ячеек на  $i_0$ -м временном слое;  $i_{\max}$  - количество временных слоев.

- Нахождение численного решения во внутренних точках расчетной сетки  $i = m \dots i_{\max}, j = n \dots j_{\max}^i$ .

В целях повышения точности найденного численного решения целесообразно применить динамическую расчетную сетку с нерегулярным шагом интегрирования. В областях с сильно нелинейным характером решений необходимо сгущать сетку, уменьшая шаги численного дифференцирования. В более «пологих» областях расчетную сетку необходимо разряжать в целях снижения требований к вычислительным ресурсам.

Для снижения требований к начальному приближению к решению целесообразно применить интервальные алгоритмы на начальном этапе поиска решения. В последствии для получения более точного приближения к решению необходимо перейти на численные поисковые и локальные методы поиска решения.

### 2.1.2. Формирование требований к средствам квазиоптимального управления

Одним из важнейших аспектов вычислительного эксперимента является обнаружение нештатных ситуаций и формирование управляющих воздействий, выводящих систему из экстремального режима. Под экстремальной ситуацией в работе понимается выполнение условия:

$$\prod_{\substack{j=0 \dots j_{\max}^i \\ i=i_{t-\tau_j} \dots i_t}} J(\mathbf{x}_j^i) > 0, 0 \leq \tau_j \leq t \quad (2.6)$$

где  $J$  - критерий экстремальности, принимает только неотрицательные значения;  $\tau_j$  - глубина расчета критерия экстремальности по времени;  $i_{t-\tau_j}$  - номер временного слоя,

соответствующий времени  $t - \tau_J$ . В случае возникновения экстремальной ситуации необходимо сформировать управляющее воздействие на канале граничных условий  $f_\Gamma$  на участке  $i = i_{t-\tau_J} \dots i_t$ , которое приведет к выполнению равенства:

$$\begin{aligned} J_{i=i_{t-\tau_J} \dots i_t, j=0 \dots j_{\max}^i}(\mathbf{x}_j^i) &= 0, 0 \leq \tau_J \leq t, \\ \mathbf{x}_{j_0}^i &= \mathbf{f}_\Gamma(t_i, l_{j_0}) + \mathbf{f}_\Gamma^*(t_i, l_{j_0}, \mathbf{p}), i = i_{t-\tau_J} \dots i_t, j_0 = 0 \dots n-1, \end{aligned} \quad (2.7)$$

где  $\mathbf{f}_\Gamma^*$  - искомая поправка к граничным условиям динамической эволюционной системы,  $\mathbf{p}$  - вектор параметров. Обобщенный критерий экстремальности, учитывающий все временные слои с индексами  $i = i_{t-\tau_J} \dots i_t$  и управления по граничным условиям (2.7), имеет вид:

$$G(\mathbf{p}, t, \tau_J) = \sum_{i=i_{t-\tau_J}}^{i=i_t} \sum_{j=0}^{j=j_{\max}^i} J(\mathbf{x}_j^i). \quad (2.8)$$

Задача поиска вектора параметров  $\mathbf{p}$  квазиоптимального управляющего воздействия  $\mathbf{f}_\Gamma^*$  ставится как оптимизационная:

$$\mathbf{p} = \arg \min_{\mathbf{p} \in \mathbf{P}} G(\mathbf{p}, t, \tau) \quad (2.9)$$

для заданного пространства оптимизации  $\mathbf{P}$ , текущего временного слоя с координатой  $t$  и заданной глубины расчета критерия экстремальности  $\tau$ . Поэтому интерактивная система моделирования динамических

режимов систем эволюционного типа должна обладать возможностью автоматического или автоматизированного подбора параметров модели для достижения оптимума в терминах критерия экстремальности.

При практическом использовании математических методов для решения задач оптимизации динамических процессов выделяют ряд последовательных этапов [33, 34]:

- **общий анализ задачи оптимизации.** Предполагает предварительный общий анализ задачи оптимизации с целью выяснения ее типа (статическая, динамическая, оптимальное управление и т.д.).
- **Определение критерия (или нескольких критериев) эффективности.** Выбором критерия оптимизации определяется основная цель функционирования динамического процесса.
- **Выбор управляемых переменных и анализ их влияния на критерий оптимизации.** Этап требует тщательного анализа и выявления возможного качественного влияния оптимизирующих или управляющих переменных на критерий оптимизации. Здесь важно, с одной стороны, учесть все существенные для оптимизации переменные, а с другой – исключить из рассмотрения несущественные переменные, мало влияющие на критерий оптимизации, так как сложность решения задачи в значительной степени определяется числом переменных, по которым производится оптимизация. Правильный учет всех требуемых ограничений на переменные системы обязателен, поскольку, как показывает опыт решения задачи оптимизации, по некоторым переменным оптимум часто находится на ограничении. Кроме того, при помощи проведенного анализа важно исключить все ограничения, которые заведомо не будут достигаться в оптимальном режиме.

- Составление математической модели. Происходит установление в математической форме связи критерия оптимизации с управляемыми переменными, а также математической трактовки всех имеющихся ограничений.
- Выбор оптимальной стратегии исследования. Этап наиболее важен для сложной системы, так как при оптимизации полный расчет и анализ функционирования системы (определение значений глобального критерия и значений функций, отвечающих ограничениям) приходится производить многократно.
- Выбор метода поиска экстремума критерия эффективности. Представляет собой математическую задачу нахождения экстремума глобального критерия в области изменений управляемых переменных, определяемой ограничениями системы.
- Проведение оптимального расчета и анализ полученных результатов.

Различаются следующие виды оптимизации динамических процессов:

- Квазистатическая оптимизация – выбор наилучшего технологического режима. Такая оптимизация предполагает, что процесс находится в установившемся состоянии.
- Динамическая оптимизация – оптимизация переходного процесса. Осуществляется оптимальным образом переход от одного стационарного состояния к другому.
- Оптимизация колебательного процесса – выбор параметров периодического управления (амплитуды, частоты) при заданном законе изменения входных параметров (например, гармоническом). Процесс находится в установившемся периодическом режиме.

- Оптимальное управления – выбор формы закона изменения, амплитуды, частоты входных параметров управления динамическим процессом.

Описание в монографиях [3, 4, 33-39] различных алгоритмов и подходов связано с тем, что многолетняя практика решения нелинейных задач оптимизации привела специалистов к достаточно единодушному мнению о невозможности создания универсального алгоритма, который бы одинаково успешно решил все задачи. Действительно, задачи нелинейной оптимизации чрезвычайно разнообразны. Они отличаются структурой вхождения нелинейности, количеством переменных и ограничений, требуемым объемом памяти. И практический опыт показывает, что существуют классы задач, в которых, казалось бы, самый неэффективный с теоретической точки зрения метод благодаря простоте реализации и специфике задачи дает хорошие результаты. В связи с этим вычислительная практика требует набора различных алгоритмов.

Сформулируем основные требования, предъявляемые к блоку оптимизации параметров управляющего воздействия, выводящего динамическую эволюционную систему из экстремальной ситуации:

- 1) должна быть предусмотрена возможность задания критерия экстремальности в виде функции от вектора параметров системы  $J(\mathbf{x})$ ;
- 2) блок оптимизации должен осуществлять однокритериальную оптимизацию в рамках заданного пространства оптимизации  $\mathbf{P}$ ;
- 3) в ходе оптимизации параметров управляющего воздействия необходимо осуществлять откат вычислительного эксперимента на заданное количество временных слоев, определяемое глубиной расчета критерия экстремальности  $\tau$ .



### **2.1.3. Постановка задачи визуализации работы отдельных компонентов системы интерактивного моделирования**

Представление сеточных функций в виде табличных данных не всегда является удобным, т.к. при больших объемах информации бывает достаточно трудно проанализировать полученные результаты. Возникает потребность в ином виде представления информации, таком как построение изображений на основе массивов численных данных, иными словами в визуализации.

Визуализация является важной частью интерактивных систем моделирования технологических процессов, она помогает осуществлять наглядный качественный контроль параметров и результатов моделирования, тем самым расширяя возможности интуитивного управления процессом на основе личного опыта оператора.

Спецификой задачи является независимость процессов моделирования и визуализации, т.к. первые могут иметь значительную продолжительность выполнения во времени, а при визуализации результатов необходимо иметь возможность оперативно изменять конфигурацию отображения и набор отображаемых параметров.

Наиболее широко используемыми способами графического представления сеточных функций в настоящее время являются двумерный график вида  $y = f(x)$  и трехмерная поверхность вида  $z = f(x, y)$ .

В процессе визуализации необходимо учитывать возможность сопоставления нескольких графиков на одной схеме, например изменение нескольких параметров во времени или в пространстве. Система визуализации должна осуществлять выделение цветом областей, на которых моделируемый объект входит в экстремальный режим. Также необходима возможность оперативного изменения

положения и размера отображаемого участка области изменения аргументов в режиме реального времени.

#### **2.1.4. Формирование требований к интерактивному режиму моделирования и взаимодействию с оператором**

При разработке систем моделирования повышенное внимание следует обращать на организацию диалогового взаимодействия человека с системой. Диалоговый режим обеспечивает существенное упрощение операций ввода-вывода информации при проведении моделирования и анализа результатов моделирования. При организации диалога необходимо выполнение следующих требований: отражение предметной области пользователя; гибкость средств для обеспечения удобства в достижении цели диалога для пользователей с разной степенью подготовки; устойчивость к ошибкам пользователей. Созданию диалоговых систем моделирования способствует появление и широкое внедрение современных технических средств интерактивного взаимодействия с вычислительной машиной.

Применение систем интерактивного взаимодействия при моделировании технологических процессов вызвано следующими причинами:

- большинство решаемых задач при исследовании технологических процессов не поддается строгой формализации и могут быть решены только пользователем;
- во многих случаях формализация задачи бывает очень громоздкой и неэффективной, в то же время пользователь может решить эту задачу значительно быстрее, используя свой опыт, интуицию и зрительный аппарат;
- появляется возможность оперативно оценивать промежуточные результаты и управлять дальнейшим ходом вычислительного процесса, контролировать и

корректировать данные. Интерактивный режим работы предполагает обмен сообщениями между пользователем и системой в соответствии с принятым языком диалога.

## **2.2. Модульная структура системы**

### **многовариантного интерактивного моделирования**

Основными функциями системы интерактивного моделирования являются:

- Задание математического описания и краевых условий динамической системы эволюционного типа.
- Численное решение системы эволюционных уравнений на нерегулярной расчетной сетке.
- Обнаружение нештатных ситуаций и формирование управляющих воздействий, выводящих систему из экстремального режима.
- Визуализация работы компонентов системы интерактивного моделирования.
- Сериализация результатов моделирования.
- Интерактивный контроль работы компонентов системы интерактивного моделирования.

Модульная структура системы интерактивного моделирования определяется приведенной выше функциональной декомпозицией системы (рис. 2.1):

- Модуль задания входных данных. Осуществляет структурированный ввод математического описания модели, краевых условий и параметров моделирования на основе специально разработанного структурного языка, позволяющего дополнять хранилище математических моделей без изменения исходного кода программы. Язык является подмножеством XML, представляет собой иерархическую декомпозицию модели на компоненты. Для обеспечения высокой скорости работы вычислительного блока, осуществляется автоматическая трансляция файлов

задания входных данных в C++ код, который компилируется и статически подключается к системе на этапе запуска.

- Модуль численного решения. Осуществляет численную аппроксимацию математической модели системы эволюционного типа, а также формирование динамической расчетной сетки на основе данных, полученных от модуля задания входных данных. Задача численного решения системы эволюционных уравнений (1) в общем случае имеет вид:

$$\begin{cases} \mathbf{F}_j^i(\mathbf{x}_{j-n}^{i-m}, \dots, \mathbf{x}_j^i, t_i, l_j) = 0; \\ \mathbf{x}_j^{i_0} = \mathbf{f}_H(t_{i_0}, l_j), j = 0 \dots j_{\max}^{i_0}, i_0 = 0 \dots m-1; \\ \mathbf{x}_{j_0}^i = \mathbf{f}_2(t_i, l_{j_0}), i = 0 \dots i_{\max}, j_0 = 0 \dots n-1; \end{cases} \quad (2.10)$$

где  $i, j$  - индексы ячейки в расчетной сетке;  $m, n$  - количество ячеек, участвующих в аппроксимации производных;  $t_i, l_j$  - значения временной и пространственной координат в ячейке;  $j_{\max}^{i_0}$  - количество ячеек на  $i_0$ -м временном слое;  $i_{\max}$  - количество временных слоев.

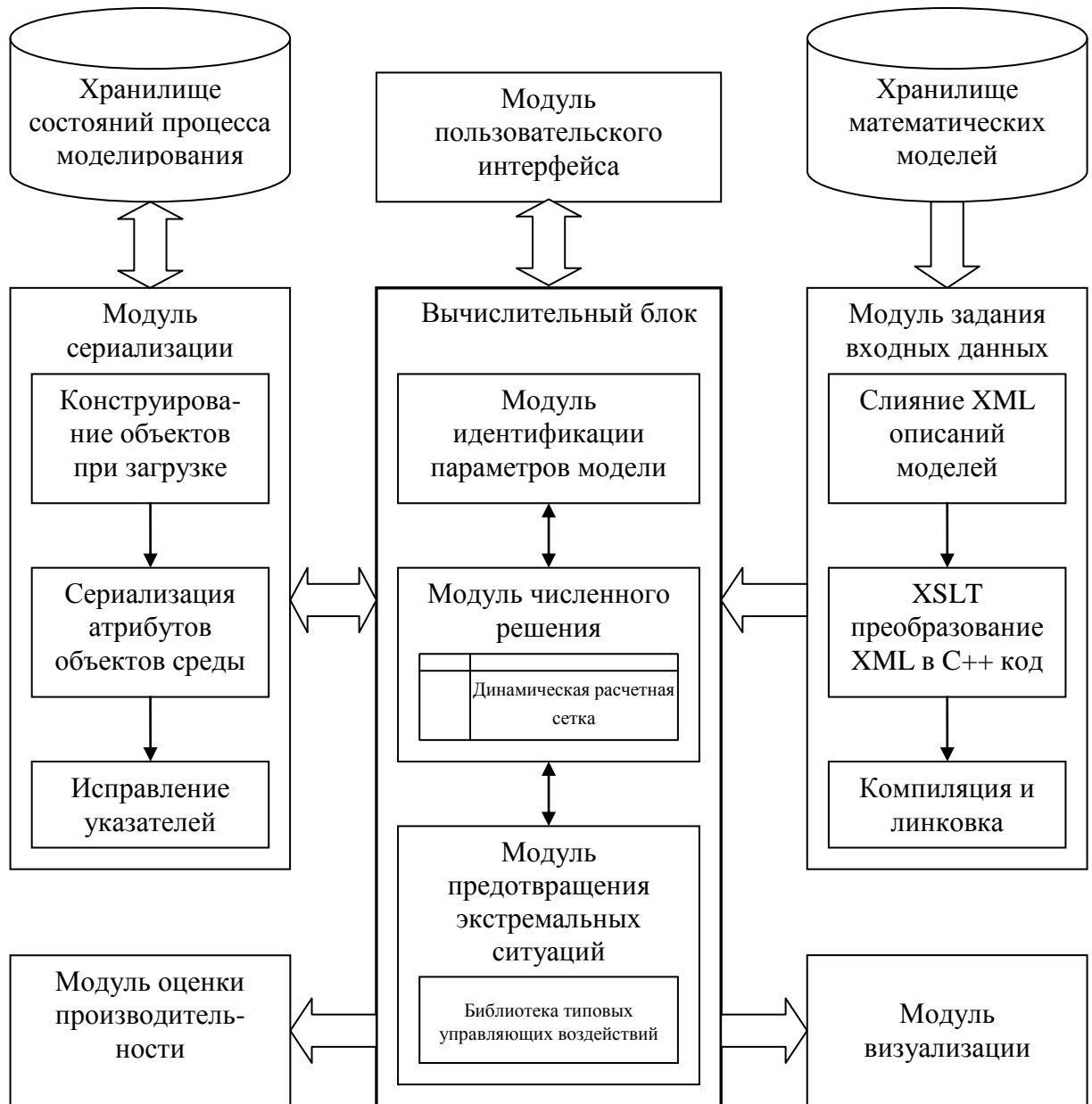


Рис. 2.1. Модульная структура системы интерактивного моделирования

Граничные условия могут задаваться как в виде функций от соответствующих координат:

$$\begin{aligned} \mathbf{x}_j^{i_0} &= \mathbf{f}_H(t_{i_0}, l_j), j = 0 \dots j_{\max}^{i_0}, i_0 = 0 \dots m - 1; \\ \mathbf{x}_{j_0}^i &= \mathbf{f}_z(t_i, l_{j_0}), i = 0 \dots i_{\max}, j_0 = 0 \dots n - 1; \end{aligned} \quad (2.11)$$

так и в результате решения системы уравнений статики:

$$\begin{cases} \mathbf{F}_j^{i_0}(\mathbf{x}_{j-n}^{i_0}, \dots, \mathbf{x}_j^{i_0}, t_{i_0}, l_j) = 0; \\ \mathbf{F}_{j_0}^i(\mathbf{x}_{j_0}^{i-m}, \dots, \mathbf{x}_{j_0}^i, t_i, l_{j_0}) = 0; \\ \mathbf{x}_{j_0}^{i_0} = \mathbf{x}_0(t_{i_0}, l_{j_0}); \\ i_0 = 0 \dots m - 1, j_0 = 0 \dots n - 1; i = m \dots i_{\max}, j = n \dots j_{\max}^{i_0}. \end{cases} \quad (2.12)$$

Решение осуществляется на динамической расчетной сетке с частичной регуляризацией при помощи многометодного подхода. Используя заданное прямоугольное пространство поиска корней, находится интервал, содержащий корень, после чего найденное решение уточняется методом обобщенного покоординатного спуска, а затем – методом Ньютона для получения точного приближения к решению.

- Библиотека типовых управляющих воздействий содержит набор типовых законов управления в виде функций временного аргумента, задающих качественный вид воздействия, а также набора коэффициентов, задающих количественные характеристики закона управления. Имеется возможность задания произвольного управляющего воздействия в виде функции временной переменной при помощи структурного языка описания математических моделей.

- Модуль управления в экстремальных ситуациях. Осуществляет циклическое численное решение динамической эволюционной системы на расчетной сетке с коррекцией коэффициентов текущих управляющих воздействий. Тип управляющего воздействия задается в модуле задания входных данных из библиотеки типовых управляющих воздействий.
- Модуль идентификации параметров модели. Производит коррекцию параметров модели с целью минимизации отклонения от заданных показателей при помощи поисковых алгоритмов параметрической оптимизации.
- Модуль визуализации. Формирует графическую диагностику работы модулей системы интерактивного моделирования. Осуществляет визуализацию решения системы эволюционных уравнений в двухмерном и трехмерном виде, шага дифференцирования, краевых условий.
- Модуль сериализации. Осуществляет чтение или запись состояния процесса моделирования в бинарный поток, что обеспечивает возможность автосохранения, приостановки и последующего продолжения расчета на любом этапе хода численного решения.
- Модуль пользовательского интерфейса. Предназначен для формирования двухсторонней связи между оператором и системой интерактивного моделирования. Модуль выполнен с использованием современных средств графической визуализации.

### **2.3. Многовариантный подход к численному решению систем эволюционных уравнений**

Наиболее универсальной схемой численного решения систем эволюционных уравнений является последовательное применение нескольких численных

методов, начиная от менее требовательных к заданию начального приближения, и заканчивая более чувствительными методами. Укрупненный алгоритм многовариантного численного решения системы эволюционных уравнений представлен на рис. 2.2.

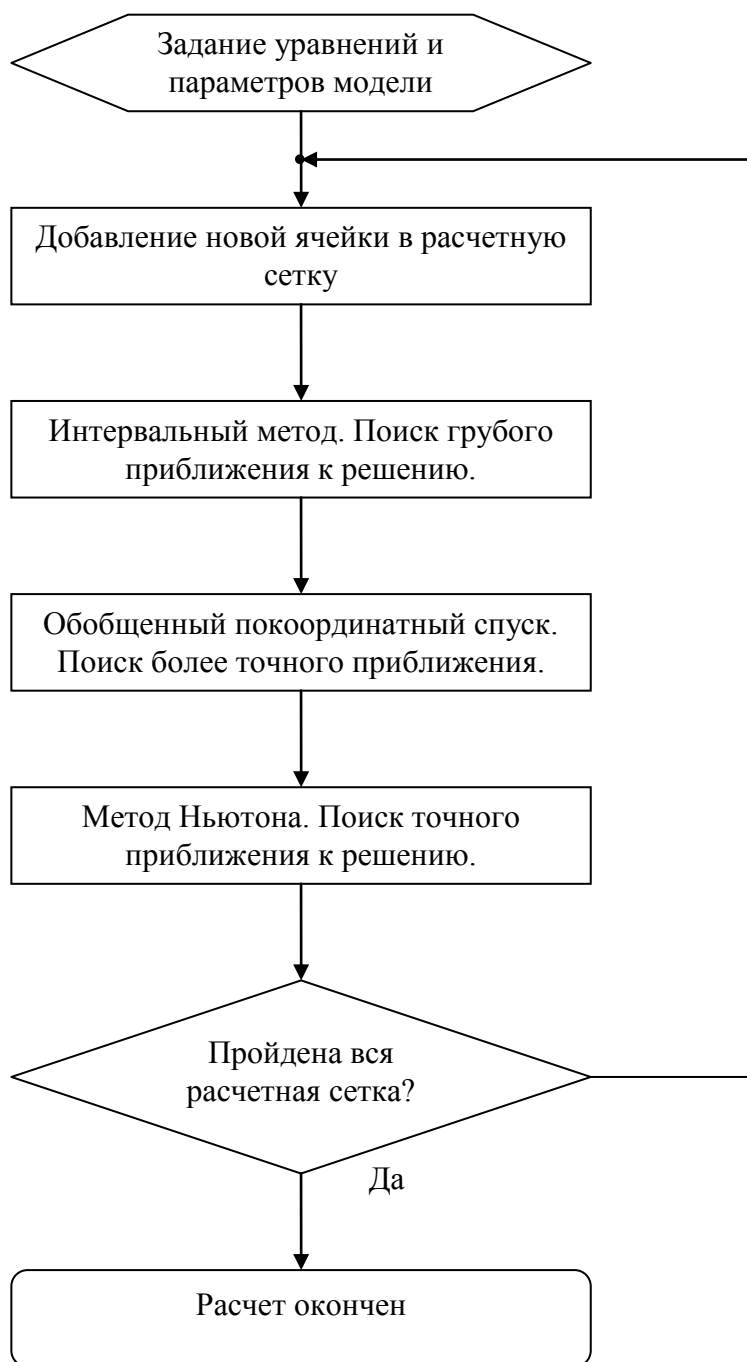




Рис. 2.2. Укрупненный алгоритм многовариантного  
численного решения систем нелинейных  
эволюционных уравнений

На начальном этапе задается пространство поиска решений, после чего при помощи интервального метода оно сужается до заданной окрестности корня. Из полученного отрезка выбирается точка, являющаяся начальным приближением для поискового алгоритма обобщенного покоординатного спуска, при помощи которого производится первое уточнение найденного решения. После чего методом Ньютона осуществляется окончательное уточнение приближенного значения корня системы эволюционных уравнений.

# ГЛАВА 3. РАЗРАБОТКА АЛГОРИТМОВ И МЕТОДОВ МНОГОВАРИАНТНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА

## 3.1. Постановка задачи численного решения систем эволюционных уравнений в частных производных

Задача численного решения системы эволюционных уравнений (1.1) в общем случае имеет вид:

$$\begin{cases} \mathbf{F}_j^i(\mathbf{x}_{j-n}^{i-m}, \dots, \mathbf{x}_j^i, t_i, l_j) = 0; \\ \mathbf{x}_j^{i_0} = \mathbf{f}_H(t_{i_0}, l_j), j = 0 \dots j_{\max}^{i_0}, i_0 = 0 \dots m - 1; \\ \mathbf{x}_{j_0}^i = \mathbf{f}_c(t_i, l_{j_0}), i = 0 \dots i_{\max}, j_0 = 0 \dots n - 1; \end{cases} \quad (3.1)$$

где  $i, j$  - индексы ячейки в расчетной сетке;  $m, n$  - количество ячеек, участвующих в аппроксимации производных;  $t_i, l_j$  - значения временной и пространственной координат в ячейке;  $j_{\max}^{i_0}$  - количество ячеек на  $i_0$ -м временном слое;  $i_{\max}$  - количество временных слоев. Динамическая расчетная сетка с нерегулярным шагом характеризуется нелинейным изменением временной и пространственной координат.

## 3.2. Методология численного решения систем эволюционных уравнений в частных производных

### 3.2.1. Переход от аналитического вида решаемой системы эволюционных уравнений к разностной аппроксимации

В общем случае, система нелинейных эволюционных уравнений в частных производных имеет вид:

$$\begin{cases} \mathbf{F}\left(\mathbf{x}, \frac{\partial^m \mathbf{x}}{\partial t^m}, \frac{\partial^m \mathbf{x}}{\partial l^m}, t, l\right) = 0, \\ \mathbf{x}(0, l) = \mathbf{f}_n(l), \\ \mathbf{x}(t, 0) = \mathbf{f}_z(t), \end{cases} \quad (3.2)$$

где  $\mathbf{F} = (F_1, \dots, F_n)$  – вектор-функция состояний, характеризующие закон эволюции системы;  $\mathbf{x} = (x_1, \dots, x_n)$  – переменные, характеризующие состояние системы,  $t$  – временная координата,  $l$  – пространственная координата,  $m$  – любое натуральное число,  $\mathbf{f}_n$  – начальные условия,  $\mathbf{f}_z$  – граничные условия.

Для нахождения приближенного численного решения системы (3.2) необходимо перейти от аналитического вида к разностному путем замены производных их аппроксимациями. Для этого вводится двухмерная расчетная сетка, в узлах которой расположены значения переменных  $\mathbf{x}$ ,  $t$ ,  $l$ . В узле с координатами  $(i, j)$  разностная аппроксимация системы эволюционных уравнений (3.2) будет иметь вид (3.1).

### 3.2.2. Функциональный анализ различных способов задания краевых условий динамической системы эволюционного типа

Начальные и граничные условия системы эволюционных уравнений могут быть получены либо с помощью некоторого детерминированного функционального закона, либо в результате решения системы уравнений статики, получаемой из исходной путем приравнивания к нулю всех частных производных по времени (для начальных условий), либо по пространству (для граничных условий).

Функциональная форма задания краевых условий имеет вид:

$$\begin{aligned} \mathbf{x}_j^{i_0} &= \mathbf{f}_n(t_{i_0}, l_j), j = 0 \dots j_{\max}^{i_0}, i_0 = 0 \dots m - 1; \\ \mathbf{x}_{j_0}^i &= \mathbf{f}_z(t_i, l_{j_0}), i = 0 \dots i_{\max}, j_0 = 0 \dots n - 1; \end{aligned} \quad (3.3)$$

При формировании краевых условий на расчетной сетке в функциональной форме, осуществляется линейризация функций  $\mathbf{f}_n$ ,  $\mathbf{f}_z$ . Выбирая шаги расчетной сетки в краевых областях, следует руководствоваться минимумом погрешности аппроксимации с одной стороны и мерой предельного сгущения сетки - с другой.

Формирование краевых условий при помощи решения уравнений статики имеет вид:

$$\begin{cases} \mathbf{F}_j^{i_0}(\mathbf{x}_{j-n}^{i_0}, \dots, \mathbf{x}_j^{i_0}, t_{i_0}, l_j) = 0; \\ \mathbf{F}_{j_0}^i(\mathbf{x}_{j_0}^{i-m}, \dots, \mathbf{x}_{j_0}^i, t_i, l_{j_0}) = 0; \\ \mathbf{x}_{j_0}^{i_0} = \mathbf{x}_0(t_{i_0}, l_{j_0}); \\ i_0 = 0 \dots m-1, j_0 = 0 \dots n-1; i = m \dots i_{\max}, j = n \dots j_{\max}^{i_0}. \end{cases} \quad (3.4)$$

В случае порядка разностной аппроксимации производных, превышающего первый, производные не приравниваются к нулю, а понижается их порядок.

### 3.2.3. Применение динамической расчетной сетки с нерегулярным шагом для решения эволюционных уравнений в частных производных

Численное решение систем нелинейных эволюционных уравнений наиболее рационально проводить на нерегулярных динамических сетках. В общем случае, динамические сетки применимы везде, где результатом решения является распределение некоторых параметров в пространстве. Применение таких сеток является более эффективным с точки зрения требований к вычислительным ресурсам.

Интерактивная система многовариантного моделирования осуществляет численное решение систем нелинейных разностных уравнений вида (3.1) при помощи многовариантного алгоритма численного решения *Solve*. Численное решение осуществляется на динамической прямоугольной двухмерной сетке с нерегулярным шагом (рис. 3.1). Шаг временной координаты между слоями, а также шаг пространственной координаты в пределах одного слоя изменяются в ходе решения по следующим правилам:

- 1) Если расчет предыдущей ячейки (или слоя) прошел успешно (заданная точность была достигнута),

уменьшается счетчик увеличения шага. Если счетчик достиг нуля, осуществляется увеличение шага.

