

**Московский физико-технический институт  
(Государственный университет)  
Факультет Молекулярной и Биологической Физики  
Кафедра Биофизики и Экологии**

**Разработка методов компьютерного моделирования  
функциональных систем организма  
(на примере сердечно-сосудистой системы)**

**Магистерская диссертация**

**Бурыкин А. А.**

**Научный руководитель**

**к. ф. - м. н. Тимин Е. Н.**

**Москва 2000**

## Содержание

1 Введение.....	2
2 Литературный обзор .....	6
2.1 Математические модели кровообращения .....	6
2.2 Математические модели организма человека .....	7
2.3 Декомпозиция сложных систем .....	9
2.4 Средства визуального моделирования сложных систем.....	12
3 Материалы и методы .....	15
4 Описание методов декомпозиции физиологических систем.....	17
5 Объектно-ориентированная библиотека классов функциональных элементов сердечно-сосудистой системы .....	25
5.1 Объектно-ориентированный подход к созданию сложных компьютерных моделей в физиологии .....	25
5.2 Описание библиотеки классов функциональных элементов сердечно-сосудистой системы. ....	28
5.3 Использование библиотеки классов .....	46
5.4 Преимущества использования объектно-ориентированной библиотеки элементов .....	50
6 Среда визуального моделирования функциональных систем организма .....	51
6.1 Описание программного продукта .....	51
6.2 Пример использования «Среды визуального моделирования».....	63
7 Выводы.....	73
8 Список литературы .....	75
9 Приложение. Модель работы сердца и замыкание модели кровеносной системы.....	78
9.1 Имитационное моделирование сердечной деятельности.....	78
9.1.1 Краткие сведения по физиологии работы сердца.....	78
9.1.2 Математическая формулировка эмпирических закономерностей .....	80
9.2 Алгоритм решения системы нелинейных уравнений замкнутой кровеносной системы .....	82
9.3 Резюме .....	85

## 1 Введение

В 1987 году Malvanу [15] своем обзоре по физиологии кровообращения отметил, что на современном этапе исследований накоплено такое количество экспериментальных данных, что разобраться и понять их взаимосвязь можно только при помощи математического моделирования. То же относится и к другим системам организма.

В данный момент существует множество математических моделей, как отдельных участков, так и всей сердечно-сосудистой системы в целом, а так же модели, объединяющие ее с другими системами организма.

Однако в связи с разнообразием таких моделей возникает целый ряд проблем, которые были сформулированы В. А. Лищуком [16] при рассмотрении различных моделей сердечно-сосудистой системы:

1. отсутствие преемственности в разработке моделей;
2. многочисленность несвязанных подходов;
3. несравнимость моделей между собой;
4. отсутствие общего подхода к моделированию контуров регуляции физиологических систем;
5. трудность выделения наиболее существенных элементов и общих соотношений, причем в такой форме, которая позволила бы получить описание данной физиологической системы различной детальности, объема и специфики;

Современные математические модели кровообращения с необходимостью являются компьютерными моделями, т.к. их исследование в силу их сложности проводится исключительно с помощью компьютера. Поэтому от эффективности компьютерного представления модели существенно зависит прогресс в исследованиях в области моделирования физиологических систем, а, следовательно, и во всей физиологии.

Одним из требований к таким (реализованным на компьютере) моделям, в дополнение к уже указанным Лищуком проблемам является организация возможности быстрого развития модели, так как модель постоянно развивается, появляются новые экспериментальные данные,

пересматриваются модели уже существующих подсистем, добавляются новые подсистемы организма.

Кроме того, поскольку компьютерная модель может быть использована для расчета задач различной вычислительной мощности, то очень желательно оказывается обеспечить переносимость компьютерной модели, т.е. возможность ее работы как на персональных компьютерах (в том случае, когда не требуется высокий уровень детализации процессов и/или расчеты на больших временах), так и на рабочих станциях типа Alpha (в противоположном случае).

Таким образом, можно выделить еще две проблемы компьютерного моделирования (как в физиологии, так, впрочем, и в других областях):

6. Трудности с модернизацией и расширением моделей;
7. Невозможность переноса компьютерной модели между компьютерами с разными вычислительными мощностями.

Существующие компьютерные реализации математических моделей кровообращения (так же как и других систем организма) в полной мере не решают вышеописанных проблем.

Таким образом, возникает необходимость в усовершенствовании подхода к компьютерному моделированию сложных физиологических систем и разработки на его основе программного продукта, который бы позволял быстро и эффективно создавать расширяемые математические модели систем организма. Причем такой программный продукт должен быть доступен для пользователя, который может не владеть в полной мере современными средствами программирования и численными методами.

Решению этих двух задач и посвящена данная работа. Для этой цели применяются современные компьютерные технологии (такие как OOA, CASE, RAD) и разрабатываются проблемно-зависимые методы представления и обработки данных по физиологии.

В качестве примера применения разработанного программного продукта приводится одна из моделей сердечно – сосудистой системы. На данном примере демонстрируются преимущества применения предлагаемого подхода к компьютерному моделированию сложных физиологических систем,

реализованного в виде описываемого в данной работе программного продукта.

В заключение хочу выразить благодарности:

Е. Н. Тимину – за научное руководство,

А. В. Евдокимову – за руководство проектом по разработке программного продукта,

А. С. Холодову – за общее руководство проектом,

А. В. Голеневу – за помощь в изучении современных информационных технологий.

## 2 Литературный обзор

### 2.1 Математические модели кровообращения

Кровеносная система является одной из основных систем организма человека и состоит из малого круга, осуществляющего газообмен с дыхательной системой, и большого круга кровообращения, питающего ткани и органы [13, 14]. В сердечно-сосудистой системе (в каждом круге) выделяется три взаимосвязанных звена: артериальное, венозное и связующее их капиллярное.

Процесс перемещения крови по сосудам подчинен тем же законам, что и движение жидкости в любых системах трубок. В физиологии кровообращения возникает задача отделить прямые проявления всеобщих законов физики от эффектов, опосредованных физиологической регуляцией.

Важнейшей физиологической ролью кровообращения является своевременная доставка кислорода и питательных веществ всем клеткам организма, и тем самым поддержание постоянства внутренней среды, окружающей клетки. Соответственно к регуляции кровообращения относят явления (реакции), направленность которых усматривается в предотвращении или уменьшении нарушений постоянства внутренней среды, возникающих при различных состояниях. Учитывают, кроме того, что во многих жизненных и искусственных ситуациях регуляция кровообращения обеспечивает также экономизацию резерва кровотока, который в организме ограничен.

При моделировании система кровообращения в целом, она подразделяется на некоторое количество подсистем (сосудистых участков, полостей сердца и т. п.), описываемых чаще всего в сосредоточенных параметрах. Метод сосредоточенных параметров широко применяется для описания сердечно-сосудистой системы [15, 16, 17]. Он состоит в том, что все связанные с гемодинамикой функции и свойства участка сосуда (объемная скорость крови, давление, объем, сопротивление и др.) сосредоточены, т. е. их численные оценки приведены к некоторой одной пространственной (координатной) точке. При этом обычно принимаются допущения о

несжимаемости крови, ламинарности ее потока и линейной зависимости вязкости от скорости.

Существуют довольно развитые теоретические разработки, касающиеся общего подхода к моделированию системы кровообращения. В частности В.А. Лищуком [16, 17] была предложена так называемая «базисная модель сердечно-сосудистой системы», единицей которой является «обобщенный сосудистый участок». Обобщенный сосудистый участок понимается как отдельный сосуд либо система сосудов некоторого участка ткани или органа, которые по условию задачи выделены в качестве относительно самостоятельной области и характеризуются усредненными показателями кровотока, давления, сопротивления, эластичности и др.

## ***2.2 Математические модели организма человека***

Одной из тенденций математического моделирования в физиологии является создание комплексных моделей взаимосвязанных систем организма человека, охватывающих регуляцию значительного числа наиболее важных физиологических функций [22, 23, 25]. Среди наиболее развитых моделей подобного рода можно назвать модель общей регуляции кровообращения Гайтона и др.[18, 21] и модель внутренней сферы организма Амосова и др. [19].

Модель общей регуляции кровообращения Гайтона содержит около 350 элементов, связывающих между собой более 200 физиологически интерпретируемых переменных. Модель описывает следующие физиологические процессы: гемодинамику системного и легочного кровообращения, автономное управление кровообращением, кислородное обеспечение тканей мышц и «немышц», регуляцию количества эритроцитов, гипертрофию и декомпенсацию сердца, экскреторную функцию почек, регуляцию потребления воды, систему ренин – ангиотензин – альдестерон, динамику транскапиллярного обмена жидкости в легких и других тканях. Модель ориентированна на воспроизведение нормальных и патологических явлений в системе кровообращения и связанных с ней системах

кислородного снабжения тканей и водно-солевого обмена на интервалах времени от нескольких минут до двух месяцев.

Большая комплексная модель регуляции физиологических систем – модель внутренней сферы организма человека – разработана А. М. Амосовым и др. Модель объединяет следующие подсистемы: кровообращение, внешнее дыхание, тканевой метаболизм, терморегуляцию и водно-солевой обмен. В целом модель внутренней сферы организма объединяет более 200 физиологических переменных и содержит более 400 элементов. Особенностью модели является ее блочная структура, позволяющая исследовать подсистемы в различных комбинациях в зависимости от цели и рассматриваемых временных интервалов. Комплексная модель предусматривает возможность исследований процессов регуляции важнейших жизненных функций при физической и тепловой нагрузках, изменениях состава вдыхаемого воздуха, потребления солей и воды, нарушения сердечной функции. Интервалы времени исследуемых на модели процессов составляют от нескольких минут до двух недель.

Значительный интерес представляет модель гомеостаза, предложенная коллективом японских ученых [24, 26].

Это крупномасштабная модель управления жидкими средами человеческого организма, которая была создана для изучения проблем, связанных с нарушениями этих сред, и жидкостной терапии. Модель, содержащая подсистемы кровообращения, дыхания, почечной функции и жидкости внутри- и внеклеточной областей, описана математически как система нелинейных дифференциальных уравнений от более чем 200 переменных и постоянных параметров.

Поведение модели для различных видов входной информации, полученное с помощью численных экспериментов на компьютере, хорошо согласуется с экспериментальными данными, относящимися к жидким средам человеческого организма и к нарушениям кислотно-щелочного равновесия.

Модель реализована в виде программы, реализованной на языке Pascal, и имеет модульную структуру.

Однако все подобные компьютерные модели обладают одним недостатком – они с трудом могут быть модернизированы даже своими разработчиками, и пользователи могут в весьма ограниченных пределах менять входные данные и параметры модели [4].

Некоторые программные пакеты, предназначенные для моделирования (такие как система «Динамоэксперт» [20]) дают возможность пользователю модифицировать модель. Однако такие программы требуют от пользователя знаний численных методов и программирования (например, знания языка Fortran), что затрудняет их использование специалистами – физиологами.

### **2.3 Декомпозиция сложных систем**

Как весь организм, так и отдельные его подсистемы, удовлетворяют пяти признакам сложной системы, описанных Г. Бучем [7]:

- Сложные системы часто являются иерархическими и состоят из взаимозависимых подсистем, которые в свою очередь также могут быть разделены на подсистемы, и т. д., вплоть до самого низкого уровня.
- Выбор, какие компоненты в данной системе считаются элементарными, относительно произволен и в большой степени оставляется на усмотрение исследователя.
- Внутрикомпонентная связь обычно сильнее, чем связь между компонентами. Это обстоятельство позволяет отделять «высокочастотные» взаимодействия внутри компонентов от «низкочастотной» динамики взаимодействия между компонентами.
- Иерархические системы обычно состоят из немногих типов подсистем, по-разному скомбинированных и организованных.
- Сложная система является результатом развития более простой системы

Таким образом, организм является типичной сложной системой, в том смысле, который придает данному термину объектно-ориентированный анализ. Действительно, в организме легко выделить иерархию взаимодействующих подсистем. Так сам организм состоит из дыхательной, кровеносной, пищеварительной и т. п. систем, кровеносная система состоит

из сосудов, сердца и т. д. Подобным же способом можно описать в применении к физиологии и остальные четыре признака сложной системы.

Для эффективного моделирования сложной системы необходима ее декомпозиция – разбиение моделируемой системы на части.

Существующие методы для декомпозиции сложных систем можно разделить на три группы [7, 8, 10]:

Метод структурного проектирования сверху вниз (SADT – Structured Analysis and Design Technique)

Метод потоков данных (DFD – Data Flow Diagrams)

Объектно-ориентированное проектирование (OOD – Object-Oriented Design)

(Первые два метода декомпозиции являются разновидностями структурного подхода к декомпозиции сложных систем. Сущность структурного подхода заключается в декомпозиции (разбиении) системы на автоматизируемые функции: система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, те – на задачи и так далее до конкретных процедур. При этом моделируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны.)

### ***3.1. Метод структурного проектирования сверху вниз***

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т. е. производимые им действия и связи между этими действиями.

Построение SADT – модели начинается с представления всей системы в виде простейшего компонента – одного блока и дуг, изображающих интерфейсы с функциями вне системы (см. рис. 1).

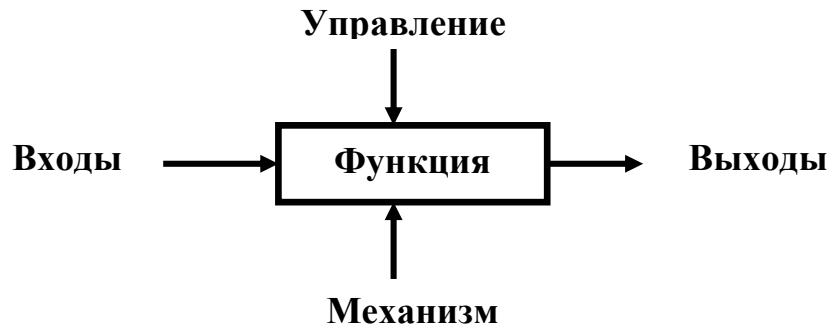


Рис.1. Функциональный блок и интерфейсные дуги.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки определяют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых показана как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом в целях большей детализации. Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию.

### **3.2. Моделирование потоков данных**

В этом методе система рассматривается как иерархия диаграмм потоков данных - преобразователей входных потоков в выходные. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы с внешними входами и выходами. Они детализируются с помощью диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут уровень декомпозиции, на котором процессы становятся элементарными и детализировать их далее невозможно.

### **3.3. Объектно-ориентированная декомпозиция**

В основе ООД лежит представление о том, что систему необходимо моделировать как совокупность взаимодействующих друг с другом объектов, рассматривая каждый объект как экземпляр определенного класса, причем классы образуют иерархию.

Сущность, имеющая значение в предметной области, обладающая набором параметров, которые однозначно ее выделяют среди других ей подобных, в информационном моделировании, называется объектом [7, 9].

При объектно-ориентированном анализе моделируемая система рассматривается как совокупность взаимодействующих объектов, причем эти объекты находятся между собой в отношениях наследования (один объект является потомком другого, т. е. произошел из другого путем добавления новых функциональных возможностей, а также изменения старых; последнее свойство называется полиморфизмом) и агрегации (один объект содержится в другом).

Организм естественно рассматривать как состоящий из взаимодействующих объектов. При таком рассмотрении каждый сосуд является объектом, который взаимодействует, например, с соседними сосудами, получая и передавая потоки крови.

Взаимодействия между объектами также естественны для физиологии, например, один объект - желудочек может содержаться в другом объекте - сердце, который является более сложным и, соответственно обладает более сложным поведением.

Иерархия объектов отражает эволюционное развитие различных подсистем в организме.

#### ***2.4 Средства визуального моделирования сложных систем***

Подавляющее большинство существующие методов декомпозиции сложных систем созданы для моделирования социальных, технических или комбинированных систем. (Однако, основываясь на кибернетическом единстве биологических, социальных и технических систем с точки зрения процессов управления ими, можно утверждать, что методы декомпозиции любых сложных систем будут похожими.) Соответственно практически все программные продукты, реализующие тот или иной метод декомпозиции (позволяющие визуально создавать модель системы), предназначены для моделирования бизнес – процессов (например, финансовых потоков или

документооборота) и используются для автоматизации деятельности предприятий.

Самым распространенным программным продуктом визуального моделирования общенаучного назначения является Simulink [27]. Однако Simulink плохо подходит для создания моделей в области физиологии. Во-первых, из-за того, что большая часть элементов модели в среде Simulink является математическими операциями и функциями (среди этих функций имеются дифференциальные и дискретные операторы, что формально позволяет графически описать любую систему уравнений). Поэтому использование такой системой специалистами – физиологами может вызывать трудности. Например, для решения с её помощью дифференциального уравнения нужно понимать смысл преобразования Лапласа. Во-вторых, из-за специфики физиологических систем строить их модели визуально непосредственно из отдельных параметров практически невозможно из-за катастрофического нарастания сложности модели (первоначально Simulink предназначался для создания в основном механических и электрических моделей, однако модель даже довольно простой механической системы – например, связанных маятников, выглядит очень громоздко).

Среди коммерчески распространяемых подобных программ, предназначенных для моделирования узких предметных областей выделяются такие программы, как OrCAD [29] и MicroSim [28]. Они используются для моделирования электрических схем, хотя основные их идеи и принципы и могут быть использованы (в силу электромеханической аналогии) для построения аналогичных программ визуального моделирования дыхания и кровообращения.

Программные продукты для визуального моделирования сложных систем в физиологии отсутствуют.

Таким образом, существующие на сегодняшний момент подходы к компьютерному моделированию в физиологии можно свести либо к использованию универсальных программ визуального моделирования (которые неприспособлены конкретно для создания физиологических

моделей), либо к использованию готовых программных продуктов, в которых уже заложена та или иная модель [30] (однако математические модели, лежащие в основе этих программ, зачастую скрыты от пользователя в программном коде, поэтому невозможно объяснить поведение модели, и, как следствие, проводить на ней какие-либо исследования).

### 3 Материалы и методы

При работе использовались язык программирования Java, программный продукт Rational Rose и система визуального программирования IBM VisualAge for Java. Созданный программный продукт тестировался на IBM – совместимых персональных компьютерах (Pentium, Pentium II и Pentium III) под управлением операционных систем Windows 95/98/2000/NT и на рабочей станции Dec Alpha под управлением операционной системы Digital Unix версии 4.0C.

Язык Java [12] разработан фирмой Sun Microsystems и выпущен в 1995 году. Таким образом, Java является самым современным из коммерчески используемых языков программирования.

Язык Java представляет собой объектно-ориентированный, интерпретируемый, переносимый язык программирования. Он особенно эффективно используется в качестве языка программирования для Internet, т. к. не зависит от используемой платформы (т.е. от архитектуры компьютера и используемой операционной системы). Данный язык позволяет создавать программный код, работающий на различных процессорах под управлением различных операционных систем. Переносимость осуществляется за счет того, работают внутри так называемой виртуальной машины Java (JVM), которая является фактически надстройкой над процессором, и поэтому программа, написанная на Java, не взаимодействует напрямую с конкретными особенностями той или иной платформы.

