

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В ОБРАЗОВАНИИ И
ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЯХ

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: «Моделирование высокоэнергетических процессов с использованием
сетевых вычислений»

Студент А.А. Морозов

Научный
руководитель д.ф.-м.н. доцент, профессор А.И. Сафронов

Руководитель программы д.пед.н., профессор, А.Н. Ярыгин

« » 20 г.

Допустить к защите

Заведующий кафедрой к.тех.н., доцент, А.В. Очеповский

« » 20 г.

Тольятти 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ.....	8
1.1 Высокоэнергетические процессы во внутренней баллистике.....	8
1.2 Модели для расчета параметров артиллерийских систем	13
1.2.1 Двухфазная двухскоростная модель сплошной среды.....	13
1.2.2 Особенности модели двухфазной двухскоростной среды.....	14
1.3 Преимущества численного моделирования при решении прикладных задач	14
1.3.1 Преимущества параллельных вычислений.....	18
1.3.2 Способы синхронизации параллельного взаимодействия.....	19
ГЛАВА 2. ОПИСАНИЕ ФИЗИКО-МАТЕМАТИЧЕСКОЙ МОДЕЛИ	22
2.1 Физическая модель.....	22
2.1.1 Модель баллистической системы с присоединенными зарядами.....	23
2.2 Математическая модель схемы метания с присоединенными зарядами на основе двухфазной модели гетерогенной среды	24
2.3 Численное решение задачи методом распада произвольного разрыва	32
2.4 Обоснование адекватности модели двухфазной двухскоростной среды с присоединенными зарядами.....	40
ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПЬЮТЕРНОЙ МОДЕЛИ	42
3.1 Описание функциональности компьютерной модели.....	42
3.2 Разработка алгоритмов компьютерной модели	45
3.2.1 Расчет параметров системы на временном слое	45
3.2.2 Подбор входных параметров с использованием алгоритмов распараллеливания	48
3.3 Разработка программного приложения.....	49
3.3.1 Структура головного модуля	50
3.3.2 Описание переменных головного модуля	51
3.3.3 Описание методов головного модуля	51

3.3.4 Структура класса ThreadCalculate	53
3.3.5 Структура класса CalculateDD	53
3.4 Интерфейс программы.....	65
3.4.1 Главное меню программы	69
ГЛАВА 4. РЕЗУЛЬТАТЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ	71
4.1 Результаты расчета системы малого калибра	71
4.2 Согласование расчётных данных с экспериментом	72
4.3 Оценка эффективности алгоритма распараллеливания	74
ЗАКЛЮЧЕНИЕ	77
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	79
ПРИЛОЖЕНИЕ 1. Программный код класса CalculateDD	85

ВВЕДЕНИЕ

Актуальность темы исследования. В настоящее время в промышленности достаточно широко применяются высокоэнергетические процессы и установки, использующие энергию взрывчатых веществ, твердых и жидких топлив. В частности они нашли применение в различных отраслях горнодобывающей, металлообрабатывающей, авиационной и строительной промышленности.

Экспериментальные исследования вышеупомянутых процессов и установок важны и их, конечно, следует проводить, например, выстрел из артиллерийского орудия. Однако они не дают требуемого объёма информации. При экспериментальном исследовании обычно проводятся замеры параметров (например, давления) в точке или в нескольких точках, а необходимо проводить замеры во множестве точек, распределённых по всей исследуемой области. Кроме того, любой физический эксперимент требует значительных ресурсных и финансовых затрат и обычно затянут во времени. Поэтому в данном случае для глубокого и всестороннего исследования естественно привлечение методов математического моделирования, в первую очередь методов, постановки вычислительного эксперимента – как наиболее точных и надёжных расчётных методов. Сопоставляя расчётную и экспериментальную информацию можно с минимальными затратами решить поставленную задачу.

В связи с этим, актуальным и необходимым становится разработка адекватных моделей систем реализующих высокоэнергетические процессы с целью повышения выходных характеристик.

Актуальность использования распределённых технологий обуславливается в сложности параметризации и расчетов систем с множеством входных и выходных параметров, что влияет на скорость получение характеристик рассматриваемой системы.

Цель исследования: создание эффективного метода моделирования высокоэнергетических процессов во внутренней баллистике с

использованием распределенных технологий.

Объект исследования: высокоэнергетические процессы во внутренней баллистике.

Предмет исследования: физико-математические модели баллистических систем с присоединенными зарядами на основе механики гетерогенных сред.

Задачи исследования:

1. Исследование баллистической системы с присоединенными зарядами;
2. Математическое моделирование рассматриваемой системы с использованием подхода гетерогенных сред для изучения высокоэнергетических процессов во внутренней баллистике;
3. Выбор численного метода решения замкнутой системы уравнений;
4. Отладка и параметрическое исследование предложенной системы;
5. Выявление параметров системы позволяющих значительно повысить энергетику установки;
6. Моделирование многопараметрических систем с использованием распараллеленного алгоритма, подходов сетевых вычислений или суперкомпьютеров;
7. Апробация полученной компьютерной модели на баллистических системах и сравнение результатов моделирования с реальными экспериментами.

Методологическая база исследования. Реализация методов решения поставленных задач осуществлялась при помощи аналитических и численных методов, методов анализа и синтеза технических систем, методов математического моделирования физических процессов и методов программной реализации алгоритмов.

Научная новизна, полученных автором результатов заключается в следующем: разработана компьютерная модель для расчетов параметров баллистических систем с использованием распределённых технологий; проведено сравнение и согласование полученных расчетных данных с

экспериментальными данными других авторов или источников; показана эффективность применения алгоритмов распараллеливания для моделирования сложных систем.

Практическая значимость. Полученные в работе данные по моделированию высокоэнергетических процессов во внутренней баллистике могут быть использованы для нахождения улучшенных выходных параметров баллистических систем.

Достоверность и обоснованность научных положений и выводов, сделанных в диссертационной работе, следует из адекватности физических и математических моделей, используемых в работе, что подтверждается сравнением результатов данной работы с известными экспериментальными и теоретическими данными, полученными ранее другими авторами.

Текст диссертационной работы состоит из введения, четырёх глав, выводов по работе, списка цитируемой литературы и приложения. Общий объём работы составляет 110 страниц и содержит 18 рисунков, список цитируемой литературы включает 58 наименований.

В первой главе диссертации проводится анализ предметной области и производится постановка задачи исследования, а также выбор модели для расчетов баллистических систем. Здесь же рассматриваются возможности численных методов при решении прикладных задач.

Вторая глава посвящена описанию физико-математической модели высокоэнергетических процессов, протекающих в камере и стволе при выстреле артиллерийского орудия. Описываются физическая и математическая модели, а также численные методы решения (конечно-разностная реализация дифференциальной математической модели).

В третьей главе описываются алгоритмы и функциональность компьютерной модели, реализованной на основе физико-математической модели, предназначенной для расчёта динамики высокоэнергетических процессов. Программный продукт состоит из головного модуля, основного модуля вычислений, а также модулей обработки и вывода результатов расчёта.

При разработке программного продукта использовались методы структурного программирования и приемы многопоточковой обработки информации.

В четвёртой главе приводятся результаты компьютерного (численного) моделирования внутрикамерных процессов при выстреле артиллерийского орудия и даётся оценка эффективности алгоритма параллельных вычислений компьютерной модели.

В приложение помещен программный код основного модуля вычислений программного продукта.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

В данной главе речь идёт об анализе предметной области и производится постановка задачи исследования. Рассматриваются возможности численного моделирования при решении прикладных задач, анализируются высокоэнергетические процессы во внутренней баллистике.

При написании главы использовались материалы из [13, 18, 10, 11, 13, 18, 24, 46 и др.].

1.1 Высокоэнергетические процессы во внутренней баллистике

Внутренняя баллистика - это наука, занимающаяся изучением процессов, которые происходят при выстреле, и, в особенности, при движении пули (метаемого элемента) по каналу ствола. **Выстрелом** называется выбрасывание пули (метаемого элемента) из канала ствола оружия энергией газов, образующихся при сгорании порохового заряда. [33].

«При выстреле из стрелкового оружия происходят следующие явления. От удара бойка по капсюлю боевого патрона, досланного в патронник, взрывается ударный состав капсюля и образуется пламя, которое через затравочные отверстия в дне гильзы проникает к пороховому заряду и воспламеняет его. При сгорании порохового (боевого) заряда образуется большое количество сильно нагретых газов, создающих в канале ствола высокое давление на дно пули, дно и стенки гильзы, а также на стенки ствола и затвор. В результате давления газов на дно пули она сдвигается с места и врезается в нарезы; вращаясь по ним, продвигается по каналу ствола с непрерывно возрастающей скоростью и выбрасывается наружу по направлению оси канала ствола. Давление газов на дно гильзы вызывает движение оружия (ствола) назад. От давления газов на стенки гильзы и ствола происходит их растяжение (упругая деформация), и гильза, плотно прижимаясь к патроннику, препятствует прорыву пороховых газов в сторону затвора. Одновременно при выстреле возникает колебательное движение (вибрация) ствола и происходит

его нагревание. Раскаленные газы и частицы несгоревшего пороха, истекающие из канала ствола вслед за пулей, при встрече с воздухом порождают пламя и ударную волну; последняя является источником звука при выстреле. [33]».

«При выстреле из автоматического оружия, устройство которого основано на принципе использования энергии пороховых газов, отводимых через отверстие в стенке ствола (например, автомат и пулеметы Калашникова, снайперская винтовка Драгунова, станковый пулемет Горюнова), часть пороховых газов, кроме того, после прохождения пулей газоотводного отверстия устремляется через него в газовую камеру, ударяет в поршень и отбрасывает поршень с затворной рамой (толкатель с затвором) назад. Пока затворная рама (стебель затвора) не пройдет определенное расстояние, обеспечивающее вылет пули из канала ствола, затвор продолжает запирает канал ствола. После вылета пули из канала ствола происходит его отпирание; затворная рама и затвор, двигаясь назад, сжимают возвратную (возвратно-боевую) пружину; затвор при этом извлекает из патронника гильзу. При движении вперед под действием сжатой пружины затвор досылает очередной патрон в патронник и вновь запирает канал ствола. [33]».