- 2) если в результате расчета предыдущей ячейки (или слоя) заданная точность не была достигнута, счетчик увеличения шага устанавливается на заданное максимальное значение, величина текущего шага уменьшается.
- 3) при переходе на новый временной слой, шаг по пространственной координате и его счетчик увеличения сбрасываются в свое начальное значение.

Новизна данного алгоритма заключается в наличии запаздывания на приращение шага за счет введения счетчиков увеличения и уменьшения шагов, что вносит в характер расчетной сетки частичную регуляризацию.

Если в ходе расчета достигается заданная точность, повышается скорость расчета за счет увеличения шага и уменьшения количества ячеек в сетке. Если же требуемой точности достигнуть не удалось, шаг уменьшается, что приводит к уменьшению погрешности разностной аппроксимации производных. При стремлении к нулю данной погрешности выполняется необходимое условие стремления к нулю глобальной ошибки, которая может накапливаться от слоя к слою. Т.е. выполняется необходимое условие устойчивости разностной схемы. К сожалению, данное условие устойчивости целиком определяется способом аппроксимации каждой отдельной задачи. Однако адаптивная подстройка шага позволяет быстро получать численное решение при удачном выборе аппроксимации задачи.

○ - начальные и граничные условия; ○ - ячейки, значения которых получены численным решением системы нелинейных уравнений; ● - ячейки, значения которых получены путем интерполяции между соседними ячейками слоя;

$x$  - пространственная координата;  $t$  - временная координата;  $s_{xi}$  - шаги расчетной сетки по пространственной координате;  $s_{tj}$  - шаги расчетной сетки по временной координате;  $x_{\min}, x_{\max}$  - пространственные границы расчетной сетки;  $t_{\min}, t_{\max}$  - временные границы расчетной сетки.

Некоторые традиционные разностные схемы аппроксимации первой производной имеют вид [9]:

$$\frac{d\mathbf{x}_{t_i}}{dt} = \frac{\mathbf{x}_{t_{i+1}} - \mathbf{x}_{t_{i-1}}}{t_{i+1} - t_{i-1}} - \text{ неявная двухточечная схема (3.5)}$$

$$\frac{d\mathbf{x}_{t_i}}{dt} = \frac{\mathbf{x}_{t_i} - \mathbf{x}_{t_{i-1}}}{t_i - t_{i-1}} - \text{ явная двухточечная схема, (3.6)}$$

$$\frac{d\mathbf{x}_{t_i}}{dt} = \frac{3 \cdot \mathbf{x}_{t_i} - 4 \cdot \mathbf{x}_{t_{i-1}} + \mathbf{x}_{t_{i-2}}}{t_i - t_{i-2}} - \text{ явная трехточечная схема. (3.7)}$$

Метод аппроксимации может быть задан явно в описании задачи, а также выбран системой автоматически. Выбирается метод с участием максимального числа ячеек, при условии, что такие ячейки существуют в расчетной сетке.

Добавление новой ячейки в сетку влечет за собой добавление ячеек на предыдущих временных слоях для расчета производных по времени. Данная процедура вносит характер частичной регуляризованности в расчетную сетку. Для повышения вычислительной устойчивости пользователь имеет возможность настраивать число слоев, на которые добавятся ячейки с текущей пространственной координатой. Значения параметров модели в точках на самых дальних временных слоях определяются линейной интерполяцией между соседними точками. Значения остальных ячеек определяются путем численного решения системы нелинейных разностных уравнений в каждой из



них. Вычисление может производиться как в каждой отдельной ячейке, так и путем объединения группы ячеек в комплексную многоточечную систему нелинейных разностных уравнений.

### **3.2.4. Разработка алгоритма добавления новой ячейки в нерегулярную расчетную сетку с использованием частичной регуляризации сетки**

Рассмотрим алгоритм добавления новой ячейки в нерегулярную динамическую расчетную сетку. В результате перехода от системы эволюционных уравнений в частных производных к ее численной аппроксимации получается разностная схема, определяющая топологию ячеек, участвующих в расчете текущей системы уравнений. Предположим, нужно найти решение уравнения в частных производных вида:

$$F\left(y, \frac{\partial y}{\partial x}, \frac{\partial y}{\partial t}\right) = 0, \quad (3.8)$$

где требуется найти распределение значений функции  $y$  по координатам  $x$  и  $t$ . Перейдем к разностной аппроксимации уравнения при помощи двухточечной формулы аппроксимации производных [9]:

$$\begin{aligned} \frac{\partial y}{\partial x} &\cong \frac{y(x) - y(x - s_x)}{s_x}, \\ \frac{\partial y}{\partial t} &\cong \frac{y(t) - y(t - s_t)}{s_t}. \end{aligned} \quad (3.9)$$

Тогда уравнение (3.8) примет вид:

$$F\left(y(x), \frac{y(x) - y(x - s_x)}{s_x}, \frac{y(t) - y(t - s_t)}{s_t}\right) = 0. \quad (3.10)$$

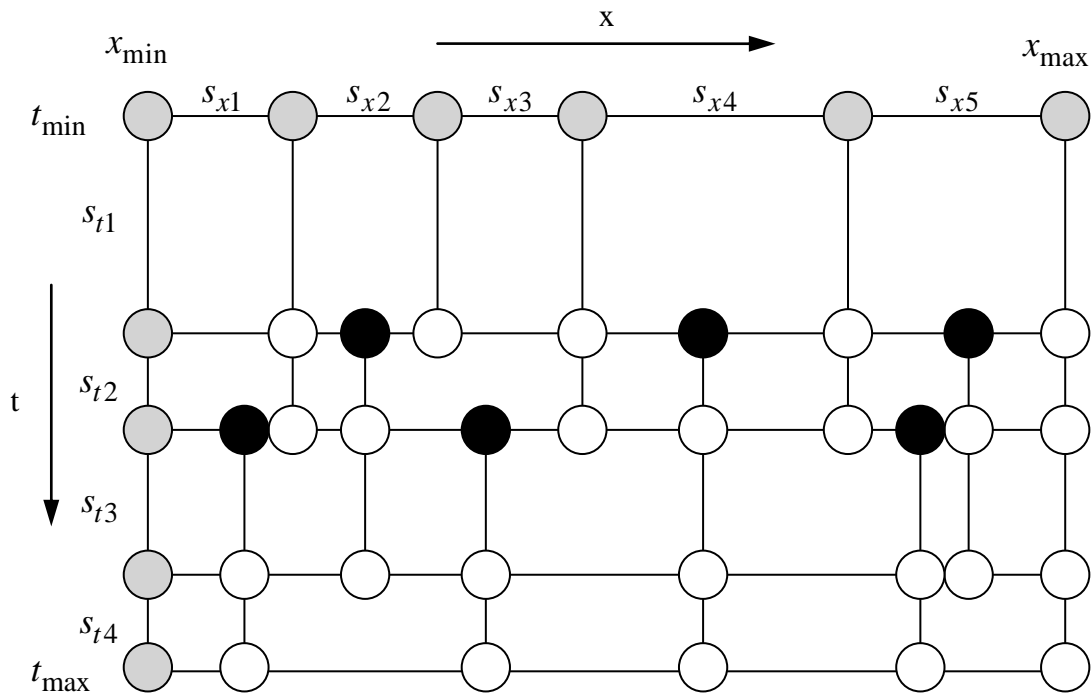


Рис. 3.1. Двухмерная нерегулярная прямоугольная сетка

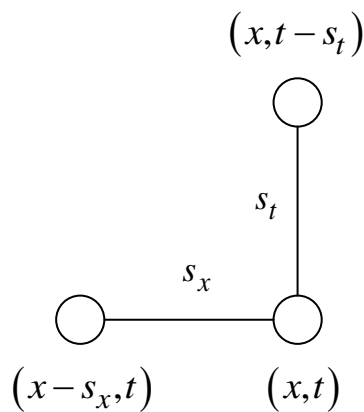


Рис. 3.2. Разностная схема уравнения (3.10)

Разностная схема, соответствующая уравнению (3.10) приведена на рисунке (3.2) и в литературе носит название левого треугольника [27]. Очевидно, при использовании явного односточечного метода расчета, расчетная сетка будет заполняться слева направо, сверху вниз (принимая за ось абсцисс пространственную переменную, а за ось ординат – временную). После добавления новой ячейки в расчетную сетку, для ее расчета необходимо наличие предыдущих ячеек по времени и по пространству, т.е. имеющих координаты  $(x, t - s_{t1})$ ,  $(x - s_{x1}, t)$ . Направление заполнения расчетной сетки гарантирует наличие ячейки  $(x - s_{x1}, t)$ , если  $(x, t)$  не находятся в начальной или граничной области.

Значение ячейки с координатами  $(x, t - s_{t1})$  может быть получено либо при помощи интерполирующей функции  $I(y_1, y_2, \dots, y_n)$  между существующими ячейками временного слоя  $t - s_{t1}$ , либо решением уравнения (3.10) в ячейке  $(x, t - s_{t1})$ , которое, в свою очередь, также зависит от ячейки  $(x, t - s_{t1} - s_{t2})$  и т.д. В параметры алгоритма заполнения расчетной сетки целесообразно вынести количество рассчитываемых таким образом дополнительных ячеек в отдельный параметр  $\Omega$ . Назовем его *глубиной пересчета слоев*. На рисунке (3.1) черным цветом обозначены ячейки, полученные при помощи интерполяции между соседними

ячейками слоя, а белым цветом – ячейки, полученные в результате решения уравнения (3.9).

Блок схема алгоритма добавления новой ячейки в расчетную сетку показана на рисунке 3.3:

- 1) Входные данные задаются на шаге 1:  $x$  и  $t$  - координаты внутренней области расчетной сетки, где требуется создать новую ячейку;  $\Omega$  - глубина пересчета слоев; обнуляется итератор  $i$ ; расчетная сетка должна быть полностью заполнена на временных слоях  $[t_{\min}, t)$  и заполнена до текущей ячейки на слое  $t$ :  $[x_{\min}, x)$ .

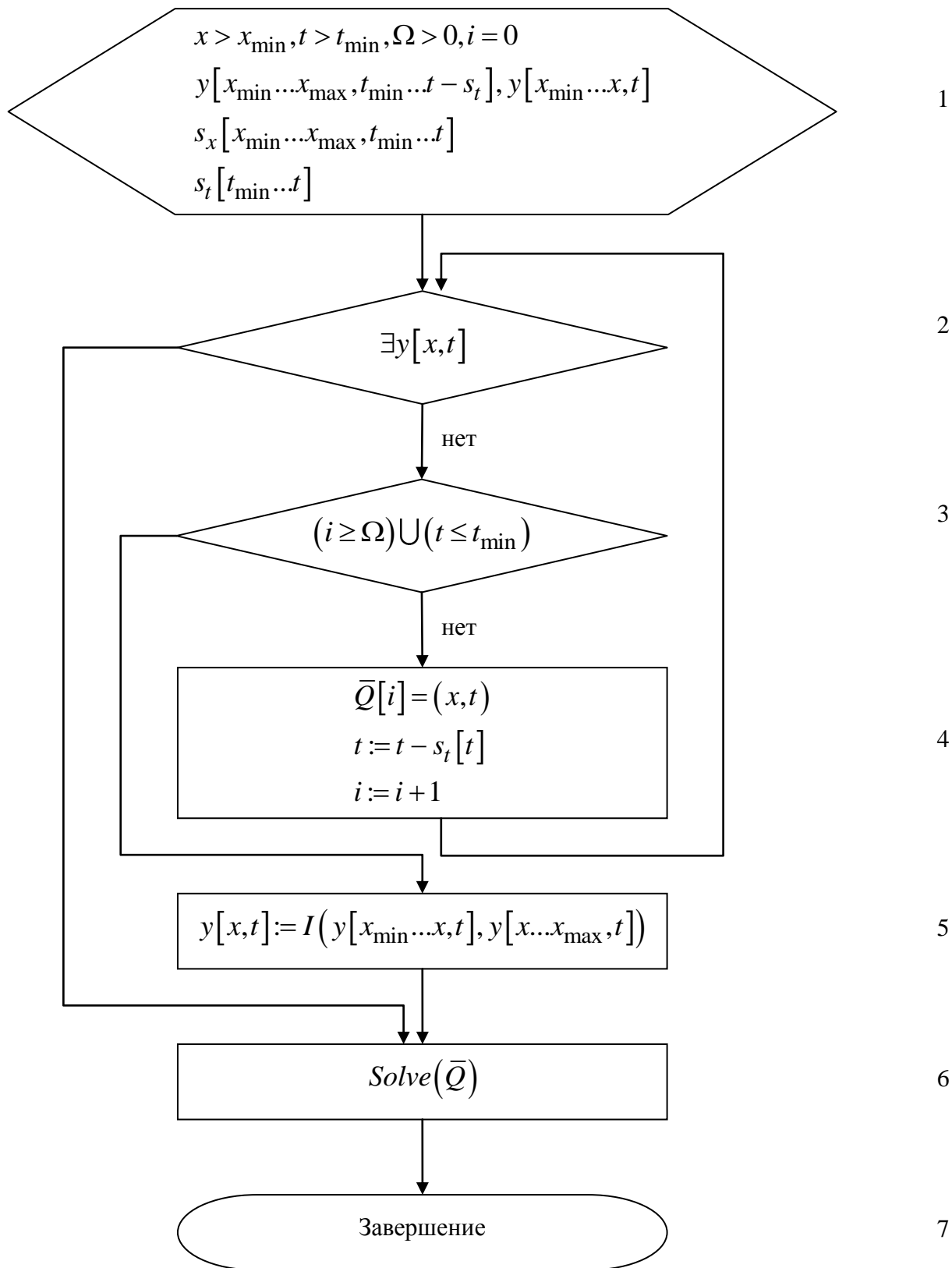


Рис. 3.3. Алгоритм *Next*, осуществляющий добавление новой ячейки в расчетную сетку

- 2) На шаге 2 выполняется проверка существования ячейки с текущими координатами в расчетной сетке. Если ячейка существует, происходит переход на шаг 6 к решению вектора выбранных ячеек  $\bar{Q}$ . Иначе происходит переход на шаг 3.
- 3) На шаге 3 осуществляется проверка переполнения вектора  $\bar{Q}$  -  $i \geq \Omega$  и проверка выхода из внутренней области расчетной сетки -  $t \leq t_{\min}$ . Если вектор переполнен или текущие координаты  $(x,t)$  не принадлежат внутренней области расчетной сетки, происходит переход на шаг 5. Иначе выполняется шаг 4.
- 4) На шаге 4 текущие координаты  $(x,t)$  заносятся в вектор решаемых ячеек  $\bar{Q}$ , происходит переход на предыдущий временной слой  $t - s_t[t]$ , и приращение индекса  $i$ . После чего происходит переход на шаг 2 алгоритма.
- 5) На шаге 5 расчет значения ячейки  $(x,t)$  осуществляется при помощи интерполирующей функции  $I$  на основе уже рассчитанных ячеек сетки. После выполнения интерполяции происходит переход на следующий шаг 6.
- 6) На шаге 6 осуществляется численное решение ячеек, помещенных в вектор  $\bar{Q}$  в обратном порядке, при помощи алгоритм многовариантного численного решения *Solve*.
- 7) Шаг 7 – завершение работы алгоритма. Результатом работы является дополнение расчетной сетки новой ячейкой  $(x,t)$  и необходимыми для ее расчета дополнительными ячейками, общим количеством не более  $\Omega$  ячеек.

### 3.2.5. Адаптация алгоритма обобщенного покоординатного спуска для нахождения численного решения в узлах нерегулярной расчетной сетки с адаптивной подстройкой шага дифференцирования

Уточнение решения, найденного интервальным методом целесообразно начинать с поисковых алгоритмов, т.к. этот класс методов менее зависим от качества задания начального приближения. В качестве такового метода выбран метод обобщенного покоординатного спуска.

Для задачи  $J(x) \rightarrow \min, x \in R^n, J \in C^2(R^n)$  в качестве минимизируемого функционала выступает норма вектор-функции системы (3.1). Переход от вектора  $x^i$  к вектору  $x^{i+1}$  по методу циклического покоординатного спуска происходит следующим образом: для  $l \in [1:n]$  компонента  $x_l^{i+1}$  определяется как

$$x_l^{i+1} \in \text{Arg} \min_{x \in R^1} J(x_1^{i+1}, x_2^{i+1}, \dots, x_{l-1}^{i+1}, x, x_{l+1}^i, \dots, x_n^i) \quad (3.11)$$

В интерактивной системе исследования динамических систем в качестве метода простого покоординатного спуска применяется метод, изложенный в работе [41]. Задается вектор начальных шагов  $h = (h_1, \dots, h_n)$  продвижений из точки  $x$  в направлении координатных ортов  $e^1, e^2, \dots, e^n$ . Далее шаги  $h_i$  модифицируются от итерации к итерации. Если выполняется неравенство  $J(x + h_i e^i) \leq J(x)$ , то текущая точка  $x$  заменяется на  $x + h_i e^i$ , а величина  $h_i$  утраивается:  $h_i = 3 \cdot h_i$ . После этого осуществляется переход к следующему номеру  $i$ . Если  $J(x + h_i e^i) > J(x)$ , то производится умножение  $h_i$  на  $-0,5$  и также осуществляется переход к следующему

координатному орту. Благодаря этому, алгоритм адаптируется к конкретным условиям численного решения за счет изменения величин и знаков шагов. Если начальные значения шагов были выбраны неудачно, то они быстро скорректируются до необходимых значений.

Однако если градиент минимизируемого функционала составляет значительный угол с ортами  $e^1, e^2, \dots, e^n$ , эффективность метода простого покоординатного спуска падает, вплоть до полного останова вследствие конечного размера машинной сетки. На помощь приходят методы обобщенного покоординатного спуска. Основной процедурой при реализации данного класса методов является процедура диагонализации матрицы  $J''(x)$  с последующим циклическим покоординатным спуском вдоль собственных векторов. Целесообразность такого подхода вытекает из геометрически очевидного факта, заключающегося в том, что оси наиболее рациональной системы координат при минимизации квадратичных функционалов (независимо от их выпуклости) методом покоординатного спуска совпадают с собственными векторами матрицы вторых производных, являющихся осями симметрии соответствующих поверхностей уровня. Как показано в [40], для целей численного решения систем нелинейных уравнений, как в выпуклой ситуации, так и в невыпуклой, достаточно вычислить произвольный ортонормированный базис в инвариантном подпространстве, отвечающем каждой изолированной группе собственных значений. При этом обстоятельстве собственные векторы могут быть вычислены со значительными погрешностями. Отвечающие этим базисам линейные оболочки с высокой точностью совпадают с истинными подпространствами, определяемыми невозмущенной диагонализируемой матрицей.



В качестве основной процедуры приведения симметричной матрицы к главным осям в методах обобщенного покоординатного спуска использован метод Якоби [42], несмотря на наличие конкурирующих, вообще говоря, более эффективных вычислительных схем [43]. Такой выбор обусловлен следующими обстоятельствами. Во-первых, вычисленные методом Якоби собственные векторы всегда строго ортонормальны с точностью, определяемой точностью компьютера, даже при кратных собственных числах. Последнее весьма существенно при использовании этих векторов в качестве базиса, так как предотвращается возможность вырождения базиса, существующая, например, в методе Пауэлла [44]. Во-вторых, многие вычислительные схемы имеют преимущество перед методом Якоби лишь при решении частичной проблемы собственных значений. В нашем же случае всегда решается полная проблема, и поэтому выигрыш во времени оказывается несущественным при существенно более сложных вычислительных схемах. В-третьих, определенное влияние на выбор алгоритма оказала простота логики метода Якоби, что приводит к компактности реализующих его программ.

В методе Якоби исходная симметричная матрица  $A$  приводится к диагональному виду с помощью цепочки ортогональных преобразований вида

$$A_{k+1} = U_k^T A_k U_k, A_0 = A, k = 1, 2, \dots, \quad (3.12)$$

являющихся преобразованиями вращения. В результате надлежащего выбора последовательности  $\{U_k\}$  получаем  $\lim A_k = D = U^T A U, k \rightarrow \infty$ , где  $D = \text{diag}(\lambda_i)$  - диагональная матрица;  $U = U_0 U_1 U_2 \dots$  - ортогональная матрица. Так как (3.12) есть преобразование подобия, то на

диагонали матрицы  $D$  расположены собственные числа матрицы  $A$ ; столбцы матрицы  $U$  есть собственные векторы матрицы  $A$ . Элементарный шаг (1) процесса Якоби заключается в преобразовании посредством матрицы  $U_k = \{u_{ij}\}$ , отличающейся от единичной элементами  $u_{pp} = u_{qq} = \cos \varphi$ ,  $u_{pq} = -u_{qp} = \sin \varphi$ . Угол вращения  $\varphi$  выбирается таким образом, чтобы сделать элемент  $a_{pq}$  матрицы  $A$  нулем. Вопросы сходимости различных численных схем, реализующих метод Якоби, рассмотрены в книге [45].

В интерактивной системе моделирования используется алгоритм *jacobi* из сборника алгоритмов линейной алгебры [46], реализующий так называемый частный циклический метод Якоби. В этом методе аннулированию подвергаются все элементы верхней треугольной части матрицы  $A$  с применением построчного выбора. При таком выборе индексы элементов  $a_{pq}$  пробегают последовательность значений  $(1,2), (1,3), \dots, (1,n); (2,3), (2,4), \dots, (2,n); \dots; (n-1,n)$ . Затем начинается новый цикл перебора элементов в том же порядке.

В интерактивной системе моделирования применен алгоритм обобщенного покоординатного спуска на основе конечно-разностных аппроксимаций производных. Достоинством подхода, основанного на применении формул численного дифференцирования, кроме его универсальности, является низкая стоимость подготовки задачи к компьютерному моделированию. От пользователя требуется лишь написать подпрограмму для вычисления значения  $J(x)$  при заданном  $x$ . Реализованные на основе разностной аппроксимации производных методы численного решения систем нелинейных уравнений оказываются, по существу, прямыми методами, т.е. методами, не использующими в своей схеме производные

от  $J$ . Действительно, заменяя  $\frac{dJ}{dx_i}$ , например, конечно-разностным отношением  $\frac{J(x + se^i) - J(x)}{s}$ , где

$e^i = (0, \dots, 1, \dots, 0), s \in R^1$ , мы фактически используем лишь значения  $J$ , вычисленные при определенных значениях аргумента. Используемые методы приближенного решения строятся на основе использования локальной квадратичной модели минимизируемого функционала, получаемой из общего разложения в ряд Тейлора. Естественно поэтому при выборе формул численного дифференцирования также руководствоваться идеей локальной квадратичной аппроксимации. Исходя из этого, целесообразно вместо формул с односторонними приращениями применять двусторонние конечно-разностные аппроксимации производных, оказывающиеся точными для квадратичных функционалов. Действительно, легко проверить, что если

$$f(x) = \frac{1}{2} \langle Ax, x \rangle - \langle b, x \rangle, \quad \text{то равенство} \quad \frac{df}{dx_i} = \frac{f(x + se^i) - f(x - se^i)}{2s}$$

оказывается точным при любом  $s \neq 0$ . Точно так же точным оказывается следующее представление для вторых производных квадратичного функционала

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{f(x + se^i + se^j) - f(x - se^i + se^j) - f(x + se^i - se^j) + f(x - se^i - se^j)}{4s^2}$$

. При использовании данных соотношений, вычислительные затраты характеризуются числом обращений к вычислению значений  $J(x)$ : для вычисления градиента -  $2n$ , для вычисления матрицы Гессе -  $2n^2 + 1$ , где  $n$  - размерность вектора  $x$ .

Для данного метода достаточно определять  $J'(x)$  и  $J''(x)$  с точностью до множителя, поэтому деление на  $2s$  и  $4s^2$  не производится. Это позволяет избежать известных вычислительных трудностей, если значение  $s$  оказывается относительно малым. В данном случае мы отказываемся от

работы с различными шагами по  $s_i$  по отдельным компонентам  $x_i$  вектора  $x$ , предполагая, что последние надлежащим образом нормализованы.

В используемом в интерактивной системе моделирование методе обобщенного покоординатного спуска шаг дискретности  $s$  определяется автоматически в зависимости от величины нормы вектора продвижения  $\|x^i - x^{i-1}\|$  в пространстве переменных  $x$  за один цикл работы алгоритма. Чем большая величина  $s$  используется, тем шире предполагаемая область справедливости локальной квадратичной модели исходного функционала.

Полученные в результате диагонализации матрицы Гессе новые координатные орты используются далее для реализации базового алгоритма циклического покоординатного спуска с процедурой выбора шагов продвижения по осям. Переход к новым осям координат целесообразно осуществлять после того, как текущие оси «исчерпали себя» и дальнейшего существенного улучшения ситуации не ожидается. В данном методе обновление осей осуществляется после того, как по каждому из координатных направлений вслед за успешным продвижением последовала неудача в смысле возрастания значения  $J(x)$ .

Процесс заканчивается по исчерпанию заданного числа обращений к процедуре вычисления  $J(x)$ , после чего при помощи метода Ньютона производится окончательное уточнение полученного результата до требуемой точности.

### **3.2.6. Модификация пространственного метода численного решения систем нелинейных уравнений с применением радиально-базисной нейронной сети для хранения информации о пространстве решения**

Интервальный алгоритм численного решения систем нелинейных уравнений определяет множество локальных решений с заданной точностью, либо констатирует отсутствие решений в ограниченном пространстве поиска. Решаемая система нелинейных уравнений имеет вид:

$$\mathbf{F}(\mathbf{x}) = 0, \quad \begin{cases} F_1(x_1, x_2, \dots, x_N) = 0, \\ F_2(x_1, x_2, \dots, x_N) = 0, \\ \dots \\ F_N(x_1, x_2, \dots, x_N) = 0, \end{cases} \quad (3.13)$$

где  $\mathbf{x}$  - вектор неизвестных,  $\mathbf{F}$  - вектор-функция, определяющая уравнения системы. Требуется найти значения вектора неизвестных, обращающие в 0 каждый элемент вектор-функции  $\mathbf{F}$ .

Специфика численного решения состоит в том, что в результате работы алгоритма находится не точка, а интервал значений в явном виде, в котором предположительно находится корень. При этом существующие алгоритмы численного решения оперируют точками, в основном неявно переходя к интервалам уже при нахождении решения: по окончании работы алгоритма говорится, что корень лежит в определенной окрестности найденной точки.

Пространственный метод решения оперирует целыми пространствами значений, а не отдельными точечными значениями. Пространство значений будем считать объединением множества интервалов:

$$D = \bigcup_{i=1}^n D_i, \quad D_i = (D_i^{\min}, D_i^{\max}); \quad (3.14)$$

где  $D_i^{\min}$ ,  $D_i^{\max}$  - нижняя и верхняя граница  $i$ -го интервала.

Требуется найти такую операцию, которая на входе принимает ограниченное пространство поиска корней вида (3.14), а возвращает пространство значений нормы функционала системы уравнений (3.13):

$$G(\mathbf{D}) = \{\|\mathbf{F}(\mathbf{x})\|\}, \forall \mathbf{x} \in \mathbf{D}, \quad (3.15)$$

где  $\mathbf{D}$  - пространство поиска корней.

Нахождение функции (3.15) в общем виде потребует использования библиотек интервальной математики, где элементарные функции работают с пространствами значений.

Исходя из того, что  $\|\mathbf{F}(\mathbf{x})\|$  является иерархической суперпозицией элементарных интервальных функций, вычисление конечного пространственного функционала системы (3.15) представляет собой рекурсивный процесс, в котором пространство значений нижней стоящей в иерархии элементарной функции является пространством аргумента элементарной функции, стоящей по иерархии на ступень выше.

Уже после первого вычисления пространственного функционала (3.15), однозначно определяется наличие хотя бы одного корня системы в рассматриваемом пространстве поиска решений. Для этого нужно всего лишь определить, попадает ли 0 в пространство  $G(\mathbf{D})$  (3.15).

Если в пространстве поиска корней имеется хотя бы один корень, оно разбивается на подпространства и в каждом из них осуществляется поиск корней. Подпространства, не содержащие корней, отбрасываются, для остальных операция продолжается пока размер подпространства не станет меньше заданной точности поиска корней.

Укрупненный пространственный алгоритм численного решения системы нелинейных уравнений (3.13):

- 1) Задается пространство поиска решений. Задается пороговая длина интервала точности определения решений.
- 2) Выбирается новое текущее подпространство из пространства поиска решений.
- 3) Определяется, есть ли корень в текущем подпространстве. Если имеется хотя бы один корень, переходим на шаг 4. Если решением являются все точки подпространства, переходим на шаг 5. Если в подпространстве нет корней, переходим на шаг 7.
- 4) Определяется размер текущего подпространства. Если размер меньше требуемого интервала точности, переходим на шаг 5. Иначе – переходим на шаг 6.
- 5) Текущее подпространство целиком добавляется в пространство решений. Происходит переход на шаг 7.
- 6) Текущее подпространство разбивается на подпространства. Новые подпространства добавляются в пространство поиска решений. Происходит переход на шаг 7.
- 7) Текущее подпространство исключается из пространства поиска решений. Происходит переход на шаг 8.
- 8) Если пространство поиска корней пусто, переходим на шаг 9, иначе – на шаг 2.
- 9) Завершение работы алгоритма. Выходными данными алгоритма является заполненное пространство решений.