Программный продукт Rational Rose [7, 8] фирмы Rational Software Corporation (США) представляет собой объектно-ориентированное CASE – средство (Computer Added Software Engineering – разработка программного обеспечения с помощью компьютера), предназначенное для автоматизации этапов анализа и проектирования программного обеспечения, а также для генерации кодов на различных языках программирования и выпуска проектной документации [7, 10]. Rational Rose использует синтез – методологию объектно-ориентированного анализа и проектирования, основанную на подходах трех ведущих специалистов в данной области: Буча [7], Рамбо и Джекобсона. Разработанная ими универсальная нотация для

моделирования объектов UML (Unified Modeling Language – Унифицированный язык моделирования) претендует на роль стандарта в области объектно-ориентированного анализа и проектирования.

Среда визуального программирования IBM VisualAge for Java [11], разработана фирмой IBM и представляет собой RAD – средство (Rapid Application Development – быстрая разработка приложений), позволяющая в визуальном режиме создавать и связывать между собой элементы графического интерфейса пользователя (GUI – graphical user interface). При работе использовалась версия 2.0 этого программного продукта, использующая версию пакетов JDK (Java Developer's Kit – комплект разработчика на Java) 1.1.8.

#### **4 Описание методов декомпозиции физиологических систем.**

В данном разделе описывается подход к декомпозиции сложных физиологических систем (таких, как весь организм человека или отдельные его подсистемы). Данный подход реализован в процессе создания программного продукта «Среда визуального моделирования физиологических систем организма», который позволяет с помощью описанных в данном разделе методов декомпозиции физиологических систем, в визуальном режиме создавать, расширять и изменять модели этих систем.

Наряду с общими чертами, присущими любой сложной системе, физиологические системы имеют некоторые существенные особенности, резко отличающие процесс создания их моделей от процесса создания моделей в физике и химии.

Во-первых, сложность (в смысле многоэлементности) физиологических систем намного превосходит сложность большинства моделируемых физических или механических систем. Поэтому при моделировании физиологических систем не возможно пользоваться подходом (развитым, в частности в Simulink), в котором модель строится из отдельных переменных («параметров»), которые сложным образом связаны друг с другом и входят во множество соотношений. При попытке применить такой подход к физиологии, мы получаем при визуальном моделировании катастрофическое нарастание сложности, при которой даже один моделируемый орган (не говоря уже о целой функциональной системе организма) становится практически не обозримым (ни на экране, ни даже на распечатке).

Во-вторых, особенностью физиологической системы является во многих случаях незнание конкретного механизма взаимодействия и (само)регуляции различных органов и подсистем в организме. Поэтому не возможно создать их модель, написав соответствующее число уравнений (такое возможно лишь при моделировании простейших задач, относящихся скорее к гидродинамике, чем собственно к физиологии), решив которые, можно получить конкретную зависимость. В связи с этим приходится использовать либо экспериментально полученные зависимости, а в

некоторых случаях вообще экспертным образом заданные кривые. Даже те регуляторные взаимодействия, конкретный молекулярный или химический механизм которых известен, все равно при моделировании на уровне целых подсистем организма приходится заменять итоговыми (опять таки упрощенными и не следующими из каких-нибудь конкретных уравнений) кривыми, для того, что бы уменьшить сложность системы и сделать ее доступной расчету.

Предложенный подход к декомпозиции физиологических систем успешно решает обе эти главные проблемы.

При данном методе декомпозиции используется иерархический подход к моделированию сложных систем, при котором они разбиваются на более простые подсистемы.

Однако, как уже указывалось выше, подобную декомпозицию нецелесообразно проводить до уровня отдельных переменных (параметров).

Параметры вместе с теми зависимостями, в которые они входят (точнее с теми из зависимостей, которые возможно записать в виде уравнений, отражающие те или иные физические законы) объединяются в готовые для применения при моделировании блоки, называемые элементами. Элементы являются наиболее характерными совокупностями взаимодействующих параметров (например, представляют собой различные модели сосудов, сердца и т. п.), из которых, как из кирпичиков, можно строить модель физиологической подсистемы. (Заметим, что совокупность элементов на уровне реализации представляет собой иерархию классов, и поэтому при необходимости добавления новых элементов их легко получить путем агрегации и наследования от уже существующих.)

Иногда, однако, при создании модели не удается инкапсулировать некоторые параметры в какие-либо элементы, например, если этот параметр относится ко всей (под)системе (например возраст, вес и т.п.), т.е. является глобальным, или же к той подсистеме, которая напрямую не моделируется (например, такие параметры как нервный тонус, относящийся к нервной подсистеме, конкретной модели которой не создается). Такая же ситуация возникает при моделировании внешних по отношению к системе факторов

(например содержания тех или иных веществ во вдыхаемом воздухе, или веществ, которые поступают (вводятся при помощи капельницы и т. п.) в кровь).

Для моделирования таких ситуаций используются так называемые оторванные (свободные, внешние, висячие) параметры. (Везде в программном продукте они называются просто параметрами, т. е. также как и «обычные» параметры, инкапсулированные в элементы, что, конечно, может создать некоторую путаницу)

При моделировании системы используются два вида связей: связи между элементами и связи между параметрами.

Связи между элементами возникают при создании физико-геометрической модели системы, и представляют собой чисто «физические» (механические, гидродинамические и т. п.) связи между группами параметров, инкапсулируемых в данных элементах. Таким образом, с помощью связей между элементами можно построить модель «нерегулируемой физиологической системы».

(При визуальном моделировании для формализации процесса создания межсистемных связей (связей между элементами разных подсистем) используется так называемый внешний элемент, который представляет собой указатель на элемент, не относящийся к данной подсистеме.)

Связи между элементами представляют собой те механизмы взаимодействия между физиологическими объектами, которые можно записать с помощью уравнений (представляют собой «точно известные» физические законы).

Связи между параметрами предназначены для моделирования регуляторных процессов, протекающих в организме (поэтому такие связи называются также регуляторными). Такие связи представляют собой функциональные зависимости с участием сколь угодно большого числа параметров, относящихся к любым элементам.

К сожалению, в подавляющем большинстве случаев из эксперимента известны только связи между двумя какими-то двумя параметрами

(двумерные зависимости), поэтому возникает проблема восстановления по этим данным исходных сложных и многомерных регуляторных зависимостей.

Далее под словом «связь» понимается математическое выражение влияния только одного параметра на другой (те данные, которые получаются в эксперименте). Связи между связями далее называются «операциями» над связями, а результат применения операций к связям - «зависимостью» данного параметра (см. рис. 2).

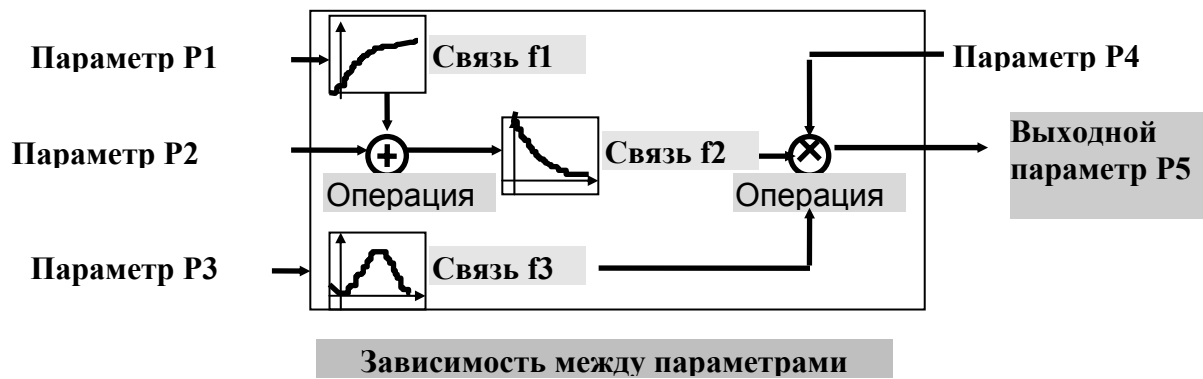


Рис. 2. Визуальное представление контуров регуляции

На рис. 2 графически представлена следующая зависимость между параметрами:

$$P5=P4*f3(P3)+f2(P2+f1(P1))$$

Сложные регуляторные механизмы описываются через зависимости, однако, экспериментальные данные можно получить только для связей и операций. Именно это диктует такую на первый взгляд сложную систему взаимосвязей параметров модели.

В случае, когда совместное влияние нескольких связей (например, связей от параметров 1 и 2 на рис. 2) не подчиняется законам аддитивности, мультипликативности (или других введенных операций), необходимо преобразовать выход одной из этих связей (в данном случае параметр 1) с помощью ещё одной специально введённой связи (в данном случае с помощью связи 1) и только после этого применять к ним операции сложения или умножения.

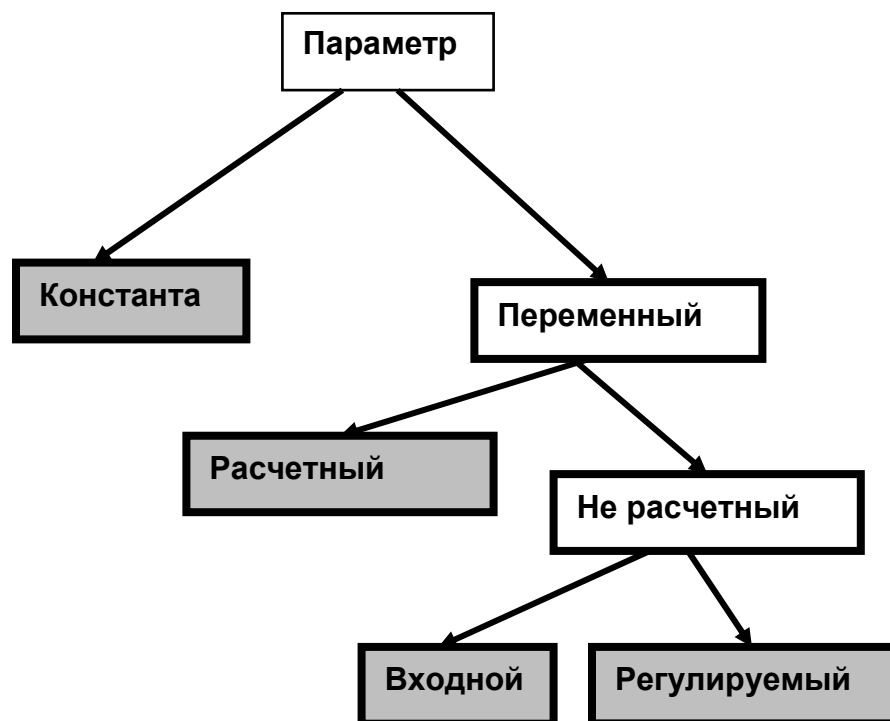
Свободные параметры могут участвовать только в регуляторных связях.

Связи между параметрами реализуют те сложные механизмы взаимодействия и регуляции в организме, которые не могут быть представлены в виде уравнений или систем уравнений и относительно которых существуют только экспериментальные данные или экспертные предположения. Поэтому связи между параметрами вводятся при создании модели явным образом, т.е. либо путем интерполяции вводимых значений, либо посредством изменения числовых значений коэффициентов в какой-нибудь функции, выбранной из библиотеки функций (которую, разумеется можно расширять и изменять).

Таким образом, разделение связей, существующих в системе на эти два вида позволяет успешно решить вторую проблему, возникающую при моделировании физиологических систем.

Ниже приводится классификация параметров (по типам) с точки зрения возможности его участия в регуляторных механизмах и с точки зрения процесса расчета модели.

Параметры могут быть четырех типов (см. рис. 3)



**Рис.3.** Типы параметров.

**Константа** – параметр не может меняться;

**Расчетный** – параметр может меняться только по тем законам, которые заложены в элементы (т. е. согласно уравнениям);

**Регулируемый** – параметр может участвовать в регуляторных связях (связях между параметрами);

**Входной** – параметр может меняться по желанию пользователя.

(Примечание. Входным может быть только регулируемый параметр, т. е. параметр может быть либо регулируемым, либо входным.)

В некоторых задачах возникает необходимость моделирования распространение различных химических веществ по организму. Для удобства проектирования моделей вещества не распределяются по элементам непосредственно, а группируются в новые объекты, называемые средами.

Среда представляет собой осмысленный и естественный с точки зрения физиологии набор веществ - воздух, кровь и т.п., которым потом и заполняются элементы (каждый элемент может заполняться только одной

средой). Поэтому в каждый элемент добавляется только один параметр (название среды), а не множество параметров (концентрации всех веществ, входящих в среду). Кроме того, при изменении состава воздуха или крови (добавления или удаления какого-нибудь вещества) достаточно один раз произвести изменение в нужной среде, а не делать изменения во всех элементах, которые содержат данное вещество.

Однако при моделировании процессов диффузии и поглощения (в отличие от процессов конвекции, для которых достаточно задание концентраций) необходимо задавать еще дополнительные параметры, такие как коэффициенты и скорости диффузии или характерное время поглощения. Эти параметры также инкапсулируются средой, т. к. по своей сути являются физическими свойствами среды по отношению к веществам. (В данной версии программного продукта среда может обладать лишь одним таким свойством. Иногда среды без таких физических свойств, а только с концентрациями называются множествами или наборами.)

Описанных выше понятий достаточно для проектирования большинства видов моделей функциональных систем организма на системном уровне (т. е. на том уровне, где элементарными объектами являются сосуд, желудочек и т. д.). Данная методология мало пригодна для создания, например сложных моделей одного сосуда и других моделей более «мелкого масштаба». Используя данный метод моделирования сложных физиологических систем с помощью «Среды визуального моделирования функциональных систем организма» можно быстро и эффективно создавать математические модели в физиологии.

## **5 Объектно-ориентированная библиотека классов функциональных элементов сердечно-сосудистой системы**

### ***5.1 Объектно-ориентированный подход к созданию сложных компьютерных моделей в физиологии***

При моделировании таких сложных систем, как организм человека в целом, методами процедурного программирования (т. е. на таких языках, как Fortran, Pascal и т. п.), появляется большое число перекрестных ссылок между компонентами, что означает зависимость одного участка программного кода от другого. Как следствие, возникает множество проблем связывания отдельных подсистем друг с другом (т.е. функции из одного фрагмента обращаются к данным из другого фрагмента, предполагая, что те имеют какую-то определенную структуру), и если произошли какие-то изменения и усовершенствования в модели одной из подсистем организма, то это приведет к необходимости переписывания большого объема программного кода в моделях других подсистем. Причем это делается только для того, что бы обеспечить прежнее взаимодействие между подсистемами. Поэтому возникают большие проблемы со сборкой и отладкой программ.

Кроме того, большие трудности возникают при модернизации и изменениях в существующей модели. Для примера рассмотрим задачу моделирования разветвленной системы сосудов (сосудистого дерева). Если была написана программа расчета гемодинамики сосудистого дерева (расчета давлений и кровотоков), то для того, что бы кроме этого иметь возможность рассчитать еще и перенос какого-либо физиологически-активного вещества по этому дереву, необходимо полностью переписывать программу [4] (лучше написать новую).

К тому же, компьютерные модели, спроектированные при помощи процедурного подхода, лишены наглядности (представляют собой фактически программы решений различных уравнений и их систем), а эта наглядность очень важна для взаимного понимания при работе разработчиков модели со специалистами в области физиологии.

Для устранения описанных проблем предлагается объектно-ориентированный подход к созданию сложных компьютерных моделей в физиологии [1]. Данный подход уже давно широко применяется для создания программных продуктов в самых разных областях, однако до сих пор не применялся для формализации такой предметной области, как физиология.

В основе объектно-ориентированного подхода лежит идея объектов, представляющих собой на уровне программной реализации абсолютно независимые (автономные) фрагменты программного кода, объединяющие какие-либо данные с методами их обработки (инкапсуляция). Такая полная автономность позволяет полностью избежать первой из проблем, а именно зависимости одного фрагмента программного кода от других, которая и приводит к трудностям со сборкой и отладкой программ.

При таком подходе каждый смысловой элемент предметной области - в данном случае физиологии - например, сосуд, моделируется с помощью объекта, в котором объединяются как данные, которыми характеризуется сосуд (такими как поток и сопротивление), так и методы расчета (обработки) этих данных (например, аналог закона Ома) (см Рис. 4).

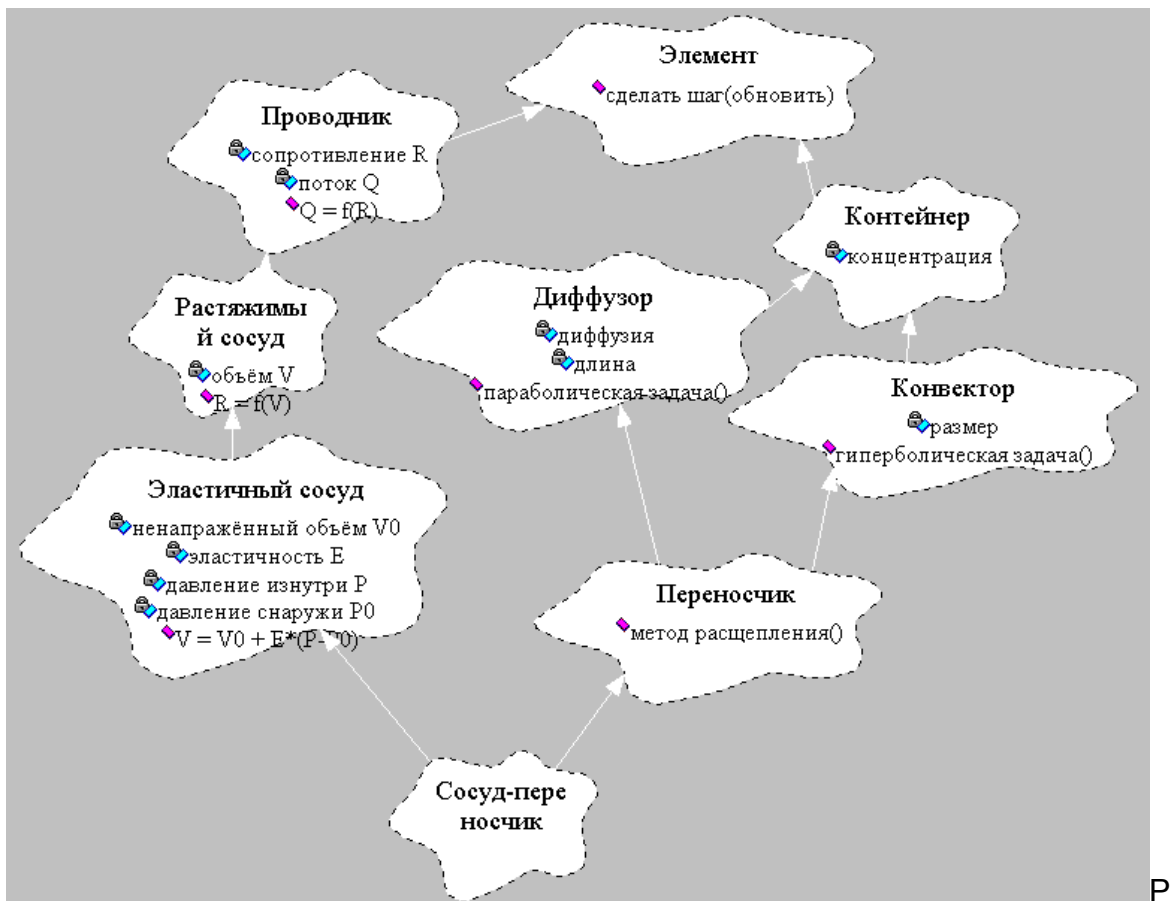


рис 4.