«При выстреле из автоматического оружия, устройство которого основано на принципе использования энергии отдачи (например, пистолет Макарова, автоматический пистолет Стечкина, автомат обр.1941 г.), давление газов через дно гильзы передается на затвор и вызывает движение затвора с гильзой назад. Это движение начинается в момент, когда давление пороховых газов на дно гильзы преодолевает инерцию затвора и усилие возвратно-боевой пружины. Пуля к этому времени уже вылетает из канала ствола. Отходя назад, затвор сжимает возвратно-боевую пружину, затем под действием энергии сжатой пружины затвор движется вперед и досылает очередной патрон в патронник. [33]».

«В некоторых образцах оружия (например, крупнокалиберный пулемет Владимиров, станковый пулемет обр. 1910г.) под действием давления пороховых газов на дно гильзы вначале движется назад ствол вместе со

сцепленным с ним затвором (замком). Пройдя некоторое расстояние, обеспечивающее вылет пули из канала ствола, ствол и затвор расцепляются, после чего затвор по инерции отходит в крайнее заднее положение и сжимает (растягивает) возвратную пружину, а ствол под действием пружины возвращается в переднее положение. Иногда после удара бойка по капсюлю выстрела не последует или он произойдет с некоторым запозданием. В первом случае имеет место осечка, а во втором затяжной выстрел. Причиной осечки чаще всего бывает отсыревание ударного состава капсюля или порохового заряда, а также слабый удар бойка по капсюлю. Поэтому необходимо оберегать боеприпасы от влаги и содержать оружие в исправном состоянии. Затяжной выстрел является следствием медленного развития процесса зажигания или воспламенения порохового заряда, Поэтому после осечки не следует сразу открывать затвор, так как возможен затяжной выстрел. Если осечка произойдет при стрельбе из станкового гранатомета, то перед его разряжением необходимо выждать не менее одной минуты. [33]».

«При сгорании порохового заряда примерно 25— 35% выделяемой энергии затрачивается на сообщение пуле поступательного движения (основная работа); 15— 25% энергии а совершение второстепенных работ (врезание и преодоление трения пули при движении по каналу ствола; нагревание стенок ствола, гильзы и пули; перемещение подвижных частей оружия, газообразной и несгоревшей частей пороха); около 40% энергии не используется и теряется после вылета пули из канала ствола. Выстрел происходит в очень короткий промежуток времени (0,001-0,06 с). При выстреле различают четыре последовательных периода: предварительный; первый, или основной, второй; третий, или период последствий газов. [33]».

«**Предварительный период** длится от начала горения порохового заряда до полного врезания оболочки пули в нарезы ствола. В течение этого периода в канале ствола создаёт давление газов, необходимое для того, чтобы сдвинуть пулю с места и преодолеть сопротивление ее оболочки врезанию в нарезы ствола. Это давление называется **давлением форсирования**; оно достигает

250-500 кг/см² в зависимости от устройства нарезов, массы пули и твёрдости её оболочки (например, у стрелкового оружия под патрон обр.1943 г. давление форсирования равно около 300 кг/см²). Принимают, что горение порохового заряда в этом периоде происходит в постоянном объеме, оболочка врезается в нарезы мгновенно, а движение пули начинается сразу же при достижении в канале ствола давления форсирования. [24, 11, 54, 33]».

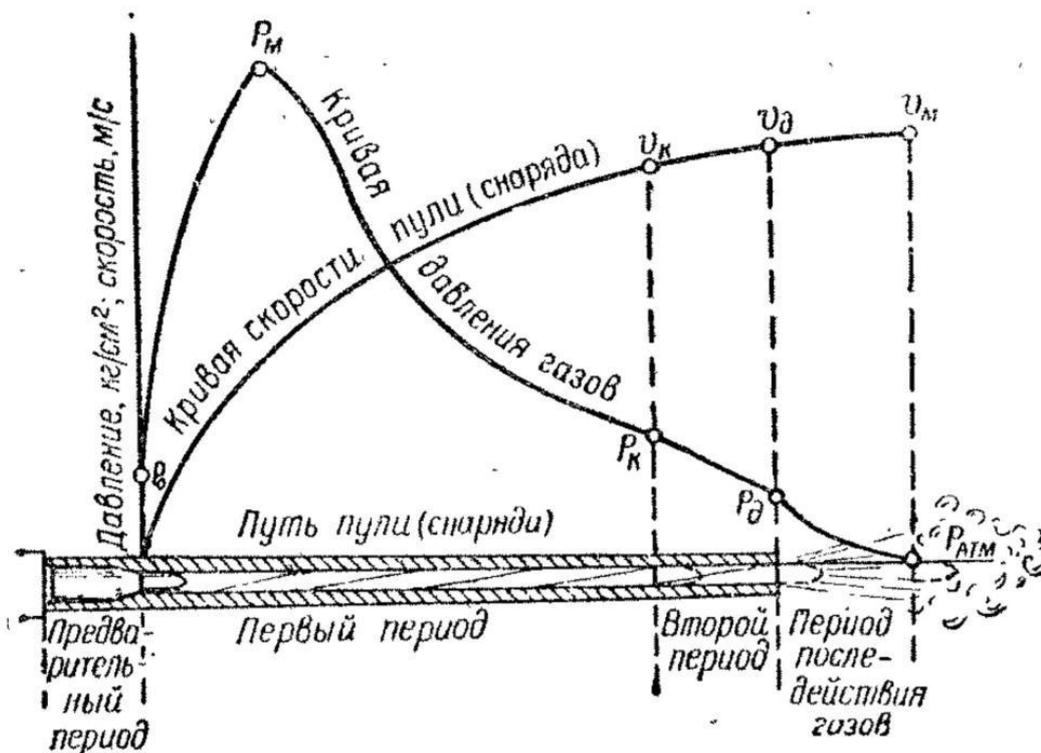


Рисунок 1.1 – Периоды выстрела: P_o – давление форсирования; P_m – наибольшее (максимальное) давление; P_k – и v_k – давление газов и скорость пули в момент конца горения пороха; P_d и v_d – давление газов и скорость пули в момент вылета её из канала ствола; v_m – наибольшая (максимальная скорость) пули; $P_{отм}$ – давление, равное атмосферному

«**Первый**, или **основной период** длится от начала движения пули до момента полного сгорания порохового заряда. В этот период горение порохового заряда происходит в быстро изменяющемся объеме. В начале периода, когда скорость движения пули по каналу ствола еще невелика, количество газов растет быстрее, чем объем за-пульного пространства (пространство между дном пули и дном гильзы), давление газов быстро повышается и достигает наибольшей величины (например, у стрелкового

оружия под патрон обр. 1943 г. — 2800 кг/см^2 , а под винтовочный патрон — 2900 кг/см^2). Это давление называется **максимальным давлением**. Оно создается у стрелкового оружия при прохождении пулей 4 — 6 см пути. Затем вследствие быстрого увеличения скорости движения пули объем запульного пространства увеличивается быстрее притока новых газов, давление начинает падать, к концу периода оно равно примерно $\frac{2}{3}$ максимального давления. Скорость движения пули постоянно возрастает и к концу периода достигает примерно $\frac{3}{4}$ начальной скорости. Пороховой заряд полностью сгорает не задолго до того, как пуля вылетит из канала ствола. [33]».

«Второй период длится от момента полного сгорания порохового заряда до момента вылета пули из канала ствола. С началом этого периода приток пороховых газов прекращается, однако сильно сжатые и нагретые газы расширяются и, оказывая давление на пулю, увеличивали скорость ее движения. Спад давления во втором периоде происходит довольно быстро и у дульного среза — дульное давление — составляет у различных образцов оружия $300\text{—}900 \text{ кг/см}^2$ (например, у самозарядного карабина Симонова — 390 кг/см^2 , у станкового пулемета Горюнова 570 кг/см^2). Скорость пули в момент вылета ее из канала ствола (дульная скорость) несколько меньше начальной скорости. [33]».

«У некоторых видов стрелкового оружия, особенно короткоствольных (например, пистолет Макарова), второй период отсутствует, так как полного сгорания порохового заряда к моменту вылета пули из канала ствола фактически не происходит. [33]».

«Третий период, или период последствия газов, от момента вылета пули из канала ствола до момента прекращения действия пороховых газов на пулю. В течении этого периода пороховые газы, истекающие из ствола со скоростью $1200\text{—}2000 \text{ м/с}$, продолжают воздействовать на пулю и сообщают ей дополнительную скорость. Наибольшей (максимальной) скорости пуля достигает в конце третьего периода на удалении нескольких сантиметров от дульного среза ствола. Этот период заканчивается в тот момент, когда давление

пороховых газов на дно пули будет уравновешено сопротивлением воздуха. [33]».

1.2 Модели для расчета параметров артиллерийских систем

Одним из важнейших вопросов в развитии ствольного оружия является повышение начальной скорости снаряда. За последние десятилетия было предложено множество разнообразных схем, позволяющих тем или иным, как правило, достаточно экзотическим, способом получать высокие скорости снарядов [33].

Расчет параметров артиллерийских систем проводится с использованием множества моделей: термодинамической, газопороховой смеси, двухфазной. В реальности оказалось, что более адекватной является модель гетерогенной среды (в данном случае двухфазная двухскоростная), которая дает результаты, согласующиеся с экспериментальными данными [13, 18].

1.2.1 Двухфазная двухскоростная модель сплошной среды

Теории движения многофазных гетерогенных сред посвящена обширная литература. В разные годы этим вопросом занимались Н. А. Слезкин, Х. А. Рахматулин, С. Трусделл, С. Г. Телетов, А. Н. Крайко, Л. Е. Стернин, Р. И. Нигматулин, В. Н. Николаевский, А. А. Шрайбер, С. Соу, М. А. Гольдштик, Р. Бусройд, Ушаков В.М., Никулин О.А, Сафронов А.И. и др [9, 10, 11, 18, 31].

Из возможных гетерогенных смесей наиболее подробно изучены вследствие своей сравнительно регулярной структуры, так называемые *дисперсные смеси* [9, 10, 11], которые состоят из двух фаз, одна из которых представляет собой капли, пузырьки или твердые частицы.

Представляет большой интерес применить подход механики многофазных многоскоростных континуумов, для моделирования работы импульсных систем, химических реакторов производящих ультра дисперсные порошки. В связи с этим необходимо вывести замкнутую систему уравнений, описывающую происходящие процессы, в частности для двухфазной и двухскоростной среды [9, 10, 11, 21].