Как видно, результатом работы алгоритма является заданный в явном виде интервал, в котором находится корень. Кроме того, на функцию не накладывается никаких требований относительно дифференцируемости и непрерывности. В случаях, когда текущее подпространство

поиска корней целиком лежит вне области определения функционала (3.15), функционал возвращает пустое пространство.

Достоинства пространственного метода численного решения систем нелинейных уравнений:

- 1) использование пространства поиска корней вместо начальной точки, используемой существующими алгоритмами;
- 2) отсутствие требований непрерывности и дифференцируемости исходной функции;
- 3) определение интервала, в котором находится корень, в явном виде;
- 4) нахождение всех корней, принадлежащих пространству поиска;
- 5) определение наличия корня в рассматриваемом пространстве поиска уже на первой итерации работы алгоритма;
- 6) экспоненциальная сходимость метода;
- 7) корректная работа с бесконечным количеством решений (алгоритм возвращает пространство корней);
- 8) поиск решения в каждом из подпространств можно осуществлять параллельно, что дает возможность применения распределенных вычислений.

Недостатком пространственного метода численного решения систем нелинейных уравнений является экспоненциальный рост вычислительных затрат с увеличением размерности решаемой задачи.

Важными факторами, определяющими масштабируемость пространственного метода численного решения систем нелинейных уравнений, являются:

- 1) выбор начального пространства поиска корней;
- 2) способ рекурсивного деления пространства поиска корней.



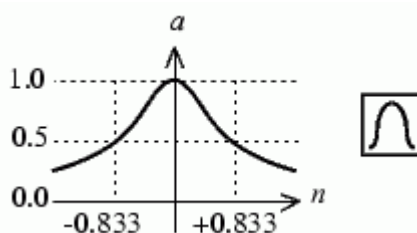
Объектом исследования математических моделей является зависимость поведения корней  $x$  от коэффициентов системы уравнений  $p$ , т.е. от параметров математической модели -  $x(p)$ . Очевидно, что такая функциональная зависимость имеет место только если система имеет один корень при данном наборе параметров.

При множественном решении системы уравнений (3.15) с разными векторами параметров  $p$ , имеет смысл использовать информацию о найденных корнях с ранее использованными векторами параметров  $p_1, p_2, \dots, p_n$ , с целью редуцирования начального пространства поиска корней. Принимая во внимание большое количество уникальных значений векторов параметров  $p$ , становится очевидно, что применение регулярных структур данных для хранения такой информации является затруднительным. Кроме того, при решении с новым вектором параметров, встает задача прогнозирования пространства возможного нахождения корней.

Целесообразным в данном случае является использование самоорганизующихся вычислительных структур, в частности, нейронных сетей с радиально-базисной передаточной функцией, т.к. такие сети хорошо подходят как для задач интерполяции, так и для задач экстраполяции данных.

Радиально-базисная функция (рис. 3.4) имеет вид:

$$r(n) = e^{-n^2}. \quad (3.16)$$



### Рис. 3.4. Радиально-базисная функция

Выход нейрона с радиально-базисной передаточной функцией (рис. 3.5) определяется по формуле:

$$a = e^{-\left(\|w-p\| \cdot b\right)^2}, \quad (3.17)$$

где  $w$  - вектор весов,  $p$  - вектор входов,  $b$  - масштабный коэффициент.

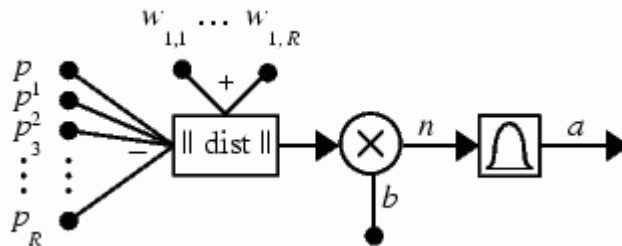


Рис. 3.5. Нейрон с радиально-базисной передаточной функцией.

При использовании аппарата искусственных нейронных сетей, одна из главных проблем – сокращение обучающей выборки, т.к. рост количество обучающих наборов ведет к резкому увеличению числа нейронов в сети и времени обучения. Основным требованием к самой сети ставится удовлетворительная работа не только на обучающей выборке, но и на промежуточных значениях, т.е., сеть должна обладать свойством генерализации.

Обычно для целей нелинейной аппроксимации применяют искусственные нейронные сети с радиальной базисной функцией (radial basis function [47]) конфигураций, представленных на рис. 3.6 и 3.7.

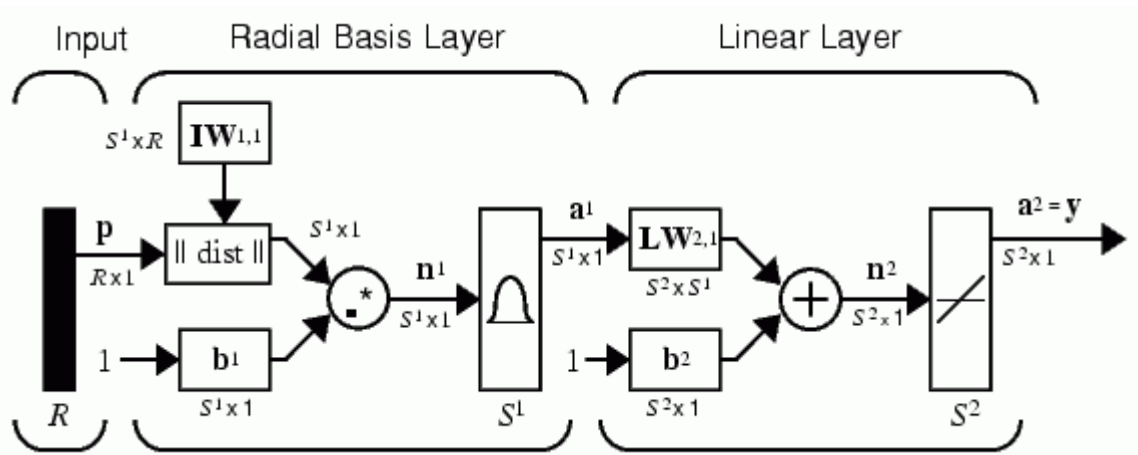


Рис. 3.6. Нейронная сеть с радиально-базисной передаточной функцией

На рис. 3.6 представлена нейронная сеть с радиально-базисной передаточной функцией (Radial Basis Neural Network – RBNN [48]). Здесь блок dist определяет расстояние между векторами входных параметров и весов текущего нейрона.

Вторую сеть еще называют сетью генерализованной регрессии (Generalized Regression Neural Network – GRNN [48]). Блок prod осуществляет векторное умножение вектора выхода первого слоя и вектора весов нейрона второго слоя.

При применении такого рода сетей важным фактором является правильная подборка значений параметра  $b$  (от bias - растяжение) в первом слое (рис. 3.6, 3.7), который определяет «растяжение» радиально-базисной функции по оси аргумента. Чем больше это значение, тем «глаже» будет функция на промежуточных значениях аргумента. Заметим, что в стандартных методах обучения радиально-базисных сетей [48], для всех нейронов первого слоя используется одинаковое значение  $b$ , следовательно, для наилучшего функционирования этих методов, необходимо равномерное распределение обучающей выборки по области определения функции.

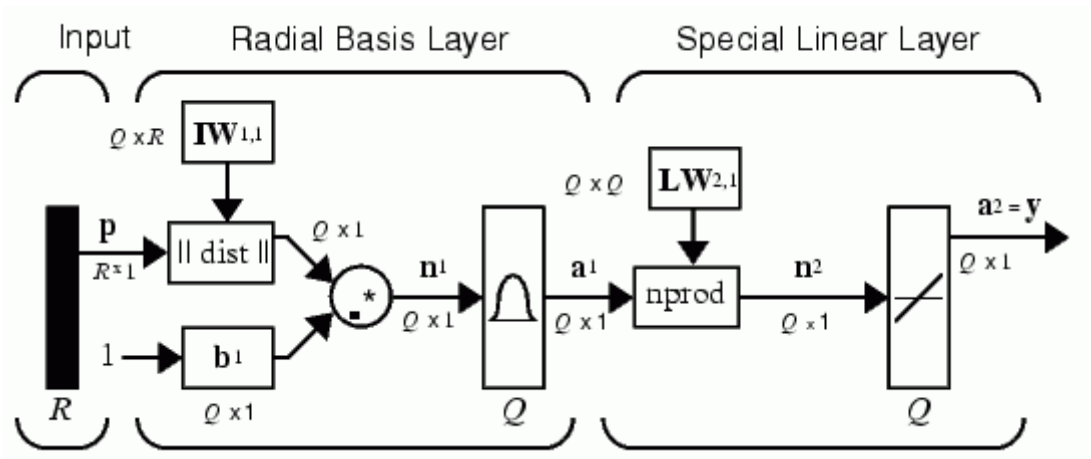


Рис. 3.7. Нейронная сеть генерализованной регрессии

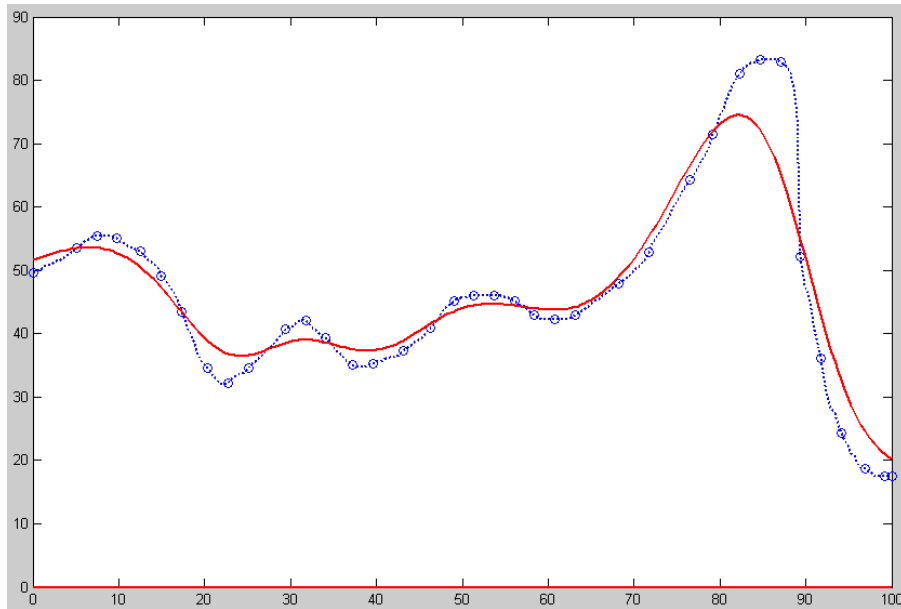


Рис. 3.8. Аппроксимация нелинейной функции при помощи сети генерализованной регрессии с радиально-базисной передаточной функцией

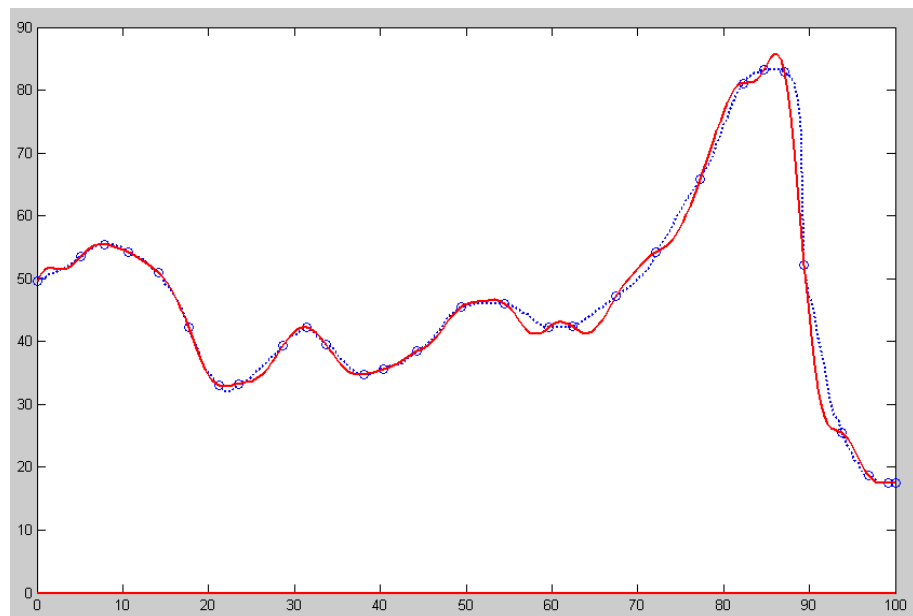


Рис. 3.9. Аппроксимация нелинейной функции при помощи радиально-базисной нейронной сети

Радиально-базисная нейронная сеть (рис.3.9) лучше воспроизводит значения аппроксимируемой функции в точках обучающей выборки, тогда как сеть генерализованной регрессии (рис.3.8) лучше интерполирует функцию на интервалах между точками обучающей выборки. При наличии неравномерности распределения обучающей выборки возможны следующие стратегии повышения качества аппроксимации:

- 1) Использование нескольких сетей для обучения, каждая из которых будет аппроксимировать равномерные участки функции.
- 2) Создание собственного метода обучения сетей, учитывающего неравномерность распределения обучающей выборки.

В случае метода пространственного решения систем нелинейных уравнений, информация о найденных интервалах расположения корней  $x$  уравнения при заданных параметрах системы  $p$  используется для обучения нейронной сети. В качестве алгоритма обучения можно использовать ортогональный метод наименьших квадратов для обучения радиально-базисных нейронных сетей [48].

В ходе обучения нейронной сети необходимо учитывать тот факт, что увеличение обучающей выборки ведет к быстрому росту числа нейронов сети, что, в свою очередь, ведет к расходу памяти и повышенным вычислительным затратам. Поэтому целесообразно намеренно расширять интервалы расположения корней  $x$ , используемые в качестве выхода нейронной сети. Такой подход увеличивает толерантность к ошибкам обучения сети и лишь в незначительной степени влияет на скорость нахождения решения:

$$\frac{|\mathbf{x}_{\max}^* - \mathbf{x}_{\min}^*|}{|\mathbf{x}_{\max} - \mathbf{x}_{\min}|} = N,$$

$$\varepsilon^* = N \cdot \varepsilon,$$

$$I^* = I + \log_2 N,$$
(3.18)

Где  $\mathbf{x}_{\max}, \mathbf{x}_{\min}$  - интервал расположения корней,  $N$  – коэффициент изменения длины интервала расположения корней,  $\varepsilon$  - среднеквадратичная ошибка обучения нейронной сети,  $I$  - количество итераций пространственного метода численного решения нелинейных уравнений, за которое достигается заданная точность. Звездочками обозначены значения названных параметров после изменения длины интервала расположения корней.

Обученная нейронная сеть используется для определения пространства поиска корней. Найденное пространство исследуется на предмет нахождения в нем корня системы уравнений (3.15). Если в найденном пространстве корней нет, осуществляется поиск в начальном пространстве поиска. После этого поиск корней продолжается по алгоритму пространственного численного решения систем нелинейных уравнений.

Предложенный метод использования информации о найденных решениях существенно снижает вычислительные затраты на поиск корней, тем самым улучшая масштабируемость пространственного алгоритма численного решения систем нелинейных уравнений.

Пространственный алгоритм численного решения систем нелинейных уравнений может быть применен к решению систем уравнений в частных производных, используя разностную аппроксимацию производных на расчетной сетке. При этом в качестве входных параметров нейронной сети будут использоваться как параметры

модели, так и значения корней на соседних узлах расчетной сетки, используемых при аппроксимации производных.

### **3.3. Повышение эффективности алгоритма адаптивной подстройки шага численного дифференцирования систем эволюционного типа**

#### **3.3.1. Формирование критерия оптимальности динамической расчетной сетки**

Применение динамических расчетных сеток с нерегулярным шагом ведет к необходимости создания метода оценки, позволяющего комплексно сравнивать алгоритмы построения сетки с позиций неравномерности распределения ячеек и требований к вычислительным ресурсам, который бы позволил усовершенствовать традиционные стратегии формирования шагов численного дифференцирования. Сформулируем критерий оптимальности расчетной сетки:

$$\kappa = \left( \frac{\max_{i=0..n_t} N_i + \min_{i=0..n_t} N_i}{\max_{i=0..n_t} N_i} \right) \cdot \sum_{i=0}^{n_t} N_i \cdot T_i, \quad (3.19)$$

где  $N_i$  - количество ячеек на  $i$ -ом временном слое;  $T_i$  - время, затраченное на расчет  $i$ -го временного слоя;  $n_t$  - количество временных. Алгоритм построения расчетной сетки будем считать тем эффективнее, чем меньше значение  $\kappa$ .

#### **3.3.2. Выявление обобщенного алгоритма адаптивной подстройки шага для двухмерной прямоугольной нерегулярной расчетной сетки**



Сузим пространство поиска алгоритмов построения расчетной сетки до класса алгоритмов адаптивной подстройки шага, а общий вид динамической сетки до двухмерной нерегулярной прямоугольной сетки (рис. 3.1). Здесь серым цветом показаны граничные ячейки расчетной сетки, белым и черным – внутренние ячейки. Значения белых ячеек находятся путем решения исходной системы уравнений (3.1). Значения черных ячеек определяются в результате интерполяции между соседними ячейками.

Рассмотрим алгоритм адаптивной подстройки шага для двухмерного случая. Назовем координаты пространства расчетной сетки условно  $x$  и  $t$ , а шаги по координатам  $s_x$  и  $s_t$  соответственно. Для шагов по каждой из координат сетки, а также для самих координат задаются минимальные и максимальные ограничения:

$$\begin{aligned} x &\in [x_{\min}, x_{\max}], \\ t &\in [t_{\min}, t_{\max}], \\ s_x &\in [s_x^{\min}, s_x^{\max}], \\ s_t &\in [s_t^{\min}, s_t^{\max}]. \end{aligned} \quad (3.20)$$

Вычисление всей сетки начинается с точки  $x = x_{\min}, t = t_{\min}$  и максимальных шагов  $s_x^{\max}, s_t^{\max}$ . При переходе на новый временной слой шагу по пространству  $x$  присваивается максимальное значение  $s_x^{\max}$ . Если вычисление заданного числа ячеек прошло успешно  $\psi_x \geq \Psi_x$ , происходит приращение счетчика успешно рассчитанных ячеек текущего слоя  $\psi_x = \psi_x + 1$  и увеличение шага по формуле:

$$s_x = \min\left(\gamma_x(s_x), s_x^{\max}\right), \gamma_x(s_x) > s_x. \quad (3.21)$$

Затем происходит переход к следующей ячейке расчетной сетки  $x = \min(x + s_x, x_{\max})$ . Если решение, найденное в текущей ячейке, не удовлетворяет заданным требованиям точности, происходит сброс счетчика успешно рассчитанных ячеек текущего слоя  $\psi_x = 0$ , откат на предыдущую ячейку  $x = x - s_x$  и уменьшение шага по формуле:

$$s_x = \max\left(\eta_x(s_x), s_x^{\min}\right), \eta_x(s_x) < s_x. \quad (3.22)$$

С учетом вычисленного шага выбирается новая текущая ячейка  $x = \min(x + s_x, x_{\max})$  и расчет повторяется.

Если в результате уменьшения шага была достигнута нижняя граница  $s_x^{\min}$ , происходит сброс счетчика успешно рассчитанных слоев  $\psi_t = 0$ , откат на предыдущий слой  $t = t - s_t$  и уменьшение шага по формуле:

$$s_t = \max\left(\eta_t(s_t), s_t^{\min}\right), \eta_t(s_t) < s_t. \quad (3.23)$$

С учетом вычисленного шага выбирается новый текущий слой  $t = \min(t + s_t, t_{\max})$  и расчет повторяется. Если была достигнута нижняя граница изменения шага  $s_t^{\min}$ , дальнейшее вычисление считается невозможным и происходит выход из алгоритма.

Если вычисление всего временного слоя прошло успешно, т.е.  $x = x_{\max}$ , происходит приращение счетчика успешно рассчитанных слоев  $\psi_t = \psi_t + 1$ . При достижении заданного порогового значения  $\psi_t \geq \Psi_t$ , происходит увеличение шага по формуле:

$$s_t = \min(\gamma_t(s_t), s_t^{\max}), \gamma_t(s_t) > s_t. \quad (3.24)$$

Затем осуществляется переход на следующий слой  $x = x_{\min}$ ,  $t = \min(t + s_t, t_{\max})$ . Если в результате была достигнута граница расчетной сетки  $t = t_{\max}$ , вычисление считается успешно завершенным и происходит выход из алгоритма.

Блок схема описанного алгоритма адаптивной подстройки шага приведена на рис. 3.10:

- 1) Входными данными являются границы расчетной области  $x_{\min}$ ,  $x_{\max}$ ,  $t_{\min}$ ,  $t_{\max}$ ; границы изменения шагов  $s_x^{\min}$ ,  $s_x^{\max}$ ,  $s_t^{\min}$ ,  $s_t^{\max}$ ; константы, определяющие количество успешно рассчитанных слоев и ячеек до увеличения шага  $\Psi_x$ ,  $\Psi_t$ . Расчет начинается с точки  $(x_{\min}, t_{\min})$  и шагов  $s_x^{\max}$ ,  $s_t^{\max}$ . Далее происходит переход на шаг 2.
- 2) На шаге 2 осуществляется вызов алгоритма *Next*, который осуществляет расчет и добавление в расчетную сетку ячейки с координатами  $(x, t)$ , а также всех дополнительных ячеек динамической сетки (рис. 3.3). В случае успешного расчета происходит переход на шаг 3, в случае неудачи – на шаг 8.
- 3) На шаге 3 выполняется проверка, является ли текущая ячейка с координатами  $(x, t)$  последней ячейкой на временном слое  $t$ . Если это последняя ячейка слоя, происходит переход на шаг 10, в противном случае – на шаг 4.
- 4) На шаге 4 выполняется приращение счетчика успешно рассчитанных ячеек  $\psi_x$  текущего слоя. Далее происходит переход на шаг 5.
- 5) На шаге 5 проверяется, рассчитано ли необходимое количество ячеек для увеличения шага  $s_x$ . Если необходимое количество ячеек рассчитано, происходит переход на шаг 6, иначе – на шаг 7.

- 6) На шаге 6 осуществляется увеличение шага  $s_x$  по формуле (3.21). Далее происходит переход на шаг 7.
- 7) На шаге 7 происходит приращение координаты  $x$  текущей ячейки расчетной сетки. Далее происходит переход на шаг 2.

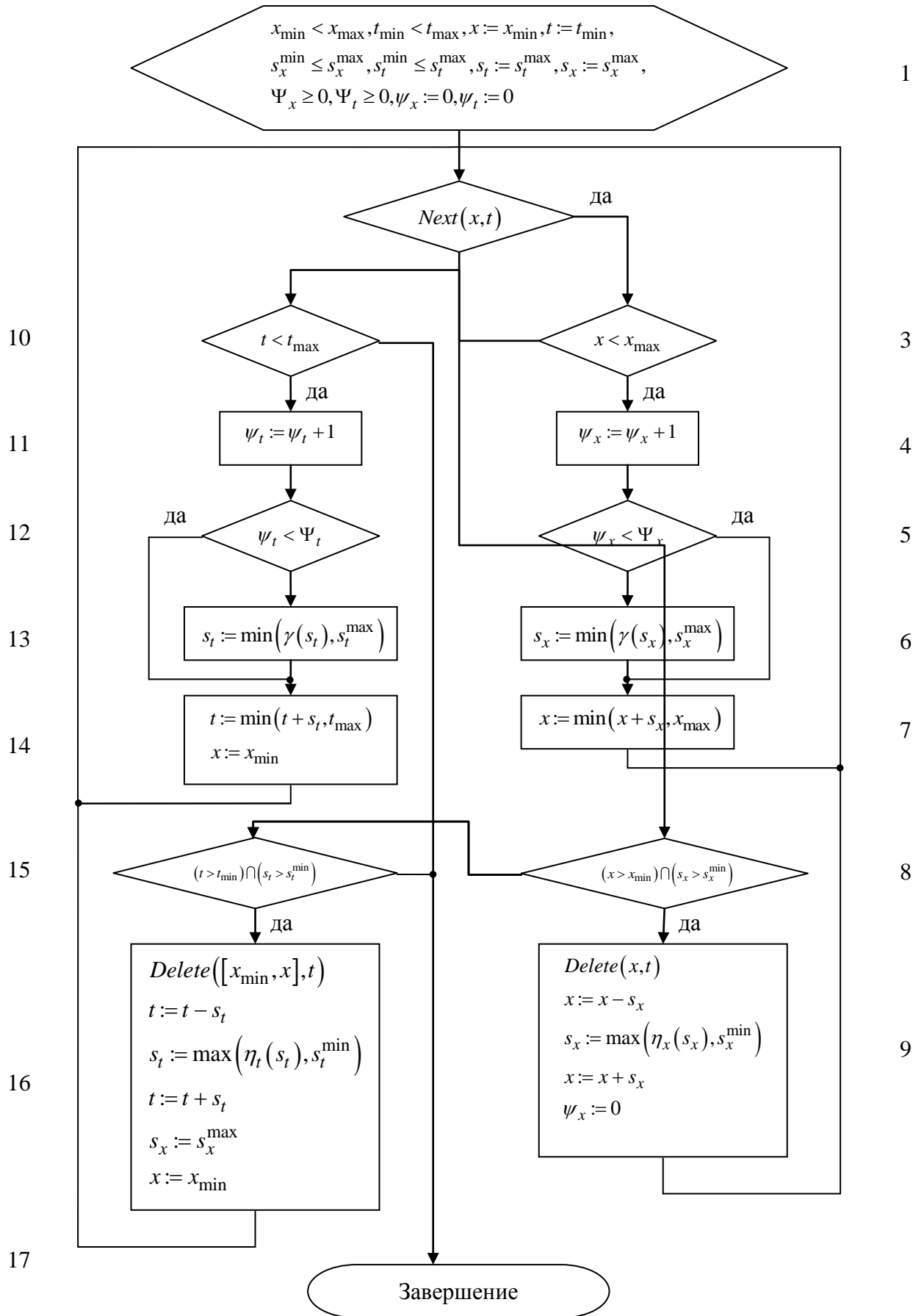


Рис. 3.10. Алгоритм адаптивной подстройки шага

- 8) На шаге 8 осуществляется проверка, не является ли текущая ячейка с координатами  $(x, t)$  первой ячейкой на временном слое  $t$  и превышает ли текущий шаг  $s_x$  минимальную границу изменения шага  $s_x^{\min}$ . Если ячейка не является первой на слое и текущий шаг больше минимальной границы, происходит переход на шаг 9, иначе – на шаг 15.
- 9) На шаге 9 происходит удаление текущей ячейки, а также всех дополнительных ее ячеек из расчетной сетки. Затем уменьшается шаг  $s_x$  по формуле (3.22), выбираются новые координаты текущей ячейки расчетной сетки и сбрасывается счетчик успешно рассчитанных ячеек  $\psi_x$ . Затем происходит переход на шаг 2.
- 10) На шаге 10 осуществляется проверка, является ли текущий временной слой последним в расчетной сетке. Если текущий слой – последний, происходит успешное завершение работы алгоритма на шаге 17, иначе – переход на шаг 11.
- 11) На шаге 11 происходит приращение счетчика успешно рассчитанных временных слоев  $\psi_t$ . Затем осуществляется переход на шаг 12.
- 12) На шаге 12 проверяется, рассчитано ли необходимое количество слоев для увеличения шага  $s_t$ . Если необходимое количество ячеек рассчитано, происходит переход на шаг 13, иначе – на шаг 14.
- 13) На шаге 13 осуществляется увеличения шага  $s_t$  по формуле (3.24). Затем происходит переход на шаг 14.
- 14) На шаге 14 происходит переход на следующий временной слой. Координаты новой текущей ячейки -  $(x_{\min}, t + s_t)$ . Затем происходит переход на шаг 2.
- 15) На шаге 15 выполняется проверка, не является ли текущий временной слой  $t$  первым в расчетной сетке и

превышает ли шаг  $s_t$  минимальную границу изменения шага  $s_t^{\min}$ . Если слой не первый и шаг превышает минимальную границу, осуществляется переход на шаг 16. Иначе происходит неудачное завершение работы алгоритма на шаге 18.

16) На шаге 16 осуществляется удаление всех основных и дополнительных ячеек текущего временного слоя из расчетной сетки. Затем уменьшается шаг  $s_t$  по формуле (3.23), выбираются новые координаты текущей ячейки расчетной сетки, сбрасываются счетчик успешно рассчитанных ячеек  $\psi_x$ , счетчик успешно рассчитанных слоев  $\psi_t$  и происходит переход на шаг 2.