Если требуется учесть растяжимость сосудов, достаточно унаследовать свойства и методы класса сосуда, добавив к ним свойство объёма и функцию расчёта сопротивления трубки через этот объём. Далее класс растяжимой сосуда можно использовать в качестве родительского класса для эластичных, пластичных и других сосудов, различающихся моделью растяжения.

А теперь предположим, что требуется решать не только задачу расчёта потоков и давлений в системе сосудов, но еще и задачу переноса веществ в этой системе. Понятно, что методы расчёта конвективного, диффузионного и конвективно-диффузионного переноса должны существовать в специальных классах «переносчиках» независимо от элементов, связанных с моделированием механики сосудов. (см. Рис. 4) Чтобы использовать эти методы для задачи переноса в сосудах, достаточно создать элемент, наследующийся от сосуда и от переносчика одновременно, причём никакого нового программного кода создавать не требуется.

## **5.2 Описание библиотеки классов функциональных элементов сердечно-сосудистой системы.**

В данном разделе описывается библиотека классов функциональных элементов сердечно-сосудистой системы. Классы, входящие в данную библиотеку, делятся на четыре группы:

- Течение жидкости (газа);
- Обмен веществ;
- Диффузионный перенос веществ;
- Конвективный перенос веществ;

Это позволяет решать линейные и нелинейные уравнения Кирхгофа, а также уравнения переноса (конвективного, диффузионного, конвективно-диффузионного), что дает возможность решать задачи газодинамики в легких, гидродинамики кровообращения и распространения химических веществ в организме.

Большинство классов описываемой библиотеки предназначено для создания моделей различных процессов, происходящих в сердечно-сосудистой системе: гидродинамика кровообращения, перенос и обмен веществ и т. п. Однако, в силу идентичности моделей, те же классы могут использоваться, например, для моделирования процессов, происходящих в дыхательной системе.

Созданные с помощью этой библиотеки математические модели могут описывать кровообращение на системном уровне, не претендуя на подробное отображение «тонкой структуры» кровообращения (в отдельном сосуде и т. п.).

В основе описываемых ниже классов лежит понятие обобщенного сосудистого участка [16]. Обобщенный сосудистый участок понимается как отдельный сосуд либо система сосудов некоторого участка ткани или органа, которые по условию задачи выделены в качестве относительно самостоятельной области и характеризуются усредненными показателями кровотока, давления, сопротивления, эластичности и др.

Ниже перечисляются все классы данной библиотеки, сгруппированные по решаемым задачам. В скобках после названия элемента приведено

название класса из библиотеки на языке Java (иерархия классов элементов приведена на рис 5.). Диаграмма классов с наследованием и агрегацией представлена на Рис 6. Указаны только поля классов, без методов.

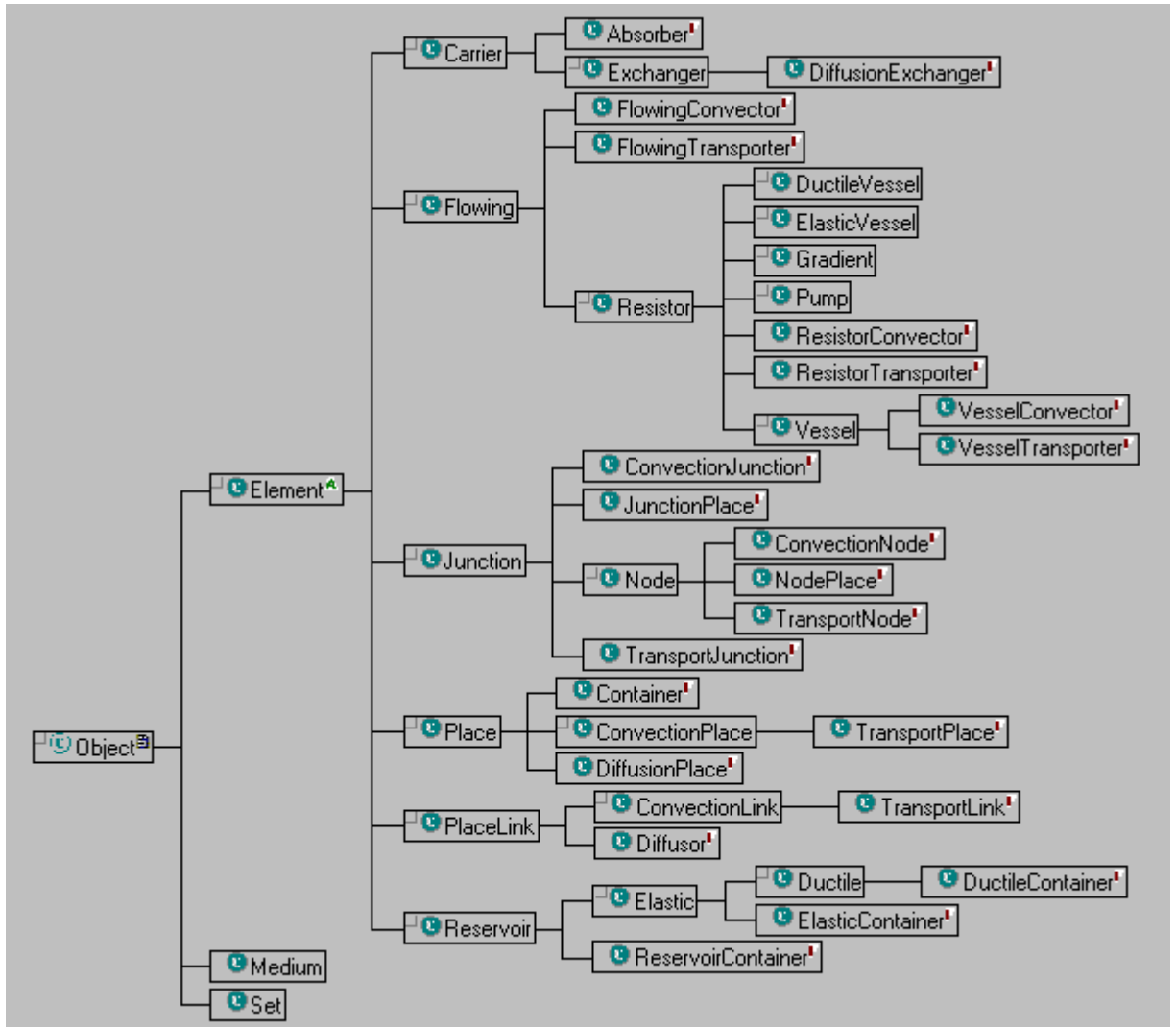


Рис.5 Структура библиотеки классов физиологических элементов на языке Java.

Для удобства для каждого из элементов приведены все его (как его собственные, так и унаследованные от суперклассов или благодаря агрегации) переменные (поля) и функции (методы). Традиционный подход к разработке библиотеки привел бы к многочисленным повторениям в коде (как это легко можно заметить из приведенного ниже списка элементов). Это еще

раз доказывает преимущество объектно-ориентированного подхода при разработке библиотеки.

(Примечание.

1. Если после названия параметра в скобках стоит *регулируется*, то этот параметр может участвовать в регуляторных связях (связях между параметрами) или же быть входным параметром, если же стоит (в различных классах Сердца) *обязательно регулируется*, то параметр должен участвовать в регуляторных связях.
2. Если обозначение параметра стоит в квадратных скобках [], то это означает массив параметров.
3. В описании библиотеки отсутствуют приведенные на рисунках классы конвективно-диффузионного переноса. С их описанием можно познакомиться в [6])

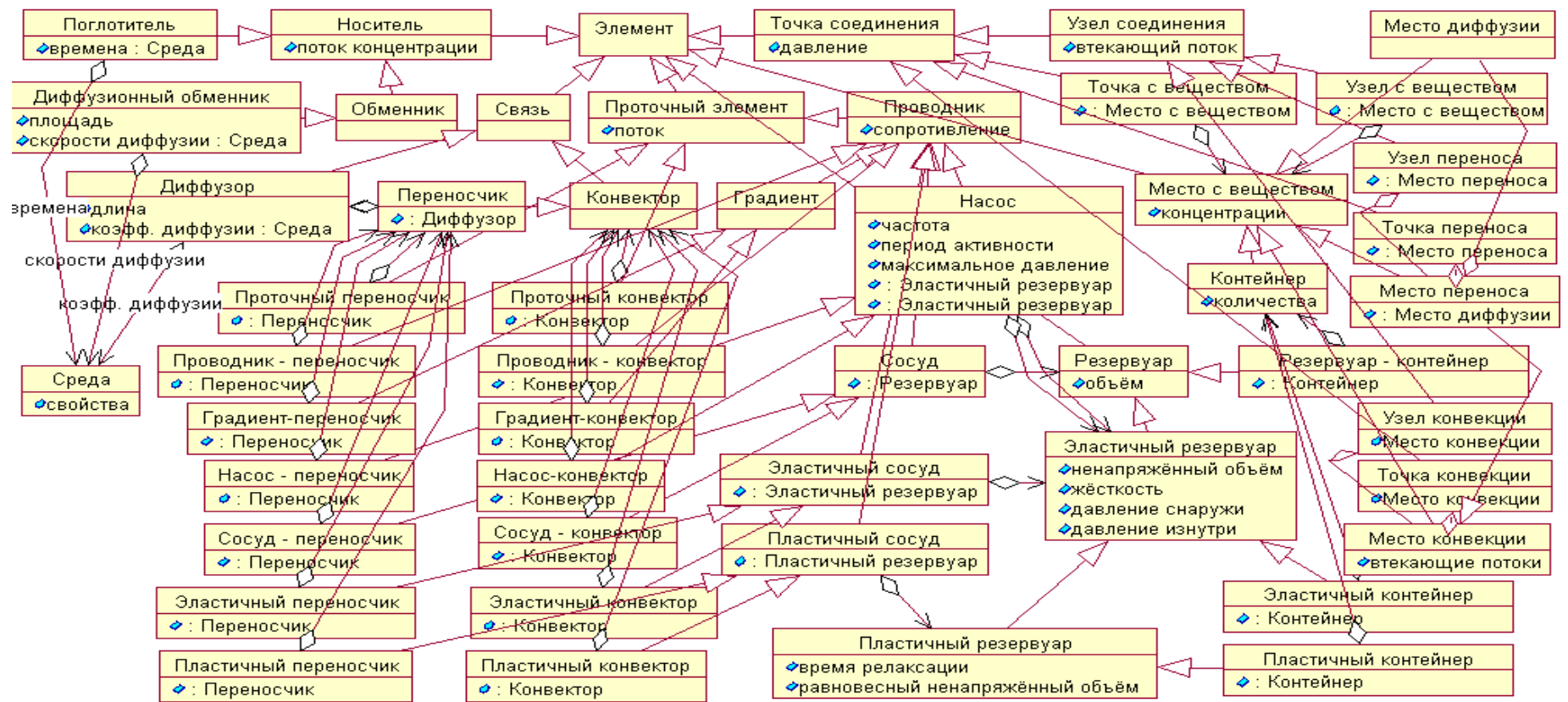


Рис 6. Диаграмма классов.

### 1. Течение жидкости и газа.

**1. Резервуар (Reservoir)** – предназначен для хранения некоторого объема жидкости или газа. Его связи (которых может быть сколько угодно) с **Проточными элементами** и с **Узлами соединения** позволяют ему обмениваться жидкостью (газом) с другими резервуарами и системами проводников соответственно.

*Поле  $V$  – объем.*

*Метод  $V=V+\left(\sum_i Q_i - \sum_j q_j\right) \cdot dt$ , где  $Q_i$  и  $q_j$  - потоки через входные и*

*выходные элементы соответственно.*

### 2. Эластичный резервуар (Elastic)

*Поля  $V$  – объем,*

*$V_0$  - ненапряженный объем (*регулируется*),*

*$E$  – жесткость (*регулируется*),*

*$P_0$  – давление снаружи (*регулируется*),*

*$P$  – давление изнутри.*

*Методы  $V=V+\left(\sum_i Q_i - \sum_j q_j\right) \cdot dt$ , где  $Q_i$  и  $q_j$  - потоки через входные и*

*выходные элементы соответственно.*

$$P = P_0 + E \cdot (V - V_0).$$

**3. Точка соединения (Junction).** Необходима для формализованного описания системы ветвящихся проводников (см. ниже). Данный элемент содержит всего один параметр - давление в точке, и сам никаких специфических действий над ним не производит. Точку соединения можно связывать с любым количеством элементов типа **Проводник** и одного из его подтипов (причём направление связи не имеет значения), однако самый обычный способ её использования - постановка на выходе элемента типа **Насос**, который в таком случае будет рассчитывать давление в точке.

*Поле  $P$  – давление (*регулируется*).*

**4. Узел соединения (Node).** Расширяет возможности **Точки соединения** и отличается от неё тем, что самостоятельно рассчитывает своё

давление на основе информации о сопротивлениях проводников, с которыми он связан, и о давлениях на их противоположных концах. При этом имеется возможность притока или оттока из узла некоторого количества жидкости (газа), для чего узел соединения содержит второй параметр - втекающий поток.

*Поля*  $P$  – давление,

$q$  – втекающий (внешний) поток (*регулируется*).

*Метод*  $P = R \cdot \left( \sum_i P_i / R_i + q - \sum_j Q_j \right)$ , где  $R_i$  – сопротивления связанных

проводников,  $P_i$  – давления на их противоположных концах,  $Q_j$  – потоки

через связанные активные проводники (насосы, градиенты),  $R = \left( \sum_i 1/R_i \right)^{-1}$  -

входное сопротивление.

**5. Проточный элемент (Flowing).** Является простейшим типом, характеризуясь только одним параметром - потоком, то есть объёмом жидкости (газа), протекающей через него в единицу времени. Связывается данный элемент каждым концом с не более чем одним элементом типа **Резервуар** или одного из его подтипов. Если связь направлена к резервуару, то его объём будет увеличиваться, если от резервуара - уменьшаться (естественно, при условии положительности потока).

*Поле*  $Q$  – поток (*регулируется*).

## **6. Проводник (Resistor)**

*Поля*  $Q$  – поток,

$R$  – сопротивление (*регулируется*).

*Метод*  $Q = (P_2 - P_1) / R$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно.

**7. Градиент (Gradient).** Обеспечивает заданную разность давлений между **Узлом соединения** (который должен быть связан с входом данного элемента) и **Точкой соединения** (с выходом). Он имеет все свойства **Проводника** (поток и сопротивление), однако расчёт ведёт по той же

формуле в обратном направлении, определяя из неё не поток, а давление на выходе (в **Точке соединения**).

*Поля Q – поток (регулируется),*

*R – сопротивление (регулируется).*

*Метод  $P_2 = P_1 + Q \cdot R$  (Расчёт давления  $P_2$  в выходном элементе через давление  $P_1$  во входном).*

**8. Сердце** (Насос, Pump) формально расширяет возможности **Проводника**, но представляет собой гораздо более сложный элемент, предназначенный для описания таких циклически меняющих свои параметры объектов, как желудочек сердца. Основной функцией насоса является расчёт потока и давления на выходе через известное давление на входе и выходное сопротивление. Насос имеет два однотипных набора параметров, относящиеся к двум фазам цикла его работы - активной (сокращение) и пассивной (расслабление). Эти наборы в точности соответствуют двум **Эластичным резервуарам** (см. ниже), однако некоторых параметров несколько отличается. В частности, ненапряжённый объём в активной фазе соответствует минимальному объёму, объём в пассивной фазе - ударному объёму, а ненапряжённый объём в пассивной фазе - просто объёму. Помимо этих параметров и рассмотренных выше *потока* и *сопротивления*, Насос характеризуется *частотой* (обратным периодом цикла), *длительностью активной фазы* и *максимальным давлением*, которое он может развивать при заданном объёме. В связи с тем, что некоторые закономерности работы такого насоса не могут быть описаны точно, но известны по экспериментальным данным, способ расчёта трёх параметров (максимального давления, минимального объёма и объёма в пассивной фазе) не фиксируется, и подлежит обязательному определению на уровне модели. Как и **Проводник**, **Насос** может быть связан с **Точкой соединения** или **Узлом соединения**, однако связь от **Насоса** к **Узлу** лишена смысла, поскольку узел рассчитывает своё давление, а в данном случае это должен делать Насос.

(Вывод и подробный анализ данной модели сердца можно найти в [5])

*Поля Q - поток,*

$R$  - выходное сопротивление (*может регулируется*),

$f$  - частота (*регулируется*),

$T_s$  - длительность активной фазы (*регулируется*),

$P_m$  - максимальное давление (*обязательно регулируется*),

$V$  - ударный объём,

$V_d$  - объём в пассивной фазе (*обязательно регулируется*),

$E$  - выходная жёсткость (*регулируется*),

$P_d$  - давление в пассивной фазе,

$P_{od}$  - давление снаружи в пассивной фазе (*регулируется*),

$V_s$  - объём в активной фазе,

$V_m$  - минимальный объём в активной фазе (*обязательно регулируется*),

$K$  - жёсткость сердца (насоса) (*регулируется*),

$P_s$  - давление в активной фазе,

$P_{os}$  - давление снаружи в активной фазе (*регулируется*).

(Примечание: активная стадия – систола, пассивная – диастола.)

*Методы*  $R = (P_2 - P_1)/Q$ , где  $P_1$  и  $P_2$  - давления во входном и в выходном элементе соответственно.

$$\Delta V = (P_m - P_d) \cdot \left[ K_1 + \frac{P_m - P_d}{V_d - V_m} \right]^{-1} \quad (K_1 = K / (1 - \exp(-E \cdot T_d / R))),$$

$$T_d = 1/f - T_s)$$

$$Q = f \cdot \Delta V$$

$$P_2 = P_d + K R \Delta V / E T_d$$

$$P_s = P_d + K_1 \cdot \Delta V$$

$$V_s = V_d - \Delta V$$

**9. Сосуд (Vessel).** Является сочетанием параметров и функциональных возможностей, присущих двум типам элементов - **Проводнику** и **Резервуару**. Новым свойством является то, что *сопротивление* сосуда как проводника зависит от его диаметра, а значит, и от *объёма* сосуда как резервуара. С другой стороны, *поток* через сосуд определяется не только стационарным законом пропорциональности между

ним и разностью давлений; при расчёте учитывается и нестационарная поправка к потоку, равная скорости изменения объёма и формально относящаяся не к сосуду, а к связанному с его выходным концом **Узлу соединения** (параметр *втекающий поток*).

Поля  $Q$  – поток,

$R^0$  – сопротивление (норма) (*регулируется*),

$R$  – сопротивление,

$V^0$  – объём (норма) (*регулируется*),

$V$  – объём (*регулируется*)

*Методы*  $R = R^0(V^0/V)^2$ , где  $R^0$  и  $V^0$  – нормальные значения.

$Q = (P_2 - P_1)/R$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно.

**10. Эластичный сосуд (ElasticVessel).** Сочетает свойства **Проводника** со свойствами **Эластичного резервуара**. Зависимость *сопротивления* и *потока* от *объёма* и скорости его изменения точно такая же, как в обычном **Сосуде**, а зависимость *объёма* от разности давлений и свойств стенки – такая же, как в Эластичном резервуаре. Отличие в том, что давления на концах сосуда как проводника напрямую определяют давление внутри сосуда как резервуара (оно принимается равным их полусумме).