1.2.2 Особенности модели двухфазной двухскоростной среды

Можно выделить несколько особенностей и достоинств модели:

1. В отличие от газо-пороховой модели двухфазная двух скоростная модель учитывает взаимодействие твердых частиц с газовой фазой [10];
2. Для газовой и твердой фаз записываются уравнения неразрывности и движения [10, 11];
3. Реализуется учет действительного взаимодействия газовой и твердой фаз;
4. В уравнениях для газовой фазы учитывается газоприток от горения твердых частиц;
5. В уравнениях для твердой фазы учитывается превращение твердых частиц в газ при горении;
6. Взаимодействие твёрдой и газовой фаз учитывается за счет введения в уравнения силы трения, которая описывает взаимодействие твердых частиц с потоком газа [10, 11].

1.3 Преимущества численного моделирования при решении прикладных задач

С появлением ЭВМ начало активно развиваться новое направление в математическом моделировании – численное моделирование. Аналогично физическому эксперименту, когда исследование проводится на специально сконструированной лабораторной установке, при численном моделировании эксперимент проводится путём вычислений на ЭВМ. В этом случае физические процессы имитируются на компьютере и описываются при помощи математической модели, которая представляет собой систему дифференциальных или интегральных уравнений. Численное решение этих уравнений позволяет предсказать динамическое поведение исследуемой системы, определить её свойства и характеристики. Целесообразность такого подхода объясняется рядом причин, благодаря чему он нашёл своё применение во множестве исследовательских областей (физика, химия, экология, медицина и др.). Помимо этого, подход имеет ряд традиционных областей применения.

«Численное моделирование особенно важно там, где не совсем ясна физическая картина изучаемого явления, не познан до конца внутренний механизм взаимодействия» [12], то есть исходная физическая картина уточняется в процессе численного эксперимента. Особенно актуально применение численного моделирования в задачах механики твёрдого тела, механики жидкости и газа, физики плазмы и т. д. [12, 25]. Часто приходится сталкиваться со сложными режимами течения, возникающими в условиях до-, транс-, сверх- и гиперзвуковых скоростей потока. Такие задачи выдвигает, например, современная практика применения летательных аппаратов. Высокие температуры, возникающие в этих условиях, ведут к эффектам диссоциации и ионизации в потоке, а в некоторых случаях и к «свечению» газа. Подобные эффекты чрезвычайно трудно воспроизвести и исследовать в лабораторных условиях, либо при проведении натурного эксперимента, поскольку уже не достаточно, чтобы выполнялись классические критерии подобия, такие, как равенство чисел Маха и Рейнольдса соответственно [12, 25]. Здесь необходимо также равенство абсолютных давлений и абсолютных температур, что достигается лишь при совпадении размеров модели с размерами исследуемого объекта. Это означает, что условия эксперимента должны быть натурными. Делать это зачастую нецелесообразно из-за резко возрастающей технической сложности и дороговизны эксперимента подобной постановки. В отдельных случаях полноразмерный подход может быть даже опасен, как, например, при изучении катастрофических или экологических явлений. К тому же опытные измерения носят весьма ограниченный характер, чаще всего не давая достаточно большого объёма и разнообразия получаемых данных. Более целесообразно при таких обстоятельствах проводить численный эксперимент, для расчёта которого применяется ЭВМ. Отдельного внимания заслуживает временной фактор. Многие физические эксперименты невозможно провести из-за их чрезмерной затянутости во времени, как, например, определение «усталости» конструкций. Бывает и обратная ситуация, когда рассматриваемые процессы протекают слишком быстро, что значительно усложняет измерения

(например, артиллерийский выстрел и др.). Численное моделирование предоставляет иной подход ко времени эксперимента, делая его подвластным исследователю. Эксперимент можно прервать в любой момент его проведения, безопасно вмешаться в его процесс, а также рассмотреть физическую картину на любом временном шаге. Другое принципиальное преимущество численного моделирования связано со сложностью теоретической постановки исследуемых вопросов. Для, например, абсолютного большинства задач газовой динамики не только не доказаны теоремы существования и единственности решения, но и, скорее всего, они не могут быть даже сформулированы [10, 9, 13, 31]. У этих задач чрезвычайно сложна математическая постановка, в точном смысле она также не сформулирована.

При решении многих задач механики сплошных сред (и прежде всего механики жидкости и газа) приходится иметь дело со сложными дифференциальными уравнениями в частных производных, либо их интегральными аналогами. В общем случае это системы уравнений смешанного эллипτικο-параболического или эллипτικο-гиперболического типа с неизвестной формой поверхности перехода [17, 30]. Кроме того, постановка задачи в общем случае подразумевает наличие подвижных границ. Поверхности (или линии), описывающие такие границы, сами определяются и изменяются в процессе вычисления, что требует отдельного математического описания. Примером служит движение снаряда в канале ствола артиллерийского орудия под действием продуктов сгорания порохового заряда – задача, решаемая в данной работе. Из-за нестационарности, нелинейности и большого числа используемых переменных, такие системы уравнений почти не поддаются аналитическому решению. Численное моделирование в таком случае представляет собой практически единственное средство получения результата. Однако при всех вышеупомянутых преимуществах численного моделирования принципиальная роль физического эксперимента ни в какой степени не должна принижаться. Существенный недостаток численных моделей состоит в том, что их применимость ограничена рамками лежащей в

их основе математической модели. По результатам проведённого физического эксперимента можно оценить допустимость применения выбранной математической модели, подтвердить либо опровергнуть её адекватность исследуемому процессу. С другой стороны, численное моделирование способно объяснить физический эксперимент, внести ясность и полноту в понимание физического явления. Наиболее целесообразным, с практической точки зрения, является сочетание численного и физического подходов. Тем не менее, физическому эксперименту надо придавать главенствующее значение, поскольку его никогда не заменит численный эксперимент.

Рассмотрим согласно [12, 17, 23, 25, 27] основные этапы численного экспериментирования и их связь с аналогичными этапами постановки физического эксперимента. Вначале проводится анализ исследуемого объекта, строится его физическая модель и делается подробное математическое описание, т. е. выбирается подходящая математическая модель. При физическом экспериментировании на основе анализа объекта исследования выбирается схема эксперимента, проектируется экспериментальная установка, уточняются элементы её конструкции. Далее для решения системы дифференциальных или интегральных уравнений выбирается метод численного интегрирования (или дифференцирования), составляется разностная схема её решения с учётом возможностей ЭВМ. После этого разрабатывается и отлаживается вычислительный алгоритм, работающий по составленной схеме. Особое внимание уделяется вопросам корректности постановки задачи, устойчивости и точности её решения. Аналогично для физического эксперимента конструируется экспериментальная установка. Далее она отлаживается и калибруется. В результате получается готовое средство для исследования в том и другом случае.

На следующем этапе происходит непосредственно экспериментирование проведение серийных расчётов с использованием программы, либо серия экспериментальных измерений на лабораторной установке. Полученные результаты позволяют при необходимости откорректировать исходную

математическую модель, метод или алгоритм решения задачи, а также конструкцию экспериментальной установки и её приборное обеспечение. Происходит накопление фактов и отбор наиболее вероятных ситуаций. На этом этапе в наибольшей степени происходит взаимодействие двух подходов моделирования друг с другом. Проверенные результаты одного подхода могут подтвердить или опровергнуть верность результатов, полученных при другом подходе.

Таким образом, реализуется широкая взаимосвязь численного и физического типов экспериментирования. В совокупности они способны проверять и обогащать друг друга, совершенствовать методологию каждого из них за счёт другого, а недостатки одного подхода могут быть компенсированы достоинствами другого. Такое совместное их использование составляет современный подход к исследованию.

С ростом производительности и доступности ЭВМ у численного моделирования появляется всё больше возможностей [12]. Вместе с вычислительной техникой совершенствуются и методы вычислительной математики. Возникает возможность строить всё более детальную картину процессов, явлений или состояний с всё большим количеством переменных. Постепенно расширяется круг задач, которые можно решать, смоделировав их численно, а накопленные таким образом результаты исследований в одних областях можно эффективно применять для решения задач из других областей. Роль численного моделирования в исследовании возрастает всё больше, и этот подход вызывает всё больший интерес и доверие у исследователей.

1.3.1 Преимущества параллельных вычислений

Теория параллельных вычислений составляет раздел прикладной теории алгоритмов. [19, 26, 28, 29, 34, 39, 40, 41].

Параллельные вычисления — способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно). Термин охватывает совокупность вопросов параллелизма в

программировании, а также создание эффективно действующих аппаратных реализаций [16].

Существуют различные способы реализации параллельных вычислений. Например, каждый вычислительный процесс может быть реализован в виде процесса операционной системы, либо же вычислительные процессы могут представлять собой набор потоков выполнения внутри одного процесса ОС. Параллельные программы могут физически исполняться либо последовательно на единственном процессоре — перемежая по очереди шаги выполнения каждого вычислительного процесса, либо параллельно — выделяя каждому вычислительному процессу один или несколько процессоров (находящихся рядом или распределённых в компьютерную сеть). Основная сложность при проектировании параллельных программ — обеспечить правильную последовательность взаимодействий между различными вычислительными процессами, а также координацию ресурсов, разделяемых между процессами. [16].

1.3.2 Способы синхронизации параллельного взаимодействия

Взаимодействие через разделяемую память: на каждом процессоре мультипроцессорной системы запускается поток исполнения, который принадлежит одному процессу. Потоки обмениваются данными через общий для данного процесса участок памяти [29, 26, 41]. Количество потоков соответствует количеству процессоров. Потоки создаются либо средствами языка (например, в Java или C#), либо с помощью библиотек явно (например, в C/C++ с помощью PThreads), либо декларативно (например, с помощью библиотеки OpenMP), или автоматически встроенными средствами компилятора (например, HighPerformanceFortran). Данный вид параллельного программирования обычно требует какой-то формы захвата управления (мьютексы, семафоры, мониторы) для координации потоков между собой [16].

Взаимодействие с помощью передачи сообщений: на каждом процессоре многопроцессорной системы запускается однопоточный процесс, который обменивается данными с другими процессами, работающими на других

процессорах, с помощью сообщений. Процессы создаются явно, путем вызова соответствующей функции операционной системы, а обмен сообщениями — с помощью библиотеки (например, реализация протокола MPI), или с помощью средств языка (например, HighPerformanceFortran, Erlang или occam). Обмен сообщениями может происходить асинхронно, либо с использованием метода «рандеву», при котором отправитель заблокирован до тех пор, пока его сообщение не будет доставлено. Асинхронная передача сообщений может быть надёжной (с гарантией доставки) либо ненадёжной. [16, 39, 40].