После рассмотрения алгоритмов, лежащих в основе построения численного решения на динамической расчетной сетке, вернемся к поиску оптимального алгоритма адаптивной подстройки шага. Задача состоит в выборе функций  $\eta_x, \eta_t, \gamma_x, \gamma_t$ , при которых достигается максимум критерия оптимальности  $\kappa$ , определяемого по формуле (3.19).

Назовем *симметричными* такие функции изменения шагов, для которых выполняются соотношения:

$$s_i = \eta_i(\gamma_i(s_i)) = \gamma_i(\eta_i(s_i)). \quad (3.25)$$

Для сравнения различных алгоритмов адаптивной подстройки шага, проводится решение ряда модельных задач процесса регенерации неподвижного слоя катализатора (глава 5, 5.1-5.10). Граничные условия по температуре  $T$  (рис. 3.11) определяются по формуле:

$$T_{l_0}(t) := T_0 + T_{amp} \cdot \sin\left(2\pi \cdot \frac{t}{\tau}\right), \quad (3.26)$$

где  $T_0$  берется из таблицы (3.1),  $T_{amp} := 5K$ ,  $\tau := 10c$ . Расчет проводится при параметрах модели из таблицы (3.1) в течение одного периода  $\tau$  функции  $T_{l_0}(t)$ . Шаг по пространству фиксирован:  $s_l := 0.01m$ . Границы изменения шага по времени:  $s_t^{\min} = 10^{-4}c$ ,  $s_t^{\max} = 10^{-1}c$ . Варьированию были подвергнуты функции изменения шага по времени  $\eta_t$  и  $\gamma_t$ . На графиках на осях ординат отражены значения шага по времени  $s_t$ , на осях абсцисс – координаты соответствующих временных слоев.

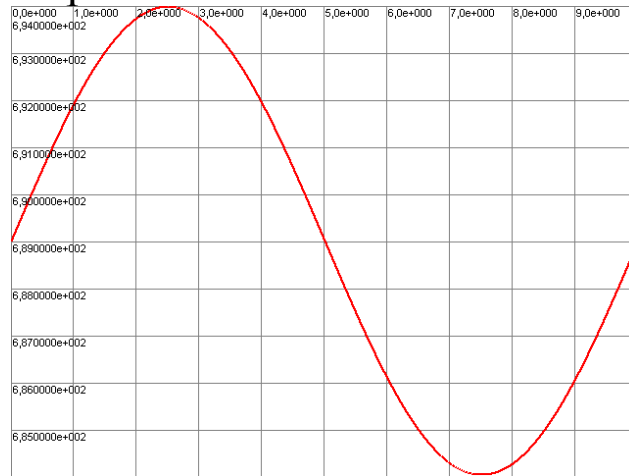


Рис. 3.11. Граничные условия по температуре

- 1) Симметричный экспоненциальный закона изменения шага (рис. 3.12):

$$\eta_t(s_t) = m \cdot s_t, 0 < m < 1; \gamma_t(s_t) = k \cdot s_t, k > 1. \quad (3.27)$$

Расчет проводился при  $m := 0.9, k := 1.1$ .

- 2) Симметричный линейный закон изменения шага (рис. 3.13):



$$\eta_t(s_t) = s_t - m; \gamma_t(s_t) = s_t + m, m > 0. \quad (3.28)$$

Расчет проводился при  $m := 10^{-4}$ .

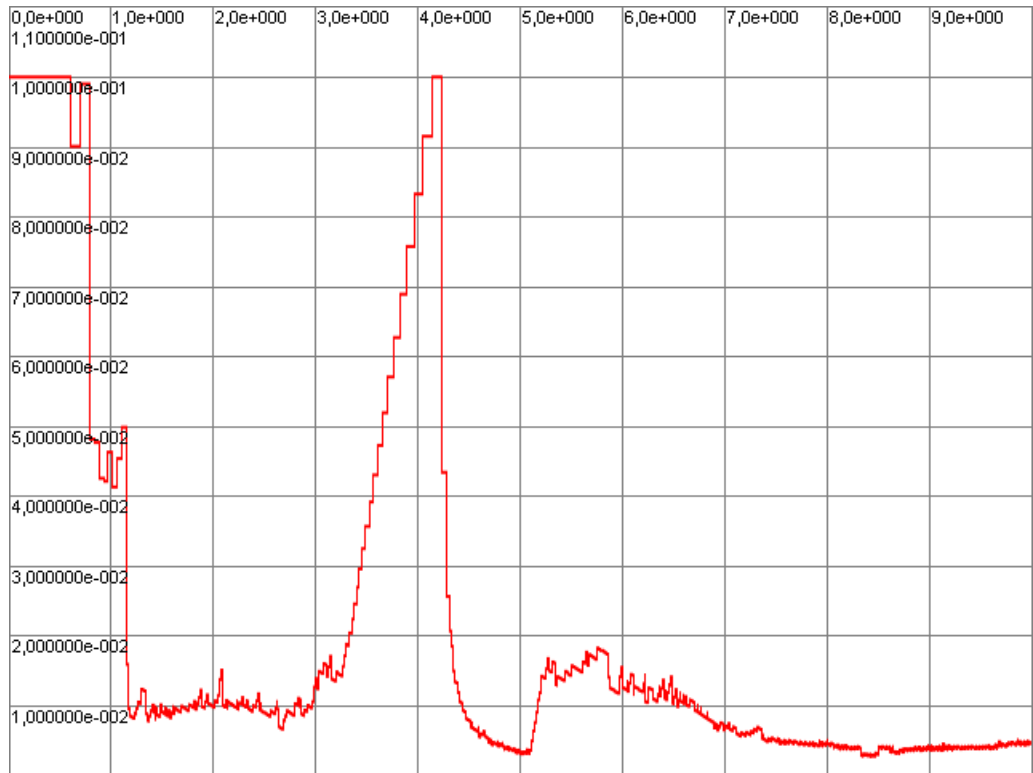


Рис. 3.12. Симметричный экспоненциальный закон изменения шага

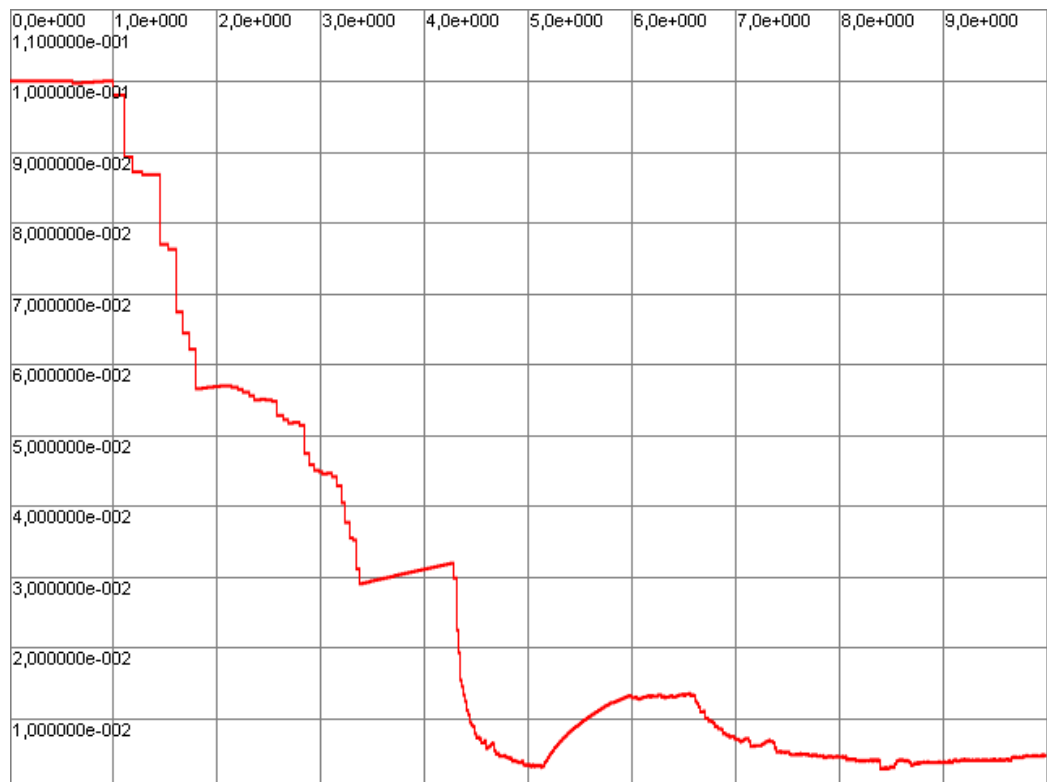


Рис. 3.13. Симметричный линейный закон изменения шага

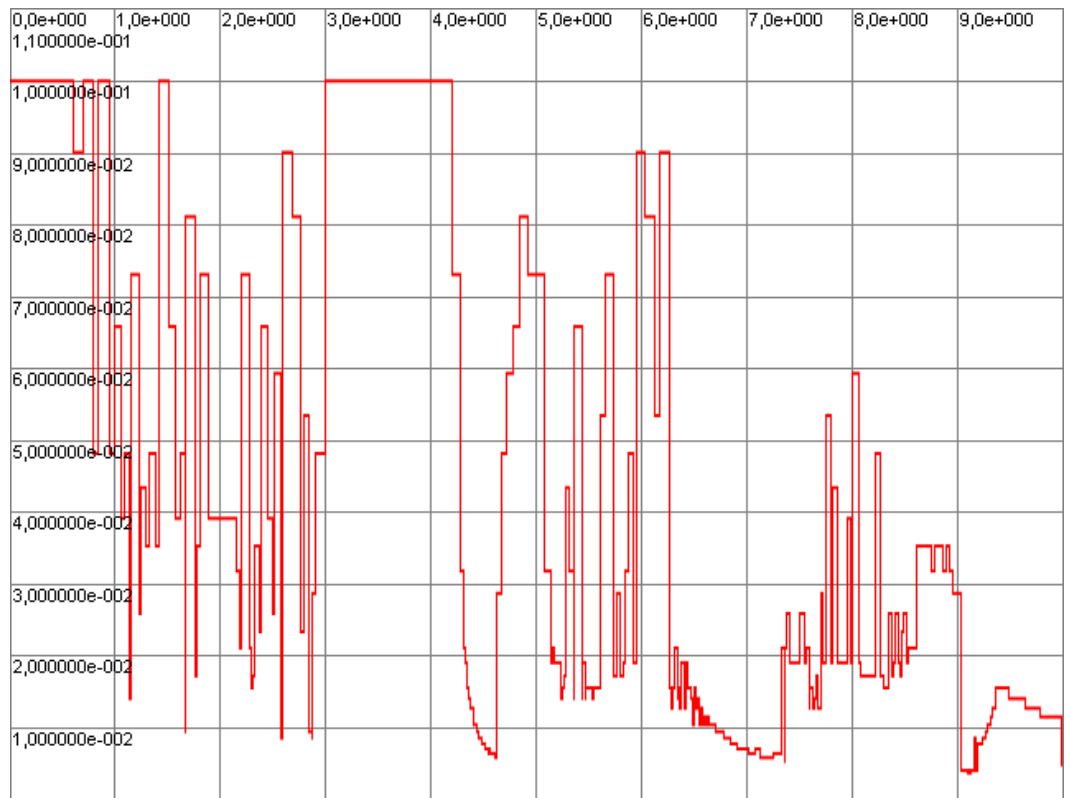


Рис. 3.14. Несимметричный экспоненциальный закон изменения шага

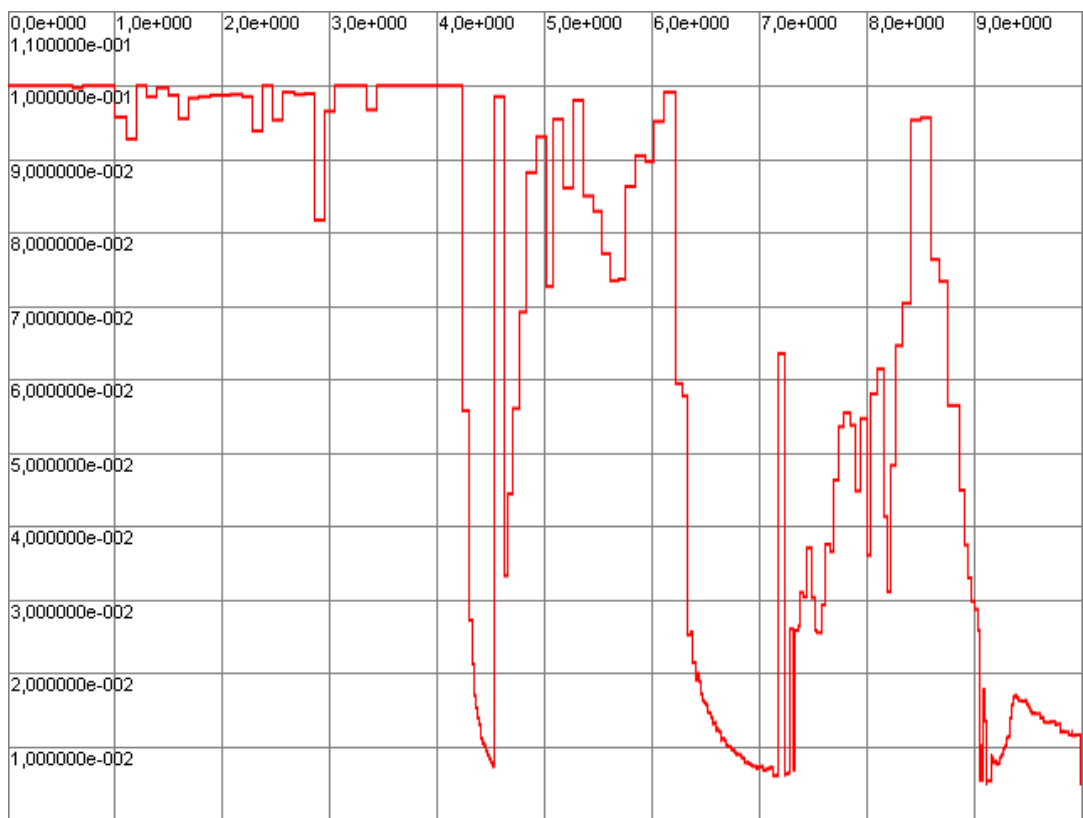


Рис. 3.15. Несимметричный линейный закон изменения шага

3) Несимметричный экспоненциальный закон изменения шага (рис. 3.14):

$$\eta_t(s_t) = m \cdot s_t, 0 < m < 1; \gamma_t(s_t) = s_t^{\max}. \quad (3.29)$$

Расчет проводился при  $m := 0.9$ .

4) Несимметричный линейный закон изменения шага (рис. 3.15):

$$\eta_t(s_t) = s_t - m, m > 0; \gamma_t(s_t) = s_t^{\max}. \quad (3.30)$$

Расчет проводился при  $m := 10^{-4}$ .

Анализ полученных результатов показал, что чем больше площадь фигуры под графиком, т.е.

$$\Lambda = \int_{t_{\min}}^{t_{\max}} s_t(t) dt, \quad (3.31)$$

тем оптимальнее с точки зрения расхода памяти расчетная сетка. Как видно из графиков, наибольшее значение  $\Lambda$  у несимметричного линейного закона изменения шага (3.31), рис. 3.15. Однако, на участках между 6 и 7 с, а также между 9 и 10 с, происходит значительная трата времени на вычисления с шагами от  $s_t^{\max}$  до выбранного  $s_t$ , что не позволяет считать его оптимальным с точки зрения критерия  $\kappa$  (3.19).

### 3.3.3. Разработка эффективного алгоритма адаптивной подстройки шага с памятью

Возникает задача повышения эффективности алгоритмов (3.29) и (3.30) в терминах критерия  $\kappa$ . Из графика 3.15 видно, что большая часть шагов лежит в диапазонах

$$[0.8 \cdot s_t^{\max}, 1.0 \cdot s_t^{\max}] \cup [0.05 \cdot s_t^{\max}, 0.2 \cdot s_t^{\max}]. \quad (3.32)$$

Требуется найти закон  $\eta_t(s_t)$ , осуществляющий быстрый переход между указанными диапазонами (3.32), но «задерживающийся» на самих диапазонах (3.32), что позволит быстро подбирать оптимальный по времени шаг на участках, где линейный закон (3.30) не эффективен с точки зрения времени расчета, и не уступать линейному закону на участках, где он эффективен. Такой закон был бы оптимален для решения рассматриваемой модельной задачи (5.1-5.7) с граничными условиями (3.26). Однако при этом полученный закон нельзя будет назвать оптимальным ни для класса задач, ни даже для других модельных задач на основе (5.1-5.7). В связи с этим, в общем случае также требуется найти алгоритм определения спектрального распределения шагов, т.е. интервалов вида (3.32).

В терминах функции  $\eta_t(s_t)$  новый оптимальный несимметричный закон адаптивной подстройки шага схематично выглядит следующим образом:

$$\begin{aligned} \left| \frac{d\eta_t(s_{t1})}{ds_t} \right| &< \left| \frac{d\eta_t(s_{t2})}{ds_t} \right|, H(s_{t1}) > H(s_{t2}), \\ \gamma_t &= s_t^{\max}, \\ \frac{d\eta_t}{ds_t} &< 0, H(s_t) \geq 0, s_t \in [s_t^{\min}, s_t^{\max}], \end{aligned} \quad (3.33)$$

где  $H(s_t)$  - функция, определяющая количество слоев сетки, рассчитанных с использованием шага  $s_t$ . Т.е. чем чаще

используется данный шаг  $s_{t1}$  в расчетной сетке, тем меньше должна быть производная функции уменьшения шага, и, таким образом, подстройка шага в окрестностях  $s_{t1}$  будет точнее. И наоборот, чем реже используется шаг  $s_{t2}$ , тем быстрее будет происходить уменьшение шага в окрестностях шага  $s_{t2}$ .

Произведена модификация линейного несимметричного закона изменения шага (3.30) с учетом требований (3.33). Для этого константа уменьшения шага  $m$  была заменена функцией  $m(s_t)$ :

$$\begin{aligned}\eta_t(s_t) &= s_t - m(s_t), m > 0, \\ \gamma_t(s_t) &= s_t^{\max}.\end{aligned}\tag{3.34}$$

При этом функция  $m(s_t)$  должна принимать значения, обратно пропорциональные количеству слоев сетки, рассчитанных с использованием шага  $s_t$ :

$$m(s_t) = \frac{a}{b + H(s_t)}, a > 0, b > 0,\tag{3.35}$$

где  $a$  и  $b$  - масштабные коэффициенты, определяющие быстроту изменения шага в зависимости от частоты его использования. С учетом (3.35) закон изменения шага (3.34) запишется следующим образом:

$$\begin{aligned}\eta_t(s_t) &= s_t - \frac{a}{b + H(s_t)}, a > 0, b > 0, \\ \gamma_t(s_t) &= s_t^{\max}.\end{aligned}\tag{3.36}$$

Функция  $H(s_t)$ , характеризующая спектральное распределение шагов по временным слоям расчетной сетки,

должна быть гладкой и непрерывной на промежутке  $[s_t^{\min}, s_t^{\max}]$ :

$$H(s_t) = \sum_{i=1}^n \frac{\alpha \cdot N[s_{ti}]}{\beta + (s_t - s_{ti})^2}, \alpha > 0, \beta > 0, \quad (3.37)$$

где  $\alpha$  и  $\beta$  - масштабные коэффициенты, задающие «пологость» функции,  $N[s_t]$  - ассоциативный массив, определяющий количество временных слоев сетки, рассчитанных с использованием шага  $s_t$ , а  $n$  - количество элементов в массиве  $N$ . Подставив выражение (3.37) в (3.36) получим:

$$\eta_t(s_t) = s_t - \frac{a}{b + \sum_{i=1}^n \frac{\alpha \cdot N[s_{ti}]}{\beta + (s_t - s_{ti})^2}}, a, b, \alpha, \beta > 0, \quad (3.38)$$

$$\gamma_t(s_t) = s_t^{\max}.$$

С учетом (3.38) был модифицирован алгоритм адаптивной подстройки шага. В начале расчета все элементы массива  $N$  обнуляются. С каждым успешно рассчитанным временным слоем, увеличивается на единицу соответствующий элемент массива  $N$ . Аналогично подстройка шага распространяется и на координату  $x$ , при этом масштабные коэффициенты будут другими. Функции изменения шагов для двухмерного случая выглядят следующим образом:

$$\eta_t(s_t) = s_t - \frac{a_t}{b_t + \sum_{i=1}^{n_t} \frac{\alpha_t \cdot N_t[s_{ti}]}{\beta_t + (s_t - s_{ti})^2}}, a_t, b_t, \alpha_t, \beta_t > 0,$$

$$\gamma_t(s_t) = s_t^{\max},$$

$$\eta_x(s_x) = s_x - \frac{a_x}{b_x + \sum_{i=1}^{n_x} \frac{\alpha_x \cdot N_x[s_{xi}]}{\beta_x + (s_x - s_{xi})^2}}, a_x, b_x, \alpha_x, \beta_x > 0,$$

$$\gamma_x(s_x) = s_x^{\max}.$$
(3.39)

В системе реализованы следующие стратегии работы алгоритма:

- 1) Массив  $N_x$  обнуляется на каждом новом временном слое. С каждой успешно рассчитанной ячейкой расчетной сетки увеличивается на единицу соответствующий элемент массива  $N_x$ .
- 2) Во втором случае массив  $N_x$  на новом слое сохраняет значения элементов на предыдущих слоях, что повышает эффективность алгоритма за счет более быстрой адаптации шага.

### **3.3.4. Исследование параметров эффективности алгоритма адаптивной подстройки шага с памятью по сравнению с широко распространенными алгоритмами**

Проведен анализ показателей эффективности алгоритма адаптивной подстройки шага с памятью на решении модельной задачи (3.27-3.30). Шаг  $s_t$  по временной координате изменялся по формуле (3.39). Шаг по пространственной координате изменялся по экспоненциальному закону, аналогично модельным задачам (3.27-3.30). Значения параметров  $a_t, b_t, \alpha_t, \beta_t$  варьировались в следующих диапазонах:



$$\eta_t(s_t) = s_t - \frac{a_t}{b_t + \sum_{i=1}^{n_t} \frac{\alpha_t \cdot N_t[s_{ti}]}{\beta_t + (s_t - s_{ti})^2}}; \quad (3.40)$$

$$\begin{aligned} a_t &:= 2.5 \cdot (i+1), i = 1, 2, 3 \dots 10; \\ b_t &:= 2.5 \cdot (j+1), j = 1, 2, 3 \dots 10; \\ \alpha_t &:= 2.5 \cdot 10^{-6} \cdot (l+1), l = 1, 2, 3 \dots 10; \\ \beta_t &:= 2.5 \cdot 10^{-6} \cdot (k+1), k = 1, 2, 3 \dots 10. \end{aligned} \quad (3.41)$$

Для сравнения работы закона изменения шага (3.39) с законами (3.29) и (3.30) была проведена серия расчетов со следующим варьированием параметров:

$$\eta_t(s_t) := s_t \cdot i \cdot 10^{-4}, i = 1, 2, 3 \dots 1000; \quad (3.42)$$

$$\eta_t(s_t) := s_t - i \cdot 10^{-4}, i = 1, 2, 3 \dots 1000. \quad (3.43)$$

При расчете с каждым набором параметров учитывались следующие характеристики: количество временных слоев расчетной сетки, время вычислений и комплексная оценка  $\kappa$ . Вычисление значения показателя эффективности в данном расчете проводится по формуле 3.19. Лучшие результаты расчетов по указанным характеристикам были сведены в таблицу.

Проанализируем полученные результаты. Последняя колонка таблицы вычислялась по формуле:

$$\frac{\mathbf{n}_i - \min(\mathbf{n})}{\min(\mathbf{n})} \cdot 100\%, \quad (3.44)$$

где  $\mathbf{n}$  - вектор значений параметра, полученный вычислениями с использованием разных алгоритмов.

Как видно из таблицы, минимальное количество ячеек в сетке, а значит и минимальный расход памяти, обеспечивает

несимметричный линейный алгоритм адаптивной подстройки шага (3.42). Однако построение такой сетки осуществляется за неприемлемо большое время. Так, при выигрыше в памяти на 18% по сравнению с алгоритмом с памятью, линейный алгоритм работает более чем в два раза дольше.

Алгоритм	Количество слоев	Время вычисления (с)	Критерий $\kappa$	Разница с лучшим показателем (%)
Минимальное количество слоев:				
Алгоритм с памятью	383	3155	1208365	18%
Экспоненциальный закон	487	1907	928709	50%
Линейный закон	<b>324</b>	6947	2250828	0%
Минимальное время вычисления				
Алгоритм с памятью	529	<b>1392</b>	736368	0%
Экспоненциальный закон	610	1612	983320	16%
Линейный закон	398	4168	1658864	200%
Лучший показатель эффективности $\kappa$				
Алгоритм с памятью	445	1451	<b>645695</b>	0%
Экспоненциальный закон	503	1783	896849	39%
Линейный закон	349	4532	1581668	145%

Вычисление по экспоненциальному закону в данном случае происходит быстрее всего, т.к. в промежутке от 6-й до 10-й секунды (рис. 3.14-3.15) в алгоритмах с медленной

подстройкой шага (3.40, 3.42) тратится значительное время на уменьшение шага от максимального значения до значения, приемлемого для расчета.

При сравнении минимального времени вычисления выигрывает алгоритм с памятью (3.40). Хотя количество ячеек в полученной сетке и больше, чем в результате с минимальным числом ячеек в предыдущем расчете, вычисление всей сетки в данном случае происходит быстрее. Объясняется это тем, что алгоритм более эффективно осуществляет уменьшение шага на участках, где вычисление с большим шагом затруднено (например, на участке от 6-й до 10-й секунды рис. 3.14-3.15). Линейный закон по-прежнему остается самым медленным, а экспоненциальный – на 16% медленнее, чем алгоритм с памятью.

Лучшим, с точки зрения показателя эффективности  $k$  (3.43) работы алгоритмов адаптивной подстройки шага, является алгоритм с памятью. Алгоритм с экспоненциальной подстройкой шага на 40% менее эффективен, что обуславливается повышенными требованиями к машинной памяти и более высокими вычислительными затратами на участках деградации шага. Алгоритм с линейной подстройкой шага, как и в предыдущих измерениях, показал наименьший расход памяти, однако ввиду больших временных затрат, он менее всего подходит для решения данной модельной задачи по сравнению с двумя другими алгоритмами.

После анализа полученных результатов можно сделать вывод, что алгоритм адаптивной подстройки шага с памятью (3.40) подходит для решения модельной задачи регенерации неподвижного слоя катализатора (5.1) лучше, чем алгоритмы (3.41) и (3.42). Однако эффективность работы алгоритма во многом зависит от значений масштабных коэффициентов  $a, b, \alpha, \beta$  (3.40), которые подбираются в соответствии с каждой конкретной модельной задачей для обеспечения лучшей эффективности алгоритма в соответствии с критерием (3.19).

Таким образом, был получен новый алгоритм адаптивной подстройки шага с памятью (3.40), более

эффективный для решения модельной задачи (5.1), чем используемые ранее алгоритмы (3.41) и (3.42). В ходе дальнейшей работы необходимо показать на каком классе задач данный алгоритм наиболее эффективен.

### **3.4. Методология параметрической оптимизации эволюционных динамических систем**

#### **3.4.1. Постановка задачи параметрической оптимизации**

Под параметрической оптимизацией далее понимается процесс однократного достижения экстремальной цели в предположении стационарности экстремальной характеристики объекта оптимизации и конечномерности пространств входных и выходных параметров. При этом сам объект оптимизации может реально существовать либо представлять собой математическую модель. С другой стороны, сам объект оптимизации можно рассматривать как статический объект оптимального управления с постоянными входными и выходными сигналами. Рассматриваемые объекты параметрической оптимизации далее будут характеризоваться совокупностью непрерывных параметров (предполагается, что ограничения типа дискретности отсутствуют), которые условно можно разделить на три группы: входные (управляемые)  $x$ , внешние  $\xi$  и выходные  $y$ .

Входными параметрами  $x = (x_1, x_2, \dots, x_n)$  называются изменяемые в процессе оптимизации параметры, играющие роль управляющего воздействия при рассмотрении объекта оптимизации как объекта управления. Внешние параметры  $\xi = (\xi_1, \xi_2, \dots, \xi_n)$  характеризуют неопределенность обстановки. Отдельные компоненты вектора  $\xi$  могут иметь как случайный, так и неслучайный характер. Содержательный смысл введения внешних параметров состоит в описании следующих характерных явлений:

- 1) влияния случайных отклонений при установке заданных (номинальных) значений  $x_i$  на реальном объекте;
- 2) случайных воздействий внешней среды на объект, изменяющих реальные выходные характеристики по отношению к расчетным;
- 3) влияний изменяющихся, как правило, не случайным, а заранее неизвестным образом, условий функционирования объекта, например таких, как температура, влажность, уровень радиации и т.д.