Поля  $Q$  – поток,

$R^0$  – сопротивление (норма) (*регулируется*),

$R$  – сопротивление,

$V^0$  – объём (норма) (*регулируется*),

$V$  – объём,

$V_0$  – ненапряжённый объём (*регулируется*),

$E$  – жесткость (*регулируется*),

$P_0$  – давление снаружи (*регулируется*),

$P$  – давление изнутри.

*Методы*  $P = (P_1 + P_2)/2$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно,

$V = V_0 + (P - P_0)/E$ ,

$R = R^0 (V^0/V)^2$ , где  $R^0$  и  $V^0$  – нормальные значения.

$Q = (P_2 - P_1)/R$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно.

## 2. Обмен веществ

**11. Контейнер (Container).** Является основой для всех элементов, способных содержать в себе набор некоторых субстанций (которые далее для определённости называются веществами). Каждое вещество в контейнере характеризуется как своей *концентрацией*, так и общим *количеством* (массой) в контейнере. При этом в данном элементе рассчитывается только количество, а концентрация вычисляется лишь в его подтипах. Любой контейнер может быть связан с несколькими элементами типов **Носитель** и **Место конвекции** (а также с элементами их подтипов), которые обеспечивают приток или отток веществ от контейнера.

*Поля*  $[c_k]$  – концентрации веществ (*регулируется*),

$[m_k]$  – количества веществ.

*Метод*  $m_k = m_k + \left( \sum_i F_{i,k} - \sum_j f_{j,k} \right) \cdot dt$ , где  $F_{i,k}$  и  $f_{j,k}$  – потоки веществ

через входные и выходные элементы соответственно.

**12. Резервуар-контейнер (Reservoir - Container)** благодаря наличию у него *объёма* (как у любого резервуара) имеет возможность рассчитать не только *количество*, но и *концентрации* веществ в контейнере.

*Поля*  $[c_k]$  – концентрации веществ,

$[m_k]$  – количества веществ.

$V$  – объём.

*Методы*  $c_k = m_k/V$ ,

$m_k = m_k + \left( \sum_i F_{i,k} - \sum_j f_{j,k} \right) \cdot dt$ , где  $F_{i,k}$  и  $f_{j,k}$  – потоки веществ

через входные и выходные элементы соответственно.

$V = V + \left( \sum_i Q_i - \sum_j q_j \right) \cdot dt$ , где  $Q_i$  и  $q_j$  – потоки через входные и

выходные элементы соответственно.

**13. Эластичный контейнер (ElasticContainer)** является **Эластичным резервуаром**, способным определять *концентрации и массы* веществ.

Поля  $V$  – объем,

$V_0$  - ненапряженный объем (*регулируется*),

$E$  – жесткость (*регулируется*),

$P_0$  – давление снаружи (*регулируется*),

$P$  – давление изнутри.

$[c_k]$  – концентрации веществ,

$[m_k]$  - количества веществ.

*Методы*

(Расчёт объёма как в эластичном резервуаре.)

$$V = V_0 + \left( \sum_i Q_i - \sum_j q_j \right) \cdot dt, \text{ где } Q_i \text{ и } q_j - \text{ потоки через входные и выходные}$$

элементы соответственно.

$$P = P_0 + E \cdot (V - V_0),$$

(Расчёт количеств веществ как в контейнере.)

$$m_k = m_k + \left( \sum_i F_{i,k} - \sum_j f_{j,k} \right) \cdot dt, \text{ где } F_{i,k} \text{ и } f_{j,k} - \text{ потоки веществ через}$$

входные и выходные элементы соответственно.

(Расчёт концентраций веществ как в резервуаре-контейнере.)

$$c_k = m_k / V.$$

**14. Носитель (Carrier)** обеспечивает отток веществ от **Контейнера** или его подтипов, поэтому он может иметь не более чем одну связь с контейнером на входе и ни одной связи на выходе. Скорость оттока формализуется с помощью параметра *поток вещества* каждого из «текущих» по данному элементу веществ, однако никаких расчётов сам Носитель не производит. Если поток вещества отрицателен, Носитель представляет собой источник вещества.

*Поле*  $[f_k]$  – поток вещества (*регулируется*).

**15. Поглотитель (Absorber)** реализует простейшую модель расчёта оттока веществ из контейнера за счёт их поглощения, распада или какого-

либо иного взаимодействия со средой. В этой модели поток концентрации пропорционален самой концентрации с коэффициентом, равным обратному характерному времени поглощения. *Время поглощения* для каждого вещества является свойством содержащейся в Поглотителе среды, а не его собственным свойством, поэтому в число параметров оно не входит.

*Поле*  $[f_k]$  – поток вещества

*Метод*  $f_k = m_k/t_k$ , где  $m_k$  - количества веществ во входном элементе,  $t_k$  - времена поглощения веществ средой.

**16. Обменник**, (Exchanger) в отличие от **Носителя**, может не только служить в качестве источника или стока вещества, но и обеспечивать обмен веществ между двумя **Контейнерами**. Для этого данный тип элемента и все его подтипы может иметь не только входную, но и выходную связь. Расчёт *потока вещества* в **Обменнике** не проводится.

*Поле*  $[f_k]$  – поток вещества (*регулируется*).

**17. Диффузионный обменник** (DiffusionExchanger) содержит в себе модель, согласно которой *поток вещества* между двумя **Контейнерами** пропорционален разности концентраций в них и эффективной *площади* обменника. В таком случае коэффициент этой пропорциональности (*скорость диффузии*, то есть коэффициент диффузии, отнесённый к эффективной толщине «стенки» обменника) можно считать свойством содержащейся в данном элементе среды, а не его собственным параметром.

*Поля*  $[f_k]$  – поток вещества,

$S$  - эффективная площадь диффузии (*регулируется*).

*Метод*  $f_k = (M_k - m_k) \cdot S \cdot d_k$ , где  $M_k$  и  $m_k$  - количества веществ во входном и выходном элементах соответственно,  $d_k$  - скорости диффузии веществ в среде.

### **3. Перенос веществ**

**18. Место с веществом** (Place) является основой для всех элементов, способных к переносу веществ. Каждое вещество в контейнере характеризуется только своей *концентрацией*, в то время как подобный данному элемент **Контейнер** снабжён также количествами веществ, которые

зачастую не могут быть определены. Само по себе Место с веществом не может быть связано ни с одним элементом, что позволяет вводить произвольные законы изменения концентрации веществ в нём на уровне модели.

*Поле* [с<sub>к</sub>] – концентрации веществ (*регулируется*).

**19. Точка с веществом** (JunctionPlace) описывается *давлением* и *концентрациями* содержащихся в ней веществ, однако в данный элемент не заложены модели их расчёта. Данный элемент может иметь связи с теми же элементами, что и обычная **Точка**.

*Поля* [с<sub>к</sub>] – концентрации веществ (*регулируется*),

*P* – давление (*регулируется*).

**20 Узел с веществом** (NodePlace) обладает всеми свойствами **Узла соединения** по отношению к расчёту своего *давления* и к связыванию с другими элементами, но также характеризуется и *концентрациями* веществ.

*Поля* [с<sub>к</sub>] – концентрации веществ (*регулируется*),

*P* – давление

*q* – втекающий (внешний) поток (*регулируется*).

*Метод*  $P = R \cdot \left( \sum_i P_i / R_i + q - \sum_j Q_j \right)$ , где  $R_i$  – сопротивления связанных

проводников,  $P_i$  – давления на их противоположных концах,  $Q_j$  – потоки

через связанные активные проводники (насосы, градиенты),  $R = \left( \sum_i 1/R_i \right)^{-1}$  –

входное сопротивление.

**21. Место диффузии** (DiffusionPlace) содержит в себе методы расчёта изменений *концентраций* содержащихся в нём веществ с помощью модели диффузионного переноса. Для реализации этой модели необходимо знать коэффициенты диффузии и эффективное расстояние между разными Местами диффузии, поэтому данный элемент может быть связан только с **Диффузорами**. Направление связей не имеет значения, количество их может быть любым. При расчётах предполагается, что соседние (связанные Диффузором) Места диффузии имеют приблизительно одинаковый объём, поэтому в случае невыполнения этого условия необходимо использовать

**Диффузионные обменники** вместо Диффузоров и **Контейнеры** вместо Мест диффузии.

*Поле*  $[c_k]$  – концентрации веществ.

*Метод*  $c_k = c_k + dt \cdot \sum_i (c_{i,k} - c_k) D_{i,k} / L_i^2$ , где  $L_i$  - эффективная длина связанных

элементов (диффузоров),  $c_{i,k}$  - концентрации на их противоположных концах,  $D_{i,k}$  - коэффициенты диффузии веществ в средах диффузоров.

**22. Диффузор** (Diffuser) , в отличие от сходного с ним **Диффузионного обменника**, потока вещества в себе не несёт и никаких расчётов производить не умеет. Он предназначен для связывания с **Местами диффузии**, которые самостоятельно рассчитывают концентрации своих веществ через хранящиеся в диффузорах *коэффициенты диффузии* (свойства среды, заполняющей Диффузоры) и эффективные *длины* самих Диффузоров. Под эффективной длиной Диффузора понимается среднее геометрическое между его реальной диффузионной длиной и отношением объёма к площади сечения.

*Поле*  $l$  – эффективная длина (*регулируется*).

*4. Конвективный перенос веществ*

**23. Место конвекции**, (ConvectionPlase) как и **Место с веществом**, содержит в себе набор *концентраций* веществ. Помимо них, данный элемент характеризуется *втекающими потоками концентрации*, с помощью которых он (и, главное, его подтипы) может связываться с **Контейнерами** и вместе с ними рассчитывать конвективный перенос веществ. Возможны также связи с элементами типа **Конвектор**, но не с элементами его подтипов, поскольку Место конвекции не имеет *давление* в числе своих параметров.

*Поля*  $[f_k]$  – втекающий поток вещества (*регулируется*),

$[c_k]$  – концентрации веществ

*Метод*  $c_k = c_k + \left( \sum_i (C_{i,k} - c_k) Q_i / V_i - \sum_j (c_{j,k} - c_k) q_j / v_j + f_k / V \right) \cdot dt$ , где  $Q_i$  и

$q_j$  - потоки через входные и выходные элементы соответственно,  $C_{i,k}$  и  $c_{j,k}$  -

концентрации веществ на их противоположных концах,  $V_i$  и  $v_j$  - их эффективные размеры,  $V = \frac{1}{2} \left( \sum_i V_i + \sum_j v_j \right)$ .

**24. Точка конвекции** (ConvectionJunction) отличается от **Места конвекции** наличием давления, благодаря которому все связанные с ней элементы подтипов **Конвектора** (обладающие сопротивлением) получают возможность рассчитывать свой поток. Этот поток, наряду с размерами конвекторов, **Точка конвекции** использует при расчёте своих концентраций.

*Поля*  $[f_k]$  – втекающий поток вещества (*регулируется*),

$[c_k]$  – концентрации веществ,

$P$  – давление (*регулируется*).

*Метод*  $c_k = c_k + \left( \sum_i (C_{i,k} - c_k) Q_i / V_i - \sum_j (c_{j,k} - c_k) q_j / v_j + f_k / V \right) \cdot dt$ , где  $Q_i$  и

$q_j$  - потоки через входные и выходные элементы соответственно,  $C_{i,k}$  и  $c_{j,k}$  - концентрации веществ на их противоположных концах,  $V_i$  и  $v_j$  - их

эффективные размеры,  $V = \frac{1}{2} \left( \sum_i V_i + \sum_j v_j \right)$ .

**25. Узел конвекции** (ConvectionNode) не только содержит *давление* среди своих параметров, но и рассчитывает его методами **Узла соединения**. Это накладывает запрет на соединение его входа с **Градиентом-конвектором** и **Насосом-конвектором**, которые сами рассчитывают своё выходное давление. Как и **Место конвекции** или **Точка конвекции**, данный элемент может быть связан с **Контейнерами**, однако благодаря наличию у него *втекающего потока* эти связи получают второе назначение, аналогичное назначению связей между **Узлом соединения** и **Резервуарами**.

*Поля*  $[f_k]$  – втекающий поток вещества (*регулируется*),

$[c_k]$  – концентрации веществ,

$P$  – давление,

$q$  – втекающий (внешний) поток (*регулируется*).

*Методы*

$$P = R \cdot \left( \sum_i P_i / R_i + q - \sum_j Q_j \right), \quad \text{где } R_i - \text{сопротивления связанных}$$

проводников,  $P_i$  – давления на их противоположных концах,  $Q_j$  – потоки через связанные активные проводники (насосы, градиенты),  $R = \left( \sum_i 1/R_i \right)^{-1}$  – входное сопротивление.

$$c_k = c_k + \left( \sum_i (C_{i,k} - c_k) Q_i / V_i - \sum_j (c_{j,k} - c_k) q_j / v_j + f_k / V \right) \cdot dt, \quad \text{где } Q_i \text{ и } q_j -$$

потоки через входные и выходные элементы соответственно,  $C_{i,k}$  и  $c_{j,k}$  – концентрации веществ на их противоположных концах,  $V_i$  и  $v_j$  – их

$$\text{эффективные объемы, } V = \frac{1}{2} \left( \sum_i V_i + \sum_j v_j \right).$$

(Примечание: в общем случае эффективные объемы равны обычным.)

**26. Конвектор (Convector)** представляет собой **Проточный элемент**, обладающий помимо *потока* ещё и *размером*. Используя эти параметры, он способен участвовать в расчёте конвективного переноса веществ, проводимом **Местами конвекции**, которые должны быть связаны с каждым из его концов.

*Поля Q* – поток (*регулируется*).

$v$  – эффективный объем (*регулируется*).

**27 Проводник-конвектор**, (ResistorConvector) являясь одновременно **Проводником** и **Конвектором**, может участвовать в расчёте не только переноса веществ, но и протекающего через него *потока*. Поэтому он должен иметь связи не с **Местом конвекции**, а с **Точкой конвекции** или с **Узлом конвекции**.

*Поля Q* – поток,

$R$  – сопротивление (*регулируется*).

$v$  – эффективный объем (*регулируется*).

*Метод*  $Q = (P_2 - P_1) / R$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно.

**28. Градиент-конвектор** (GradientConvector) участвует в связях с **Точкой конвекции** и **Узлом конвекции** аналогично тому, как обычный **Градиент** связывается с **Точкой соединения** и **Узлом соединения**. По отношению к переносу веществ его роль совпадает с ролью любого конвектора.

*Поля Q – поток (регулируется).*

*R – сопротивление (регулируется).*

*v – эффективный объем (регулируется).*

*Метод  $P_2 = P_1 + Q \cdot R$  (Расчёт давления  $P_2$  в выходном элементе через давление  $P_1$  во входном).*

**29. Насос – Конвектор** (Сердце, PumpConvector) соединяет в себе функциональность **Насоса** со способностью **Конвектора** по переносу веществ. Он может иметь точно такие же связи, как и **Градиент**.

*Поля Q - поток,*

*R - выходное сопротивление (может регулируется),*

*f - частота (регулируется),*

*$T_s$  - длительность активной фазы (регулируется),*

*$P_m$  - максимальное давление (обязательно регулируется),*

*V - ударный объём,*

*$V_d$  - объём в пассивной фазе (обязательно регулируется),*

*E - выходная жёсткость (регулируется),*

*$P_d$  - давление в пассивной фазе,*

*$P_{od}$  - давление снаружи в пассивной фазе (регулируется),*

*$V_s$  - объём в активной фазе,*

*$V_m$  - минимальный объём в активной фазе (обязательно регулируется),*

*K - жёсткость сердца (насоса) (регулируется),*

*$P_s$  - давление в активной фазе,*

*$P_{os}$  - давление снаружи в активной фазе (регулируется).*

*v – эффективный объем (регулируется).*

*(Примечание: активная стадия – систола, пассивная – диастола.)*

*Методы*  $R = (P_2 - P_1)/Q$ , где  $P_1$  и  $P_2$  - давления во входном и в выходном элементе соответственно.

$$\Delta V = (P_m - P_d) \cdot \left[ K_1 + \frac{P_m - P_d}{V_d - V_m} \right]^{-1} \quad (K_1 = K / (1 - \exp(-E \cdot T_d / R))),$$

$$T_d = 1/f - T_s)$$

$$Q = f \cdot \Delta V$$

$$P_2 = P_d + K R \Delta V / E T_d$$

$$P_s = P_d + K_1 \cdot \Delta V$$

$$V_s = V_d - \Delta V$$

**30. Сосуд-конвектор**, (VesselConvector) обладая *объёмом* (как и любой **Сосуд**), имеет возможность изменять в зависимости от него свои свойства по отношению к переносу веществ.

*Поля* Q – поток,

$R^0$  – сопротивление (норма) (*регулируется*),

R – сопротивление,

$V^0$  – объем (норма) (*регулируется*),

V – объем (*регулируется*),

v – эффективный объем (*регулируется*).

*Методы*  $R = R^0 (V^0 / V)^2$ , где  $R^0$  и  $V^0$  – нормальные значения.

$Q = (P_2 - P_1)/R$ , где  $P_1$  и  $P_2$  - давления во входном и в выходном элементе соответственно.

**31. Эластичный конвектор** (ElasticConvector), по сути, является **Сосудом-конвектором** с упругой стенкой, и подобно ему участвует в связях с **Точками конвекции** и (или) **Узлами конвекции**.

*Поля* Q – поток,

$R^0$  – сопротивление (норма) (*регулируется*),

R – сопротивление,

$V^0$  – объем (норма) (*регулируется*),

V – объем,

$V_0$  - ненапряженный объем (*регулируется*),

$E$  – жесткость (*регулируется*),

$P_0$  – давление снаружи (*регулируется*),

$P$  – давление изнутри.

$v$  – эффективный размер (*регулируется*).

*Методы*  $P = (P_1 + P_2)/2$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно,

$$V = V_0 + (P - P_0)/E,$$

$$R = R^0 (V^0/V)^2, \text{ где } R^0 \text{ и } V^0 - \text{ нормальные значения.}$$

$Q = (P_2 - P_1)/R$ , где  $P_1$  и  $P_2$  – давления во входном и в выходном элементе соответственно.

### **5.3 Использование библиотеки классов**

Описанные выше классы из библиотеки можно использовать при разработке собственных моделей. При этом проявляется еще одно математическое достоинство излагаемого подхода, оно заключается в его инвариантности относительно типа решаемой задачи. Является ли задача статической, динамической или вообще обратной задачей идентификации параметров модели – во всех этих случаях используются одни и те же классы элементов. Различие между этими задачами заключается только в алгоритме обновления состояния элементов. В случае динамической задачи обновление происходит до конца времени расчета, в случае статической – до получения заданной точности. В случае обратной задачи получение заданной точности происходит много раз в цикле, итерации которого соответствуют различным значениям искомых параметров модели, а изменение этих параметров происходит согласно какому-либо методу решения нелинейных уравнений.

```

main
{
for (i=0 ; i<n ; i++)
{//инициализация экземпляров классов физиологических элементов e[i]
(i=1..n) с помощью конструкторов
//связывание элементов между собой:
e[i].addNeighbor(e[j])
//Статическая задача
//задание точности расчетов  $\varepsilon_0$ 
do {
 $\varepsilon = 0$ ;
for (i=0 ; i<n ; i++)
{
e[i].update();
 $\varepsilon = \varepsilon + e[i].getDifference()$  ;
}
 $\varepsilon = \varepsilon / n$  ;
}
while (  $\varepsilon > \varepsilon_0$  );
//Динамическая задача
//задание периода времени T и шага по времени  $\Delta t$ 
while (t<T)
{
for (i=0 ; i<n ; i++)
{
e[i].update(t,  $\Delta t$ );
}
}
e[j].getValue();
}

```

Из приведенного листинга видно, что сама программа не зависит от используемых классов и не меняется при замене одних классов на другие (например, получаемые при наследовании).