«Параллельные системы, основанные на обмене сообщениями, зачастую более просты для понимания, чем системы с разделяемой памятью, и обычно рассматриваются как более совершенный метод параллельного программирования. Существует большой выбор математических теорий для изучения и анализа систем с передачей сообщений, включая модель факторов и различные виды исчислений процессов. Обмен сообщениями может быть эффективно реализован на симметричных мультипроцессорах как с разделяемой когерентной памятью, так и без неё. У параллелизма с распределенной памятью и с передачей сообщений разные характеристики производительности. Обычно (но не всегда), накладные расходы памяти на процесс и времени на переключение задач у систем с передачей сообщений ниже, однако передача самих сообщений более накладна, чем вызовы процедур. Эти различия часто перекрываются другими факторами, влияющими на производительность. [16]».

«Гибридный способ: на многопроцессорных системах с распределённой памятью (DM-MIMD), где каждый узел системы представляет собой мультипроцессор с общей памятью (SM-MIMD), можно использовать гибридный метод программирования [29,34]. На каждом узле системы запускается многопоточный процесс, который распределяет потоки между процессорами данного узла. Обмен данными между потоками на узле осуществляется через общую память, а обмен данными между узлами — через передачу сообщений. В этом случае количество процессов определяется

количеством узлов, а количество потоков — количеством процессоров на каждом узле. Гибридный способ программирования более сложен (требуется особым образом переписывать параллельную программу), но наиболее эффективен в использовании аппаратных ресурсов каждого узла многопроцессорной системы. [16]».

«Разумеется, в такой системе можно также использовать и исключительно метод передачи сообщений, то есть запустить на каждом процессоре каждого узла отдельный процесс. В этом случае количество процессов (и потоков) будет равно количеству процессоров на всех узлах. Этот способ проще (в параллельной программе надо только увеличить количество процессов), но является менее эффективным, так как процессоры одного и того же узла будут обмениваться друг с другом сообщениями, словно они находятся на разных машинах. [16]».

ГЛАВА 2. ОПИСАНИЕ ФИЗИКО-МАТЕМАТИЧЕСКОЙ МОДЕЛИ

В данной главе производится постановка задачи – физико-математической модели высокоэнергетических процессов, протекающих в камере и стволе при срабатывании артиллерийского орудия. Описываются физическая и математическая модели, а также численные методы решения (конечно-разностная реализация дифференциальной математической модели).

При написании главы использовались материалы из [8, 9, 10, 11, 12, 13, 18, 24, 25, 31, 30, 46, 45, 47, 52 и др.].

2.1 Физическая модель

На рисунке 2.1 изображена принципиальная схема компоновки ствола для артиллерийского орудия среднего калибра, рассматриваемого в комплексной задаче срабатывания артиллерийского орудия.

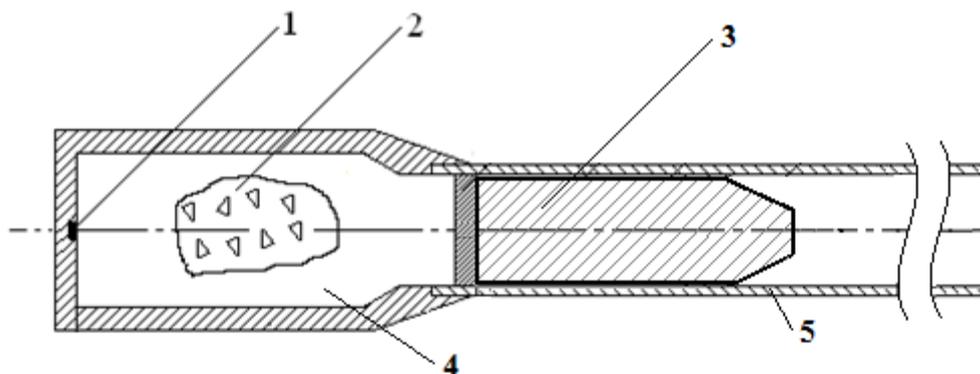


Рисунок 2.1 – Общая схема компоновки артиллерийского орудия: 1 – воспламенитель, 2 – пороховой заряд, 3 – снаряд, 4 – камера, 5 – ствол.

После зажигания воспламенителя пиропатроном, которая расположена внутри порохового заряда у дна камеры, пороховой заряд прогревается и постепенно воспламеняется. Горение порохового заряда происходит в нестационарном и турбулентном режиме. Так как, в канале ствола между пространствами за снарядом и перед снарядом полная герметичность отсутствует, продукты сгорания порохового заряда просачиваются из заснарядного пространства в предснарядное. При достижении продуктами

сгорания некоторого уровня давления, называемого давлением страгивания, в камере начинается движение снаряда. Происходит постепенное врезание ведущего пояска снаряда в нарезы ствола, в процессе которого снаряд в режиме скольжения входит в ствол. За счёт интенсивного поступления продуктов сгорания с поверхности заряда и наличия свободного объёма, в заснарядном пространстве возникают нестационарные колебания давления (и других газодинамических параметров). За счёт нарастания давления снаряд динамично ускоряется, продолжая своё движение по стволу артиллерийского орудия. Постепенно вырождаются нестационарность процесса горения и колебательный режим течения продуктов сгорания порохового заряда. Также по длине порохового заряда постепенно выгорает горящий свод [11, 24, 46, 47].

2.1.1 Модель баллистической системы с присоединёнными зарядами

Повышения скорости метания можно добиваться при использовании в баллистической системе одной или нескольких присоединённых зарядов к метаемому телу. Схема метания с присоединёнными зарядами (СПЗ) рассмотрена в [46, 47, 52]. Работу систем рассмотренной схемы можно обеспечить за счёт реализации зажигания присоединённого заряда в строго определённый момент времени (задержки зажигания) после зажигания основного заряда, например, за счёт воспламенения присоединённого заряда иницирующими газами, перетекающими из камеры основного заряда.

Имеются различные способы обеспечения перетока иницирующих газов из камеры основного заряда в область заряда присоединённой камеры.

Анализ применимости способов иницирования заряда присоединённой камеры можно проводить на основе решения прямой задачи внутренней баллистики (ПЗВБ) с учетом начального периода в области присоединённой камеры системы.

Воспламенение заряда присоединённой камеры производится горячими газами основного заряда. Достоинством такого способа воспламенения заряда присоединённой камеры является то, что обеспечивается стабильность начала

постепенного перетока инициирующих газов основного заряда в область заряда присоединенной камеры.

После воспламенения заряда присоединённой камеры и достижения перед поршнем давления более высокого, чем давление за поршнем, в канале системы возникает сложная волновая картина, которая приводит к перераспределению энергии заряда на повышение скорости метаемого элемента по сравнению с классической схемой метания [10, 11, 18, 24].

Система с одним присоединённым (дополнительным) зарядом к метаемому телу представлена на рисунке 2.2.

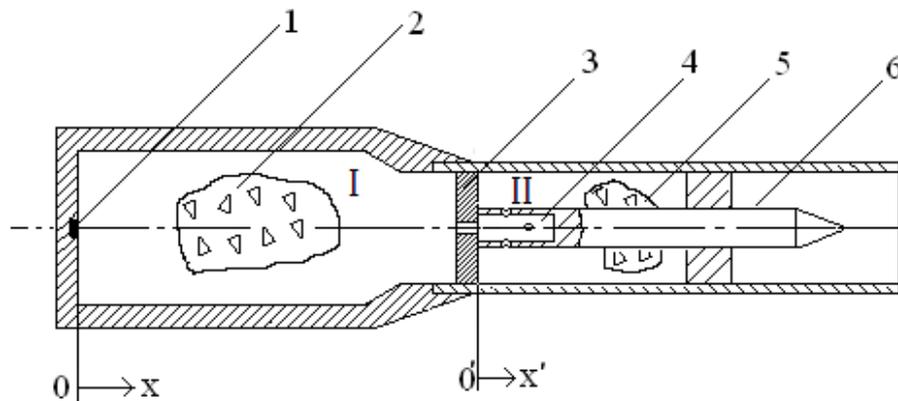


Рисунок 2.2 – Схема метания СПЗ: I–камера основного заряда; II–камера присоединённого заряда; 1– капсюль-воспламенитель; 2 – основной заряд; 3 – поршень с отверстием; 4 – регулирующий объём для перетока; 5 – присоединённый к метаемому телу заряд; 6 – метаемый элемент.

Поршень и метаемый элемент жёстко скреплены на сжатие и легко разделяются при растяжении. Будем называть область, где располагаются основной заряд и продукты его горения, областью 1, область между поршнем и метаемым элементом – областью 2.

2.2 Математическая модель схемы метания с присоединёнными зарядами на основе двухфазной модели гетерогенной среды

В соответствии с физической постановкой задачи данной в пункте 2.1.1 необходимо обеспечивать задержку зажигания присоединённого заряда после инициирования основного заряда системы для рассматриваемой схемы. Одним из достаточно простых и эффективных способов обеспечения задержки

зажигания является способ зажигания присоединенного заряда перетекающими продуктами горения основного заряда. Для обеспечения стабильного воспламенения присоединенного заряда с заданной задержкой зажигания необходимо проводить исследование влияния различных факторов на процессы функционирования нетрадиционных схем [46, 47].

Для описания процесса воспламенения присоединенного заряда [46, 47, 52] пороховой баллистической установки (ПБУ) горячими продуктами горения основного заряда будем использовать двухфазную двухскоростную модель [9, 10, 46, 47, 52] при следующих допущениях:

- движение фаз одномерно;
- пороховые частицы в среднем одинаковых размеров. Столкновениями между ними пренебрегаем;
- вне поверхностей разрыва расстояния, на которых характеристики течения меняются существенно, много больше расстояний между частицами;
- вязкость и теплопроводность существенны лишь в процессах взаимодействия фаз;
- теплоотдача и трение газа о стенки не учитываются;
- параметры газа внутри и вне пороховых элементов в данном сечении одинаковы;
- теплоотдачей на воспламенившуюся часть поверхности пренебрегаем;
- пороховые газы основного и присоединенного зарядов имеют одинаковую природу;
- частицы в воспламеняющем газе основного заряда отсутствуют;
- отверстие в поршне перекрыто диафрагмой, прорывающейся при достижении определенного давления со стороны основного заряда.