Вектор выходных параметров  $y = (y_1, y_2, \dots, y_m)$  позволяет количественно оценить основные характеристики оптимизируемого объекта. С точки зрения общей теории управления, вектор  $y$  характеризует состояние объекта оптимизации. Далее предполагается, что существует функциональная связь

$$y = \varphi(x, \xi), \quad (3.45)$$

определяющая оператор объекта оптимизации и позволяющая по заданным  $x$  и  $\xi$  рассчитать соответствующий вектор  $y$  (задача анализа). Как правило, оператор объекта  $\varphi$  задан алгоритмически. Заметим, что мы здесь, вообще говоря, отличаем объект оптимизации от объекта управления в общей схеме управления.

Задача параметрической оптимизации в общем случае ставится как многокритериальная задача с ограничениями:

$$y_i(x, \xi) \rightarrow \min_x, i \in [1:k], x \in D \subset R^n, \quad (3.46)$$

$$D = \{x \in R^n \mid g_i(x, \xi) \leq 0, i \in [1:m], g_j(x, \xi) = 0, j \in [m+1:S]\}.$$

Множество  $\{y_1, y_2, \dots, y_k\}$  образует множество критериальных выходных параметров, имеющих смысл частных критериев оптимальности и характеризующих

качество оптимизации. Наличие нескольких частных критериев, по существу, отражает ту неопределенность цели, которая явно или неявно присутствует при оптимизации любого сколько-нибудь сложного объекта.

Интуитивный смысл задачи (3.46) состоит в выборе такого вектора  $x$  из допустимого множества  $D$ , чтобы каждый из критериальных выходных параметров принял по возможности меньшее значение. Математическое решение задачи (3.46) в виде конкретного вектора  $x^* \in D$ , вообще говоря, не существует, так как критериальные выходные параметры отражают противоречивые (конфликтные) требования к объекту оптимизации, и минимумы соответствующих функционалов достигаются в различных точках. Это предположение правомерно, так как в противном случае, если, например, две выходные функции  $y$  имеют минимумы в одной и той же точке, то одна из них может не рассматриваться.

Допустимое множество  $D$  формируется на основе трех групп содержательно различных ограничений, имеющих вид равенств либо неравенств и задаваемых в (3.46) с помощью функций  $g_i, i \in [1:S]$ .

Прямые или аргументные ограничения накладываются непосредственно на компоненты вектора входных параметров:

$$a_i \leq x_i \leq b_i; a_i, b_i \in R^1. \quad (3.47)$$

В более общем случае границы интервалов могут быть функциями от других входных параметров:

$$a_i(x_j) \leq x_i \leq b_i(x_j); i \neq j. \quad (3.48)$$

Ограничения (3.47), (3.48) встречаются наиболее часто и вызываются причинами, связанными в основном с условиями

физической реализуемости необходимых входных параметров.

Функциональные ограничения включают условия работоспособности, имеющие принципиальное значение при оценке правильности функционирования объекта оптимизации, исходя из его функционального назначения. Эти ограничения имеют вид

$$y_i \leq t_i; t_i \in R^1; i \in [1: L]. \quad (3.49)$$

Критериальные ограничения имеют вид

$$y_l \leq t_l; t_l \in R^1; l \in [1: K]. \quad (3.50)$$

и отражают требования к характеристикам качества объекта оптимизации, подлежащие безоговорочному выполнению и приобретающие по существу характер функциональных ограничений.

Основное отличие функциональных ограничений от критериальных заключается в следующем. Как правило, выполнение неравенств (3.49) с большим запасом не требуется; важно только гарантировать их выполнение. С другой стороны, в силу критериального характера выходных параметров  $y$  в (3.6) необходимо добиваться максимально возможных запасов при выполнении соответствующих неравенств.

Ввиду сложности нахождения аналитического решения задачи (3.46), в основном осуществляют численное решение. В интерактивной системе моделирования предусмотрена возможность оптимизации модели с использованием аргументных ограничений вида (3.47). Входными параметрами оптимизации являются глобальные параметры модели, которые могут играть роль параметров функций изменения граничных условий или коэффициентов модели.

Критерии оптимизации могут быть как функциями, определенными на текущей ячейке, так и функциями, учитывающими множество ячеек расчетной сетки. Условно можно выделить три режима параметрической оптимизации в интерактивной системе моделирования EQU:

- 1) оптимизация в каждой ячейке,
- 2) оптимизация на каждом временном слое,
- 3) оптимизация на всей расчетной сетке.

Оптимизация в ячейке дает распределение оптимальных значений параметров по времени и пространству. Результаты такой оптимизации можно использовать скорее на стадии проектирования, чем для управления в реальном времени. Например, можно геометрически оптимальным образом построить реактор (входным параметром оптимизации является радиус реактора на каждом участке длины).

Оптимизация на временном слое дает распределение оптимальных настроек параметров во времени. Такая оптимизация наиболее подходит для моделирования управления в реальном времени, т.к., по сути, ее результатом является форма оптимального управляющего воздействия. Однако не всегда удается подобрать критерии, удовлетворение которых в каждый момент времени приводит к удовлетворению глобальных критериев, отражающих весь моделируемый временной промежуток.

Результатом оптимизации на всей расчетной сетке в общем случае является такое распределение входных параметров по времени и пространству, которое приводит к удовлетворению глобальных критериев оптимизации. Например, в качестве такого критерия может выступать наименьшее время протекания реакции. Практическую сложность вызывает большая размерность вектора входных параметров и необходимость многократно проводить полное моделирование на всей расчетной сетке на каждом шаге оптимизации. Поэтому на практике рациональнее минимизировать количество управляющих параметров.



Например, задавать не значение граничных условий в каждый момент времени, а параметры аналитически заданной функции, значения которой будут определять граничные условия.

### 3.4.2. Редуцирование задачи многокритериальной оптимизации к однокритериальной задаче

Возьмем в качестве примера задачу безусловной оптимизации по трем критериям:

$$\begin{aligned} I_1(\mathbf{x}) &\rightarrow \min, \\ I_2(\mathbf{x}) &\rightarrow \max, \\ a_1 &\leq I_3(\mathbf{x}) < a_2, \end{aligned} \quad (3.51)$$

т.е. в процессе оптимизации критерий  $I_1$  необходимо минимизировать, критерий  $I_2$  - максимизировать, а критерий  $I_3$  при этом должен находиться в интервале  $[a_1, a_2)$ . Пространство оптимизации – вектор параметров модели  $\mathbf{x}$ .

Требования к критериям оптимизации зачастую бывают противоречивы. Например, в задаче (3.51), значение вектора  $\mathbf{x}$ , при котором достигается глобальный минимум критерия  $I_1$ , не обязательно будет совпадать со значением  $\mathbf{x}$ , при котором критерий  $I_2$  достигает глобального максимума. Обычно довольствуются тем, что ищут оптимум одного критерия, накладывая на остальные критерии ограничения. Поэтому, в общем виде, задача многокритериальной оптимизации сводится к задаче однокритериальной оптимизации. Задачу (3.51) в терминах одного критерия можно переписать в виде:

$$\begin{cases} \mathbf{x} = \arg \min_{\mathbf{x}} I(\mathbf{x}), \\ I(\mathbf{x}) = |\alpha_1 \cdot I_1(\mathbf{x})| + |\alpha_2 \cdot I_2^{-1}(\mathbf{x})| + |\alpha_3 \cdot I_3^*(\mathbf{x})|, \\ I_3^*(\mathbf{x}) = 1, a_1 \leq I_3(\mathbf{x}) < a_2, \\ I_3^*(\mathbf{x}) = 0, a_1 > I_3(\mathbf{x}) \geq a_2, \end{cases} \quad (3.52)$$

где  $\alpha_1$ ,  $\alpha_2$  и  $\alpha_3$  - весовые коэффициенты, определяющие влияние каждого критерия на результат оптимизации.

Наличие ограничений на вектор  $\mathbf{x}$  предполагает введение дополнительных критериев оптимизации, учитывающих эти ограничения. Рассмотрим задачу нелинейной оптимизации вида

$$J(\mathbf{x}) \rightarrow \min_{\mathbf{x} \in D}, D = \{ \mathbf{x} \in \mathbb{R}^n \mid h_i(\mathbf{x}) = 0, i \in [1:q] \}, \quad (3.53)$$

не содержащую ограничений в виде неравенств. Наиболее перспективным общим методом учета ограничений считается метод модифицированных функций Лагранжа [49]. Применительно к задаче (3.53) он формулируется следующим образом: введем в качестве обобщенного критерия оптимальности функционал

$$M(\mathbf{x}, \lambda, \sigma) = J(\mathbf{x}) + \langle \lambda, h(\mathbf{x}) \rangle + \frac{\sigma}{2} \|h(\mathbf{x})\|^2, \quad (3.54)$$

где  $\sigma > 0$  - параметр метода. Тогда алгоритм оптимизации сводится к итерационному процессу

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg \min_{\mathbf{x}} M(\mathbf{x}, \lambda^k, \sigma), \\ \lambda^{k+1} &= \lambda^k + \sigma h(\mathbf{x}^{k+1}), h(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_q(\mathbf{x})], \end{aligned} \quad (3.55)$$

обобщающему методы штрафных функций и множителей Лагранжа. Основная особенность сформулированного алгоритма по сравнению с методом штрафных функций заключается в отсутствии неограниченно растущего коэффициента  $\sigma$ . В методе модифицированных функций Лагранжа величина  $\sigma$  влияет лишь на скорость сходимости, но не на сам факт сходимости последовательности  $\{x_k\}$  к оптимуму. При решении практических задач значение  $\sigma$  целесообразно подбирать в интерактивном режиме, так как надежные методы априорного задания  $\sigma$  в настоящее время отсутствуют.

### 3.4.3. Явление овражности и методы распознавания овражной ситуации

Рассмотрим следующий пример критерия оптимальности, зависящего от двух управляемых параметров  $x_1, x_2$  [50]:

$$J(x_1, x_2) = g_0^2(x_1, x_2) + \sigma g_1^2(x_1, x_2) \rightarrow \min_x, \quad (3.56)$$

где  $\sigma$  - достаточно большое положительное число. Рассмотрим также уравнение

$$g_1(x_1, x_2) = 0, \quad (3.57)$$

определяющее в простейшем случае некоторую зависимость  $x_2 = \varphi(x_1)$ . Тогда при стремлении параметра  $\sigma$  к бесконечности значение функционала  $J$  в каждой точке, где  $g_1(x_1, x_2) \neq 0$ , будет неограниченно возрастать по абсолютной величине, оставаясь ограниченным и равным  $g_0^2(x_1, x_2)$  во всех

точках на кривой  $x_2 = \varphi(x_1)$ . То же самое будет происходить с нормой вектора градиента  $J'(x) = \left[ \frac{\partial J}{\partial x_1}, \frac{\partial J}{\partial x_2} \right]$ , где

$$\frac{\partial J}{\partial x_1} = 2g_0(x_1, x_2) \frac{\partial g_0}{\partial x_1} + 2\sigma g_1(x_1, x_2) \frac{\partial g_1}{\partial x_1};$$

$$\frac{\partial J}{\partial x_2} = 2g_0(x_1, x_2) \frac{\partial g_0}{\partial x_2} + 2\sigma g_1(x_1, x_2) \frac{\partial g_1}{\partial x_2}.$$

Линии уровня  $J(x) = \text{const}$  для достаточно большого  $\sigma$  представлены на рис. 3.16. Там же стрелками показано векторное поле антиградиентов, определяющее локальные направления наискорейшего убывания  $J(x)$ .

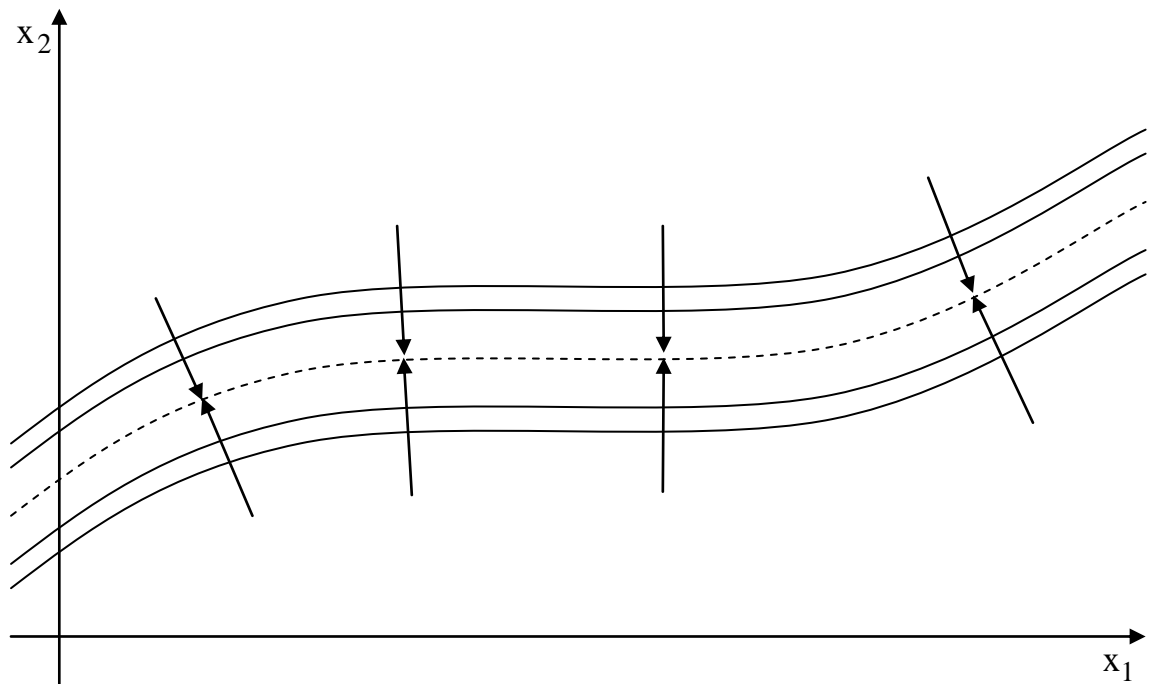


Рис. 3.16. Явление овражности

Ясно, что при достаточно больших  $\sigma$  минимальные значения  $J(x)$  следует искать вдоль зависимости  $x_2 = \varphi(x_1)$ , определяющей так называемое *дно оврага* [41]. Из (3.56) следует, что изменение  $J(x)$  вдоль дна задается выражением  $g_0^2(x_1, x_2)$  и не зависит от величины параметра  $\sigma$ . Таким образом, задача минимизации  $J(x)$  сводится к минимизации функционала  $g_0^2(x_1, \varphi(x_1))$  от одной переменной  $x_1$ . В общем случае уравнение  $x_2 = \varphi(x_1)$  обычно неизвестно.

Приведенный пример *овражной ситуации* является достаточно простым, хотя и из него уже видны принципиальные трудности, связанные, например, с применением широко распространенных методов спуска по антиградиенту. Действительно, из рис. 3.16 следует, что направления поиска, задаваемые антиградиентами, оказываются неэффективными. Приводя довольно быстро процесс поиска на дно оврага, они в окрестности дна начинают осциллировать, оставаясь почти перпендикулярными направлению в точку минимума.

Пусть решается задача  $J(x) \rightarrow \min_x, J \in C^2(D), x \in D \subset \mathbb{R}^n$ . Будем предполагать далее, что функционал  $J(x)$  ограничен снизу на  $D$ . Траектория наискорейшего спуска  $x(\tau)$  функционала  $J(x)$  задается дифференциальным уравнением:

$$\frac{dx}{d\tau} = -J'(x), J'(x) \square \left[ \frac{\partial J}{\partial x_1}, \dots, \frac{\partial J}{\partial x_n} \right]. \quad (3.58)$$

**Определение 1.** Функционал  $J(x)$  называется *овражным*, если отвечающая ему система дифференциальных уравнений (3.58) – жесткая [41].

**Определение 2.** Функционал  $J(x) \in C^2(D), D \subset \mathbb{R}^n$  называется *овражным* в множестве  $Q \in D$ , если найдутся такие числа  $\delta > 0, \sigma \gg 1$  и множество  $Q_\delta \subset D$ , что

$$\begin{aligned}
& 1. \forall x \in Q_\delta, \lambda_1[J''(x)] \geq \sigma |\lambda_n[J''(x)]|; \\
& 2. \forall x \in Q, \text{Arg} \min_{x' \in X_\delta(x)} J(x') \subset Q; \\
& 3. \forall x \in Q, L[X_\delta(x) \cap Q] \leq \sigma^{-1} L[X_\delta(x)],
\end{aligned}
\tag{3.59}$$

где  $X_\delta(x) = \{x' \in \mathbb{R}^n \mid \|x' - x\| \leq \delta\}$ ;  $Q_\delta = \bigcup_{x \in Q} X_\delta(x)$ ,  $\lambda_i[A]$  - собственные числа матрицы  $A = J''(x)$ , упорядоченные по убыванию:

$$\lambda_1[A] \geq \lambda_2[A] \geq \dots \geq \lambda_n[A];$$

$L[S]$  - константа Липшица в соотношении

$$\|J'(x') - J'(x)\| \leq L[S] \|x' - x\|, \forall x', x \in S \subset \mathbb{R}^n.$$

Множество  $Q$  называется *дном оврага*.

**Определение 3.** Пусть  $\forall x \in Q, \det J''(x) \neq 0$ . Наименьшее из чисел  $\sigma$ , удовлетворяющих определению 2, называется *степенью овражности*  $J(x)$  в  $Q$  и обозначается  $\eta(Q)$ .

Отношение  $\eta(x) = \lambda_1(x) \left| \min_i \lambda_i(x) \right|^{-1}$ ,  $x \in Q$ , называется *локальной степенью овражности* в точке  $x$ . Для вырожденных матриц  $J''(x)$  величина  $\eta(x)$  полагается равной  $\infty$  [51].

Согласно [41], овражные задачи считаются плохо обусловленными экстремальными задачами. Традиционно принято классифицировать задачу как плохо обусловленную, если  $\log_2 \eta > t$ , где  $t$  - длина разрядной сетки вычислительной машины. Однако и при меньших значениях  $\eta$  для целого ряда алгоритмов могут возникать значительные вычислительные трудности, особенно если овражность сопровождается отсутствием выпуклости  $J(x)$ .

Рассмотрим практические методы распознавания овражной ситуации, играющие роль *критериев овражности*. Наиболее существенной характеристикой оказывается значение показателя  $\eta$  в допустимой области изменения управляемых параметров.

Своеобразным индикатором может служить метод простого градиентного спуска, реализуемый по схеме

$$x^{k+1} = x^k - hJ'(x^k) \quad (3.60)$$

с постоянным шагом  $h \in \mathbb{R}^1$ .

Принадлежность  $J(x)$  к классу овражных в этом случае проявляется в необходимости применения относительно малых значений  $h$ . Попытки увеличения  $h$  вызывают потерю свойства релаксационности последовательности  $\{J(x^k)\}$ , и значения  $J(x^k)$  начинают резко возрастать. Если для некоторого фиксированного  $h$  (наибольшего из возможных) удалось заставить процесс (3.60) протекать без полной остановки, то по результатам работы метода можно количественно оценить величину  $\eta$ .

Для этого процесс (3.60) продолжается до тех пор, пока отношение  $\frac{\|J'(x^{k+1})\|}{\|J'(x^k)\|}$  не стабилизируется около некоторого значения  $\mu$ . Тогда справедливо следующее равенство:

$$\eta \cong \frac{2}{|1-\mu|}. \quad (3.61)$$

Соотношение (3.61) справедливо независимо от выпуклости функционала  $J(x)$  и является основным для грубой практической оценки степени овражности решаемой задачи в окрестности текущей точки. Доказательство соотношения (3.61) дано в [41].

В силу изложенного целесообразно начинать процесс оптимизации с помощью метода простого градиентного спуска. Если задача простая и степень овражности невелика, то уже этот метод довольно быстро приведет в малую окрестность оптимума. В противном случае будет получена

оценка  $\eta$ , что позволит правильно оценить ситуацию и выбрать наиболее рациональный алгоритм.

#### **3.4.4. Методология решения общей задачи однокритериальной оптимизации динамических систем эволюционного типа**

Далее рассматривается класс однокритериальных задач безусловной оптимизации, к которым, как было показано ранее, сводятся многие более сложные ситуации. Основные факторы, которые учитываются при выборе метода, - это машинное эpsilon  $\varepsilon_m$ , характеризующее точность вычислений, размерность пространства поиска  $n$  и степень овражности  $\eta$  минимизируемого функционала (рис. 3.17).

После ввода исходных данных определяется машинное эpsilon. Далее применяются стартовые алгоритмы на основе метода циклического покоординатного спуска и метода простого градиентного спуска. Если задача алгоритмически проста, то уже на этом этапе может быть получено разумное приближение к искомому результату. На практике обычно происходит резкое замедление сходимости задолго до достижения приемлемых точек в пространстве поиска. В этом случае, как это было показано ранее, на основе результатов работы метода простого градиентного спуска может быть получена оценка степени овражности минимизируемого функционала. После этого необходимо осуществить переход к более мощным вычислительным процедурам. Если размерность решаемой задачи превышает некоторый заданный порог  $N$ , определяемый объемом доступной памяти компьютера, то производим переход к алгоритмам типа RELCH [41]. В противном случае определяется «информативность» матрицы Гессе на основе проверки известного неравенства, содержащего степень овражности и машинное эpsilon.



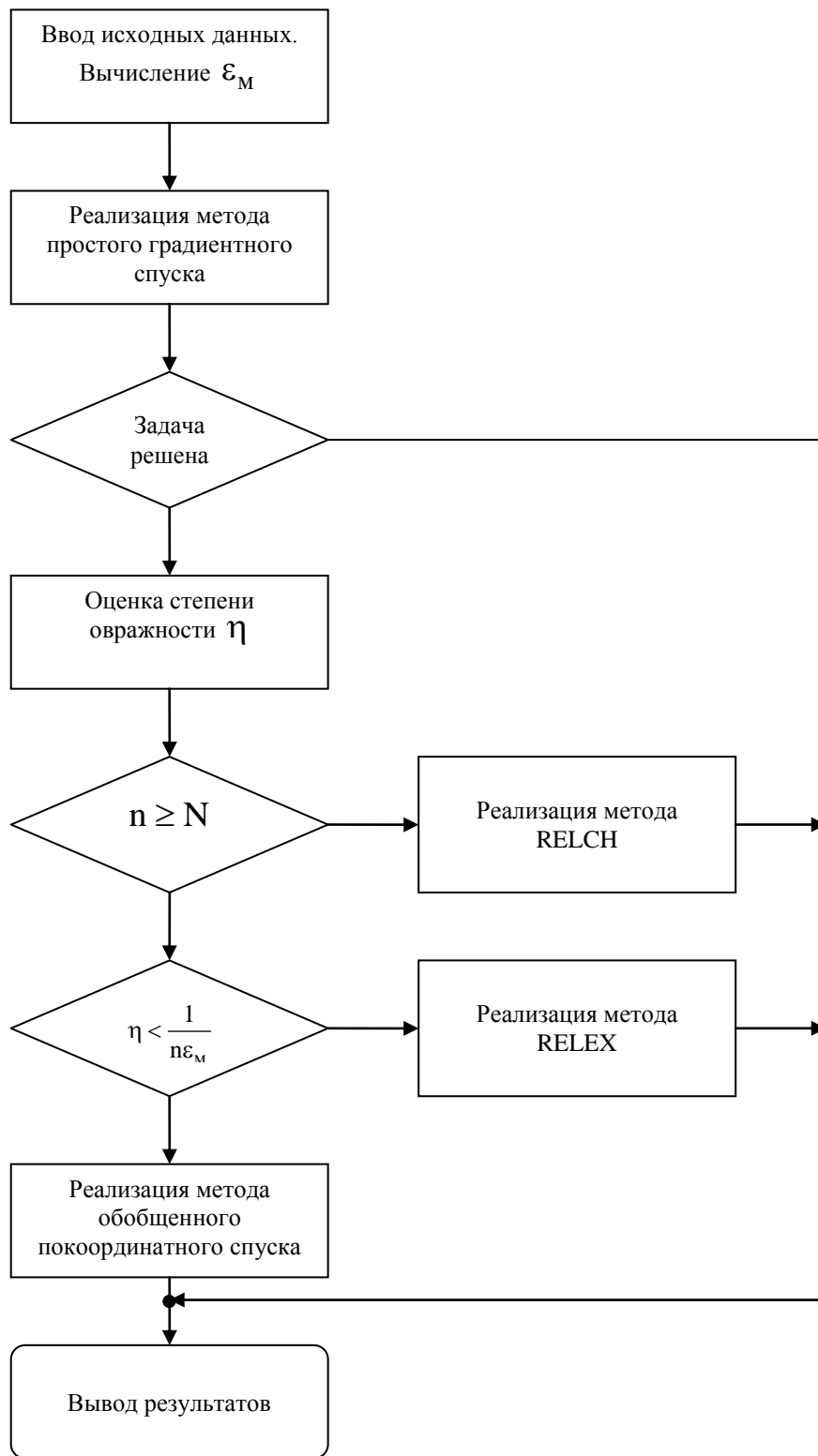


Рис. 3.17. Тактика решения общей задачи однокритериальной параметрической оптимизации

Если матрица Гессе информативна, обращаемся к методам с экспоненциальной релаксацией RELEX [41], в противном случае – к различным вариантам методов обобщенного покоординатного спуска, как к наиболее мощным и универсальным процедурам.

### **3.5. Вывод эволюционных динамических систем из экстремальных ситуаций**

Важной составляющей исследования динамических эволюционных систем является обнаружение экстремальных ситуаций и формирование управляющих воздействий, выводящих систему из экстремального режима в ходе вычислительного эксперимента. Такого рода экстремальными ситуациями могут являться: выход из допустимого температурного диапазона, критическое падение производительности системы, превышение максимально допустимой нормы расхода сырья и т.д. Возникает необходимость осуществлять обнаружение экстремальной ситуации, а также формировать необходимую корректировку граничных условий эволюционной динамической системы в автоматическом режиме.

Численное решение задачи (1.1) осуществляется на динамической расчетной сетке, представленной множеством ячеек с координатами  $D = \{0 \dots i_{\max}, 0 \dots j_{\max}^i\}$ , причем временная граница  $i_{\max}$  является плавающей и определяется критерием останова расчета. Множество ячеек, в которых решение найдено, имеет координаты  $M = \{0 \dots i-1, 0 \dots j_{\max}^i\} \cup \{i, 0 \dots j\}$ ,  $M \subset D$ , где  $i, j$  - координаты текущей ячейки. Под экстремальной ситуацией будем понимать выполнение условия:

$$\prod_{\substack{j=0 \dots j_{\max}^i \\ i=i_{t-\tau_j} \dots i_t}} J(x_j^i) > 0, 0 \leq \tau_j \leq t, \quad (3.62)$$

где  $J$  - критерий экстремальности, принимает только неотрицательные значения;  $\tau_J$  - глубина расчета критерия экстремальности по времени;  $i_{t-\tau_J}$  - номер временного слоя, соответствующий времени  $t - \tau_J$ . В случае возникновения экстремальной ситуации необходимо сформировать управляющее воздействие на канале граничных условий  $f_\Gamma$  на участке  $i = i_{t-\tau_J} \dots i_t$ , которое приведет к выполнению равенства:

$$\begin{aligned} & \sum_{j=0 \dots j_{\max}^i} J \left( \mathbf{x}_j^i \right) = 0, 0 \leq \tau_J \leq t, \\ & \mathbf{x}_{j_0}^i = \mathbf{f}_\Gamma \left( t_i, l_{j_0} \right) + \mathbf{f}_\Gamma^* \left( t_i, l_{j_0}, \mathbf{p} \right), i = i_{t-\tau_J} \dots i_t, j_0 = 0 \dots n-1, \end{aligned} \quad (3.63)$$

где  $\mathbf{f}_\Gamma^*$  - искомая поправка к граничным условиям динамической эволюционной системы,  $\mathbf{p}$  - вектор параметров. Существует несколько подходов к расчету критерия экстремальности  $J$  на этапе обнаружения экстремальных ситуаций.

- 1) Простейший метод предполагает расчет критерия  $J$  непосредственно на множестве найденных решений  $M$ . В случае возникновения экстремальной ситуации, проводится перерасчет области  $\{i_{t-\tau} \dots i_t, 0 \dots j_{\max}^i\}$  с учетом поправки к граничным условиям  $\mathbf{f}_\Gamma^*$ .
- 2) Другой метод предполагает одновременное осуществление численного решения задачи (1.1) на дополнительной расчетной сетке, представленной множеством  $G = \{i_t \dots i_{t+\tau}, 0 \dots j_{\max}^i\}$ , с более грубым шагом в целях повышения производительности. В случае возникновения экстремальной ситуации, осуществляется перерасчет области  $G$  с учетом поправки к граничным условиям  $\mathbf{f}_\Gamma^*$ , на что затрачивается меньше времени, нежели в методе 1, за счет более грубого используемого шага.