И вообще не нужно никаких структур данных (типа массивов и т. п.) для хранения входных и выходных данных, все данные хранятся в самих объектах [3], т. е. объекты сами хранят данные о себе и сами же их обрабатывают. Потом также по циклу они опрашиваются и опять таки сами выдают свои значения (для вывода на график и т. п.). Поэтому ничего не меняется при изменении структуры модели или при добавлении новых элементов. Это преимущество является следствием фундаментального свойства объектов, а именно объединения данных с методами их обработки.

Необходимо также отметить, что при этом выходные данные модели соответствуют структуре предметной области (значения параметра «давление» хранятся в объекте «сосуд», а не в каком-то массиве). Это можно сказать и о самом процессе создания модели. В этом проявляется еще одно преимущество объектно-ориентированного подхода к моделированию: он резко повышает наглядность самого процесса создания модели в физиологии. Действительно, гораздо легче, наглядней и понятней при создании модели оперировать понятиями сосуда, резервуара и т. д., чем понятиями систем уравнений. При объектно-ориентированном подходе сами системы уравнений как таковые исчезают.

Большинство классов физиологических элементов из рассматриваемой библиотеки лишены всех регуляторных функций и вообще нелинейности. Это специально сделано потому, что в отличие от линейных физических моделей, которые меняются относительно редко, количество и конкретный вид регуляторных функций и вообще нелинейностей сильно зависит от конкретной модели и должно определяться ее разработчиком. Это достигается путем наследования новых классов от уже существующих в данной библиотеке. Таким образом, построен своеобразный универсальный каркас, легко приспособляемый для решения большинства конкретных задач моделирования, как отдельных систем, так и всего организма.

Кроме того еще одно большое преимущество объектно-ориентированного подхода заключается в возможности, используя одну и ту же библиотеку классов, проводить вычисления как на локальном компьютере, так и (при необходимости больших вычислительных ресурсов) в сети.

Данная библиотека реализована на языке Java, что позволяет использовать ее на компьютерах с различными архитектурами и операционными системами. Поэтому появляется уникальная возможность, не меняя ничего в самой библиотеке, использовать ее при написании программ как на персональном компьютере (в том случае, когда не требуется высокий уровень детализации процессов и/или расчеты на больших временах), так и на значительно более мощных компьютерах, например Alpha (в противоположном случае).

Данная библиотека классов может быть использована несколькими способами:

- Во-первых, она может использоваться непосредственно при разработке компьютерных моделей организма, путем использования в программе либо непосредственно самих классов из библиотеки, либо унаследованных от них. Несколько таких (тестовых) моделей были созданы с использованием данной библиотеки классов. Эти модели являются скорее просто тестовыми примерами, которые были созданы только для того, что бы продемонстрировать работоспособность предложенного подхода. Они практически не несут в себе никакого физиологического содержания, и поэтому не будут здесь рассматриваться.
- Во-вторых, данная библиотека классов может быть использована как основа для создания различных сред визуальной разработки моделей, в которой пользователь мог бы в визуальном режиме соединять эти объекты, вводить конкретные численные значения для каждого объекта, и таким образом визуально создавать модель. Подобная «Среда визуального моделирования» будет описана в следующем разделе.

#### **5.4 Преимущества использования объектно-ориентированной библиотеки элементов**

В заключение суммируем те преимущества, которые дает использование объектно-ориентированной библиотеки классов (реализованной на Java) для создания сложных компьютерных моделей в физиологии:

- Наглядность (объектами легче оперировать, чем уравнениями);
- Легкость отладки, изменения и модернизации;
- Экономия программного кода (за счет наследования и агрегации);
- Возможность как локальных, так и сетевых (параллельных) вычислений;
- Переносимость (PC, Alpha, ...).

## 6 Среда визуального моделирования функциональных систем организма

### 6.1 Описание программного продукта

В предыдущем разделе была описана библиотека классов физиологических элементов и было упомянуто о том, что ее можно использовать для создания так называемых сред визуальной разработки математических моделей в физиологии. Такой программный комплекс «Среда визуального моделирования функциональных систем организма» (СВМ) [2], позволяющий пользователю (эксперту в области физиологии) в визуальном режиме, не прибегая к программированию численных методов и интерфейса, создавать математические модели физиологических систем организма, и описывается в данном разделе.

Визуальное моделирование физиологической системы (организма или его подсистем, т. к. СВМ предназначена для создания моделей на уровне подсистем организма) для удобства пользователя проходит в три этапа (См. Табл. 1 и Рис. 7), условно называемыми этапами создания Схемы, Модели, и Сценария.

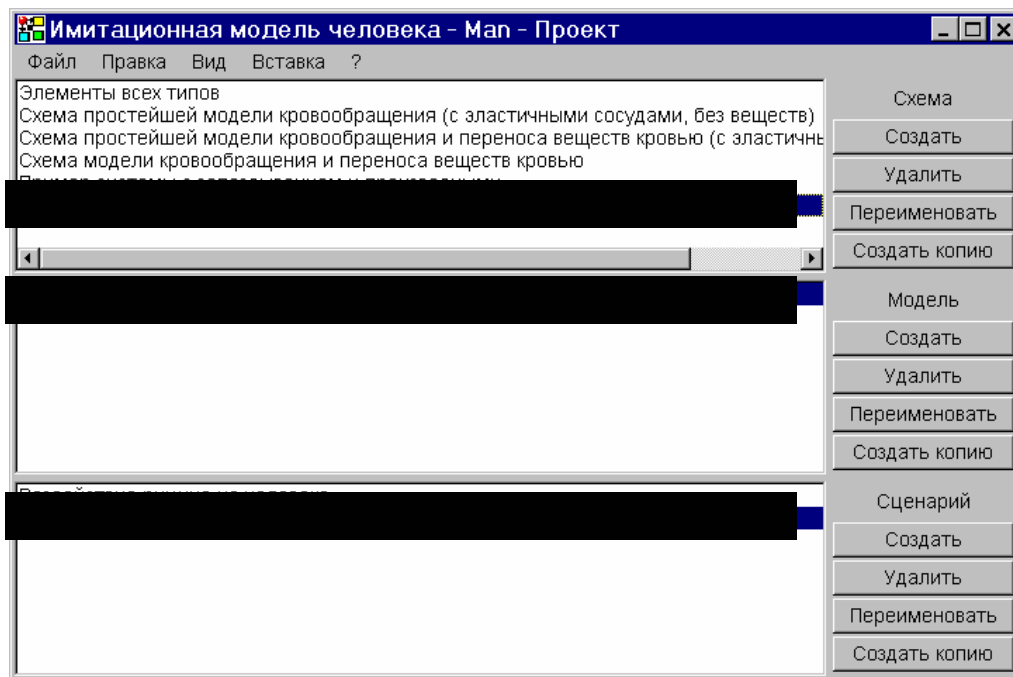


Рис. 7. Главное окно программы

<b>Уровень</b>	<b>Вид вводимых данных</b>
<b>Схема</b>	<u>Качественные:</u> <ul style="list-style-type: none"> <li>• физические;</li> <li>• структурно-геометрические</li> </ul>
<b>Модель</b>	<u>Количественные:</u> <ul style="list-style-type: none"> <li>• значения переменных в норме;</li> <li>• нелинейности и регуляции</li> </ul>
<b>Сценарий</b>	<u>Качественные:</u> <ul style="list-style-type: none"> <li>• выбор входных и выходных переменных</li> </ul> <u>Количественные:</u> <ul style="list-style-type: none"> <li>• динамика входных переменных</li> </ul>

Таблица 1.

На уровне Схемы пользователь сначала разбивает моделируемую систему на подсистемы, визуально определяя иерархию вложенности их друг в друга.

Затем пользователю предлагается список различных физиологических элементов, из которых он может строить свою модель. (См. Табл. 2) Эти элементы представляют собой те самые классы, которые входят в библиотеку классов, описанных в предыдущем разделе.

<b>№</b>	<b>Элемент</b>	<b>Рис.</b>	<b>№</b>	<b>Элемент</b>	<b>Рис.</b>
1	Резервуар		24	Место диффузии	
2	Эластичный резервуар		25	6.1.1.1.1 Диффузор	
3	Пластичный резервуар		26	Место конвекции	
4	Точка соединения		27	Точка конвекции	
5	Узел соединения		28	Узел конвекции	
6	Проточный элемент		29	Конвектор	
7	Проводник		30	Проводник-конвектор	
8	Градиент		31	Градиент-конвектор	
9	Насос		32	Насос-конвектор	
10	Сосуд		33	Сосуд-конвектор	
11	Эластичный сосуд		34	Эластичный конвектор	
12	Пластичный сосуд		35	Пластичный конвектор	
13	Контейнер		36	Место переноса	
14	Резервуар-контейнер		37	Точка переноса	
15	Эластичный контейнер		38	Узел переноса	
16	Пластичный контейнер		39	Переносчик	
17	Носитель		40	Проводник-переносчик	
18	Поглотитель		41	Градиент-переносчик	
19	Обменник		42	Насос-переносчик	
20	Диффузионный обменник		43	Сосуд-переносчик	
21	Место с веществом		44	Эластичный переносчик	
22	Точка с веществом		45	Пластичный переносчик	
23	Узел с веществом				

Таблица 2

Пользователь выбирает необходимые ему элементы, помещает их в соответствующие подсистемы и связывает их между собой с помощью мышки (см. рис 8.). Структура модели создается как из элементов конструктора.

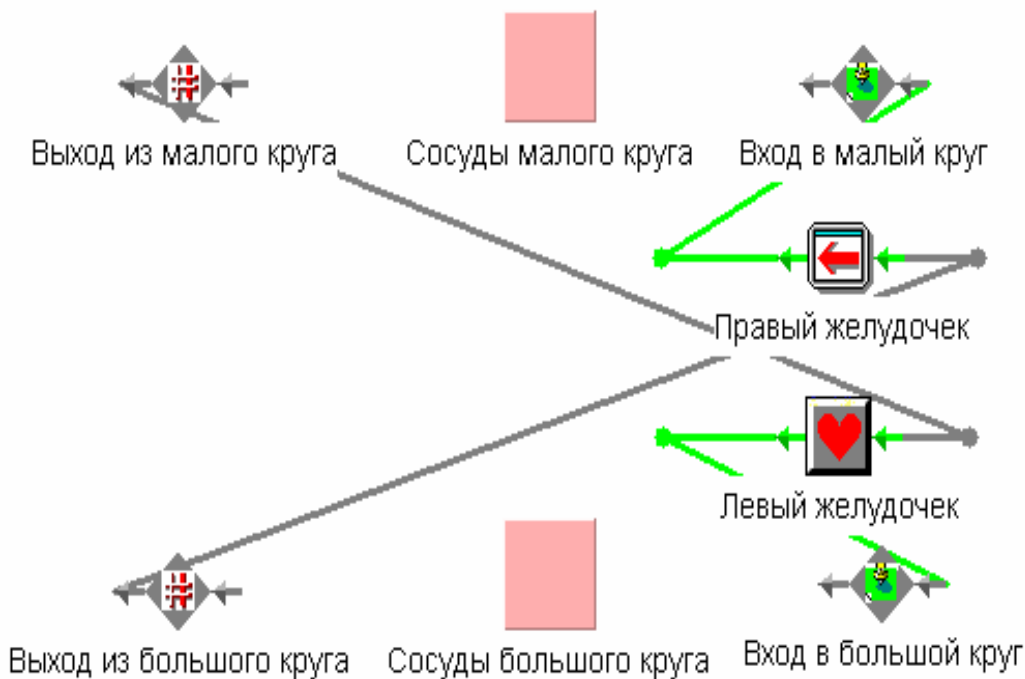


Рис 8. Схема модели.

Таким образом, на уровне Схемы вводятся качественные данные о системе, а именно физические (тип элемента) и структурно-геометрические (иерархия вложенности и связи между элементами).

На следующем этапе моделирования, условно названном Моделью, вводятся уже количественные данные о моделируемой системе.

Эти данные делятся на две группы:

Во-первых, это ввод значений всех переменных модели в норме.

Во-вторых, это ввод различных регуляторных функций (определение контуров регуляции) и вообще всех нелинейностей, присущих системе. Дело в том, что в предлагаемых пользователю элементах реализованы лишь линейные физические законы, а все нелинейности и регуляции должен вводить сам пользователь. Это сделано намеренно, т. к. в отличие от «физической» составляющей модели, «физиологическая» часть сильно зависит от задачи и от предположений эксперта (пользователя), поэтому заранее это предусмотреть не возможно. (Такое разделение основывается на предположении о том, что физиологическую систему можно разделить на управляемую и управляющую части.)

Ввод всех нелинейностей и регуляторных связей полностью автоматизирован.

Зависимость между двумя переменными (параметрами) строится визуально с помощью стрелочек. (См. Рис. 9).

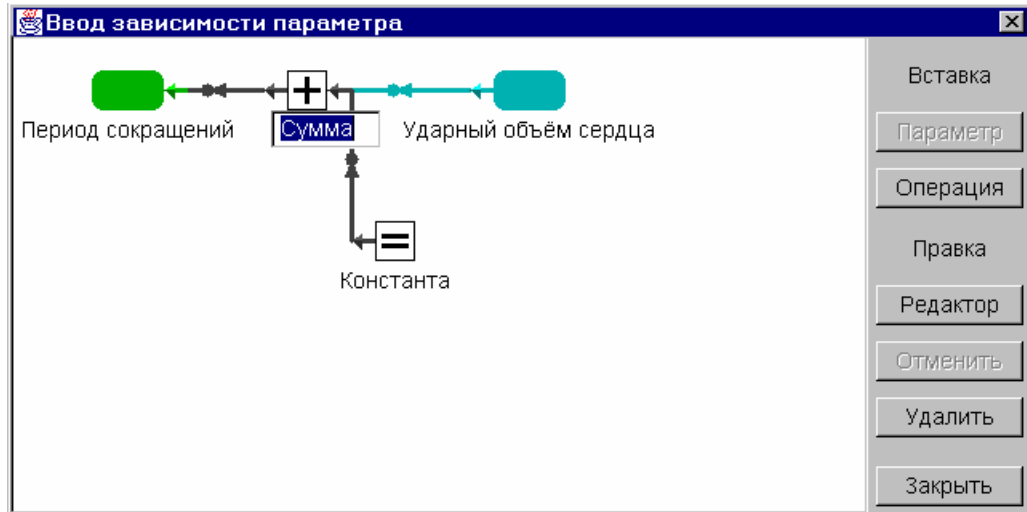


Рис. 9. Редактор зависимости

Пользователю предлагается набор функциональных зависимостей (типов функций) между переменными, (См Рис. 10) причем возможно использование функции с запаздыванием, а также возможность получения зависимости не от переменной, а от ее производной любого порядка.

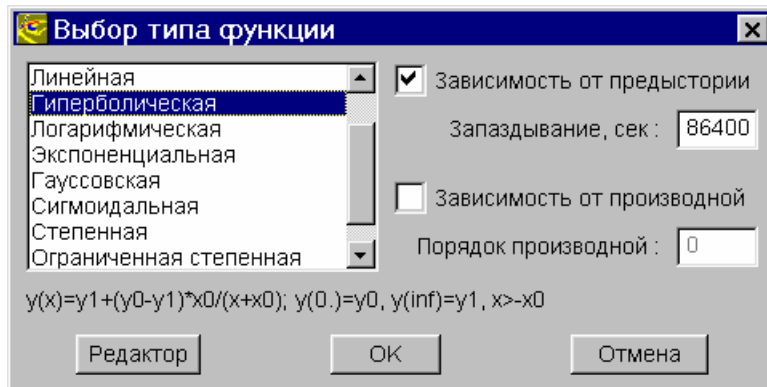


Рис. 10. Определение типа функции

Конкретный вид функции того или иного типа (Рис. 11.) может быть получен визуально (либо конечно, можно вводить напрямую значения коэффициентов функции). Кроме того (кнопка «Настроить») предусмотрена возможность автоматически настроить график функции так, что бы он проходил через определенную точку (крестик)-значения переменных в норме.

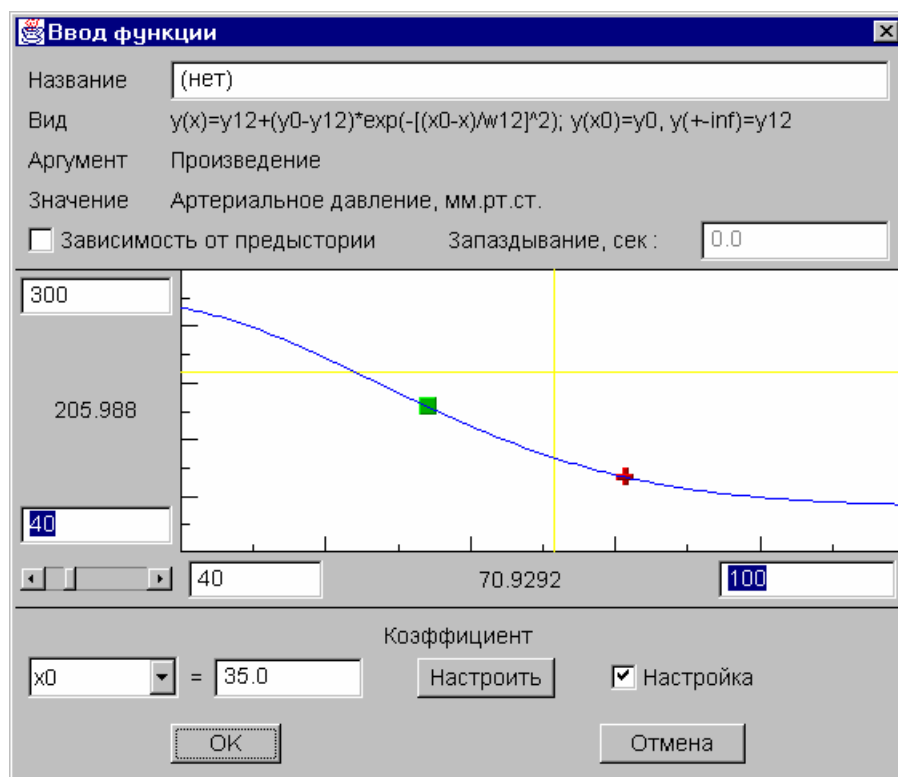


Рис. 11. Редактор функции с неопределёнными коэффициентами.

В случае если пользователю не подходит ни один из предлагаемых типов функций, существует возможность задать табличную функцию (См Рис. 12.) где можно мышкой ввести экспериментально полученные точки, они автоматически будут интерполированы и внесены в модель.