Движение двухфазной среды реагирующих частиц и газообразных продуктов реакции моделируется на основе уравнений механики гетерогенных сред [9, 10, 31, 46], записанных для среды, имеющей две фазы: 1 - газ, являющийся продуктом химической реакции на поверхности частиц

конденсированного вещества, 2 - реагирующие частицы конденсированного вещества. В уравнениях, описывающих процессы в такой среде, учитывается межфазный обмен массой, импульсом и энергией соответствующей записью источниковых членов в правых частях уравнений системы [10, 11, 18].

Будем называть область между неподвижной левой стенкой канала и поршнем областью I, а область между поршнем и снарядом - областью II. Система уравнений, описывающая процессы в области I, записывается в инерциальной системе координат (x, t) . Система уравнений, описывающая процессы в области II, записывается в неинерциальной системе координат, связанной с первым поршнем (x', t) . Ниже приводится система уравнений с начальными и граничными условиями, которая при $N=0$ описывает процессы в области I, а при $N = 1$ и замене переменных и параметров на переменные и параметры со штрихом «'» ($U \equiv U'$ и т. д.) описывает процессы в области II с момента времени t_1 , когда на поверхности всех частиц области II инициирована химическая реакция. Частицы конденсированной фазы предполагаются неподвижными относительно поршней до момента их разделения (до момента времени t_2).

«Математическое моделирование схемы с присоединённым зарядом [46, 47, 52] основано на подходе механики гетерогенных сред [9, 10], модифицированном для внутренней баллистики ствольных систем в [11, 24], и проводится при следующих основных допущениях:

- движение камеры с присоединённым зарядом (поршень ↔ присоединённый заряд ↔ метаемый элемент) начинается при достижении давления форсирования;
- начальный период для основного и присоединённого зарядов не учитывается;
- горение частиц пороха происходит по геометрическому закону;
- вязкость и теплопроводность существенны только в процессах взаимодействия фаз;

- до момента разделения сборки частицы присоединенного заряда неподвижны относительно сборки;
- после разделения сборки частицы присоединенного заряда могут выпадать на поршень; при равенстве пористости некоторой предельной величине вблизи поршня частицы начинают двигаться со скоростью поршня до тех пор, пока пористость не станет выше предельной;
- при движении элементов схемы метания (камеры с присоединенным зарядом, затем поршня и метаемого элемента) не учитываются трение и сопротивление воздуха в стволе;
- присоединенный заряд воспламеняется мгновенно в момент времени, соответствующий времени задержки зажигания;
- после воспламенения присоединенного заряда перетоки между областями I и II отсутствуют. [46]».

Для удобства расчета газодинамических параметров систему уравнений с учетом постепенного воспламенения заряда при принятых допущениях будем записывать в системе координат, связанной с контейнером присоединенного заряда. Балансовые соотношения для произвольного участка канала переменного сечения фиксированного объема V , заключенного между сечениями с координатами $\xi_1 = x'$, $\xi_2 = x''$ и ограниченного поверхностью Σ (рисунок 2.3), в любой момент времени $t > 0$ запишем с учетом обмена массой, импульсом, энергией в соответствии с рассматриваемым движением двух фаз: пороховых газов присоединенного заряда и газа – воспламенителя с одной стороны, пороховых частиц присоединенного заряда – с другой.

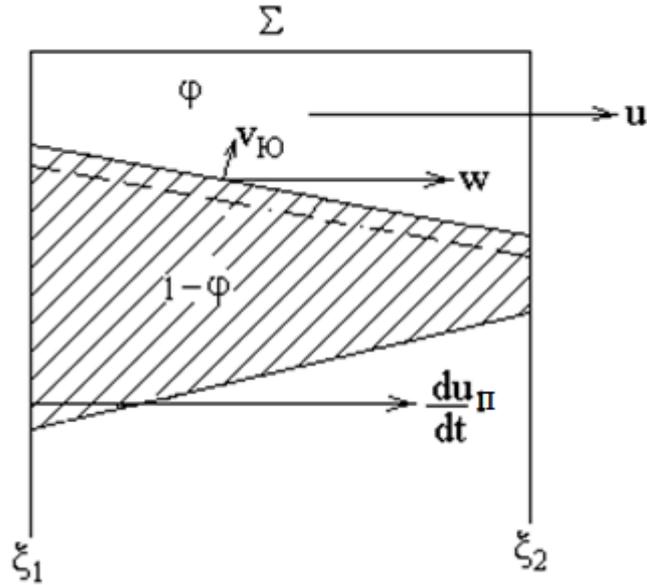


Рисунок 2.3 – Участок канала переменного сечения.

$$\frac{\partial}{\partial t}(\rho S \varphi) + \frac{\partial}{\partial x}(\rho S \varphi U) = M \quad ; \quad (2.1)$$

$$\frac{\partial}{\partial t}(\rho S \varphi U) + \frac{\partial}{\partial x}(\rho S \varphi U^2 + P S \varphi) = M \omega - \tau_{mp} + P \frac{\partial \varphi S}{\partial x} - N \rho \varphi S \frac{dU_{II}}{dt} \quad ; \quad (2.2)$$

$$\frac{\partial}{\partial t}(\rho S \varphi E) + \frac{\partial}{\partial x}(S \varphi U (\rho E + P)) = -P \frac{\partial(1-\varphi)S\omega}{\partial x} - \tau_{mp} \omega + M \left(Q + \frac{U^2}{2} \right) - N \rho \varphi S U \frac{dU_{II}}{dt} \quad ; \quad (2.3)$$

$$\frac{\partial}{\partial t}(\rho_2 S (1-\varphi)) + \frac{\partial}{\partial x}(\rho_2 S (1-\varphi) \omega) = -M \quad ; \quad (2.4)$$

$$\frac{\partial}{\partial t}(\rho_2 S (1-\varphi) \omega) + \frac{\partial}{\partial x}(\rho_2 S (1-\varphi) \omega^2) + (1-\varphi) S \frac{\partial P}{\partial x} = \tau_{mp} - M \omega - N \rho_2 (1-\varphi) S \frac{dU_{II}}{dt} \quad ; \quad (2.5)$$

$$\frac{\partial z}{\partial t} + \omega \frac{\partial z}{\partial x} = \frac{a_1 P}{e_a} \quad ; \quad (2.6)$$

$$P \left(\frac{1}{\rho} - \alpha \right) = RT \quad ; \quad (2.7)$$

$$E = \varepsilon + \frac{U^2}{2} \quad ;$$

$$\varphi = 1 - \Lambda_0 n (1 - \psi(z)) \quad ;$$

$$\psi(z) = \chi_1 z (1 + \lambda_1 z) \quad ;$$

Здесь: t – время; x – координата; u – скорость газа; ω – скорость твёрдой фазы; p – давление; ρ – плотность газа; ρ_2 – плотность вещества пороха; T – температура; φ – пористость; E, ε – полная и внутренняя энергия единицы объёма газа; S – площадь поперечного сечения канала; z – относительная толщина сгоревшего свода; M – скорость массоприхода от горения пороха; $\tau_{ГР}$ – сила взаимодействия между фазами; du_{II}/dt – ускорение поршня; Q – тепловой эффект горения пороха; R – универсальная газовая постоянная; α – коволюм; e_B – толщина горящего свода зерна пороха; a_1, ν – коэффициенты в законе скорости горения; n – концентрация; L_0 – начальный объём частицы пороха; $\Psi(z)$ – относительный сгоревший объём частицы пороха; κ_1, λ_1 – коэффициенты формы частиц пороха.

В системе уравнений (2.1)-(2.7) уравнение [10, 11]:

- (2.1) уравнение сохранения массы газа;
- (2.2) уравнение импульса газа;
- (2.3) уравнение энергии газа;
- (2.4) уравнение сохранения массы частиц, конденсированной фазы;
- (2.5) уравнение импульса к-фазы;
- (2.6) уравнение для относительной толщины прореагировавшего слоя частиц;
- (2.7) уравнение состояния.

Начальные условия для области I:

$$T(x, 0) = TG; \quad P(x, 0) = P_\phi; \quad U(x, 0) = \omega(x, 0) = 0;$$

$$\varphi(x, 0) = \varphi_H;$$

$$\psi_H = \frac{1/\Delta - 1/\rho_2}{f/P_\phi + \alpha - 1/\rho_2};$$

$$z_H = \frac{2\psi_H}{\chi_1(1 + \sigma_H)}; \tag{2.8}$$

$$\sigma_H = \sqrt{1 + 4 \frac{\lambda_1}{\chi_1} \psi_H};$$

Начальные условия для области II:

$$T'(x', t_1) = T_G'(x'); \quad P'(x', t_1) = P'(x'); \quad U'(x', t_1) = U'(x'); \quad (2.9)$$

$$\omega'(x', t_1) = 0; \quad \varphi'(x', t_1) = \varphi'(x'); \quad z'(x', t_1) = z'(x').$$

Граничные условия:

$$U(0, t) = \omega(0, t) = 0; \quad U(x_{II}, t) = \omega(x_{II}, t) = U_{II}; \quad (2.10)$$

$$U'(0, t) = \omega'(0, t) = 0; \quad U'(x_{II}', t) = \omega'(x_{II}', t) = U_{II}';$$

Значения переменных x_{II} , x_{II}' , U_{II} , U_{II}' определяются интегрированием уравнения движения поршней до и после момента времени t_2 (момента разделения поршней):

$$\left. \begin{aligned} Q \frac{dU_{II}}{dt} &= P_1 S \\ q'_{II} \frac{dU'_{II}}{dt} &= 0 \end{aligned} \right\}, t \leq t_2 \quad (2.11)$$

$$\left. \begin{aligned} q_{II} \frac{dU_{II}}{dt} &= (P_1 - P_2) S \\ q'_{II} \frac{dU'_{II}}{dt} &= P_3 S - q'_{II} \frac{dU_{II}}{dt} \end{aligned} \right\}, t > t_2 \quad (2.12)$$

где

P_1 - давление на поршень слева;

P_2 - давление на поршень справа;

P_3 - давление на снаряд слева.