3) Еще один способ основан на вычислении временных слоев  $i_t \dots i_{t+\tau_J}$  при помощи методов экстраполяции. В данном случае осуществляется вычисление критерия экстремальности  $J(x_j^i), i=0 \dots i_t, j=0 \dots j_{\max}^i$ . Т.к. при этом критерий  $J$  не зависит от множества точек  $x_j^i, i=i_t \dots i_{t+\tau_J}, j=0 \dots j_{\max}^i$ , данный метод непригоден для коррекции граничных условий  $f_r$  с целью формирования управляющего воздействия для вывода системы из экстремального режима, хотя вычислительные затраты на этапе обнаружения при использовании метода экстраполяции временных слоев меньше, чем у методов 1 и 2.

Обобщенный критерий экстремальности, учитывающий все временные слои с индексами  $i=i_{t-\tau_J} \dots i_t$  и управления по граничным условиям (3.63), имеет вид:

$$G(\mathbf{p}, t, \tau_J) = \sum_{i=i_{t-\tau_J}}^{i=i_t} \sum_{j=0}^{j=j_{\max}^i} J(x_j^i). \quad (3.64)$$

Задача поиска вектора параметров  $\mathbf{p}$  квазиоптимального управляющего воздействия  $f_r^*$  ставится как оптимизационная:

$$\mathbf{p} = \arg \min_{\mathbf{p} \in \mathbf{P}} G(\mathbf{p}, t, \tau) \quad (3.65)$$

для заданного пространства оптимизации  $\mathbf{P}$ , текущего временного слоя с координатой  $t$  и заданной глубины расчета критерия экстремальности  $\tau$ .

Формирование управляющего воздействия, выводящего систему из экстремального режима, является частным случаем параметрической оптимизации. В качестве критериев выступают запасы прочности отдельных компонентов системы. Например, мера принадлежности текущей

температуры допустимому температурному диапазону, отношение текущего давления в реакторе к максимально допустимому давлению и т.п. (3.66).

$$J_1(T) = \left| \frac{T - T_{\min}}{T_{\max} - T_{\min}} \right|, \quad (3.66)$$
$$J_2(P) = \frac{P}{P_{\max}}.$$

Условимся считать критерии запаса прочности системы таким образом, что значения критерия в диапазоне  $[0,1)$  являются допустимыми, а значения  $(1,\infty]$  означают экстремальную ситуацию.

Собственно сами критерии такого рода, как правило, включают в обычные задачи оптимизации по различного рода экономическим показателям. Однако существуют применения для критериев запаса прочности самих по себе:

- 1) рекомендации оператору в ходе проведения вычислительного эксперимента в ручном режиме;
- 2) вывод системы из экстремального режима.

В первом случае в ходе вычисления протяженного во времени процесса периодически производится расчет критериев запаса прочности системы. Также делается прогноз о дальнейшем поведении таких критериев.

$$J_0 := J(\mathbf{x}), \quad (3.67)$$
$$J_1 := J(\mathbf{x} + \Delta\mathbf{x}),$$

где  $J$  - критерий запаса прочности системы,  $\mathbf{x}$  - вектор текущих параметров системы,  $\Delta\mathbf{x}$  - прогнозируемое изменение параметров системы в ходе расчета через некоторое время. Если в ходе дальнейшего расчета в

результате прогнозирования ожидается увеличение критерия  $J$ , т.е. выполняется условие  $J_1 > J_0$ , осуществляется прогнозирование ожидаемого времени до достижения критерием  $J$  экстремального значения 1. Оператору при этом на экран выводится предупреждение и предложение сменить стратегию управления параметрами системы.

В случае вывода системы из экстремального режима требуется определить такое воздействие  $\sigma x$ , при котором значение критерия запаса прочности системы  $J$  вернется в допустимый диапазон  $[0,1)$ . При этом следует учитывать, что зачастую система обладает инертностью, и на скорость изменения вектора параметров  $x$  могут быть наложены ограничения:

$$\left\| \frac{dx}{dt} \right\| < \varepsilon_x. \quad (3.68)$$

Для решения задач формирования управляющего воздействия, выводящего систему из экстремального режима, могут быть использованы поисковые алгоритмы. Как наиболее общие, среди них можно выделить методы обобщенного покоординатного спуска и метод Чебышева с экспоненциальной релаксацией [41].

# **ГЛАВА 4. РАЗРАБОТКА АЛГОРИТМОВ И МЕТОДОВ ФУНКЦИОНИРОВАНИЯ МОДУЛЕЙ СИСТЕМЫ МНОГОВАРИАНТНОГО ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ**

## **4.1. Разработка пользовательского интерфейса управления процессом моделирования в интерактивном режиме**

### **4.1.1. Формирование графического пользовательского интерфейса с учетом управления в экстремальных ситуациях**

Разработка графического пользовательского интерфейса осуществлена с целью удовлетворения основным требованиям: контроль выполнения всех стадий исследования моделируемого процесса, отражение предметной области, минимизация действий пользователя для осуществления требуемой операции.

С помощью графического пользовательского интерфейса оператор осуществляет настройку компонентов каждой математической модели из единой интерактивной системы. Интерфейс программы состоит из двух частей: окно выбора и окно настройки модели. Окно настройки содержит набор вкладок, отражающий компоненты модели и тип исследования: численное моделирование, измерения, уравнения, краевые условия, коэффициенты, неизвестные, функции.

Окно численного моделирования содержит описание модели, органы управления процессом формирования и заполнения расчетной сетки численными решениями в ячейках, средства формирования текстовых отчетов о ходе и результатах моделирования, средства сериализации состояния моделирования, графическое и текстовое отображение статистики хода моделирования.

Окно измерений содержит настройки пространственно-временных измерений расчетной сетки, такие как границы измерений, предельные изменения шагов, графическая гистограмма распределения шагов на текущем временном слое, текстовое отображение шагов и значений измерений в текущей ячейке расчетной сетки.

Окно уравнений содержит набор систем уравнений, описывающих различные области расчетной сетки: начальные условия, граничные условия, внутреннюю область расчетной сетки. Для каждой системы уравнений задаются критерии останова и точности найденных решений, а также представлены участвующие в системе уравнений функции и неизвестные. Переход к описанию интересующей функции или неизвестного на специальной вкладке окна настройки модели осуществляется двойным кликом на соответствующие элементы списка уравнений и неизвестных текущей системы уравнений математической модели.

Окно краевых условий позволяет осуществлять настройку начальных и граничных условий неизвестных, для которых краевые условия задаются функциональной зависимостью. В окне представлен список всех доступных краевых условий, задающихся функционально. Для каждого краевого условия имеется возможность выбрать один из заданных функциональных законов изменения неизвестного, аргументом в которых выступает одно из пространственно-временных измерений расчетной сетки. Законы могут содержать коэффициенты, настройка которых осуществляется в специальном окне. В нижней части окна краевых условий графически представлен один период выбранной краевой функции.

В окне коэффициентов представлены все константные параметры математической модели. В левой части окна осуществляется выбор интересующего параметра, в правой – задание начального и текущего значений.



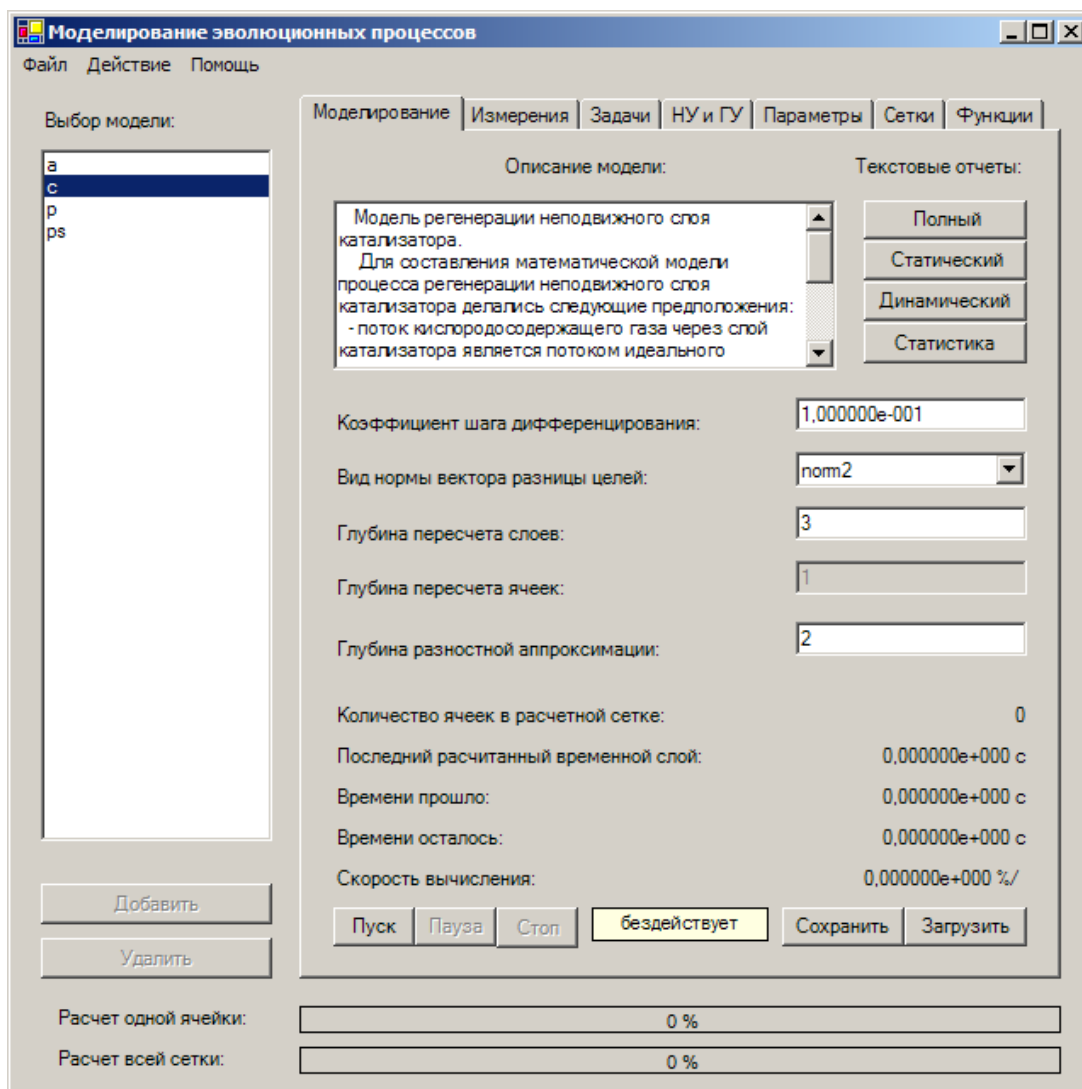


Рис. 4.1. Меню численного счета системы интерактивного моделирования

Разделение на начальное и текущее значение реализовано из-за того, что в ходе моделирования значения некоторых параметров могут меняться. Например, оптимизируемые параметры в ходе оптимизации, либо параметры, определяющие критерий останова расчета.

В окне неизвестных осуществляется визуальное отображение найденных значений неизвестных параметров в ячейках расчетной сетки, а также погрешности вычислений в каждой ячейке. Имеется возможность настройки количества отображаемых на графике слоев.

Окно функций содержит справочную информацию о функциях математической модели, задающих решаемые уравнения, краевые условия, критерии останова и оптимизации.

#### **4.1.2. Удаленное управление и текстовый интерфейс работы с системой интерактивного моделирования**

С целью автоматизации вычислительного процесса система интерактивного моделирования была снабжена текстовым интерфейсом, полностью включающим в себя функциональность графического интерфейса. Текстовый интерфейс представляет собой двусторонний обмен сообщениями по протоколу Telnet, что делает возможным удаленное управление процессом вычисления.

Входными сообщениями системы интерактивного моделирования являются консольные команды, представляющие собой строковый идентификатор и набор буквенных и числовых аргументов. Выходными сообщениями являются информационные сообщения о ходе выполнения запрошенной команды. Помимо консольных команд, в текстовом интерфейсе системы интерактивного моделирования, реализованы операции с переменными: получение значения и присваивание. Переменные могут быть как строкового, так и численного типа с двойной точностью.

## **4.2. Разработка специализированного языка описания математических моделей для автоматизации задания входных данных**

Описание математических моделей осуществляется на специальном структурном языке, являющимся подмножеством языка XML, представляется в виде иерархической структуры, построенной на базе тэгов.

Анализ предметной области выявил следующие объекты математических моделей:

- 1) Глобальные параметры – коэффициенты модели, не привязанные к расчетной сетке. Физически каждый параметр такого рода представляется числом с плавающей точкой.
- 2) Локальные параметры – параметры модели, распределенные во времени и пространстве. Значения данных параметров хранятся в каждой ячейке расчетной сетки и определяются в ходе численного решения.
- 3) Измерения – координаты расчетной сетки. На данном этапе система моделирования может работать с расчетными сетками, имеющими только одно или два измерения.
- 4) Функции – объекты, отражающие функциональные зависимости между глобальными параметрами, локальными параметрами и измерениями модели.
- 5) Задачи – постановка задач численного решения, хранящая информацию о минимизируемых функциях и пространстве поиска решений. Задача считается решенной, когда норма функций задачи не превышает заданный порог, определяющий точность вычислений. В каждой ячейке выбирается задача с подходящей топологией разностной схемы.
- 6) Краевые условия – объекты, отражающие функциональный способ формирования начальных и граничных условий.

## **4.3. Реализация вычислительного модуля системы**

## **моделирования динамических систем эволюционного типа**

### **4.3.1. Осуществление автоматического перехода от математического описания модели к численной аппроксимации**

Запись математических выражений в системе интерактивного моделирования осуществляется с использованием синтаксиса C++, что обеспечивает гибкость при комбинировании различных компонентов модели и использовании высокоуровневых конструкций описания производных, позволяющих осуществлять разностную аппроксимацию в автоматическом режиме.

Переход от естественной записи математических законов к вычислительным алгоритмам, использующим численную аппроксимацию, осуществляется в два этапа:

- 1) Трансляция описания структуры модели с языка XML в C++ код, в ходе которой тэги отдельных компонентов модели преобразуются в вызовы фабрик соответствующих классов.
- 2) Переопределение математических операций для работы с численными аппроксимациями на разностной сетке.

Подобный подход позволяет абстрагироваться от вида численной аппроксимации на стадии формирования математической модели, что дает возможность модифицировать алгоритм численного счета в ходе вычисления, не меняя при этом исходный вид математической модели.

Применение специализированного структурного языка в составе системы интерактивного моделирования позволяет дополнять хранилище моделей, не внося изменений в исходный код программы.

### **4.3.2. Реализация методов задания краевых условий динамической системы**

Краевые условия в системе интерактивного моделирования могут формироваться следующими способами:

- 1) Определение значений краевых ячеек расчетной сетки на основе функциональной зависимости от изменяемой пространственно-временной координаты расчетной сетки.
- 2) Численное решение системы уравнений, не содержащей производных по неизменной пространственно-временной координате заданной краевой области.

При формировании краевых условий на основе функциональной зависимости от пространственно-временной координаты, необходимо учитывать влияние шага расчетной сетки на погрешность численной аппроксимации функции задания краевых условий.

Для реализации формирования краевых условий на основе численного решения системы уравнений, был введен контекстный выбор решаемой задачи на основе паттерна соседних ячеек, суть которого состоит в следующем:

- 1) Производится сортировка всех систем уравнений данной задачи в порядке увеличения числа ячеек разностной схемы.
- 2) Выбор решаемой задачи в текущей ячейке осуществляется циклическим перебором отсортированного множества задач, пока топология соседних ячеек не будет удовлетворять разностную схему текущей задачи.

Данный подход позволяет гибко выбирать решаемую задачу при многоточечной численной аппроксимации, где толщина краевой области расчетной сетки составляет несколько ячеек.

#### **4.3.3. Формирование и хранение динамической**

## **расчетной сетки с нерегулярным шагом дифференцирования**

Наличие динамического шага расчетной сетки приводит к различному числу ячеек на временных слоях, что, в свою очередь, делает неэффективным применение регулярных типов хранилищ данных. Целесообразным для представления динамической расчетной сетки в памяти компьютера является применение интрузивных контейнеров, таких как многосвязные списки. Каждая ячейка прямоугольной расчетной сетки должна содержать 4 указателя:

- 1) Соседний элемент текущего временного слоя с меньшей пространственной координатой.
- 2) Соседний элемент текущего временного слоя с большей пространственной координатой.
- 3) Соседний элемент временного слоя с меньшей временной координатой и с совпадающей пространственной координатой.
- 4) Соседний элемент временного слоя с большей временной координатой и с совпадающей пространственной координатой.

Сохранение интрузивных типов данных на жестком диске связано с известными трудностями, вызванными наличием динамических указателей. Данная проблема разрешается введением дополнительной индексной адресации объектов и таблицы соответствия динамических адресов в памяти и индексов объектов.

### **4.3.4. Нахождение численного решения системы эволюционных уравнений в узлах расчетной сетки**

Нахождение численного решения в узле расчетной сетки представляет собой циклическое применение интервальных, поисковых и градиентных алгоритмов с целью коррекции значений неизвестных в ячейке для минимизации нормы функционала решаемой системы уравнений.

Вычислительный модуль осуществляет численное решение как в одной ячейке, так и автоматический синтез и решение генерализированной многоточечной системы уравнений.

После завершения расчета каждого временного слоя осуществляется проверка на останов. Критерий останова задается оператором на этапе формирования математической модели в виде двоичной логической функции.

#### **4.4. Сохранение и подсчет частоты использования шагов дифференцирования**

На каждом временном слое ведется подсчет частоты использования шагов расчетной сетки по пространственной координате. Процедура подсчета должна удовлетворять следующим требованиям:

- 1) Адекватно отражать топологию распределения шагов дифференцирования на текущем временном слое.
- 2) Минимизировать расход памяти на структуру, хранящую топологию распределения.
- 3) Минимизировать время запроса частоты использования заданного шага.

С целью удовлетворения приведенных требований, в качестве структуры, хранящей топологию распределения шагов пространственной координаты расчетной сетки, был выбран составной тип данных, содержащий кластеризованный массив шагов, отсортированный по значению шага. При добавлении нового элемента в кластер массива, осуществляется адаптивное редуцирование элементов, близких по значению к данному, в один элемент.

#### **4.5. Квазиоптимальное управление параметрами**

## модели через граничные условия при помощи типовых законов управления

В состав системы многовариантного моделирования входят некоторые типовые управляющие воздействия, а также имеется возможность задавать произвольное управляющее воздействие при помощи структурного языка описания математических моделей:

- линейный закон  $u(t) = a \cdot t + b$ ;
- гармонический закон  $u(t) = a \cdot \sin(b \cdot t + c) + d$ ;
- трапециевидный закон  $u(t) = \begin{cases} \tau = -0.5 \cdot a + (b \cdot t + c) \bmod d; \\ \min(e, \max(f, -g \cdot \tau + h)), \tau < 0; \\ \min(e, \max(f, i \cdot \tau + j)), \tau > 0. \end{cases}$

Реализована возможность автоматизированного подбора коэффициентов выбранного закона управления с целью вывода системы из экстремальной ситуации (3.62) по всей длине профиля. Критерий экстремальности (3.62) реализован в виде границ области изменения параметров, соответствующих нормальному режиму работы:

$$J(P) = \begin{cases} P_{\min} - P, P < P_{\min}; \\ P - P_{\max}, P > P_{\max}; \\ 0, P_{\min} \leq P \leq P_{\max}. \end{cases} \quad (4.1)$$

Входные данные, задаваемые пользователем:

- тип закона управления;
- управляемый параметр  $P_u$ , на канал граничных условий которого будет подаваться управляющее воздействие  $u$ ;
- минимальная  $P_{\min}$ , максимальная  $P_{\max}$  границы и удерживаемый в них параметр  $P$ ;
- настройки подбора коэффициентов закона управления: начальное, конечное значения, шаги.
- ширина временного окна  $\Omega$ .



После расчета каждой ячейки осуществляется проверка  $P_{\min} \leq P \leq P_{\max}$ . В случае выхода  $P$  из допустимого диапазона включается управляющее воздействие на канале ГУ  $P_u$ , либо корректируются коэффициенты активного управляющего воздействия, после чего происходит откат процесса моделирования на  $\Omega$  секунд назад и процесс моделирования повторяется.

## **4.6. Методология визуализации результатов моделирования динамических систем эволюционного типа**

### **4.6.1. Синтез нерегулярной самоорганизующейся структуры данных для хранения численной информации при визуализации**

При выборе формата представления сеточной функции следует руководствоваться возможностью произвольного ее заполнения. Формат представляет собой динамическую иерархическую структуру, т.е. дерево, глубина которого соответствует размерности сеточной функции. Каждый узел дерева содержит набор ветвей, причем в любой момент времени может быть добавлена новая ветвь, или может быть запрошен указатель на уже существующую. Общее количество ветвей формально не ограничено. Для подобных целей поиска и динамического формирования как нельзя лучше подходят самобалансирующиеся бинарные деревья. Процедура включения, восстанавливающая идеально сбалансированное дерево, вряд ли будет выгодна, поскольку восстановление дерева после случайного включения – довольно сложная операция. Возможным выходом из положения будет введение менее строгого определения сбалансированности. Воспользуемся определением, предложенным Г.М. Адельсоном-Вельским и Е.М. Ландисом. Их критерий сбалансированности сформулирован так: дерево называется сбалансированным тогда и только тогда, когда

высоты двух поддеревьев каждой из его вершин отличаются не более чем на единицу. Такое не совсем совершенное определение приводит к более простой процедуре переупорядочения за счет лишь незначительного усложнения поиска. Итак, в качестве структуры, содержащей ветви вершины дерева размерности сеточной функции, выбрано самобалансирующееся дерево, использующее вышеназванный критерий сбалансированности – AVL-дерево.

#### **4.6.2. Разработка метода проекции высокочастотных численных данных на пространство визуализации**

Процесс визуализации представляет собой циклическое изменение аргументов в отображаемой области изменения, вычисление соответствующих значений функции, и осуществление графического отображения, при котором пространственные координаты определяются значениями функции и аргументов графика. Если размерность функции больше, чем количество осей графика, координаты по которым определяются значениями аргументов, аргументы функции можно разделить на два класса – привязанные к осям и свободные. Значения свободных аргументов задаются жестко и сохраняются на всей области построения графика. В процесс настройки параметров отображения необходимо включить возможность задания жесткого значения для каждого аргумента и возможность оперативного изменения набора свободных и привязанных аргументов. В случае одновременного отображения нескольких графиков, каждый график уникально определяется вектором значений всех своих свободных аргументов и изображается отдельным выбираемым цветом.

Проблемой визуализации двумерного графика вида  $y=f(x)$  является масштабирование участка сеточной функции, количество элементов которого значительно превосходит количество экранных пикселей, задающих ширину области отображения. Получается ситуация, когда несколько

значений функции необходимо отобразить в области графика шириной в один пиксель. Для адекватного представления целесообразно в каждой экранной позиции отображать отрезок, соответствующий минимальному и максимальному из конкурирующих значений сеточной функции. Такой подход приведет к некоторому падению производительности в целях повышения качества визуализации, которое является важным критерием. В процессе отображения каждого вертикального отрезка проверяется выход за допустимые границы изменения, и в случае необходимости производится выделение цветом вышедших участков.

Трехмерная поверхность вида  $z=f(x,y)$  представляет собой так называемую карту высот. Проблемой визуализации трехмерной поверхности является схожий с вышеописанным случай, когда на прямоугольной области аргументов сеточной функции, соответствующей элементарному прямоугольному участку основания карты высот, находится несколько значений. В этом случае для адекватного визуального отображения сеточной функции целесообразно строить две поверхности, одна из которых будет соответствовать максимальным, а другая – минимальным значениям функции на рассматриваемом промежутке. Размер элементарного прямоугольного участка основания карты высот задает степень детализации изображения. Очевидно, что вычислительные затраты на создание высокодетализированного изображения будут велики. Это приводит к дилемме. С одной стороны необходимо обеспечить высокое качество изображения, а с другой – возможность оперативного изменения положения и масштаба отображаемого участка области изменения аргументов сеточной функции. Оптимальным решением проблемы является введение двух уровней детализации, один из которых, более низкий, будет использоваться во время скроллинга окна аргументов, а другой, более высокий – при выходе из этого режима.

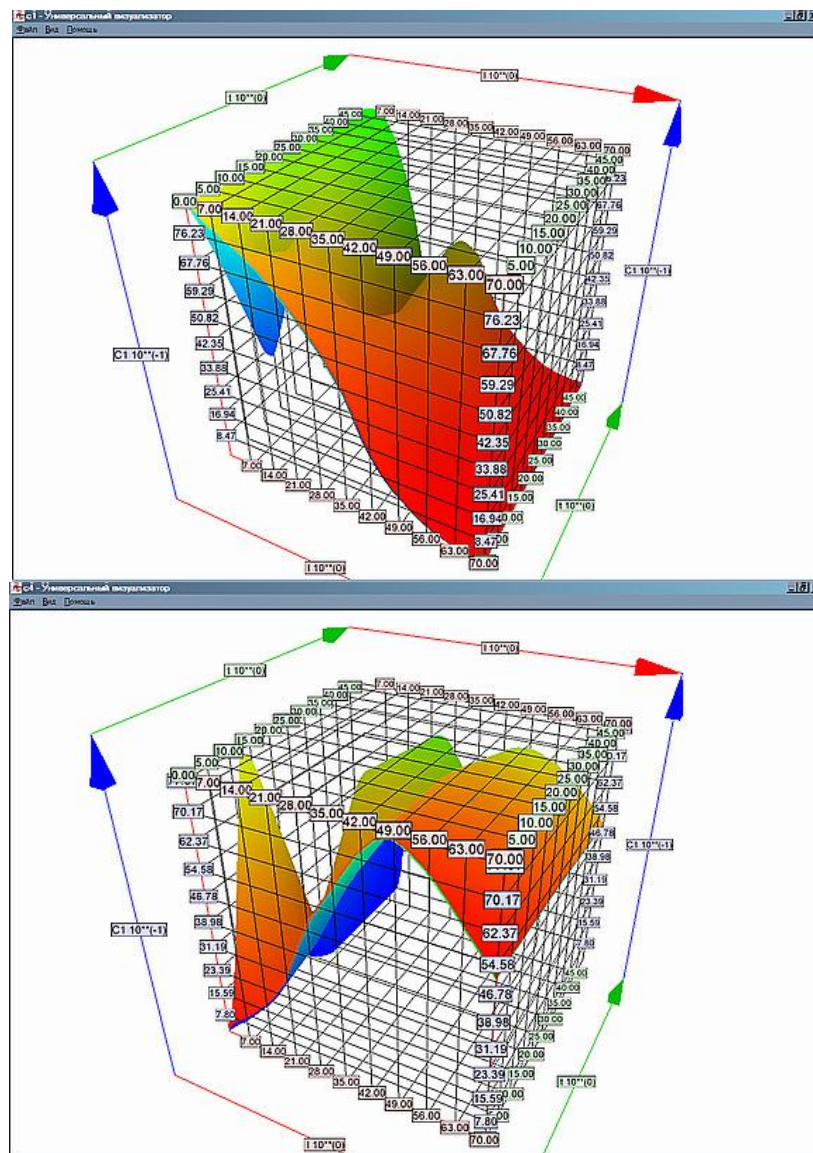


Рис. 4.2. Трехмерная визуализация распределения температуры в процессе регенерации неподвижного слоя катализатора при различных начальных условиях

## **ГЛАВА 5. ПРИМЕНЕНИЕ СИСТЕМЫ МНОГОВАРИАНТНОГО ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ ДЛЯ ИССЛЕДОВАНИЯ ЭВОЛЮЦИОННЫХ СИСТЕМ**

### **5.1. Апробация работы интерактивной системы моделирования на модельной задаче процесса регенерации неподвижного слоя катализатора**

В химической промышленности широко распространены процессы, которые осуществляются в реакторах с неподвижным слоем катализатора. Такие процессы сопровождаются побочными реакциями, приводящими к закоксовыванию катализаторов. Отлагающийся кокс существенно меняет свойства катализатора и снижает его активность и селективность. В производственных условиях это вызывает необходимость периодического отключения реакторов для регенерации катализатора, то есть удаления с него кокса [52].

В целом коксоотложение значительно усложняет промышленный процесс, снижает продуктивность единицы объема катализатора, резко увеличивает эксплуатационные расходы и себестоимость продукта. Отсюда вытекает, что весьма актуальными задачами являются поиск путей снижения закоксовыванности и оптимальных режимов регенерации катализаторов.

Регенерация катализатора заключается в выжиге кокса подачей воздуха с водяным паром. В основе регенерации лежат химические реакции кокса с кислородом, приводящие к удалению кокса путем образования газообразных веществ.

Для составления математической модели процесса регенерации неподвижного слоя катализатора делались следующие предположения:

- поток кислородосодержащего газа через слой катализатора является потоком идеального вытеснения;

- температура по поперечному сечению реактора и зерна катализатора не меняется;
- массой газа, находящейся в порах зерна катализатора, можно пренебречь по сравнению с массой катализатора;
- можно пренебречь изменением массы кислородосодержащего газа в ходе процесса.