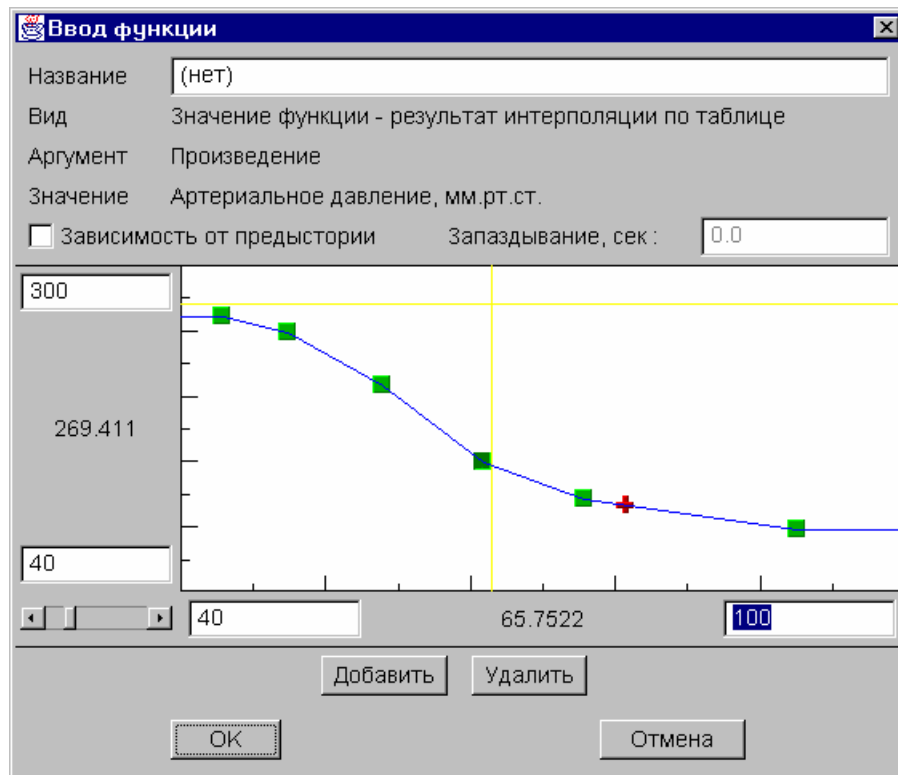


Рис. 12. Редактор табличной функции.

Список функций, поддерживаемых данной версией СВМ:

1. **Тождественная:** Значение функции равно её аргументу
2. **Табличная:** Значение функции - результат интерполяции по таблице
3. **Линейная:**  $y(x)=y_0+y_1*(x-x_0)$ ;  $y(x_0)=y_0$
4. **Гиперболическая:**  $y(x)=y_1+(y_0-y_1)*x_0/(x+x_0)$ ;  $y(0)=y_0$ ,  $y(\infty)=y_1$ ,  $x > -x_0$
5. **Логарифмическая:**  $y(x)=y_0+\ln((x_0-y_1)/(x-y_1))$ ;  $y(x_0)=y_0$ ,  $y(y_1)=+\infty$
6. **Экспоненциальная:**  $y(x)=y_1+(y_0-y_1)*\exp(x_0-x)$ ;  $y(x_0)=y_0$ ,  $y(+\infty)=y_1$
7. **Гауссова:**  $y(x)=y_{12}+(y_0-y_{12})*\exp(-[(x_0-x)/w_{12}]^2)$ ;  $y(x_0)=y_0$ ,  $y(\pm\infty)=y_{12}$
8. **Сигмоидальная:**  $y(x)=y_0+(y_{12}-y_0)*2/\pi*\text{atan}(|x-x_0|/w_{12})$ ;  $y(x_0)=y_0$ ,  $y(\pm\infty)=y_{12}$
9. **Степенная:**  $y(x)=y_0+y_{12}*(x-x_0)^{w_{12}}$ ;  $y(x_0)=y_0$
10. **Ограниченная степенная:**  $y(x)=y_0+y_{12}*(x-x_0)^{w_{12}}$ ;  $y(x_0)=y_0$ ,  $y(x > x_1)=y(x_1)$ ,  $y(x < x_2)=y(x_2)$

Список операций над связями:

1. **Разность:** Вычитание аргумента справа из аргумента сверху
2. **Отношение:** Деление аргумента сверху на аргумент справа
3. **Сумма:** Суммирование аргументов
4. **Произведение:** Произведение аргументов
5. **Среднее арифметическое:** Нахождение среднего арифметического аргументов
6. **Среднее геометрическое:** Нахождение среднего геометрического аргументов
7. **Минимум:** Выбор среди аргументов минимального
8. **Максимум:** Выбор среди аргументов максимального
9. **Константа:** Присвоение результату некоторого значения.

Наконец, на последнем этапе создания модели, условно названным этапом Сценария производится, во-первых, разделение данных на входные и выходные (произвольные комбинации), а во-вторых, задание динамики входных данных (также визуальное, с помощью мышки, См. Рис. 13.).



Рис. 13. Графический редактор входного параметра.

После этого, с помощью так называемого диалога моделирования (См Рис 14.), модель можно запустить на выполнение, и будут рисоваться графики зависимостей выходных данных от времени.

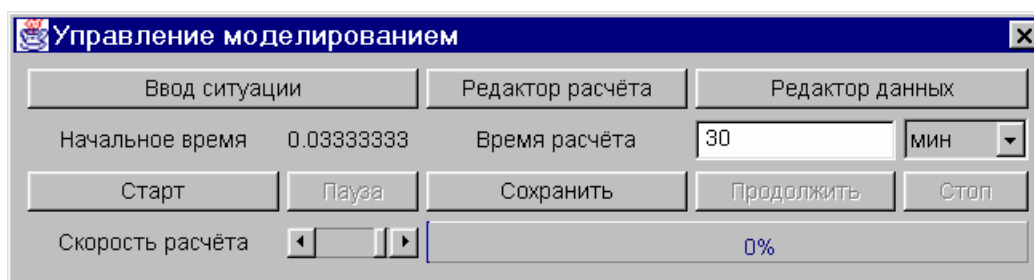


Рис. 14. Диалог управления моделированием.

Можно делать паузы в расчетах, возможны также изменение в режиме реального времени, т. е. во время моделирования, значений переменных, а также можно регулировать скорость расчета - ползунком - что бы можно было следить за графиками выходных данных и управлять моделированием.

Для облегчения работы по созданию модели предусмотрены возможности создания копий Схем, Моделей, и Сценариев (вообще говоря они образуют ориентированный граф (См Рис. 15.), для того что бы можно было бы в любой момент вернуться на предыдущий шаг создания модели, если внесенные в нее изменения были неудачными, а также для того, что бы можно было бы одновременно исследовать два варианта модели, отличающиеся наличием разных контуров регуляции и т. п.

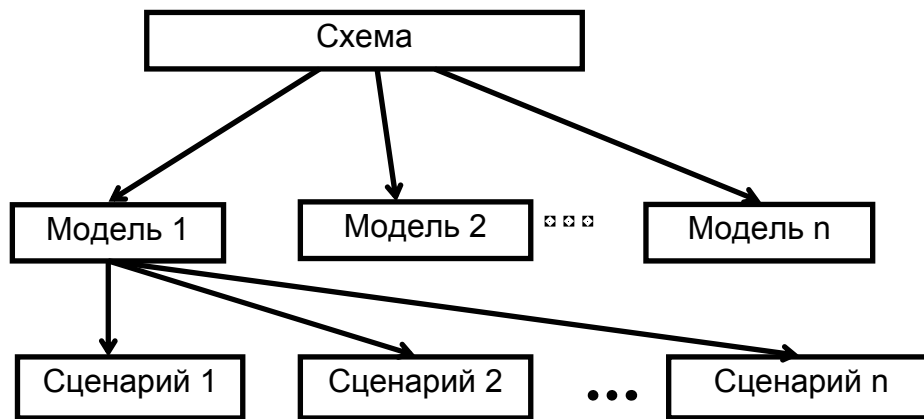


Рис 15.

Собственно поэтому и не был реализован механизм экспорта готовых моделей из среды, т. к. их хранение в среде позволяет создавать расширяемые модели, т.е. если появился новый факт, например новый контур регуляции, то его очень легко внести в Модель, и ничего больше вообще менять не нужно.

Кроме того, на всем протяжении создания модели ведется так называемый Журнал Моделирования, в который пользователь может заносить свои комментарии относительно своих действий, а также куда заносятся различные сообщения, генерируемые самой средой визуального моделирования в процессе создания и запуска модели.

СВМ предназначена для моделирования процессов, проходящих во всем организме (или в одной или нескольких физиологических системах организма) а также моделирование внешних воздействий (физической, химической или биологической природы), т.е. для визуального создания и развития моделей типа Гайтона [18, 21] или Амосова [19].

СВМ позволяет визуально (без написания уравнений) создавать модели, которые математически описываются линейными и нелинейными уравнениями Кирхгофа, а также уравнениями переноса (конвективного, диффузионного, конвективно-диффузионного), что позволяет решать задачи газодинамики в легких, гидродинамики кровообращения и распространения химических веществ в организме.

С помощью «Среды визуального моделирования» принципиально возможно моделировать следующие процессы:

- 1) Функционирование в норме
  - подсистемы организма
  - организма в целом
- 2) Патологии и воздействия (возможны комбинированные варианты)
  - Физические
    - патологии строения внутренних органов (пороки сердца и т. п.)
    - механические - раны (кровопотеря);
    - другие;
  - Химические
    - эндогенные изменения (патологии) обмена веществ;
    - экзогенные влияния:
      - изменения состава воздуха;
      - изменение состава пищи;
      - ввод веществ в кровь;
  - Биологические =Физические + Химические (бактерии, вирусы - диффузия и т. п.)

В данном подходе к созданию Среды визуальной разработки сделана попытка объединить физический (математический) и эмпирический (физиологический) подходы.

Наряду с многочисленными достоинствами, описанными выше, у данного подхода имеется один существенный недостаток.

Дело в том, что, в отличие от аналогичных по назначению электротехнических прикладных пакетов, где модели используемых элементов (сопротивление, индуктивность и т.п.) известны и не нуждаются в уточнениях и исправлениях, модели (здесь речь идет только о «физических» моделях, без учета регуляции) любых физиологических систем (органов, сосудов и т.п.) постоянно уточняются и изменяются. Более того, существуют множество «физических» моделей для одной физиологической системы или органа (например, модели сердца как генератора непрерывного либо

пульсирующего потока), в зависимости от поставленной задачи, и число таких моделей постоянно растет.

Поэтому возникает необходимость постоянного добавления все новых и новых базовых объектов, отражающих новые физические модели физиологических систем. Однако, в связи с тем, что при разработке как программного продукта в целом, так и исходных элементов был применен объектно-ориентированный подход, введение новых элементов легко осуществляется при помощи наследования от уже существующих элементов, и в принципе, при знании организации программы может быть выполнено даже и без участия разработчиков программного продукта (что было бы очень сложно, если бы объектно-ориентированный подход не применялся).

## 6.2 Пример использования «Среды визуального моделирования»

В данном разделе приводится пример использования «Среды визуального моделирования функциональных систем организма»: визуальное построение простейшей модели сердечно-сосудистой системы.

При создании этой модели ставилась задача не столько получить какие-либо новые физиологические данные, сколько продемонстрировать преимущества визуального моделирования в физиологии и положить основу целому семейству математических моделей в физиологии кровообращения.

Данная модель сердечно-сосудистой системы содержит в себе модель сердца, которая подробно описана в [5], а также в приложении к данной работе.

Ниже будет описана последовательность действий при создании этой модели.

### 1. Уровень Схемы.

Данная модель состоит из трех подсистем: собственно сердечно-сосудистой системы (см. рис. 15) и входящих в нее подсистем сосудов большого (рис. 16) и малого (рис. 17) кругов.

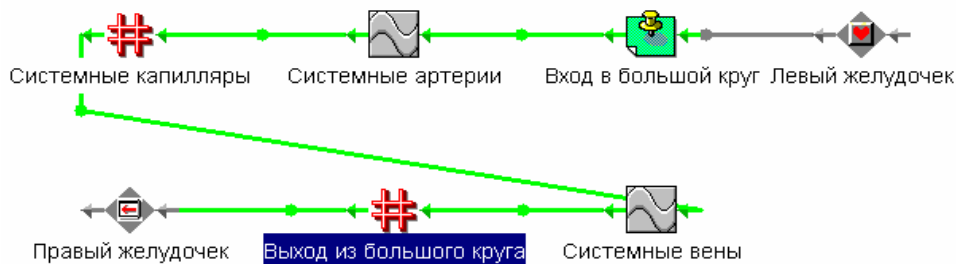


Рис. 16.

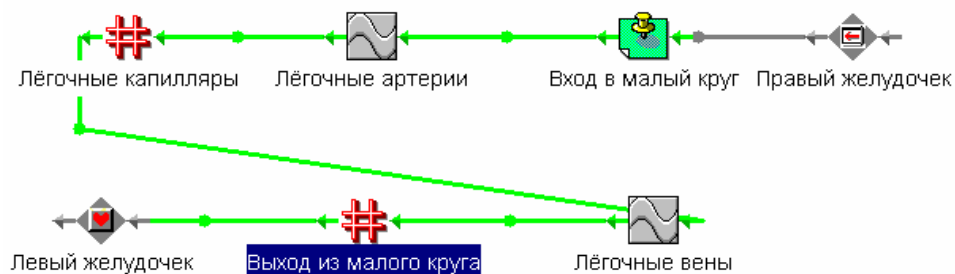


Рис. 17.

Модель состоит из двенадцати элементов следующих типов:

**Название и типы элементов**

1. Правый желудочек – Градиент;
2. Вход в малый круг – Точка соединения;
3. Лёгочные артерии – Эластичный сосуд;
4. Лёгочные капилляры – Узел соединения;
5. Лёгочные вены – Эластичный сосуд;
6. Выход из малого круга – Узел соединения;
7. Левый желудочек - Сердце (насос);
8. Вход в большой круг – Точка соединения;
9. Системные артерии – Эластичный сосуд;
10. Системные капилляры – Узел соединения;
11. Системные вены – Эластичный сосуд;
12. Выход из большого круга – Узел соединения.

Кроме того, в модели содержатся четыре выделенных (не содержащихся ни в каких элементах) параметра:

**Выделенные параметры**

1. Симпатический тонус
2. Парасимпатический тонус
3. Верхнее артериальное давление
4. Нижнее артериальное давление

**2. Уровень модели.**

На уровне модели с помощью диалога ввода нормальных значений (рис. 18) вводятся значения всех параметров в норме:

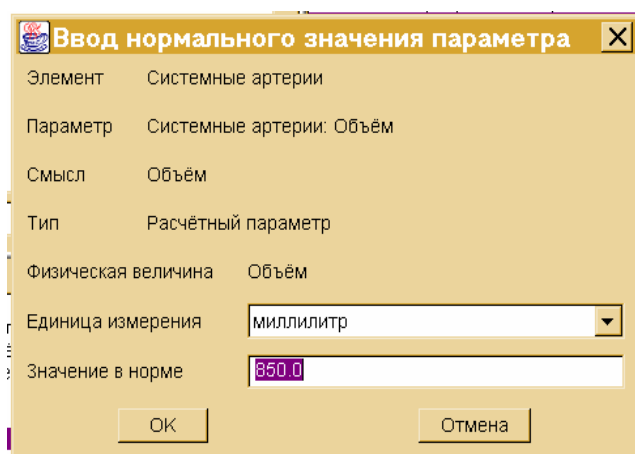


Рис. 18.

Единицы измерения:

Время – с

Частота – 1/мин

Давление – мм. рт. ст.

Объем – мл

Поток – мл/с

Сопротивление – мм. рт. ст./(мл/с)

Значения параметров в норме

1. Симпатический тонус 50
2. Парасимпатический тонус 50
3. Правый желудочек: Сопротивление -0.14
4. Правый желудочек: Поток 70
5. Вход в малый круг: Давление 13
6. Лёгочные артерии: Давление снаружи 0
7. Лёгочные артерии: Давление изнутри 10
8. Лёгочные артерии: Жёсткость 0.23
9. Лёгочные артерии: Ненапряжённый объём 133
10. Лёгочные артерии: Объём 180
11. Лёгочные артерии: Сопротивление 0.085
12. Лёгочные артерии: Поток 70
13. Лёгочные капилляры: Втекающий поток 0
14. Лёгочные капилляры: Давление 7

15. Лёгочные вены: Давление снаружи	0
16. Лёгочные вены: Давление изнутри	6.5
17. Лёгочные вены: Жёсткость	0.1
18. Лёгочные вены: Ненапряжённый объём	205
19. Лёгочные вены: Объём	260
20. Лёгочные вены: Сопротивление	0.015
21. Лёгочные вены: Поток	70
22. Выход из малого круга: Втекающий поток	0
23. Выход из малого круга: Давление	6
24. Левый желудочек: Давление снаружи в систоле	0
25. Левый желудочек: Систолическое давление	120
26. Левый желудочек: Жёсткость насоса	0.54
27. Левый желудочек: Минимальный систолический объём	40
28. Левый желудочек: Систолический объём	70
29. Левый желудочек: Давление снаружи в диастоле	0
30. Левый желудочек: Диастолическое давление	6
31. Левый желудочек: Выходная жёсткость	0.8
32. Левый желудочек: Диастолический объём	130
33. Левый желудочек: Ударный объём	60
34. Левый желудочек: Максимальное давление	280
35. Левый желудочек: Длительность систолы	0.28
36. Левый желудочек: Частота	70
37. Левый желудочек: Выходное сопротивление	1.4
38. Левый желудочек: Поток	70
39. Вход в большой круг: Давление	100
40. Системные артерии: Давление снаружи	0
41. Системные артерии: Давление изнутри	70
42. Системные артерии: Жёсткость	0.8
43. Системные артерии: Ненапряжённый объём	770
44. Системные артерии: Объём	850
45. Системные артерии: Сопротивление	1.2
46. Системные артерии: Поток	70

- 47. Системные капилляры: Втекающий поток 0
- 48. Системные капилляры: Давление 20
- 49. Системные вены: Давление снаружи 0
- 50. Системные вены: Давление изнутри 11.5
- 51. Системные вены: Жёсткость 0.012
- 52. Системные вены: Ненапряжённый объём 2400
- 53. Системные вены: Объём 3350
- 54. Системные вены: Сопротивление 0.25
- 55. Системные вены: Поток 70
- 56. Выход из большого круга: Давление 3
- 57. Выход из большого круга: Втекающий поток 0
- 58. Верхнее артериальное давление 120
- 59. Нижнее артериальное давление 80

Вводится период обновления данных – 3 с., период сохранения данных – 50 с. – они принимаются одинаковыми для всех систем модели (см. Рис. 19).

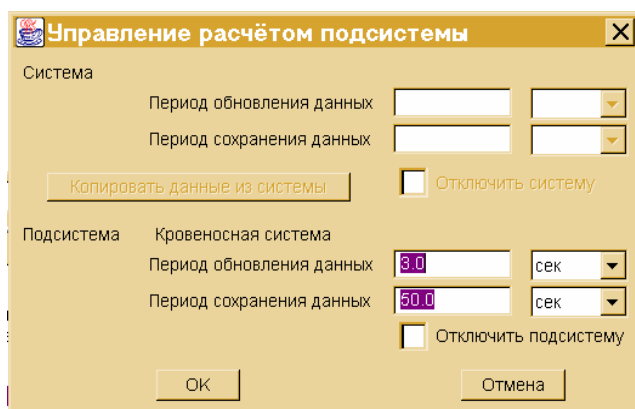


Рис. 20.