Время t_2 соответствует моменту выполнения условия

$$P_3 > \frac{q'_{II}}{q_{II}} (P_1 - P_2). \quad (2.13)$$

Начальные условия для области II системы уравнений (2.1) - (2.12) определялись из решения задачи инициирования частиц конденсированной фазы потоком горячих газов из области I через регулирующий объём (рисунок 2.2). Предполагается, что входное отверстие I (рисунок 2.2) в регулирующий объём 2 вскрывается при достижении некоторого давления форсирования $P_{\phi 1}$, а выходные отверстия 3 - при достижении некоторого давления форсирования $P_{\phi 2}$. Ниже приведена система уравнений, записанная в неинерциальной

системе координат (x', t) , связанной с поршнем, которая описывает процессы в регулирующем объеме и области II до момента времени t_1 , когда на поверхности всех частиц области II инициирована химическая реакция. Система уравнений записана в системе координат (x', t) с началом на входе отверстия в регулирующей объем. Длина регулирующего объема l . По длине регулирующего объема расположены равномерно распределенные боковые отверстия, вскрываемые при давлении форсирования $P_{\phi 2}$. Частицы конденсированной фазы в области II до момента времени t_1 предполагаются неподвижными относительно поршней.

$$\frac{\partial}{\partial t}(\rho'S'\varphi') + \frac{\partial}{\partial x}(\rho'U'S'\varphi') = Me(t - t_c) \cdot e(x' - l) - Ge(l - x') ; \quad (2.15)$$

$$\frac{\partial}{\partial t}(\rho'S'\varphi'U') + \frac{\partial}{\partial x}(\rho'\varphi'U'^2S' + P'\varphi'S') = \left(-\tau_{mp} + P' \frac{\partial S'\varphi'}{\partial x'} \right) \cdot e(x' - l) - U'Ge(l - x') - \rho'S'\varphi' \frac{dU_{II}}{dt} ;$$

(2.16)

$$\frac{\partial}{\partial t}(\rho'\varphi'S'E') + \frac{\partial}{\partial x}(\rho'\varphi'S'U'E' + \varphi'U'S'P') = Q'Me(t - t_3) - q'_T e(t_3 - t) \cdot e(x' - l) - G \left(E' + \frac{P'}{\rho'} \right) \cdot e(l - x') - \rho'\varphi'U'S' \frac{dU_{II}}{dt} ; \quad (2.17)$$

$$\frac{dz'}{dt} = \frac{a'P'^v}{e'_g} e(t - t_g) \cdot e(x' - l) ; \quad (2.18)$$

$$P' \left(\frac{1}{\rho'} - \alpha \right) = R'T' \quad (2.19)$$

В системе уравнений (2.15)-(2.19) уравнение [9, 10, 11]:

- (2.15) уравнение сохранения массы газа;
- (2.16) уравнение импульса газа;
- (2.17) уравнение энергии газа;
- (2.18) уравнение для относительной толщины прореагировавшего слоя частиц;
- (2.19) уравнение состояния газа, $e(x)$ - функция Хевисайда, $x < 0$, $e(x) = 0$; $x' \geq 0$, $e(x) = 1$.

При $x' < l$ система уравнений описывает течение газа в реагирующем объёме с боковым отводом газа через отверстия, а при $l \leq x \leq L$ - описывает течение газа в области II, занятой зёрнами конденсированной фазы.

Начальные условия:

$$\begin{aligned}
 P'(x', 0) &= 1 \text{ атм}; & T'(x', 0) &= T_H; & U'(x', 0) &= 0; \\
 z'(x', 0) &= 0; & \varphi'(x', 0) &= \begin{cases} 1, 0 < x' < l, \\ \varphi'_H, l \leq x' \leq L \end{cases}
 \end{aligned} \tag{2.20}$$

Граничные условия: на границе справа:

$$U'(l, t) = 0; \quad U'(L, t) = 0.$$

На входе в реагирующий объём определяются потоки массы, импульса и энергии газа:

$$\begin{aligned}
 G' &= R' U' S'(0); \\
 I' &= (R' U'^2 + P') S'(0); \\
 \varepsilon &= U' (R' E' + P') S'(0);
 \end{aligned} \tag{2.21}$$

Расчет постепенного инициирования зёрен конденсированной фазы проводится на основе локальной твердофазной модели зажигания

$$c_2 \rho_2 \frac{\partial T_2}{\partial t} = \lambda_2 \frac{\partial^2 T}{\partial y^2} + Q n'_0 e^{-E/RT_2} \tag{2.22}$$

Краевые условия:

$$\begin{aligned}
 \lambda_2 \frac{\partial T_2(x', 0, t)}{\partial y} &= \alpha_T (T_I - T_{\bar{A}}); \\
 \frac{\partial T_2(x', \infty, t)}{\partial y} &= D; \\
 T_2(x', y, 0) &= T_H
 \end{aligned} \tag{2.23}$$

Химическая реакция на поверхности частицы считается начавшейся когда температура поверхности превысит некоторую критическую температуру T_z .

2.3 Численное решение задачи методом распада произвольного разрыва

Система уравнений (2.15)-(2.21) решается совместно с задачей зажигания (2.22)-(2.23), решаемой в каждой точке расчетной области и позволяет

определить картину постепенного инициирования конденсированной фазы в области II. После определения полей температуры, скорости, давления, относительной толщины прореагировавшего слоя частиц на момента времени t_1 , когда химическая реакция инициирована на всех зернах к-фазы, эта информация используется в качестве начальных условий для системы уравнений (2.1)-(2.7) начальные условия (2.9).

Системы уравнений (2.1)-(2.8) и (2.15)-(2.19) с соответствующими начальными и граничными условиями решалась численно методом С.К. Годунова [8], где потоки массы, импульса и энергии на границах расчетных ячеек в уравнениях (2.1)-(2.3) и (2.15)-(2.17) определялись из решения задачи о распаде произвольного разрыва параметров газа на скачке площади сечения [8]. В уравнениях (2.4)-(2.5) потоки массы и импульса находились из решения задачи о распаде произвольного разрыва для среды, не имеющей "собственного" давления [31]. Уравнение (2.6) решалось методом, аппроксимирующим конвективный член разностями против потока. Уравнения (2.11), (2.12) решались методом Эйлера, где давления P_1, P_2, P_3 определялись из решения задачи о распаде произвольного разрыва параметров газа на непроницаемой поверхности. В расчетной ячейке, прилегающей к подвижной границе параметры среды находились по методу С.К.Годунова, примененному для случая подвижных границ ячейки [8]. При увеличении последней ячейки до величины, большей $2h$ (где h - величина расчетной ячейки) проводится её разделение: выделяется целое число ячеек h и одна ячейка h_N , прилегающая к подвижной границе, удовлетворяющая условию $h \leq h_N < 2h$.

Расчетные области I, II и регулирующего объёма покрывается разностной сеткой с шагами h, h', h_{p0} соответственно. На начальном этапе при определении начальных условий (2.9) для уравнений (2.1)-(2.7) в области II решается совместно система уравнений (2.1)-(2.7) для области I (при $N=0$) и система уравнений (2.15)-(2.19); (2.22)-(2.23). Затем после инициирования всех частиц к-фазы в области II полученные на этот момент времени поля температуры, скорости газа, давления, относительной толщины

прореагировавшего свода частиц и пористости подставляются в (2.9) и становятся начальными условиями для уравнений (2.1-2.7) при $N=1$. После этого совместно решаются системы уравнений (2.1)-(2.7) при $N=0$ и при $N=1$. При этом величина шага в области II выбирается так, чтобы $h' \approx h$.

Разностные формулы для уравнений (2.1)-(2.5) и (2.15)-(2.17) записываются в виде:

$$A_i^{n+1} = A_i^n - \left(A_{i+1/2} - A_{i-1/2} \right) \frac{\tau}{h} + F_i \tau; \quad (2.24)$$

где

$$A_i = \begin{bmatrix} \rho S \varphi \\ \rho S \varphi U \\ \rho S \varphi E \\ \rho_2 S (1 - \varphi) \\ \rho_2 S (1 - \varphi) \omega \end{bmatrix};$$

$$A_{i+1/2} = \begin{bmatrix} R_L S_i \varphi_i U_L \\ S_i \varphi_i (R_L U_L^2 + P_L) \\ S_i \varphi_i U_L \left(R_L P_L \left(\frac{1}{R_L} - \alpha \right) \frac{1}{k-1} + R_L \frac{U_L^2}{2} + P_L \right) \\ (R_2 W_2 S)_{i+1/2} \\ (R_2 W_2^2 S)_{i+1/2} \end{bmatrix}; \quad (2.25)$$

$$A_{i-1/2} = \begin{bmatrix} R_p S_i \varphi_i U_p \\ S_i \varphi_i (R_p U_p^2 + P_p) \\ S_i \varphi_i U_p \left(R_p P_p \left(\frac{1}{R_p} - \alpha \right) \frac{1}{k-1} + R_p \frac{U_p^2}{2} + P_p \right) \\ (R_2 W_2 S)_{i-1/2} \\ (R_2 W_2^2 S)_{i-1/2} \end{bmatrix}$$

где

τ - величина шага по времени;

F_i - столбец правых частей уравнений (2.1)-(2.5) и (2.15)-(2.17);

$P_L, P_p, U_L, U_p, R_L, R_p$ - параметры газа справа и слева на скачке сечения после распада произвольного разрыва параметров газа на скачке сечения;

R_2 , W_2 - параметры к-фазы после распада произвольного разрыва для среды частиц.

Для уравнений (2.15)-(2.17) в выражениях для A и отсутствуют две последние строки.

Уравнение (2.6) в разностном виде записывается следующим образом:

$$z_i^{n+1} = z_i^n - \frac{\tau}{2h} \left((\omega_i^n + |\omega_i^n|)(z_i^n - z_{i-1}^n) + (\omega_i^n - |\omega_i^n|)(z_{i+1}^n - z_i^n) + F_{zi}^n \cdot \tau \right)$$

После определения A_i^{n+1} определяются параметры течения в ячейке на новом временном слое:

$$\begin{aligned} \varphi_i^{n+1} &= 1 - \frac{(\rho_2 S(1-\varphi))_i^{n+1}}{\rho_2 S}; \\ \rho_i^{n+1} &= \frac{(\rho S \varphi)_i^{n+1}}{S_i \varphi_i^{n+1}}; \\ U_i^{n+1} &= \frac{(\rho S \varphi U)_i^{n+1}}{\rho_i^{n+1} S_i \varphi_i^{n+1}}; \\ E_i^{n+1} &= \frac{(\rho E S \varphi)_i^{n+1}}{\rho_i^{n+1} S_i \varphi_i^{n+1}}; \\ P_i^{n+1} &= \left(E_i^{n+1} - \frac{U_i^{n+1}}{2} \right) \cdot \frac{k-1}{1/\rho_i^{n+1} - \alpha}; \\ \omega_i^{n+1} &= \frac{(\rho_2 \omega S(1-\varphi))_i^{n+1}}{(\rho_2 S(1-\varphi))_i^{n+1}} \end{aligned} \quad (2.27)$$

Для уравнений (2.11)-(2.12) разностные формулы записываются в виде:

$$\left. \begin{aligned} U_i^{n+1} &= U_i^n + \frac{S}{Q} P_1 \tau \\ U'_s &= 0 \end{aligned} \right\}, t \leq t_2 \quad (2.28)$$

$$\left. \begin{aligned} U_i^{n+1} &= U_i^n + \frac{S}{q_i} (P_1 - P_2) \tau \\ U'_s &= U_s^n + \frac{S}{q_s} P_3 \tau - (U_i^{n+1} - U_i^n) \end{aligned} \right\}, t > t_2 \quad (2.29)$$

Задача инициирования химической реакции на поверхности частиц (2.22)-(2.23) решалась по неявной разностной схеме на неравномерной сетке, сгущающейся к поверхности частицы, методом прогонки. Разностные формулы записываются в виде:

$$\frac{T_{ji}^{n+1} - T_{ji}^n}{\tau} = \chi_2 \frac{(T_{ji+1}^{n+1} - T_{ji}^{n+1})/h_i - (T_{ji}^{n+1} - T_{ji-1}^{n+1})/h_{i-1}}{(h_i + h_{i-1})/2} + \frac{Q_z}{c_2 \rho_2} e^{\frac{-E}{RT_j^n}} \quad (2.30)$$

Граничные условия (2.23) аппроксимировались со вторым порядком точности по координате.