При этих предположениях математическую модель рассматриваемого процесса можно представить системой уравнений материального (по кислороду и коксу) и теплового балансов:

$$\begin{cases} \frac{\partial M}{\partial t} + U \frac{\partial M}{\partial l} = -\frac{W}{\rho_1}, \\ \frac{\partial G}{\partial t} = -\frac{W}{\rho_2 \alpha_2}, \\ \frac{\partial T}{\partial t} + U^* \frac{\partial T}{\partial l} = \frac{QW}{\rho_1 C} - \frac{\lambda S_0 (T - T_H)}{\rho_1 C S}. \end{cases} \quad (5.2)$$

Здесь введены следующие обозначения:

$$\begin{aligned} \rho_1 &= \varepsilon \rho_1^0; \rho_2 = (1 - \varepsilon) \rho_2^0; C = C_1 + \rho_2 c_2 / \rho_1; \\ U &= V_0 / \rho_1 S; U^* = C_1 U / C; \end{aligned}$$

где  $V_0$  - масса газа, поступающего в реактор за 1 с;

$\rho_1^0$  - плотность газа;

$C_1$  - удельная теплоемкость газа;

$C_2$  - теплоемкость катализатора;

$\varepsilon$  - доля свободного объема в слое катализатора;

$S$  - площадь поперечного сечения слоя;

$S_0$  - площадь боковой поверхности слоя на единицу его длины;

$Q$  - тепловой эффект реакции;

$W$  - скорость реакции, измеренная по кислороду и отнесенная к единице объема реактора;  
 $T_n$  - температура внешней среды;  
 $\lambda$  - коэффициент теплопередачи через стенку;  
 $a$  - константа, характеризующая соотношение скоростей реакции по кислороду и коксу;  
 $M$  - содержание кислорода в газе;  
 $G$  - содержание кокса в катализаторе;  
 $T$  - температура процесса;  
 $U$  - средняя массовая скорость потока в поровом пространстве катализатора;  
 $t, l$  - временная и пространственная (в направлении потока) координаты.

Скорость реакции, измеренная по кислороду, определяется по формуле:

$$W = K(T)M \left( \frac{G}{G_0} \right)^r, \quad (5.3)$$

где  $K(T)$  - константа скорости реакции окисления кокса;  
 $G_0$  - начальная концентрация кокса;  
 $r$  - константа, характеризующая область протекания реакции:

$r = 0$  соответствует кинетической и внешнедиффузионной области;

$r = 1$  соответствует внутридиффузионной области.

Константа скорости реакции представляется в виде:

$$K(T) = a_1 e^{-\frac{E}{RT}(1-\alpha T)}, \quad (5.4)$$

где  $a_1, \alpha$  - константы, присущие конкретным условиям регенерации;

$E$  - энергия активации;

$R$  - универсальная газовая постоянная.

Для системы уравнений (5.2) задаются краевые условия: начальные при  $t = 0, 0 \leq l < L$ ;

$$M(l, 0) = 0, G(l, 0) = G_0(l), T(l, 0) = T_0(l) \quad (5.5)$$

и граничные при  $t \geq 0$  и  $l = 0$ :

$$M(0, t) = M^{ex}(t), T(0, t) = T^{ex}(t) \quad (5.6)$$

Для адиабатического реактора  $\lambda = 0$ , то есть отсутствует теплообмен с окружающей средой. Примем  $r = 1, \alpha = 0$ . Обозначим

$$K_1 = -\frac{1}{\rho_1}; K_2 = -\frac{1}{\rho_2 \alpha_2}; K_3 = \frac{Q}{\rho_1 C}.$$

В этом случае система уравнений (5.2) переписывается в виде:

$$\begin{cases} \frac{\partial M}{\partial t} + U \frac{\partial M}{\partial l} = K_1 W, \\ \frac{\partial G}{\partial l} = K_2 W, \\ \frac{\partial T}{\partial t} + U^* \frac{\partial T}{\partial l} = K_3 W. \end{cases} \quad (5.7)$$

Начальные условия (5.5) по кислороду  $M$  имеют разрыв. С целью их сглаживания предполагается, что в процессе регенерации допускается квазистационарное изменение концентрации кислорода, то есть когда



$$\left| \frac{\partial M}{\partial t} \right| \ll u \frac{\partial M}{\partial l}. \quad (5.8)$$

В этом случае первое уравнение системы (5.7) переписывается в виде

$$u \frac{\partial M}{\partial l} = K_1 W. \quad (5.9)$$

И для получения гладких начальных условий решается следующая задача:

$$\begin{cases} \frac{\partial M}{\partial l} = \frac{K_1 W}{u}, \\ M(0) = M_0. \end{cases} \quad (5.10)$$

Значения параметров модели, используемые при моделировании:

Параметр	Значение	Параметр	Значение
$\rho_1^0$	3.4 кг/м <sup>3</sup>	$\rho_2^0$	655 кг/м <sup>3</sup>
$C_1$	1045 Дж/кг	$C_2$	1045 Дж/кг
$\varepsilon$	0.5	$Q$	128e+4
$\alpha$	0	$a_1$	233e+3
$a_2$	0.38	$U$	0.46
$G_0$	0.011	$M_0$	0.0219
$T_0$	689 К	$L$	1 м
$E$	705e+5 Дж/К·моль	$R$	8314.3 Дж/К·моль

В качестве примера использования интерактивной системы моделирования динамических режимов процессов эволюционного типа EQU рассмотрим численное

моделирование процесса регенерации неподвижного слоя катализатора.

Для модельной задачи (использовалась длина реактора 1м) проведем расчеты с различными законами изменения концентрации кислорода на входе в реактор. Окончание процесса – когда среднее значение концентрации остаточного кокса не превышает  $10^{-4}$  весовой доли. Условимся считать допустимым температурным диапазоном интервал от  $670^{\circ}\text{K}$  до  $710^{\circ}\text{K}$ . При выходе за нижнюю границу интервала выжиг кокса прекращается, при превышении верхней границы происходит разрушение реактора, т.е. имеет место экстремальный режим.

Проведены следующие вычислительные эксперименты (рис. 5.1-5.5):

- 1) Отложение кокса равномерно по всей длине реактора. На рис. 5.1б видно распределение температуры рядом на всех временных профилях. Температура меняется в диапазоне от  $690^{\circ}\text{K}$  до  $745^{\circ}\text{K}$ .
- 2) Отложение кокса линейно уменьшается по длине реактора. На рис. 5.2б виден ярко выраженный пик температуры, смещающийся с течением времени в сторону выхода реактора. Температура меняется в диапазоне от  $690^{\circ}\text{K}$  до  $735^{\circ}\text{K}$ .
- 3) Отложение кокса и температура катализатора линейно уменьшаются по длине реактора. На рис. 5.3б виден ярко выраженный пик температуры, смещающийся с течением времени в сторону выхода реактора. Температура меняется в диапазоне от  $690^{\circ}\text{K}$  до  $735^{\circ}\text{K}$ .
- 4) Отложение кокса равномерно по всей длине реактора, но осуществляется управление граничными условиями концентрации кислорода по линейному закону  $M(0,t) = a \cdot t + b$  с целью удержания температуры  $T$  на всей длине профиля в интервале  $[T_{\min}, T_{\max}]$ , где  $T_{\min} = 670^{\circ}\text{K}$ ,  $T_{\max} = 710^{\circ}\text{K}$ . Температура процесса целиком лежит в допустимых границах (от  $690^{\circ}\text{K}$  до  $710^{\circ}\text{K}$ ) на

всех профилях. На рис. 5.5 показан вид управляющего воздействия.

При моделировании использовались следующие настройки вычислительного блока: минимальное значение целевого функционала:  $1e-15$ ; максимальное число итераций метода спуска: 100; начальный шаг по времени: 0.1с; начальный шаг по пространству: 0.01м.

На графиках синим цветом показаны профили в начальный момент времени, красным и желтым – в последний момент времени.

## **5.2. Представление модели регенерации неподвижного слоя катализатора на языке описания математических моделей в интерактивной системе моделирования.**

Описание модели начинается с объявления главного xml узла “model”, где указывается название фабрики моделей и описание модели. Фабрика моделей используется для создания экземпляров описанного класса моделей.

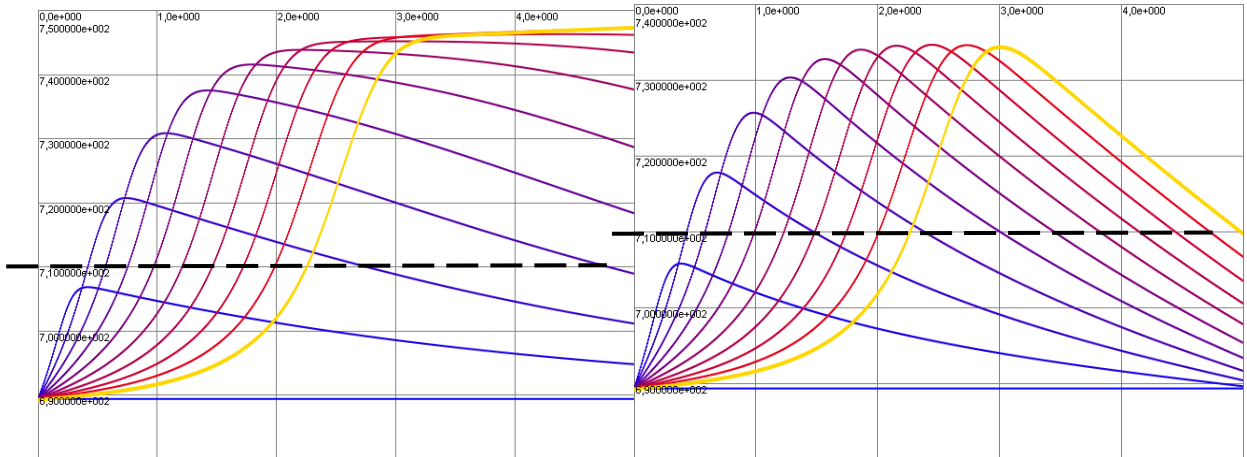
```
<model name="catalyst" desc="
```

Модель регенерации неподвижного слоя катализатора.



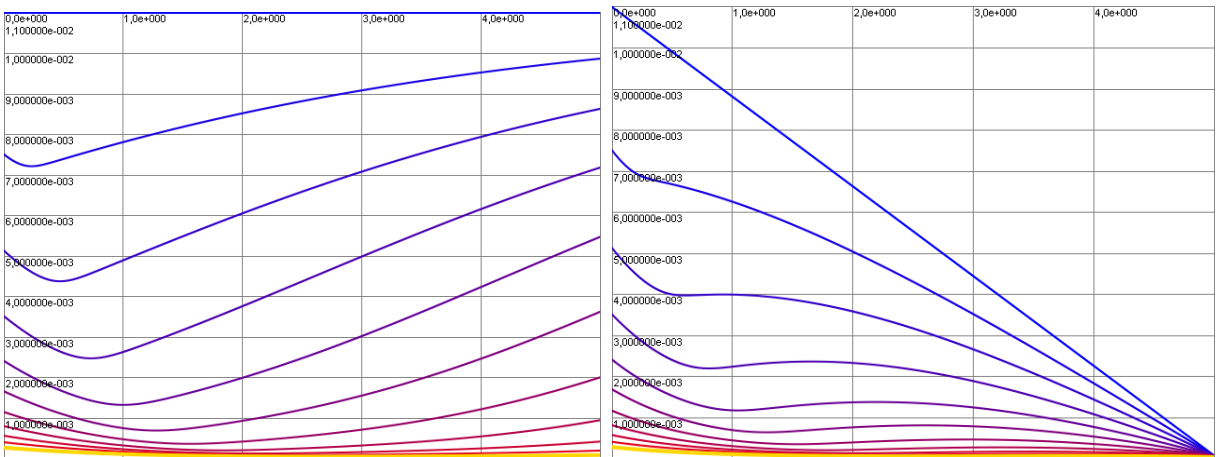
а

а



б

б



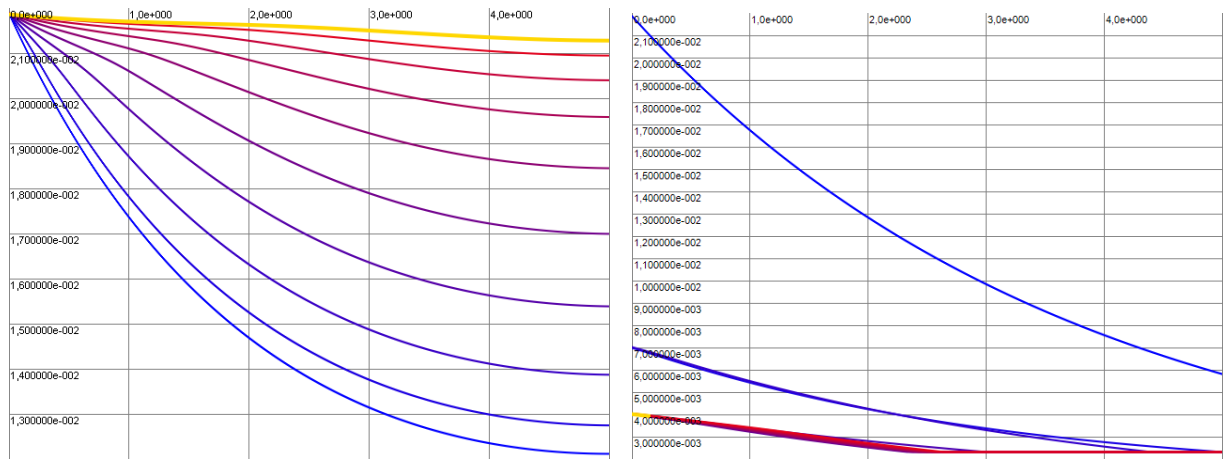
в

в

Рис. 5.1. Отложение кокса равномерно уменьшается по длине реактора

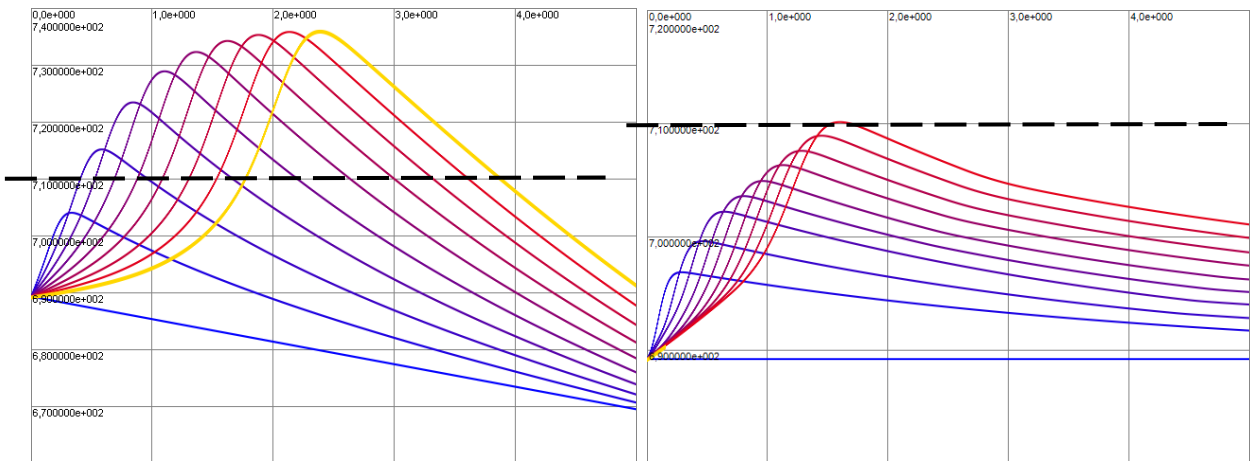
Рис. 5.2. Отложение кокса линейно

(а – профили  $M$  в различные моменты времени; б – профили  $T$  в различные моменты времени; в – профили  $G$  в различные моменты времени)



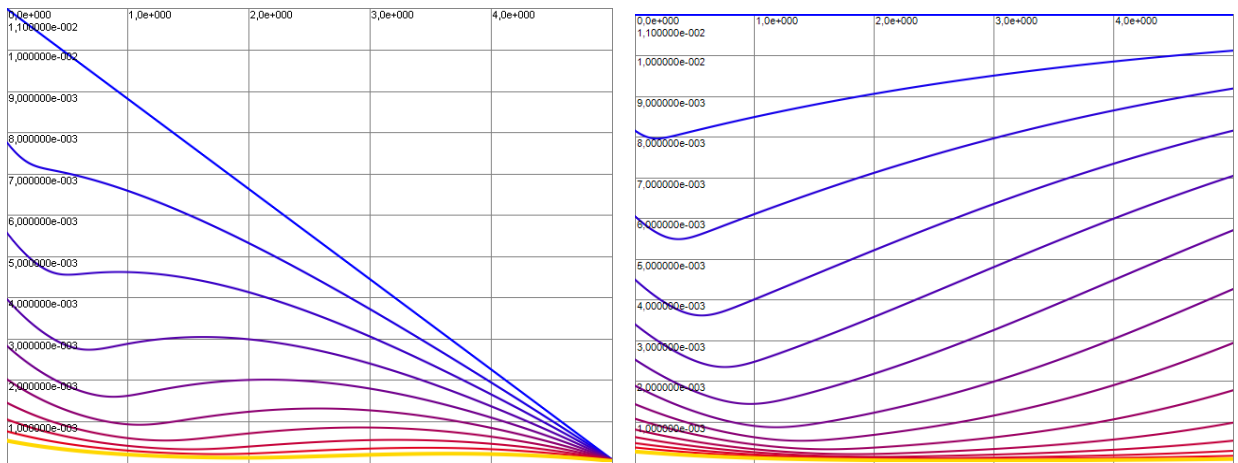
а

а



б

б



в

в

Рис. 5.3. Отложение кокса и температура линейно уменьшаются по длине

Рис. 5.4. Управление граничными условиями по кислороду

(а – профили М в различные моменты времени; б – профили Т в различные моменты времени; в – профили G в различные моменты времени)

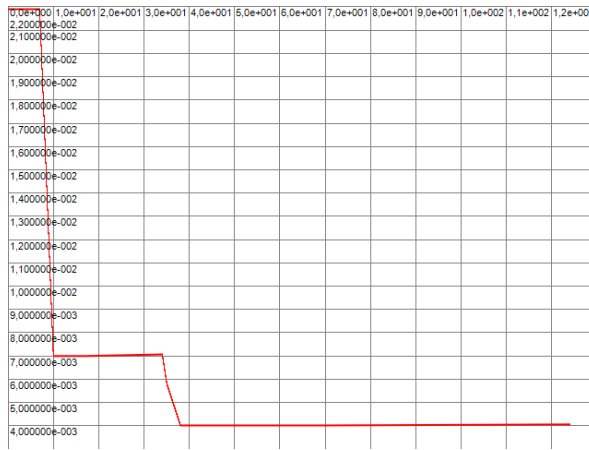


Рис. 5.5 Граничные условия по кислороду

Для составления математической модели процесса регенерации неподвижного слоя катализатора делались следующие предположения:

- поток кислородосодержащего газа через слой катализатора является потоком идеального вытеснения;
- температура по поперечному сечению реактора и зерна катализатора не меняется;
- массой газа, находящегося в порах зерна катализатора, можно пренебречь по сравнению с массой катализатора;
- можно пренебречь изменением массы кислородосодержащего газа в ходе процесса.

Далее идет описание глобальных параметров, т.е. констант и коэффициентов, определяющих настройки модели.

```

<prmg name="ro01" value="3.4" desc="плотность газа"/>
<prmg name="ro02" value="655" desc="плотность катализатора"/>
<prmg name="C1" value="1045" desc="удельная теплоемкость газа"/>
<prmg name="C2" value="1045" desc="удельная теплоемкость катализатора"/>
<prmg name="e" value="0.5" desc="доля свободного объема в слое катализатора"/>

```

<prmng name="Q" value="128e+4" desc="тепловой эффект реакции"/>

<prmng name="a" value="0" desc="константа, характеризующая соотношение скоростей реакции по кислороду и коксу"/>

<prmng name="a1" value="233e+3" desc="константа, присущая конкретным условиям регенерации"/>

<prmng name="a2" value="0.38" desc="константа, присущая конкретным условиям регенерации"/>

<prmng name="U" value="0.46" desc="средняя массовая скорость потока в паровом пространстве катализатора"/>

<prmng name="G0" value="0.011" desc="начальное содержание кокса в катализаторе"/>

<prmng name="M0" value="0.0219" desc="начальное содержание кислорода в газе"/>

<prmng name="T0" value="689" desc="начальная температура процесса"/>

<prmng name="Gmin" value="1e-4" desc="средняя концентрация кокса, при достижении которой прекращать расчет"/>

<prmng name="Gcur" value="1000" desc="текущая средняя концентрация кокса. рассчитывается автоматически каждый слой"/>

<prmng name="E" value="705e+5" desc="энергия активации"/>

<prmng name="R" value="8314.3" desc="универсальная газовая постоянная"/>

<prmng name="tmax" value="500" desc="расчитываемый временной промежуток"/>

<prmng name="lmax" value="1" desc="длина реактора"/>

За глобальными параметрами описываются локальные параметры. Их значения являются распределенными во времени и пространстве и хранятся в каждой ячейке расчетной сетки. Решение модели сводится к нахождению значений локальных параметров.

<prml name="M" value="M0" desc="содержание кислорода в газе"/>

<prml name="G" value="G0" desc="содержание кокса в катализаторе"/>

<prml name="T" value="T0" desc="температура процесса"/>

Далее следует описание измерений, где задаются границы расчетной сетки и настройки изменения шагов по координатам в ходе решения.

<dim name="l" low="0" high="lmax" step="1e-2" smin="1e-7" smax="1e-2" rinc="2" rdec="0.5" retry="1" next="0" desc="пространственная (в направлении потока) координата"/>

<dim name="t" low="0" high="tmax" step="1e-1" smin="1e-7" smax="1e+0" rinc="2" rdec="0.5" retry="0" next="0" desc="временная координата"/>

После объявления всех числовых элементов модели, идет описание функций. Синтаксис описания функций соответствует языку C++.

<func name="W" exp="K \* M \* G / G0" desc="скорость реакции, измеренная по кислороду и отнесенная к единице объема реактора"/>

<func name="K" exp="a1 \* exp( - ( E / ( R \* T ) ) \* ( 1.0 - a \* T ) )" desc="константа скорости реакции окисления кокса"/>

<func name="k1" exp="- 1.0 / ro1"/>

<func name="k2" exp="- 1.0 / ( ro2 \* a2 )"/>

<func name="k3" exp="Q / ( ro1 \* C )"/>

<func name="C" exp="C1 + ( ro2 / ro1 ) \* C2"/>

<func name="ro1" exp="e \* ro01"/>

<func name="ro2" exp="( 1.0 - e ) \* ro02"/>

<func name="UU" exp="C1 \* U / C"/>

<func name="EqM\_d" exp="Diff( M, t ) + U \* Diff( M, l ) - k1 \* W" desc="уравнение динамического материального баланса по кислороду"/>



```
<func name="EqG_d" exp="Diff( G, t ) - k2 * W"
desc="уравнение динамического материального баланса по
коксу"/>
```

```
<func name="EqT_d" exp="Diff( T, t ) + UU * Diff( T, l ) -
k3 * W" desc="уравнение динамического теплового
баланса"/>
```

```
<func name="EqM_s" exp="U * Diff( M, l ) - k1 * W"
desc="уравнение статического материального баланса по
кислороду"/>
```

В системе предусмотрена возможность задания специальных функций изменения краевых условий. Далее показан пример описания функции вида:

$$f(x) = A \cdot \sin(\phi + 2\pi \cdot x \cdot p)^{\xi} + O$$

```
<chopfunc
  name      ="esin"
  litname   ="ExpSin"
  desc     ="экспоненциальный синус"
  exp      ="esin_amp * pow ( sin ( esin_phs +
2*M_PI*m_cChopArg*esin_prd ), esin_exp ) +
esin_ofs"
  period   ="esin_prd"
  min      ="esin_ofs + esin_amp * min ( 0.01, pow ( -1.01,
esin_exp ) )"
  max      ="esin_ofs + esin_amp"/>
  <chopparam name="esin_amp" value="1" litname="Amp"
desc="Амплитуда"/>
  <chopparam name="esin_phs" value="1" litname="Phase"
desc="Фаза"/>
  <chopparam name="esin_prd" value="1" litname="Period"
desc="Период"/>
  <chopparam name="esin_exp" value="1" litname="Power"
desc="Степень"/>
  <chopparam name="esin_ofs" value="1" litname="Offset"
desc="Смещение"/>
```

В случае выхода решений за область допустимых значений, проводится повторный расчет. В данном случае областью допустимых значений для  $M$ ,  $G$  и  $T$  является все множество неотрицательных чисел.

```
<retry cond="M < 0.01 || G < 0.01 || T < 0.01"/>
```

Расчет останавливается, если остаточный слой кокса не превышает значения  $G_{min}$ . В данном случае разница между минимальной концентрацией кокса и средней на последнем временном слое сохраняется в параметр  $G_{cur}$ .

```
<stop cond="Gcur = LastLayerAvg ( G ) - Gmin"/>
```

Далее идет описание расчетных задач. Порядок объявления задач важен, т.к. он определяет процесс поиска подходящей задачи для каждой ячейки в процессе решения. Поэтому описание начинается с наиболее общей задачи динамики, затем следует описание начальной задачи по кислороду, и завершает описание граничная задача по коксу.

```
<problem name="dynamic" emin="1e-15" emax="1e-10"  
dmin="1e-10" it="100" desc="модель динамики">
```

```
<param name="M"/>
```

```
<param name="G"/>
```

```
<param name="T"/>
```

```
<equ name="EqM_d"/>
```

```
<equ name="EqG_d"/>
```

```
<equ name="EqT_d"/>
```

```
</problem>
```

```
<problem name="static" emin="1e-15" emax="1e-10"  
dmin="1e-10" it="100" desc="модель статики">
```

```
<param name="M"/>
```

```
<equ name="EqM_s"/>
```

```
</problem>
```

```
<problem name="cox" emin="1e-15" emax="1e-10"  
dmin="1e-10" it="100" desc="выгорание кокса">
```

```
<param name="G"/>
```

```
<equ name="EqG_d"/>
```

```
</problem>
```

Для локальных параметров, значения которых определяется непосредственно функцией, краевые условия задаются далее. Задается как функция, так и ее настройки для каждого локального параметра.

```
<chop name="initial G" prml="G" func="m_cChopConst"
type="initial" desc="начальные условия по коксу">
  <param name="m_cChopConstOffset" value="G0"/>
  <param name="m_cChopSinOffset" value="G0"/>
  <param name="m_cChopSawOffset" value="G0"/>
  <param name="esin_ofs" value="G0"/>
</chop>
<chop name="initial T" prml="T" func="m_cChopConst"
type="initial" desc="начальные условия по температуре">
  <param name="m_cChopConstOffset" value="T0"/>
  <param name="m_cChopSinOffset" value="T0"/>
  <param name="m_cChopSawOffset" value="T0"/>
  <param name="esin_ofs" value="T0"/>
</chop>
<chop name="boundary M" prml="M"
func="m_cChopConst" type="boundary" desc="граничные
условия по кислороду">
  <param name="m_cChopConstOffset" value="M0"/>
  <param name="m_cChopSinOffset" value="M0"/>
  <param name="m_cChopSawOffset" value="M0"/>
  <param name="esin_ofs" value="M0"/>
</chop>
<chop name="boundary T" prml="T" func="m_cChopConst"
type="boundary" desc="граничные условия по
температуре">
  <param name="m_cChopConstOffset" value="T0"/>
  <param name="m_cChopSinOffset" value="T0"/>
  <param name="m_cChopSawOffset" value="T0"/>
  <param name="esin_ofs" value="T0"/>
</chop>
```

Параметры обнаружения нештатных ситуаций и вывода системы из экстремального режима задаются узлом

коррекции граничных условий, где указывается канал управления, контролируемый параметр, вид управляющего воздействия и границы изменения его коэффициентов:

```
<chopcor name="TbyM"  
  chop="boundary M"  
  prml="T"  
  minlimit="T0-10"  
  maxlimit="T0+10"  
  minmag="M0*0.1"  
  maxmag="M0*1.0"  
  timespread="5"  
  threshold="0.4"  
  tolerance="0.8"  
  correction="t" />  
</model>
```

При запуске интерактивной системы моделирования, данный файл автоматически преобразуется в C++ код, который компилируется в отдельный модуль, который в свою очередь подключается к модулю самой интерактивной системы. В результате в системе регистрируется фабрика моделей “catalyst”, которая в процессе работы может быть использована для создания экземпляров модели регенерации неподвижного слоя катализатора. Одновременно в системе может работать несколько экземпляров модели, для которых пользователем могут быть заданы независимые настройки.

## ЗАКЛЮЧЕНИЕ

В результате проведенной работы по разработке математического и программного обеспечения для моделирования и визуализации динамических режимов систем эволюционного типа, осуществляющего управление в экстремальных ситуациях, были получены следующие результаты.