Модель содержит пять зависимостей (т. е. пять ее параметров нелинейно связаны с другими, см. раздел данной работы, посвященный декомпозиции сложных физиологических систем):

1. Левый желудочек: минимальный систолический объем,
2. Левый желудочек: диастолический объем,
3. Левый желудочек: максимальное давление,
4. Левый желудочек: частота,

## 5. Нижнее артериальное давление.

Зависимости 1 – 3 представляют собой «вырожденный» случай (см. рис. 21), они сводятся к одной функции. Эти функции представлены на рис 22, 23, и 24.

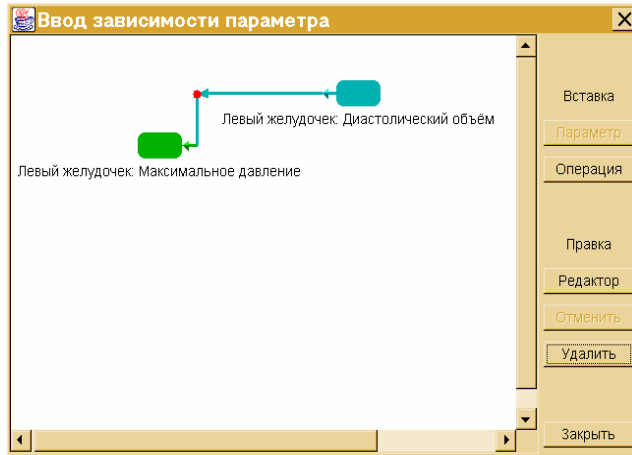


Рис. 21.



Рис. 22.

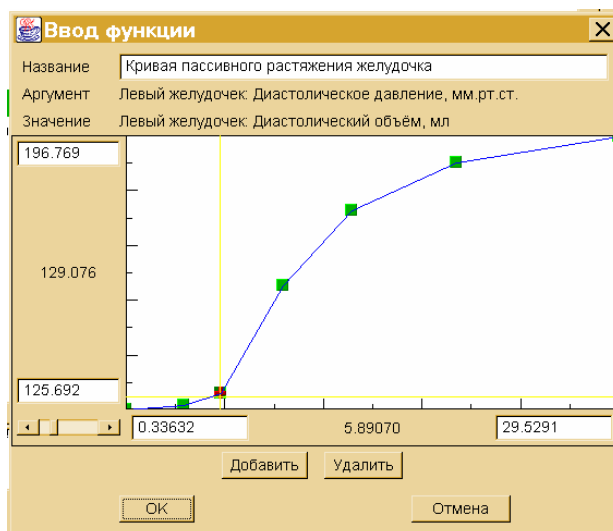


Рис. 23

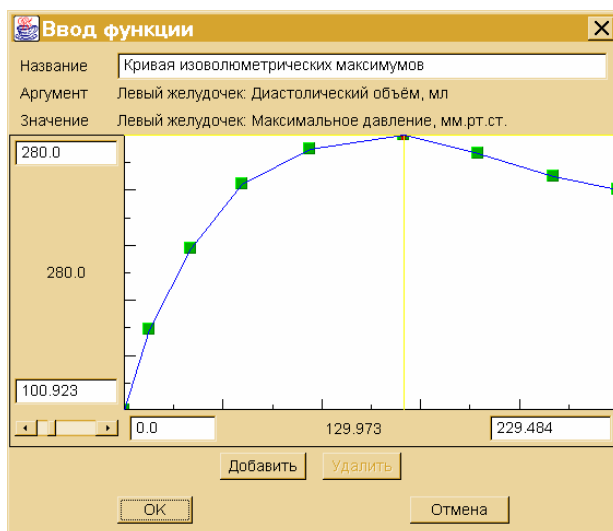


Рис. 24

Зависимость 4 представлена на рис. 25.

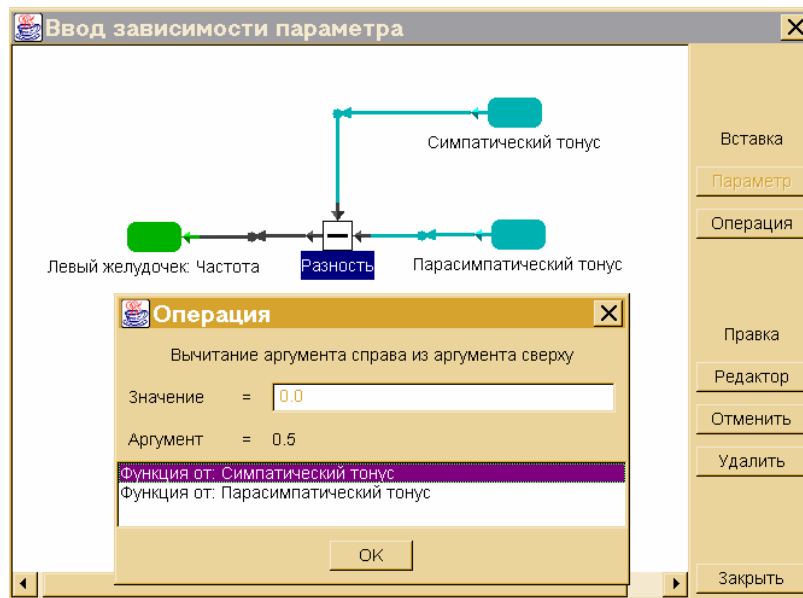


Рис. 25.

Она состоит из трех функций и одной операции. Две функции являются тождественными, а третья представлена на рис. 26.

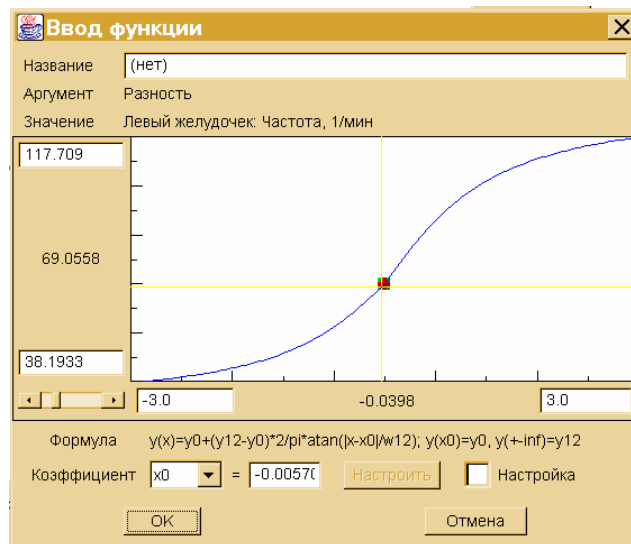


Рис. 26.

Зависимость 5 представлена на рис. 27.

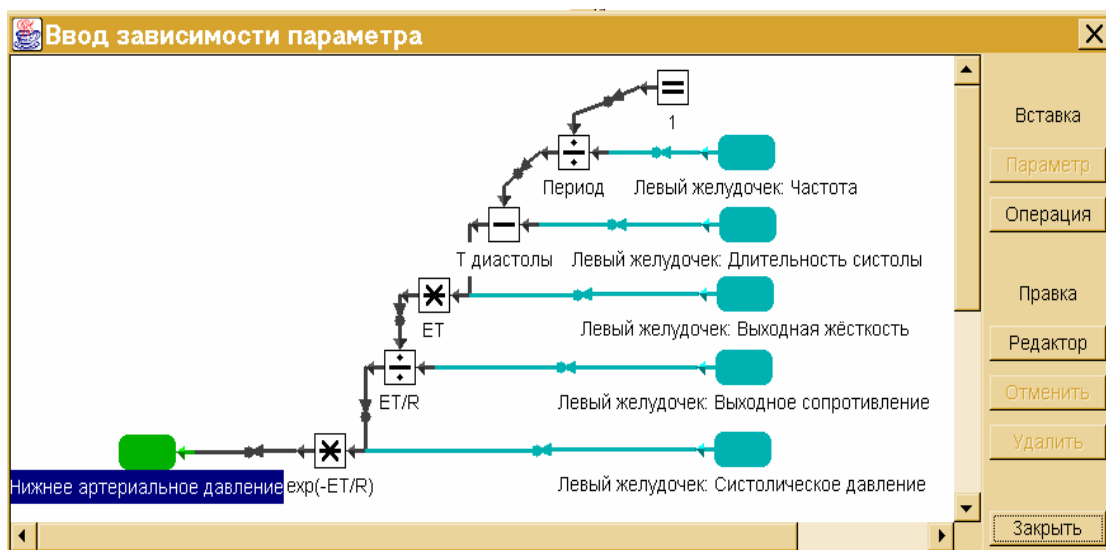


Рис. 27.

### 3. Уровень сценария.

В данном случае создается модель функционирования сердечно-сосудистой системы в норме, поэтому входные параметры отсутствуют. Достаточно просто запустить модель на выполнение (с помощью диалога управления моделированием, см. рис. 14). В качестве примера на рис. 28, 29 и 30 приводятся графики некоторых параметров модели.

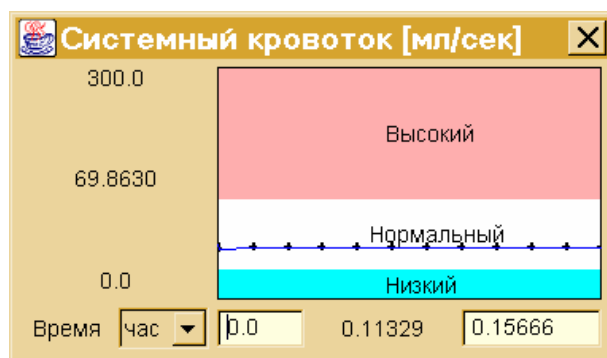


Рис. 28.

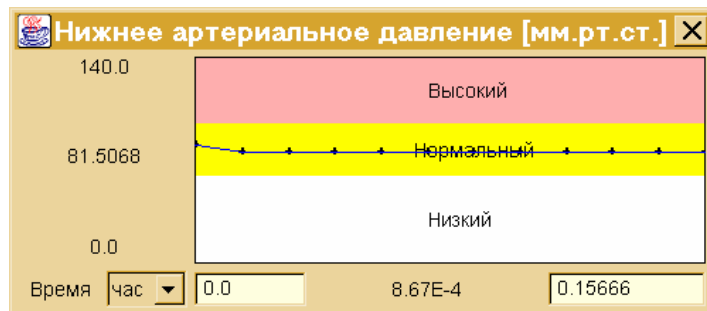


Рис. 29.

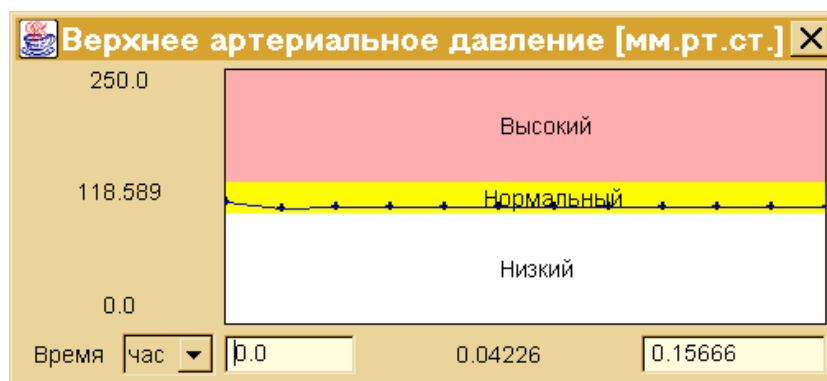


Рис. 30.

Таким образом, в визуальном режиме, с помощью «Среды визуального моделирования» создана простейшая модель сердечно-сосудистой системы. Фактически, создан «шаблон» для моделей. «Среда визуального моделирования» дает возможность легкого их расширения и дополнения новыми данными (добавления новых элементов и контуров регуляции) при одновременном сохранении общего подхода, обеспечении преемственности и сравнимости моделей.

## 7 Выводы

Разработан метод декомпозиции сложных физиологических систем, позволяющий эффективно формализовывать процесс создания моделей различных систем организма.

Создана объектно-ориентированная библиотека основных элементов сердечно-сосудистой системы.

Создан программный продукт «Среда визуального моделирования функциональных» (СВМ), позволяющий в визуальном режиме, не прибегая к программированию вычислительных алгоритмов и интерфейса, создавать математические модели физиологических систем организма. (Таким образом, в СВМ реализована концепция моделирования без уравнений.)

СВМ предназначена для создания математических моделей на уровне подсистем организма или всего организма в целом. С помощью СВМ можно создавать модели многих систем организма, однако, основной, наиболее разработанной ее частью являются элементы, относящиеся к моделированию сердечно-сосудистой системы.

При создании моделей с помощью данного программного продукта все проблемы, которые были сформулированы во Введении к настоящей работе, получают свое решение. А именно:

- предложен подход к единообразному моделированию контуров регуляции (путем явного отделения «физики» от «физиологии»);
- предложены (в виде библиотеки) наиболее существенные элементы, из которых (при необходимости добавляя новые) можно строить модели физиологических систем различной детальности, объема и специфики;
- поскольку все модели, созданные с помощью данного программного продукта, основываются на одних и тех же базовых элементах и принципах построения, то таким образом обеспечен единый подход к моделированию систем организма, модели легко сравнивать между собой, и обеспечена полная преемственность, т. к. новые модели можно создавать, просто дополняя и расширяя уже существующие.

- т. к. «Среда визуального моделирования» реализована на языке Java, это позволяет использовать ее на компьютерах с различными архитектурами и операционными системами.

В качестве примера использования программного продукта приведена одна из моделей сердечно – сосудистой системы. При создании этой модели ставилась задача не столько получить некие новые физиологические данные, сколько продемонстрировать преимущества визуального моделирования в физиологии. Таким образом, создан «шаблон» для моделей. «Среда визуального моделирования» дает возможность легкого их расширения и дополнения новыми данными при одновременном сохранении общего подхода, обеспечении преемственности и сравнимости моделей.

Таким образом, разработаны новые методы моделирования и на их основе создан полностью готовый коммерческий программный продукт, который позволяет, как проводить исследования в области физиологии, так и решать многие прикладные задачи в области экологии и клиники.

## 8 Список литературы

1. Бурыкин А.А., Евдокимов А.В. О применении объектно-ориентированного анализа при создании сложных компьютерных моделей в физиологии. Тезисы докладов XLII научной конференции МФТИ, Москва - Долгопрудный, 1999.
2. Бурыкин А.А., Евдокимов А.В. Среда визуального моделирования функциональных систем организма. Тезисы докладов XLII научной конференции МФТИ, Москва - Долгопрудный, 1999.
3. Евдокимов А.В., Бурыкин А.А. О хранении и представлении данных в системах моделирования. Тезисы докладов XLII научной конференции МФТИ, Москва - Долгопрудный, 1999.
4. Бурыкин А.А., Холодов А.С., Евдокимов А.В., Будник С.В. Математическая модель с распределенными параметрами функционирования и взаимосвязи физиологических систем организма. - Тезисы докладов XLI научной конференции МФТИ, Долгопрудный, 1998.
5. Евдокимов А. В. Численное моделирование осредненного по времени кровообращения человека. Диплом бакалавра. (МФТИ, Долгопрудный, 1998.)
6. Евдокимов А. В. Объектно-ориентированный подход в вычислительной математике и имитационном моделировании. Магистерская диссертация. (МФТИ, Долгопрудный, 2000).
7. Буч Г. Объектно-ориентированный анализ и проектирование. «Издательство Бинум», «Невский диалект» 1998.
8. Новоженев Ю. В. Объектно-ориентированные технологии разработки сложных программных систем. – М. 1996.
9. Бадд Т. Объектно-ориентированное программирование в действии. СПб., «Питер», 1997.
10. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. М., «Финансы и статистика», 1998.
11. Чекмарев А. Средства визуального проектирования на Java. СПб.: BHV – Санкт-Петербург, 1998.

12. Вебер Дж. Технология Java в подлиннике. СПб.: BHV – Санкт-Петербург, 1997.
13. Физиология человека. Под ред. Р. Шмидта и Г. Тевса. Тт.1-3. М. Мир, 1996.
14. Физиология кровообращения: Физиология сосудистой системы. Ред. Б. И. Ткаченко. Л.: Наука, 1984.
15. Каро К., Педли Т., Шротер Р., Сид У. Механика кровообращения. - М.: Мир, 1981.
16. Лищук В. А. Математическая теория кровообращения. М. «Медицина», 1991.
17. Бураковский В. И., Бокерия Л. А., Газизова Д. Ш., Лищук В. А., Люде М. Н., Работников В. С., Соколов М. В., Цховребров С. В. Компьютерная технология интенсивного лечения: контроль, анализ, диагностика, лечение, обучение. М. 1995.
18. Гайтон А. Минутный объём сердца и его регуляция. - М.: Мир, 1969.
19. Амосов Н.М., Палец Б.Л., Агапов Б.Т., Ермакова И.И., Лябах Е.Г., Пацкина С.А., Соловьев В.П. Теоретическое исследование физиологических систем. Математическое моделирование. - К.: Наукова думка, 1977.
20. Шакин В. В., Карасев А. А., Комиссаров К. В., Пржиялковский В. В. Динамоэксперт – система применения персональных ЭВМ для построения и исследования моделей в медицине. //Модели в экологии и медицине (физиологические системы человека в экстремальных условиях). Москва. Вычислительный центр АН СССР.
21. Guyton A. C., Coleman J. C., Granger H. G. Circulation: Overall regulation.- Ann. Rev. Physiol., 1972, p. 13-41.
22. Биологическая и медицинская кибернетика. Справочник. Киев, «Наукова думка» , 1986.
23. Coleman T. J. A mathematical model of the human body in heart, disease and during treatment // Proceedings Joint. Autom. Cont. Conf. – Philadelphia, 1978. – Voll. 3 – 4. –P. 77 – 86.
24. Ikeda N., Marimo F., Shiratara M., Sato T. A model of overall regulation of body fluids.// Ann. Biomed. Eng. 1979. Vol. 7. P. 135-166.

25. Белотелов Н. В., Саранча Д. А. Системный подход к моделированию процессов питания и пищеварения. // Использование вычислительных средств в экологии, экономике и медицине. Саратов: Изд-во Саратовского ун-та. 1988.
26. Куликов Г. Ю. Реализация модели гомеостаза на персональной ЭВМ. // Модели в экологии и медицине (физиологические системы человека в экстремальных условиях). Вычислительный центр АН СССР. М. 1989.
27. [www.mathworks.com](http://www.mathworks.com)
28. <http://www.microsim.com>
29. <http://www.orcad.com>
30. Body Simulation. Instruction Manual (Version 4.0) Advanced Simulation Corporation. 1996.

## 9 Приложение. Модель работы сердца и замыкание модели кровеносной системы

(Из [5])

Любая модель сосудистой системы одного круга кровообращения позволяет связать одним уравнением три величины: кровоток через неё  $Q$ , давление на входе  $P_{in}$  и на выходе  $P_{out}$ . Так как в некоторых задачах разница кровотоков через большой и малый круг может быть существенной, мы имеем два уравнения для шести неизвестных. Следовательно, чтобы система уравнений была замкнутой, модель сердца должна привести к ещё четырём соотношениям, связывающим центральные давления и кровотоки.