Величина шага по времени определяется из условий устойчивости Куранта-Фридрихса-Леви и устойчивости по правым частям для уравнений (2.2) (при $N=0$ и при $N=1$) и (2.16). Для счета шага задачи по времени выбирался максимальный шаг по времени, удовлетворяющий всем этим условиям устойчивости.

Геометрия канала задается таблично: задаются координаты оснований цилиндрических и конических частей канала, отсчитываемые от левой границы канала ($x=0$) и соответствующие диаметры оснований этих частей.

Все исходные данные задаются в системе единиц ДЕЦИМЕТР-КИЛОГРАММ-СЕКУНДА. В программе предусмотрена распечатка всех исходных данных для визуального контроля введенных данных с соответствующими комментариями и подписью размерности, в которой следует вводить исходные данные.

Расчет задачи проводится для уравнений, записанных в безразмерной форме. В качестве масштабов приняты следующие величины: масштаб плотности R^* - объемная плотность вещества в области I, масштаб давления P^* - максимальное давление в постоянном объеме после прохождения всей реакции при плотности R^* , масштаб скорости $U^*=(P^*/R^*)^{0,5}$, масштаб площади $S^*=\pi D^2/4$, масштаб длины L^* - расстояние между левой границей канала и дном первого поршня.

Счет шага задачи по времени складывается из следующей последовательности действий:

- определяется из условий устойчивости величина шага по времени;
- проводится расчет параметров течения среды во внутренних ячейках расчетной области I, (2.24)-(2.27);
- проводится расчет движения первого поршня в соответствии с (2.28)-(2.29) в зависимости от величины t ;
- проводится расчет параметров течения среды в ячейках, прилегающей к подвижной правой границе расчетной области I;
- если вскрылось входное отверстие в регулирующий объём, проводится расчет параметров течения газа в регулирующем объёме, (2.15)-(2.19);
- если вскрылись выходные отверстия регулирующего объёма, проводится расчет параметров течения газа в области II с учетом постепенного инициирования химической реакции на поверхности частиц к-фазы (2.15)-(2.19), (2.22)-(2.23).
- если произошло инициирование химической реакции на поверхности всех частиц к-фазы, то выполняются вместо двух предыдущих пунктов три следующих;
- проводится расчет параметров течения среды во внутренних ячейках расчетной области II, (2.24)-(2.27);
- проводится расчет движения второго поршня в соответствии с (2.28) или (2.29) в зависимости от величины времени t ;
- проводится расчет параметров течения среды в ячейке, прилегающей к подвижной границе расчетной области II;
- если выполнилось условие вывода на печать, то проводится вывод на печать результатов расчета на этом временном слое.

Таким образом получается решение на следующем временном слое через интервал времени τ . Циклическое повторение счета шага по времени в соответствии с вышеописанной последовательностью действий дает информацию о нестационарном развитии процесса во времени.

В основе численного интегрирования приведенной выше системы уравнений методом Годунова положено использование консервативной

разностной схемы и соотношений распада произвольного разрыва. В соответствии с методом «распад разрыва» параметры газа в канале распределяются в виде кусочно-постоянных функций с разрывами на границах ячеек. Изменение параметров газа в $(j+1/2)$ -ой ячейке на новом временном слое $(j+1)$ определяется потоками массы M^* , M_{j+1} , импульса I_j , и энергии в Γ_{j+1} через граничные сечения ячейки (узлы сетки) с координатами x_j , x_{j+1} , а также влиянием сил трения и теплоотдачи. Потоки отмеченных субстанций в пределах расчетного шага Δt образуются в результате распада разрыва параметров на границах соседних ячеек [8].

В результате взаимодействия двух сред с различным состоянием образуется возмущенная область, которая с каждой стороны отделена от сред с исходным состоянием ударной волной или волной разрежения. В области распада слева и справа от контактной поверхности (условной границы раздела двух исходных сред) устанавливаются постоянные значения скорости U и давления P . В то же время плотность газа на контактной поверхности претерпевает разрыв. Поэтому в сечении узла плотность газа зависит от направления скорости U в области распада [8].

Для определения плотности газа в узле можно применить следующее правило: если в рассматриваемом узле скорость среды направлена в рассчитываемую ячейку, то плотность соответствует состоянию распада за контактной поверхностью по направлению ее перемещения.

Итерационный процесс определения параметров фазы в области распада разрыва конструируется следующим образом:

1. Задается (или вычисляется) предварительное значение давления в области распада.
2. Вычисляются значения массовой скорости для левой и правой волн возмущения.
3. Вычисляется новое приближение давления в области распада.
4. После достижения требуемой сходимости значений определяется скорость газа в области распада разрыва.

5. Вычисляется плотность газа в сечении узла.

Приведенная схема используется при определении параметров распада разрыва в каждом из внутренних узлов расчетной сетки газообменного канала [8].

Затем по формулам вычисляются потоки массы, импульса и энергии через узловые сечения и с помощью разностных уравнений производится осреднение газодинамических параметров в пределах каждой расчетной ячейки.

Универсальность метода «распад разрыва» обусловлена выполнением законов сохранения при ведении сквозного счета как на протяженных участках, содержащих разрывные течения (ударные волны и контактные поверхности), так и в узлах местных сопротивлений. Здесь, в силу единообразия расчетных формул, течение в гладкой области рассчитывается, как и в окрестности разрыва. Схема предполагает наличие разрыва в каждом расчетном интервале, что приводит к появлению аппроксимационной вязкости. При этом отпадает необходимость специального введения вязкости в уравнение газовой динамики. Разрывы параметров хотя и размазываются по ячейкам расчетной сетки, но их положение и амплитуда определяются достаточно точно.

При наличии в потоке больших деформаций в ряде случаев, в том числе при моделировании процессов в напоробменных каналах ротора волнового обменника давления, возникает необходимость более точного описания конфигурации разрыва. Существует несколько специальных способов расчетного выделения скачков газодинамических параметров, один из которых, разработанный и апробированный авторами, изложен в последующих разделах.

Теория метода «распад произвольного разрыва» хорошо отражает сущность процессов в граничных сечениях протяженных участков и часто используется при моделировании течений в клапанной щели и местах сопряжения каналов с различными поперечными сечениями. В рамках одномерной модели точное описание пространственной картины течения в этих элементах не представляется возможным. При этом для выполнения

интегральных законов сохранения необходим учет местных сопротивлений. Основная составляющая местных потерь связана с образованием отрывных зон, условные размеры которых устанавливаются на основании опытных данных. Однако использование эмпирических зависимостей заметно снижает универсальность задания граничных условий. В этой связи заслуживают внимания попытки аналитического определения параметров газа на участках отрывных течений, основанные на использовании соотношений распада разрыва [8].

2.4 Обоснование адекватности модели двухфазной двухскоростной среды с присоединенными зарядами

Для успешного использования математической модели необходимо, чтобы модель достаточно верно описывала качественно и количественно свойства объекта моделирования, т. е. она должна быть адекватна моделируемому объекту. Для проверки адекватности математической модели реальному процессу нужно сравнить результаты измерения на процессе с результатами предсказания модели в идентичных условиях (при определенных значениях входных и управляющих параметров) и подстроить параметры математического описания объекта. Такая проверка позволяет оценить точность математической модели и, следовательно, возможность ее применения для решения различных задач [27].

В данной работе была исследована математическая модель двухфазной двухскоростной среды с присоединенными зарядами для расчета параметров баллистических систем. Можно выделить две составляющие (подмодели):

- 1) Модель баллистической системы с присоединенными зарядами;
- 2) Модель двухфазной двухскоростной среды для расчета параметров баллистических систем.

Для каждой подмодели сформулированы обоснования ее адекватности реальным системам, исходя из данных реальных экспериментов или статистических анализов.

Обоснование адекватности модели баллистической системы с присоединенными зарядами:

1. Была выдвинута идея, что при разделении заряда и организации задержки зажигания, которая примыкает к снаряду, получается прирост скорости метаемого элемента по сравнению с классической схемой метания [24, 46, 47];

2. Проведенные численные расчеты в [24, 46, 47, 52], показали, что при использовании разделения заряда, реализуется значительное ускорение метаемого элемента (до 30%, подробнее в п. 4.1).

3. Расчеты с помощью двухфазной двухскоростной модели согласуются с результатами реального эксперимента (подробнее п. 4.2)

Обоснование адекватности модели двухфазной двухскоростной среды:

- Модель адаптирована для артиллерийских систем [9, 10, 24].
- Модель с подходом гетерогенных сред (в данном случае двухфазная двухскоростная) дает результаты, наилучше согласующиеся с экспериментальными данными [46, 47, 45, 52, 53];
- С одной областью двухфазная двухскоростная модель полностью адекватна реальным экспериментам [18, 21].

В результате можно сделать вывод что, модель двухфазной двухскоростной среды с присоединенными зарядами адекватна реальной системе.

ГЛАВА 3. РЕАЛИЗАЦИЯ КОМПЬЮТЕРНОЙ МОДЕЛИ

В данной главе описываются алгоритмы и функциональность компьютерной модели, предназначенной для расчёта динамики высокоэнергетических процессов, протекающих в камере и стволе при срабатывании артиллерийского орудия.