- 1) Сформирована структура системы многовариантного моделирования и визуализации динамических режимов систем эволюционного типа, позволяющая осуществлять обнаружение нештатных ситуаций и формирование квазиоптимальных типовых управляющих воздействий, выводящих систему из экстремального режима.
- 2) Разработан алгоритм формирования частично регуляризированной расчетной сетки, учитывающий топологию разностной аппроксимации исходной системы уравнений в частных производных.
- 3) Произведена модификация метода адаптивной подстройки шага для учета неравномерности распределения ячеек на всех временных слоях расчетной сетки.
- 4) Предложен критерий эффективности расчетной сетки, отражающий требования к вычислительным ресурсам, характеризующий неравномерность распределения шагов и позволяющий комплексно сравнивать различные алгоритмы построения расчетной сетки.
- 5) Разработан специальный структурный язык описания математических моделей эволюционных динамических систем, являющийся подмножеством языка XML. Описания моделей автоматически транслируются в C++ код и подключаются к системе на этапе загрузки.
- 6) Разработана специальная программная система многовариантного моделирования, апробированная при исследовании процесса регенерации неподвижного слоя катализатора в экстремальных условиях по температуре.

## ПРИЛОЖЕНИЕ

### Тэги языка описания моделей среды интерактивного моделирования

#### **model**

Корневой узел описания модели. Остальные тэги описания данной модели содержатся внутри данного тэга.

Пример: `<model name="acoustic" desc="модель акустики">`

Атрибуты:

**name** [обязательно] – уникальное имя фабрики данной модели. При подключении описания модели к системе, пользователь имеет возможность создавать экземпляры моделей при помощи фабрик. Отдельные модели могут различаться значениями параметров, однако функциональное наполнение их будет одинаковым.

**desc** [опционально] – описание модели. Отображается в графическом интерфейсе.

#### **prmg**

Узел глобального параметра модели. Параметр представляет собой одно число, не привязан к расчетной сетке. Узел не имеет потомков.

Пример: `<prmg name="p0" value="0.5"/>`

Атрибуты:

**name** [обязательно] – уникальное в пределах модели имя глобального параметра. Имя глобального параметра не должно совпадать с именами функций, измерений и локальных параметров.

**value** [обязательно] – числовое значение параметра. Допустимы все форматы записи, принятые стандартом C++.

**desc** [опционально] – описание параметра. Отображается в графическом интерфейсе.

#### **Prml**

Узел локального параметра модели. Параметр хранит свои значения в каждой ячейке расчетной сетки. Узел не имеет потомков.

Пример: `<prml name="p" value="p0"/>`

Атрибуты:

**name** [обязательно] – уникальное в пределах модели имя локального параметра. Имя локального параметра не должно совпадать с именами функций, измерений и глобальных параметров.

**value** [обязательно] – начальное значение локального параметра. Используется для определения значения начальной точки расчетной сетки, если не объявлены краевые условия модели. Может быть как числом, так и вычисляемым выражением, например значение объявленного ранее элемента модели.

**desc** [опционально] – описание локального параметра. Отображается в графическом интерфейсе.

**dim**

Узел измерения модели. Измерение является координатой расчетной сетки модели. Относительно них происходит численное дифференцирование локальных параметров модели. Узел не имеет потомков.

Пример: `<dim name="x" low="0" high="1" step="1e-4" smin="1e-7" smax="1e-2" rinc="2" rdec="0.5" retry="1" next="0" desc="пространственная координата"/>`

Атрибуты:

**name** [обязательно] – уникальное в пределах модели имя измерения. Имя измерения не должно совпадать с именами функций, а также локальных и глобальных параметров.

**low** [обязательно] – определяет нижнюю границу расчетной сетки по данному измерению.

**high** [обязательно] – определяет верхнюю границу расчетной сетки по данному измерению.

**step** [обязательно] – определяет начальное значение шага расчетной сетки по данному измерению.

**smin** [обязательно] – определяет минимальное значение шага расчетной сетки по данному измерению.

**smax** [обязательно] – определяет максимальное значение шага расчетной сетки по данному измерению.

**rinc** [обязательно] – коэффициент увеличения шага расчетной сетки по данному измерению при успешном расчете текущей ячейки. При  $rinc > 1$  шаг адаптивно подстраивается под ход решения. При  $rinc=1$  и  $rdec=1$  шаг фиксирован и расчетная сетка имеет регулярную структуру.

**rdec** [обязательно] – коэффициент уменьшения шага расчетной сетки по данному измерению при невозможности расчета текущей ячейки. При  $rdec < 1$  шаг адаптивно подстраивается под ход решения. При  $rinc=1$  и  $rdec=1$  шаг фиксирован и расчетная сетка имеет регулярную структуру.

**retry** [обязательно] – на сколько ячеек задерживать следующее приращение шага при неудаче расчета текущей ячейки. Обычно выбирается 1 для пространственной и 0 для временной координаты.

**next** [обязательно] – на сколько ячеек задерживать следующее приращение шага при успешном расчете текущей ячейки. Обычно выбирается 0 для обеих координат.

**desc** [опционально] – описание измерения. Отображается в графическом интерфейсе.

### **func**

Узел функции модели. Функция не хранит своего значения, а получает его на основе некоторой формулы. Узел не имеет потомков.

Пример: `<func name="Eq1" exp="Diff1(u,t) + (1.0 / b) * Diff1(p,x)"/>`

Атрибуты:

**name** [обязательно] – уникальное в пределах модели имя функции. Имя функции не должно совпадать с именами измерений, а также локальных и глобальных параметров.

**exp** [обязательно] – выражение, вычисляющее значение функции. Фактически, здесь может быть записана любая C++ функция, в том числе с циклами и ветвлениями. Доступ к



остальным элементам модели, таким как измерения, локальные и глобальные параметры, а также другие функции, осуществляется по их именам.

**desc** [опционально] – описание функции. Отображается в графическом интерфейсе.

### **problem**

Узел описания задачи модели. Хранит перечисление функций, к минимизации которых сводится численное решение, и параметров, в пространстве которых проводить минимизацию. Задачи должны быть объявлены в порядке уменьшения сложности, т.е. требований по соседним ячейкам, т.к. поиск подходящей задачи ведется последовательно начиная с первой. Например сначала нужно объявлять задачу динамики, а затем задачу статики, т.к. задача статики может быть решена в любой ячейке расчетной сетки, тогда как задача динамики может быть решена только во внутренних ячейках. Узел имеет потомков: как минимум один узел **param** и как минимум один узел **equ**.

Пример: `<problem name="dynamic" emin="1e-15" emax="1e-10" dmin="1e-10" it="100" desc="модель динамики">`

Атрибуты:

**emin** [обязательно] – минимальная ошибка вычислений, при которой прекращается численный счет.

**emax** [обязательно] – максимальная ошибка вычислений, при которой ячейка считается успешно рассчитанной, даже если процедура численного счета исчерпала себя.

**dmin** [обязательно] – если разница целевого функционала на соседних итерациях становится меньше этого значения, численный счет по текущему алгоритму прекращается.

**it** [обязательно] – количество итераций метода обобщенного циклического покоординатного спуска, после завершения которых происходит переход к методу Ньютона.

**desc** [опционально] – описание задачи

**param** в контексте **problem**

Узел объявления параметра модели неизвестным в данной задаче. Узел не имеет потомков.

Пример: `<param name="p"/>`

Атрибуты:

**name** [обязательно] – имя локального параметра модели.

**equ** в контексте **problem**

Узел объявления функции минимизируемой в данной задаче. Узел не имеет потомков.

Пример: `<equ name="Eq1"/>`

Атрибуты:

**name** [обязательно] – имя функции модели.

**chop**

Узел объявления краевых условий. Хранит функцию, определяющую значения локального параметра в зависимости от координат на заданных ячейках расчетной сетки. Узел имеет потомков: опциональный узел `param`.

Пример: `<chop name="initial p" prml="p" func="m_cChopConst" type="initial" desc="начальные условия для p">`

Атрибуты:

**name** [обязательно] – имя краевого условия, уникальное среди имен краевых условий.

**prml** [обязательно] – имя локального параметра модели, краевое условие для которого задается.

**func** [обязательно] – имя функции модели, определяющей закон изменения краевого условия.

**type** [обязательно] – тип краевого условия. Возможные значения: “initial” – начальные условия, “boundary” – граничные условия.

**desc** [опционально] – описание краевого условия. Отображается в графическом интерфейсе.

**param** в контексте **chop**

Узел параметра краевых условий. Каждое краевое условие хранит набор параметров функций изменения краевых условий (например амплитуда, частота, фаза и т.п.). Узел не имеет потомков.

Пример: `<param name="m_cChopConstOffset" value="p0"/>`

Атрибуты:

**name** [обязательно] – имя параметра краевого условия.

**value** [обязательно] – значение параметра краевого условия.

**chopfunc**

Узел объявления закона изменения краевых условий. Представляет собой периодическую функцию. Узел не имеет потомков.

Пример: `<chopfunc name="esin"`

`litname="ExpSin"`

`desc="экспоненциальный синус"`

`exp="esin_amp * pow ( sin ( esin_phs + 2*M_PI*m_cChopArg*esin_prd ), esin_exp ) + esin_ofs"`

`period="esin_prd"`

`min="esin_ofs + esin_amp * min ( 0.01, pow ( -1.01, esin_exp ) )"`

`max="esin_ofs + esin_amp"/>`

Атрибуты:

**name** [обязательно] – уникальное имя среди имен законов изменения краевых условий.

**litname** [обязательно] – литературное имя закона изменения краевых условий. Отображается в графическом интерфейсе.

**exp** [обязательно] – собственно сам закон изменения краевых условий. Значение координаты-аргумента автоматически подставляется в параметр `m_cChopArg`.

**period** [обязательно] – период закона изменения краевых условий.

**min** [обязательно] – минимальное значение закона изменения краевых условий.

**max** [обязательно] – максимальное значение закона изменения краевых условий.

**desc** [опционально] – описание закона изменения краевых условий. Отображается в графическом интерфейсе.

## **chopparam**

Узел объявления параметра закона изменения краевых условий. Экземпляр такого параметра копируется в каждый закон изменения краевых условий. Узел не имеет потомков.

Пример: `<chopparam name="esin_amp" value="1" litname="Amp" desc="Амплитуда"/>`

Атрибуты:

**name** [обязательно] – имя параметра закона изменения краевых условий. Уникально среди себе подобных.

**value** [обязательно] – значение параметра по умолчанию.

**litname** [обязательно] – литературное имя параметра. Отображается в графическом интерфейсе.

**desc** [опционально] – описание параметра закона изменения краевых условий. Отображается в графическом интерфейсе.

## **retry**

Узел условия на перерасчет ячейки. При выполнении данного условия ячейка считается не рассчитанной, происходит уменьшение шага и расчет повторяется. Узел не имеет потомков.

Пример: `<retry cond="M &lt; 0.01 || G &lt; 0.01 || T &lt; 0.01"/>`

Атрибуты:

**cond** [обязательно] – условие на перерасчет ячейки. Результат приводится к логическому типу.

## **stop**

Узел условия на останов расчета. При выполнении данного условия расчет считается успешно завершенным. Узел не имеет потомков.

Пример: `<stop cond="Gcur = LastLayerAvg ( G ) - Gmin"/>`

Атрибуты:

**cond** [обязательно] – условие на останов расчета. Если результат выражения положительный, расчет продолжается. Иначе – завершается.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Павленко В.Г. Основы механики жидкости. / В.Г. Павленко. – Л.: Судостроение, 1988. – 240 с.
2. Тучинский М.Р. Математическое моделирование и оптимизация пиролизных установок. / М.Р. Тучинский, Ю.В. Родных. – М.: Химия, 1979.
3. Задорожний В.Г. Оптимизация процесса пиролиза бензина / В.Г. Задорожний Е.Л., Ульянова, Ю.В. Родных // Дифференциальные уравнения и их приложения. – Воронеж: ВГУ, 1985. – С. 25-32.
4. Подвальный С.Л., Барабанов В.Ф. Моделирование периодического управления расходом сырья процесса пиролиза углеводородов. / – Воронеж, 1988. -13 с. –Деп. В ВИНТИ, № 57-ХП88.
5. Самарский А.А. Вычислительный эксперимент. / А.А. Самарский, Ю.П. Попов –М.: Знание. 1983.
6. Афанасьев В.Н. Математическая теория конструирования систем управления. / В.Б. Колмаковский, В.Р. Носов – М.: Высш. шк., 1989. -477 с.
7. Математическое моделирование. Нелинейные дифференциальные уравнения математической физики. – М.: Наука, 1987.
8. Романов А.Н. Имитаторы и тренажеры в системах отладки АСУТП. / А.Н. Романов, В.П. Жабеев – М.: Энергоатомиздат, 1987. – 112 с.
9. Самарский А.А. Современные проблемы математической физики и вычислительной математики. / А.А. Самарский – М.: Наука, 1982. – С. 332.
10. Справочник проектировщика АСУТП / Г.Л. Смилянский, Л.З. Амлинский, В.Я. Баранов и др.; под ред. Г.Л. Смилянского. – М.: Машиностроение, 1983. – 527 с.
11. Норкин С.Б. Дифференциальные уравнения второго порядка с запаздывающим аргументом. / С.Б. Норкин - М.: Наука, 1965.

12. Солодов А.В. Система с переменным запаздыванием. / А.В. Солодов, Е.А. Солодова - М.: Наука, 1980. 384 с.
13. Эльсгольц Л.Э. Введение в теорию дифференциальных уравнений с отклоняющимся аргументом. / Л.Э. Эльсгольц, С.Д. Норкин. - М.: Наука, 1971.
14. Янушевский Р.Т. Управление объектами с запаздыванием. / Р.Т. Янушевский – М.: Наука, 1978. – 416 с.
15. Айзерман М.А. Абсолютная устойчивость регулируемых систем. / М.А. Айзерман, Ф.Р. Гантмахер. – М.: Изд-во АН СССР, 1963.
16. Андронов А.А. Теория колебаний. / А.А. Витт, С.Э. Хайкин. – М.: Наука, 1981.
17. Барбашин Е.А. Введение в теорию устойчивости. / Е.А. Барбашин. – М.: Наука, 1967.
18. Неймарк Ю.И. Стохастические и хаотические колебания. / Неймарк Ю.И., П.С. Ланда – М.: Наука, 1987.
19. Нелепин Р.А. Точные аналитические методы в теории нелинейных автоматических систем. / Р.А. Нелепин – Л.: Судостроение, 1967.
20. Попов Е.П. Теория нелинейных систем автоматического регулирования и управления. / Е.П. Попов – М.: Наука, 1979. 351 с.
21. Бутенин Н.В. Точные аналитические методы исследования нелинейных автоматических систем. / Н.В. Бутенин, Р.А. Нелепин // Прикладная математика и кибернетика. М.: Наука, 1973. С. 39-45.
22. Четаев Н.Г. Устойчивость движения. / Н.Г. Четаев – М.: Гостехиздат, 1946.
23. Дэннис Дж. Численные методы безусловной оптимизации и решения нелинейных уравнений. / Дж. Дэннис, Р.Шнабель – М.: Мир, 1988.

24. Мак-Кракен Д. Численные методы и программирование на Фортране. / Д. Мак-Кракен, У. Дорн – М.: Мир, 1977.

25. Островский А.М. Решение уравнений и систем уравнений. / А.М. Островский – М.: ИЛ, 1963.

26. Трауб Дж. Итерационные методы решения уравнений. / Дж. Трауб – М.: Мир, 1985.

27. Вержбицкий В.М. Основы численных методов. / В.М. Вержбицкий – М.: Высш. шк., 2002 – 840 с.

28. Васильев Ф.П. Численные методы решения экстремальных задач. \ Ф.П. Васильев – М.: Наука, 1988.

29. Галлеев Э.М. Оптимизация: теория, примеры, задачи. \ Э.М. Галлеев, В.Н. Тихомиров – М.: Эдиториал УРСС, 2000.

30. Гилл Ф. Практическая оптимизация / Ф.Гилл, У. Мюррей, М.Райт – М.: Мир, 1985.

31. Лесин В.В. Основы методов оптимизации. / В.В. Лесин, Ю.П. Лисовец – М.: Изд-во МАИ, 1995.

32. Neumaier A.; Interval Methods for Systems of Equations // Cambridge University Press, Cambridge, 1990 – С. 143-145.

33. Цирлин А.М. Оптимальные циклы и циклические режимы. / А.М. Цирлин – М.: Энергоатомиздат, 1985. – 264 с.

34. Кафаров В.В. Оптимизация теплообменных процессов и систем. / В.В. Кафаров, В.Г. Мешалкин, Л.В. Гурьева – М.: Энергоатомиздат, 1988.

35. Цирлин А.М. Вариационные методы оптимизации управляемых объектов. \ А.М. Цирлин, В.С. Балакирев, Е.Г. Дудников – М.: Энергия, 1976.

36. Балакирев В.С. Оптимальное управление процессами химической технологии (Экстремальные задачи в АСУ). \ В.С. Балакирев, В.М. Володин, А.М. Цирлин – М.: Химия, 1978.–383 с.

37. Кафаров В.В. Методы кибернетики в химии и химической технологии. / В.В. Кафаров – М.: Химия, 1985. – 448 с.

38. Бодров В.И. Оптимизация процесса пиролиза с использованием нестационарной математической модели / В.И. Бодров, Е.Г. Вильский. // Теор. основы хим. технол. – 1977. - № 45. – С. 750-756.

39. Федоренко Р.П. Приближенное решение задач оптимального управления. / Р.П. Федоренко – М.: Наука, 1978. – 448 с.

40. Розенброк Х. Вычислительные методы для инженеров-химиков. / Х. Розенброк, С. Стори – М.: Мир, 1968.

41. Черноруцкий И. Г. Методы оптимизации в теории управления. / И.Г. Черноруцкий – СПб.: Питер, 2004.

42. Уилкинсон Дж. Х. Алгебраическая проблема собственных значений. / Дж.Х. Уилкинсон – М.: Наука, 1970.

43. Парлетт Б. Симметричная проблема собственных значений. / Б. Парлетт – М.: Мир, 1983.

44. Химмельблау Д. Прикладное нелинейное программирование. / Д. Химмельблау – М.: Мир, 1975.

45. Фадеев Д. К. Вычислительные методы линейной алгебры. / Д.К. Фадеев, В.Н. Фадеева – М.: Физматгиз, 1963.

46. Уилкинсон Дж. Х. Справочник алгоритмов на языке АЛГОЛ. / Дж.Х. Уилкинсон, С. Райнш. – М.: Машиностроение, 1976.

47. Chen, S., C.F.N. Cowan, P. M. Grant.; Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks // 1991 – С. 302-309.

48. James A. Freeman, David M. Skapura.; Neural networks : algorithms, applications, and programming techniques // 1991. – С. 103-106.

49. Гилл Ф. Практическая оптимизация. / Ф, Гилл, У. Мюррей – М.: Мир, 1985.



50. Ракитский Ю.В. Численные методы решения жестких систем. / Ю.В. Ракитский, С.М. Устинов, И.Г. Черноруцкий – М.: Наука, 1979.

51. Черноруцкий И. Г. Оптимальный параметрический синтез: электротехнические устройства и системы. / И.Г. Черноруцкий – Л.: Энергоатомиздат, 1987.

52. Барабанов В.Ф. Интерактивные средства моделирования сложных технологических процессов. / В.Ф. Барабанов, С.Л. Подвальный – Воронеж: ВГТУ, 2000.

53. Барабанов В.Ф. Моделирование и визуализация динамических систем эволюционного типа. / В.Ф. Барабанов, О.С. Плахотнюк. // Вестник Воронежского государственного технического университета. – 2005. – Том 1, №5. –С. 40-42.

54. Плахотнюк О.С. Применение эффективного алгоритма адаптивной подстройки шагов дифференцирования с памятью для численного решения уравнений эволюционного типа. / О.С. Плахотнюк, В.Ф. Барабанов. // Вестник Воронежского государственного технического университета. – 2007. – №1.

55. Плахотнюк О.С. Прогнозирование и предотвращение экстремальных ситуаций в ходе моделирования динамических систем эволюционного типа. / О.С. Плахотнюк, В.Ф. Барабанов. // Вестник Воронежского государственного технического университета. – 2007. – №1.

56. Барабанов В.Ф., Плахотнюк О.С. Программный модуль “Система визуализации численных данных на плоскости”, ФАП ВНТИЦ № 2649916100139 от 22.12.2003.

57. Плахотнюк О.С. Универсальный визуализатор числовых данных. / О.С. Плахотнюк, А.В. Барабанов. // Современные проблемы информатизации в системах моделирования, программирования и телекоммуникаций: сб. тр. - Воронеж: Научная книга, 2004. Вып. 9. -С. 370-371.

58. Плахотнюк О.С. Система визуализации численных данных на плоскости. / О.С. Плахотнюк, А.В. Барабанов. //

Информационные технологии моделирования и управления: междунар. сб. науч. тр. - Воронеж: Научная книга, 2004. Вып. 12. -С. 110-114.

59. Барабанов В.Ф. Система визуализации численных данных в трехмерном пространстве. / В.Ф. Барабанов, О.С. Плахотнюк. // Информационные технологии моделирования и управления: междунар. сб. науч. тр. - Воронеж: Научная книга, 2004. Вып. 12. -С. 43-46.

60. Барабанов В.Ф. Визуализация результатов моделирования. / В.Ф. Барабанов, О.С. Плахотнюк. // Новые технологии в научных исследованиях, проектировании, управлении, производстве: труды всерос. конф.- Воронеж, 2004. -С. 23-24.

61. Барабанов В.Ф. Графическое представление результатов моделирования. / В.Ф. Барабанов, О.С. Плахотнюк. // Современные проблемы механики и прикладной математики: сборник трудов международной школы- семинара. Воронеж:, ВГУ, 2004. -С. 63-66.

62. Барабанов В.Ф. Интерактивная система моделирования динамических режимов систем эволюционного типа. / В.Ф. Барабанов, О.С. Плахотнюк. // Современные проблемы информатизации в непромышленной сфере экономики: Сб. трудов. – Воронеж: Научная книга, 2005. Вып. 10. –С. 91-92. материалы X междунар. науч. конф..

63. Плахотнюк О.С., Хмелевской К.Г. Разработка интерактивной среды численного исследования математических моделей.

64. Плахотнюк О.С. О декомпозиции математической модели в интерактивной среде моделирования. / О.С. Плахотнюк, В.Ф. Барабанов. // Современные проблемы информатизации в моделировании и программировании: материалы XI междунар. науч. конф. - Воронеж: Научная книга, 2006. Вып. 11. -С. 197-198.

65. Плахотнюк О.С. Использование радиально-базисной нейронной сети для численного решения систем

нелинейных уравнений. / О.С. Плахотнюк, В.Ф. Барабанов.  
// Прикладные задачи моделирования и оптимизации:  
межвуз. сб. науч. трудов. – Воронеж: ВГТУ, 2006. -С. 122-  
128.

66. Барабанов В.Ф. Исследование динамических режимов систем эволюционного типа. / В.Ф. Барабанов, О.С. Плахотнюк. // Новые технологии в научных исследованиях, проектировании, управлении, производстве: труды всерос. конф. – Воронеж: 2006. -С. 31.

# ОГЛАВЛЕНИЕ

## ВВЕДЕНИЕ

3

## 1. ПРОБЛЕМАТИКА МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ РЕЖИМОВ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА.. 6

1.1. Особенности моделирования динамических режимов эволюционных процессов..... 6

1.2. Аппарат динамических систем эволюционного типа..... 9

1.2.1. Математические модели динамических систем эволюционного типа ..... 9

1.2.2. Методы исследования динамических режимов систем эволюционного типа..... 10

1.2.3. Методы численного решения систем нелинейных уравнений ..... 15

1.3. Сравнительный обзор распространенных средств математического моделирования динамических режимов систем эволюционного типа..... 17

## 2. ФОРМИРОВАНИЕ СТРУКТУРЫ СИСТЕМЫ МНОГОВАРИАНТНОГО ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА ..... 23

2.1. Формирование требований к функциональности системы многовариантного интерактивного моделирования... 23

2.1.1. Формирование вычислительных требований к моделированию динамических систем эволюционного типа... 23

2.1.2. Формирование требований к средствам квазиоптимального управления ..... 25

2.1.3. Постановка задачи визуализации работы отдельных компонентов системы интерактивного моделирования.....	29
2.1.4. Формирование требований к интерактивному режиму моделирования и взаимодействию с оператором .....	30
2.2. Модульная структура системы многовариантного интерактивного моделирования.....	31
2.3. Многовариантный подход к численному решению систем эволюционных уравнений .....	35

### 3. РАЗРАБОТКА АЛГОРИТМОВ И МЕТОДОВ МНОГОВАРИАНТНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ ЭВОЛЮЦИОННОГО ТИПА . 38

3.1. Постановка задачи численного решения систем эволюционных уравнений в частных производных .....	38
3.2. Методология численного решения систем эволюционных уравнений в частных производных .....	39
3.2.1. Переход от аналитического вида решаемой системы эволюционных уравнений к разностной аппроксимации .....	39
3.2.2. Функциональный анализ различных способов задания краевых условий динамической системы эволюционного типа .....	40
3.2.3. Применение динамической расчетной сетки с нерегулярным шагом для решения эволюционных уравнений в частных производных .....	41
3.2.4. Разработка алгоритма добавления новой ячейки в нерегулярную расчетную сетку с использованием частичной регуляризации сетки .....	44
3.2.5. Адаптация алгоритма обобщенного покоординатного спуска для нахождения численного решения в узлах нерегулярной расчетной сетки с адаптивной подстройкой шага дифференцирования .....	49
3.2.6. Модификация пространственного метода численного решения систем нелинейных уравнений с	

применением радиально-базисной нейронной сети для хранения информации о пространстве решения .....	54
3.3. Повышение эффективности алгоритма адаптивной подстройки шага численного дифференцирования систем эволюционного типа .....	65
3.3.1. Формирование критерия оптимальности динамической расчетной сетки .....	65
3.3.2. Выявление обобщенного алгоритма адаптивной подстройки шага для двухмерной прямоугольной нерегулярной расчетной сетки .....	65
3.3.3. Разработка эффективного алгоритма адаптивной подстройки шага с памятью .....	76
3.3.4. Исследование параметров эффективности алгоритма адаптивной подстройки шага с памятью по сравнению с широко распространенными алгоритмами .....	79
3.4. Методология параметрической оптимизации эволюционных динамических систем .....	83
3.4.1. Постановка задачи параметрической оптимизации .....	83
3.4.2. Редуцирование задачи многокритериальной оптимизации к однокритериальной задаче .....	88
3.4.3. Явление овражности и методы распознавания овражной ситуации .....	90
3.4.4. Методология решения общей задачи однокритериальной оптимизации динамических систем эволюционного типа .....	95
3.5. Вывод эволюционных динамических систем из экстремальных ситуаций .....	97
4. РАЗРАБОТКА АЛГОРИТМОВ И МЕТОДОВ ФУНКЦИОНИРОВАНИЯ МОДУЛЕЙ СИСТЕМЫ МНОГОВАРИАНТНОГО ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ.....	102

4.1. Разработка пользовательского интерфейса управления процессом моделирования в интерактивном

режиме .....	102
4.1.1. Формирование графического пользовательского интерфейса с учетом управления в экстремальных ситуациях.....	102
4.1.2. Удаленное управление и текстовый интерфейс работы с системой интерактивного моделирования .....	105
4.2. Разработка специализированного языка описания математических моделей для автоматизации задания входных данных .....	106
4.3. Реализация вычислительного модуля системы моделирования динамических систем эволюционного типа..	107
4.3.1. Осуществление автоматического перехода от математического описания модели к численной аппроксимации .....	107
4.3.2. Реализация методов задания краевых условий динамической системы .....	108
4.3.3. Формирование и хранение динамической расчетной сетки с нерегулярным шагом дифференцирования.....	109
4.3.4. Нахождение численного решения системы эволюционных уравнений в узлах расчетной сетки.....	109
4.4. Сохранение и подсчет частоты использования шагов дифференцирования.....	110
4.5. Квазиоптимальное управление параметрами модели через граничные условия при помощи типовых законов управления.....	111
4.6. Методология визуализации результатов моделирования динамических систем эволюционного типа..	112
4.6.1. Синтез нерегулярной самоорганизующейся структуры данных для хранения численной информации при визуализации .....	112
4.6.2. Разработка метода проекции высокочастотных численных данных на пространство визуализации .....	113

5. ПРИМЕНЕНИЕ СИСТЕМЫ МНОГОВАРИАНТ-  
НОГО ИНТЕРАКТИВНОГО МОДЕЛИРОВАНИЯ ДЛЯ  
ИССЛЕДОВАНИЯ ЭВОЛЮЦИОННЫХ СИСТЕМ ..... 116

5.1. Апробация работы интерактивной системы  
моделирования на модельной задаче процесса  
регенерации неподвижного слоя катализатора..... 116

5.2. Представление модели регенерации неподвижного  
слоя катализатора на языке описания математических  
моделей в интерактивной системе моделирования ..... 122

Заключение ..... 132

Приложение Тэги языка описания моделей системы  
интерактивного моделирования..... 133

Библиографический список ..... 141