### 9.1 Имитационное моделирование сердечной деятельности

#### 9.1.1 Краткие сведения по физиологии работы сердца

Рассмотрим один из желудочков сердца, отталкиваясь от принятого в физиологии способа описания его работы. Свойства миокарда принято сводить к трём эмпирическим диаграммам «давление-объём», для которых из литературы можно хотя бы косвенно извлечь их количественные характеристики:

- 1) кривая пассивного растяжения – зависимость давления в расслабленном желудочке (в диастоле)  $P_d$  от его объёма  $V_d$ ;
- 2) кривая изоволюметрических максимумов – соотношение между начальным объёмом желудочка  $V_d$  и максимальным давлением  $P_m$ , которое он может развить при данном (постоянном) объёме;
- 3) кривая изотонических максимумов – соотношение между начальным давлением в желудочке  $P_d$  и минимальным объёмом  $V_m$ , до которого он может сократиться при данном (постоянном) давлении.

Реальное сокращение сердца, с момента открытия аортального и легочного клапанов, происходит по ауксотоническому закону, то есть по некоторому промежуточному между изоволюметрическим и изотоническим. На диаграмме цикла работы сердца (см. рис. 6) в первом приближении этот факт можно описать наклонным отрезком прямой. В момент перехода от систолы к диастоле этот отрезок достигает т. н. «кривой сокращений с постнагрузкой» положение которой на PV-

диаграмме определяется через параметры начальной (конечнодиастолической) точки рабочего цикла.

При фиксированных свойствах миокарда (т. е. при неизменных параметрах всех трёх сердечных кривых) сердце следующим образом приспосабливается к изменению входных и выходных параметров – притока крови из центральных вен и давления в центральных артериях. Повышение венозного возврата («нагрузка объёмом») растягивает желудочек в период диастолы, так что ударный объём и систолическое давление увеличиваются (см. рис. 8). Повышение артериального давления («нагрузка давлением») сначала сопровождается падением ударного объёма (клапаны открываются позже – см. рис. 9), однако при постоянном венозном возврате это приводит к диастолическому растяжению желудочка. Растяжение же, согласно первому приспособительному механизму, снова увеличивает ударный объём (возвращая его к прежнему значению) и ещё более увеличивает систолическое давление. Следует подчеркнуть, что все эти явления, сопровождающиеся увеличением размеров сердца при нагрузке, относятся к изолированному сердцу и в таком виде не вполне соответствуют закономерностям живого организма.

Влияние вегетативной нервной системы (а также действие растворённых в крови химических факторов) приводит к другим механизмам приспособления сердечной деятельности к нагрузкам. Основой для них служат повышение сократимости миокарда (инотропный эффект) и учащение пульса (хронотропный эффект). Инотропный эффект проявляется в смещении кривой изоволюметрических максимумов вверх (см. рис. 7). Это позволяет сердцу в случае нагрузки объёмом увеличивать ударный объём без роста диастолического, а в случае нагрузки давлением – выбрасывать против возросшего давления тот же ударный объём. Следовательно, в отличие от реакции изолированного сердца, во втором случае размеры сердца при нагрузке не изменяются, а в первом – даже уменьшаются за счёт падения систолического объёма. Хронотропный эффект заключается в уменьшении периода сердечных сокращений. Вследствие этого, в частности, давление в центральных артериях к концу диастолы не успевает сильно снизиться по сравнению со своим систолическим значением, и возникает дополнительная нагрузка давлением. С другой стороны, хронотропный эффект увеличивает кровоток через сердце (даже при неизменном ударном объёме), падение давления в сосудистой

системе растёт (даже при неизменном сопротивлении сосудов), и венозное давление может уменьшиться, несмотря на рост артериального.

Несмотря на вегетативные рефлексy, авторегуляторные механизмы приспособления сердечной деятельности к нагрузкам играют большую роль в организме, поскольку существует множество воздействий, при которых вегетативный тонус не изменяется (перемена положения тела, переливание крови, первичное влияние изменений периферического сопротивления и т. п.).

### 9.1.2 Математическая формулировка эмпирических закономерностей

Стандартные PV-диаграммы работы сердца проще всего ввести в модель, используя линейную интерполяцию между экспериментальными точками. В результате имеем:

1.  $V_d = f_1(P_d)$  – кривая пассивного растяжения;
2.  $P_m = \beta \cdot f_2(V_d)$  – кривая изоволюметрических максимумов ( $\beta$  – коэффициент, описывающий инотропный эффект нервной системы);
3.  $V_m = f_3(P_d)$  – кривая изотонических максимумов.
4. Кривая сокращений с пострелазкой с достаточной точностью может быть приближена прямой  $P_S = \frac{P_m - P_d}{V_d - V_m} \cdot V_S + \frac{P_d V_d - P_m V_m}{V_d - V_m}$ , где индексом «S» обозначены систолические значения давления и объёма.

Из предположения о линейности отрезка PV-диаграммы сердца, соответствующего периоду изгнания, следует соотношение  $P_S - P_a = K \cdot (V_d - V_S)$ . На основании сравнения различных диаграмм можно заключить, что наклон этого отрезка  $K$  изменяется лишь незначительно даже в результате нервных влияний на миокард.

Помимо ударного объёма ( $V_d - V_S$ ), систолическое давление  $P_S$  зависит от диастолического давления в аорте или лёгочной артерии  $P_a$ , которое противодействует открытию соответственно аортального или лёгочного клапанов, тем самым определяя начало периода изгнания. Эти две величины можно связать ещё одним соотношением, описывающим процесс релаксации давления в системе сосудов. Для этой цели сосуды можно условно разделить на амортизирующие

(крупные артерии) и резистивные (все остальные), первые характеризуются интегральным модулем объёмной упругости  $E = dP/dV$ , а вторые – суммарным гемодинамическим сопротивлением  $R = dP/dQ$ , где  $Q = dV/dt$ . Усредняя волновые процессы, происходящие в артериях после сердечного сокращения, можно считать, что вплоть до следующего сокращения они находятся в состоянии упругого растяжения. Тогда, с учётом пропорциональности между перепадом давления и кровотоком через резистивные сосуды, можно получить  $\frac{dP}{dt} = -E \frac{dV}{dt} = -E \frac{P - P_d}{R}$ , откуда  $P_a = P_d + (P_s - P_d) \cdot \exp\left[-\frac{E}{R} \tau_d\right] \approx P_s \cdot \exp\left[-\frac{E}{R} \tau_d\right]$ , где длительность диастолы  $\tau_d$  равна разности между периодом сердечного цикла  $\tau$  и продолжительностью систолы  $\tau_s$ .

Наконец, кровоток через желудочек  $Q$  очевидным образом связан с ударным объёмом:  $Q = \frac{V_d - V_s}{\tau}$ , что замыкает (в указанном ниже смысле) систему из 7 уравнений для одного желудочка, содержащей 8 неизвестных ( $P_s, P_a, P_d, P_m, V_s, V_d, V_m, Q$ ).

Таким образом, были сделаны следующие упрощения модели осреднённой по времени работы сердца:

- 1) кривые пассивного растяжения и изотонических максимумов для обоих желудочков сердца не изменяются ни при каких воздействиях на модель;
- 2) влияние внешних факторов на кривую изоволюметрических максимумов сводится к её умножению на некоторый коэффициент;
- 3) кривая сокращений с постнагрузкой приближается с помощью линейной функции (ввиду отсутствия количественных данных о нелинейных поправках к ней);
- 4) активные участки рабочей диаграммы сердца также аппроксимируются отрезками прямых, причём их наклоны для каждого желудочка фиксированы;
- 5) давление в центральных артериях после выброса крови из сердца падает по экспоненциальному закону на протяжении всей диастолы;

б) длительность систолы сохраняется при всех изменениях в модели, то есть сердечный цикл может укорачиваться или удлиняться только за счёт диастолы (это предположение необязательно, но вполне соответствует поведению миокарда в норме).

## **9.2 Алгоритм решения системы нелинейных уравнений замкнутой кровеносной системы**

Будем обозначать индексом 1 все величины, относящиеся к левому отделу сердца и большому кругу кровообращения, а индексом 2 – к правому сердцу и лёгочному кругу.

Пусть задано начальное приближение для диастолического давления в левом сердце  $P_d = P_{out}^2$ . Через это значение находится диастолический объём  $V_d$  (по кривой пассивного растяжения) и минимальный систолический объём  $V_m$  (по кривой изотонических максимумов). Из кривой изоволюметрических максимумов (учитывающей инотропный эффект ВНС) по  $V_d$  рассчитывается максимальное систолическое давление  $P_m$ . С помощью соотношений  $P_s - P_a = K \cdot (V_d - V_s)$  и

$P_a = P_d + (P_s - P_d) \cdot \exp\left[-\frac{E}{R} \tau_d\right]$ , находится систолическое давление

$P_s = P_d + \frac{K \cdot (V_d - V_s)}{1 - e^{-x}}$  и эффективное давление на входе в сосудистую систему

большого круга  $P_{in}^1 = P_d + \frac{1}{\tau_d} \int_0^{\tau_d} (P_s - P_d) \exp\left(-\frac{E}{R} t\right) dt = P_d + K \cdot (V_d - V_s) \frac{R}{E \cdot \tau_d}$ , т. е.

$P_{in}^1 = P_d + \frac{K \tau}{E \tau_d} RQ$ . Здесь  $R$  – сопротивление этой системы,  $E$  – модуль упругости её

артерий,  $x = \frac{E \tau_d}{R}$ . С учетом того, что кривую сокращений с постнагрузкой можно

аппроксимировать отрезком с концами в точках  $(V_m, P_d)$  и  $(V_d, P_m)$ , получается следующее выражения для кровотока через левый желудочек:

$$Q = \frac{P_m - P_d}{\tau} \cdot \left[ \frac{K}{1 - e^{-x}} + \frac{P_m - P_d}{V_d - V_m} \right]^{-1}.$$

С помощью модели сосудистой системы большого круга по найденным таким образом  $Q(P_{out}^2)$  и  $P_{in}^1(Q)$  рассчитывается центральное венозное давление (то есть давление в правом предсердии)  $P_{out}^1$ :  $P_{out}^1 = P_{in}^1 - RQ + \Delta P = \left( \frac{K\tau}{E\tau_d} - 1 \right) RQ + \Delta P$ . Здесь дополнительное давление  $\Delta P$  возникает вследствие негидродинамических механизмов венозного возврата (мышечный и дыхательный «насосы», «присасывание» предсердием крови из полых вен) и определяется амплитудами колебаний соответствующих давлений. Следует заметить, что  $\Delta P$  можно рассчитать с помощью моделей насоса, подобных модели желудочка (лишь несколько упрощённых).

Согласно описанному выше алгоритму для лёгочного круга можно вычислить другое значение общего кровотока  $Q^2(P_{out}^1)$ , однако оно, вообще говоря, не будет совпадать с  $Q(P_{out}^2)$ , как должно быть при согласованной работе обоих желудочков. Причина такого рассогласования заключается в том, что PV-диаграммы левого и правого сердца до сих пор считались независимыми и брались из не вполне точных физиологических данных, в то время как в реальности сократительные свойства обоих желудочков связаны именно условием равенства их ударных объёмов.

Согласовать работу правого желудочка (для которого форма PV-диаграмм известна хуже, чем для левого) с выбросом левого желудочка можно варьированием коэффициента  $\beta'$  кривой изоволюметрических максимумов правого сердца (см. раздел 9.1.2). В этом случае  $\beta'$  не задаётся извне модели, определяя инотропный эффект ВНС (как для левого сердца), а находится из условия минимального отличия кровотока через правый желудочек  $Q^2(\beta')$  от кровотока через левый желудочек  $Q$ :  $\beta = \arg \min |Q^2(\beta') - Q|$ . Однако вычисление  $\beta'$ , требующее реализации алгоритма поиска минимума, не является необходимым условием решения системы уравнений кровообращения; оно представляет интерес лишь для определения чисто «сердечных» переменных типа систолического объёма правого желудочка. Основные же давления для лёгочного круга  $P_{in}^2$  и  $P_{out}^2$ , интересные с точки зрения решения задачи в сосудах, зависят только от кровотока  $Q$ , уже известного из системы

уравнений левого сердца, и не зависят от  $\beta'$  (т. е. от кривой изоволюметрических максимумов). Поэтому они сразу рассчитываются по формулам  $P_{in}^2 = \frac{K\tau}{E\tau_d} RQ$ ,

$$P_{out}^2 = \left( \frac{K\tau}{E\tau_d} - 1 \right) RQ + \Delta P, \text{ где под константами } E, R, K, \Delta P \text{ понимаются параметры}$$

малого круга кровообращения: модуль упругости лёгочных артерий, суммарное сопротивление лёгочных сосудов, наклон рабочей диаграммы правого желудочка и дополнительный (не гидродинамический) перепад давления, прежде всего за счёт механизма дыхательного «насоса».

При некоторых патологиях сердца согласование деятельности желудочков может происходить в условиях перетекания части крови из одного желудочка в другой. Такой шунт проще всего рассматривать как эффективное сопротивление, соединённое параллельно с сопротивлением большого круга, и при этом сохранить полезное для расчётов условие равенства кровотока через правое и левое сердце. Однако поскольку кровоток через шунт определяется не средним артериальным, а средним желудочковым давлением, более точной является модель, в которой кровоток через лёгочные сосуды превышает системный кровоток. Поэтому в данном случае перед расчётом давлений в малом круге к  $Q$  следует добавить величину

$$\Delta Q = \frac{1}{R\tau} \int_0^{\tau} (P^1 - P^2) dt \approx \frac{1}{R\tau} \left[ \tau_d (P_d^1 - P_d^2) + (\tau - \tau_d) (P_s^1 - P_s^2) \right].$$

Таким образом, систему нелинейных уравнений кровообращения можно решить с помощью простых итераций всего лишь по одной переменной – диастолическому давлению в левом сердце  $P_{out}^2$ . Формально, один шаг итераций  $P^{n+1} = f(P^n)$  состоит из трёх вычислений значений функций, аппроксимирующих эмпирические кривые  $(V_d, V_m, P_m)$ , и одного расчёта нелинейного выражения  $P_{out}^2(Q(V_d, V_m, P_m))$ . С точки зрения, которая более соответствует физиологии задачи, каждую итерацию можно разделить на 2 этапа. На первом этапе через значение  $P_{out}^2$  из предыдущей итерации, а также через параметры большого круга и левого сердца вычисляется  $Q$ , а на втором – через  $Q$  и параметры малого круга находится новое значение  $P_{out}^2$ .

Сходимость итерационного процесса  $P^{n+1} = f(P^n)$  гарантируется тем, что его алгоритм не является искусственным, а соответствует механизмам приспособления сердца к изменению внешних параметров во времени. Однако поскольку временная динамика такого приспособления в рамках любой квазистационарной модели не является предметом исследования, скорость сходимости итераций можно искусственным образом ускорить. В случае применения, например, метода Ньютона

$$P^{n+1} = P^n - \frac{f(P^n) - P^n}{\frac{f(P^n + dP) - f(P^n)}{dP} - 1}$$

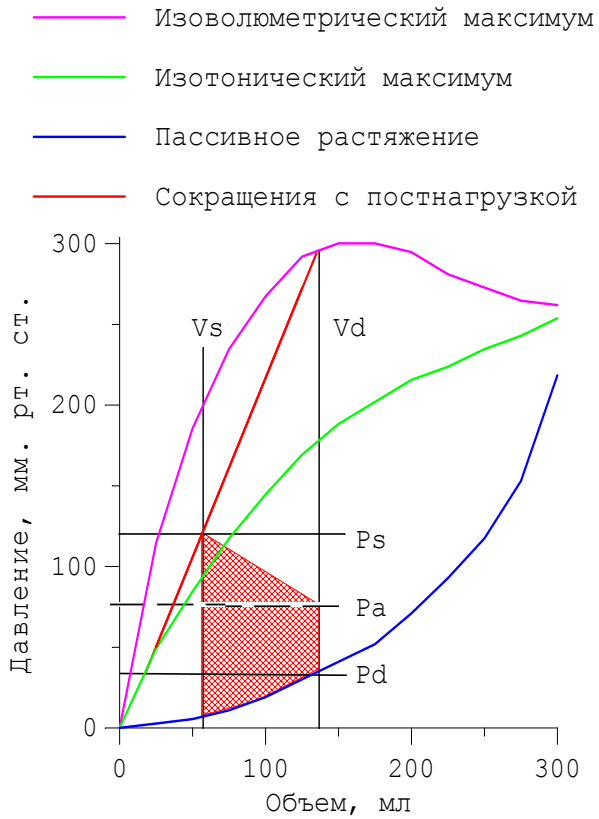
итерации имеют квадратичную скорость

сходимости.

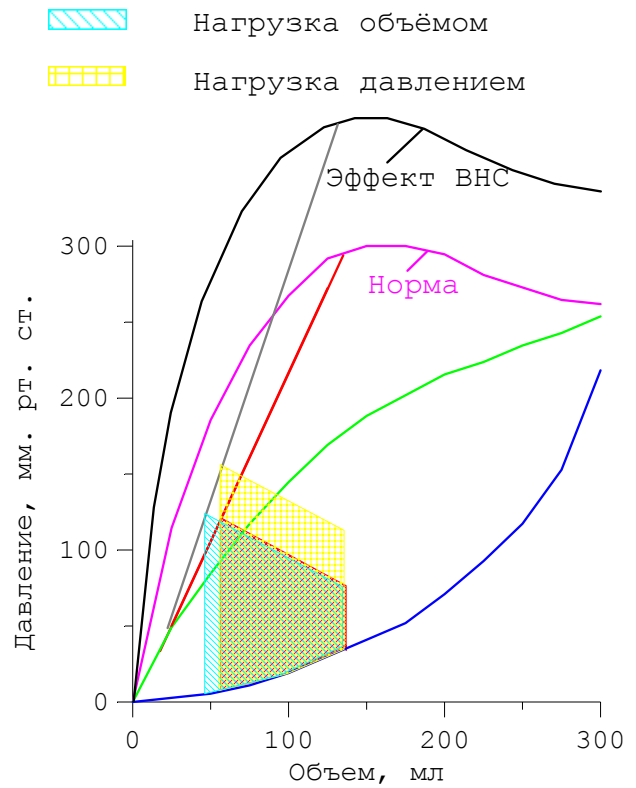
### **9.3 Резюме**

Предложенный подход к замыканию модели кровообращения моделью сердечной деятельности можно свести к следующему.

- Работа сердца описывается системой из десяти уравнений с восемью неизвестными, в число которых входят системный кровоток и 4 давления на входах и выходах сосудистых систем обоих кругов кровообращения.
- Объединение этой системы с двумя конечными уравнениями периферического кровообращения позволяет исключить из модели все неопределённые параметры, по крайней мере, те, которые зависят только от состояния кровеносной системы, а не связывают её с другими системами организма.
- Модель сердца основана на нескольких эмпирических зависимостях, причём некоторые параметры этих зависимостей удобно использовать для моделирования всевозможных внешних воздействий на кровообращение.
- Расчёт модели сердца сводится к решению одного нелинейного уравнения, итерационный алгоритм которого подобен процессу приспособления сердца к изменениям параметров, что гарантирует сходимость итераций.
- При необходимости описания патологий сердца или его естественного шунтирования модель легко обобщается на случай неодинакового кровотока через желудочки.



**Рис. 6 PV-диаграмма работы денервированного левого желудочка**



**Рис. 7 Влияние вегетативной нервной системы на приспособление сердца к нагрузкам**

Рис. 8 Авторегуляция работы сердца при нагрузке объёмом

Рис. 9 Авторегуляция работы сердца при нагрузке давлением