При написании главы использовались материалы из [8, 12, 14, 15, 17, 19, 21-28, 29, 30, 34-39, 41 и др.].

3.1 Описание функциональности компьютерной модели

Математическая модель, описанная в п. 2.2, преобразуется в алгоритмическую модель, которая в свою очередь реализуется в программный код. Из функций и процедур исходного кода формируется класс вычислений. Класс вычислений принимает входные параметры, на основе которых будут производиться расчеты. Входные параметры, которые были введены пользователем или загружены из файла, проверяются на корректность (поиск ошибок при вводе) и совместно с классом вычислений передаются в созданный головной программой поток (п. 1.3.2). Поток запускает расчетный алгоритм класса вычислений (если потоков несколько они запускаются параллельно) и после завершения возвращает результаты в качестве массивов данных. Результаты хранятся в системе в рамках запущенного сеанса, если они не были сохранены в файл. На основе полуденных числовых результатов программа строит графики и находит максимальные выходные параметры в массивах (Максимальное давление, скорость).

На рисунке 3.1 продемонстрирована функциональная схема информационной модели заложенной в приложение. На ней показаны основные функции информационной модели и их взаимодействия между собой. Про режимы параметризации подробнее расписано п 3.2.2.

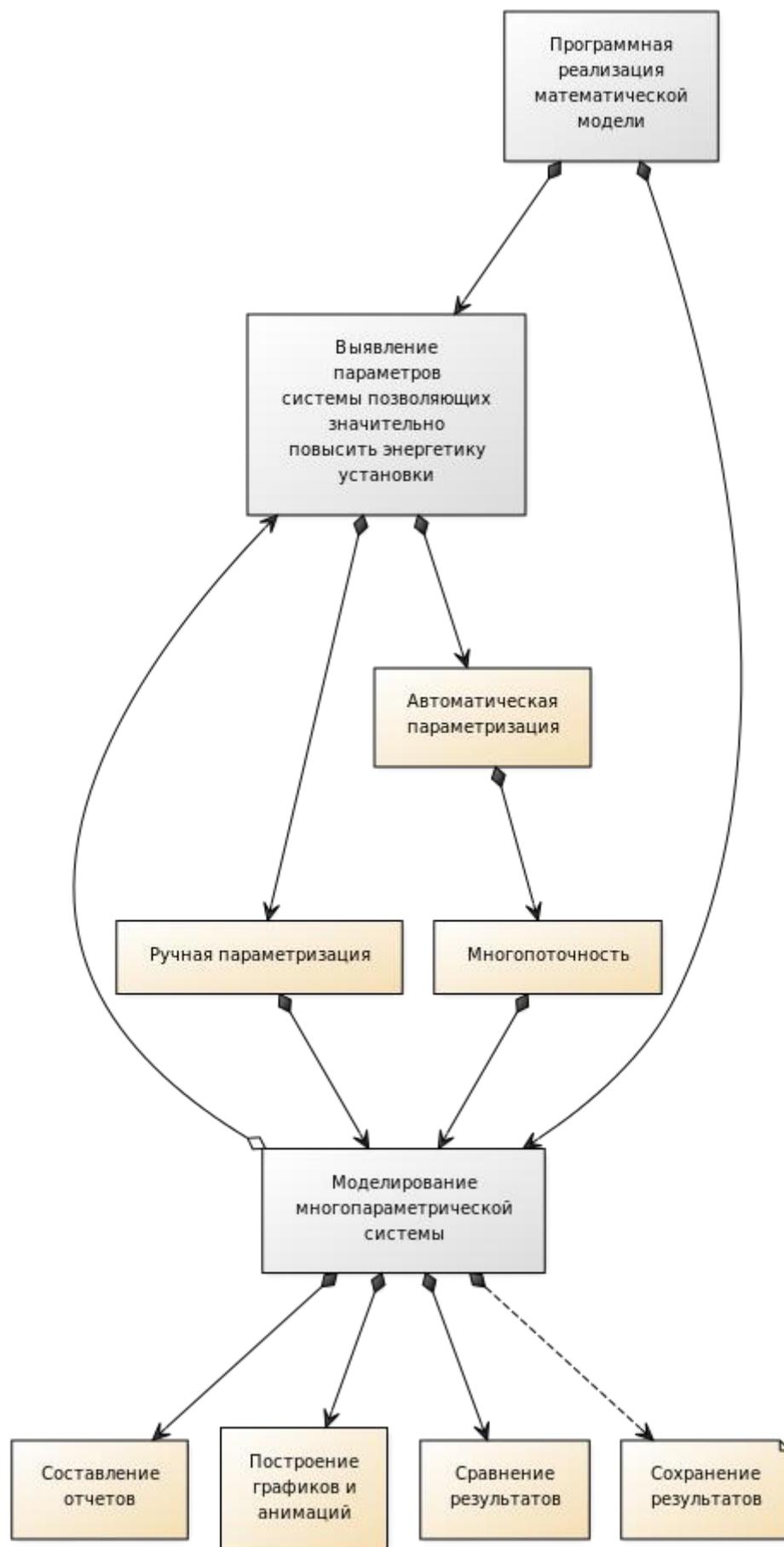


Рисунок 3.1 – Функциональная схема программного продукта

На рисунке 3.2 продемонстрирована контекстная диаграмма информационной модели заложенной в приложение. На ней показаны элементы информационной модели и их типы информации для взаимодействия между собой.

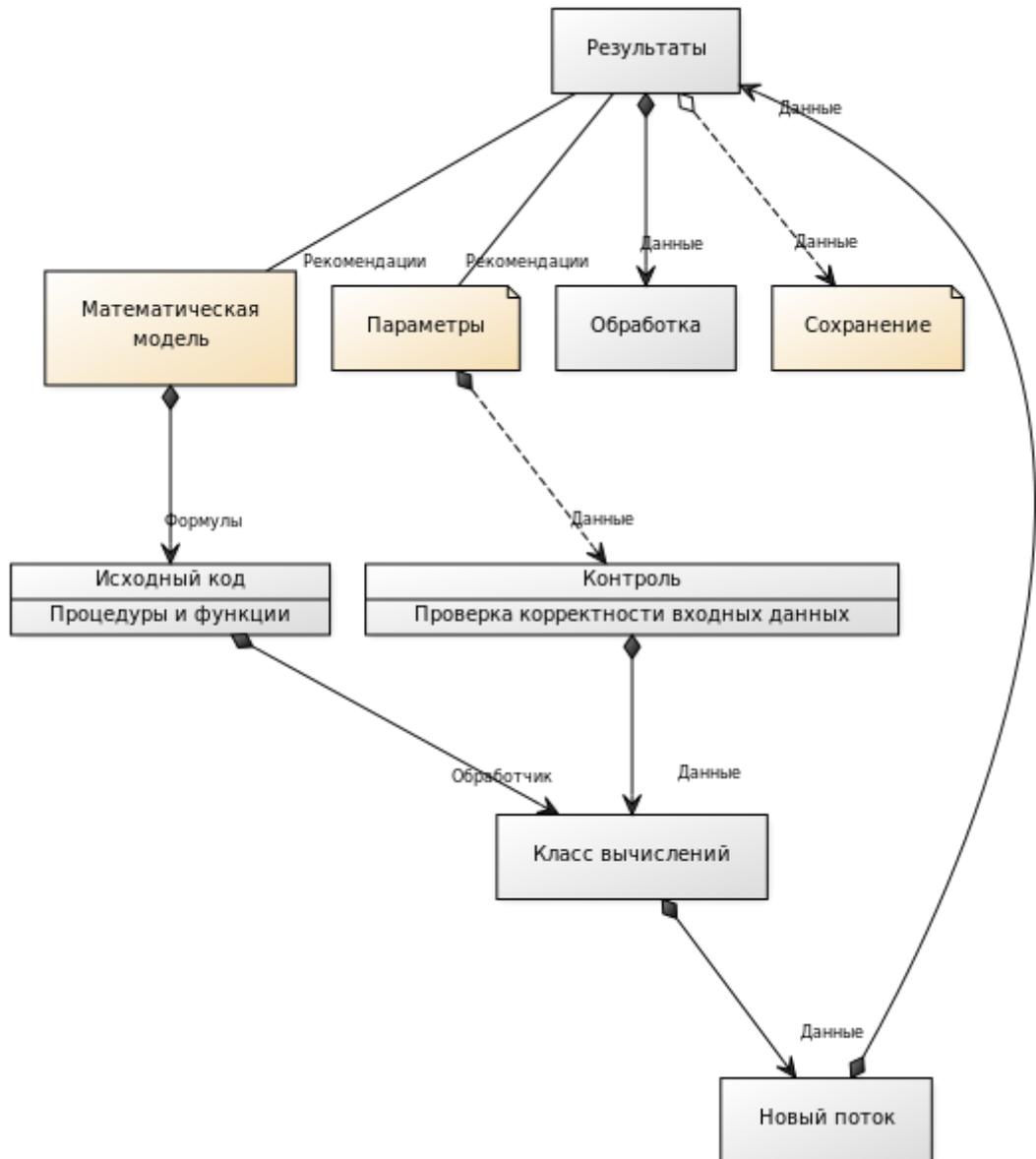


Рисунок 3.2 – Контекстная диаграмма программного продукта

3.2 Разработка алгоритмов компьютерной модели

Основной алгоритм работы приложения выглядит следующим образом:

- 1) Запуск программы;
- 2) Выбор действия;
 - a. заполнение входных данных;
 - b. загрузка входных данных;
 - c. загрузка выходных данных;
- 3) Выбор действия исходя из выбора предыдущего пункта;
 - a. расчеты на основе введенных или загруженных входных данных;
 - b. автоматический подбор на основе введенных входных данных и приращений для параметров, если их ввел пользователь;
- 4) построение графиков;
- 5) сохранение данных файл по усмотрению пользователя;
- 6) просмотр всех расчетов в терминале.

Алгоритм разработан в соответствии с рисунком 3.1, отражающего функциональную схему приложения [15].

3.2.1 Расчет параметров системы на временном слое

Алгоритм расчета параметров системы основан на решении разностных уравнений механики сплошных сред двухфазной смеси методом Годунова (распад произвольного разрыва, основанный на методе сеток) [8].

«Идея метода конечных разностей (метода сеток) [30] известна давно, с соответствующих трудов Эйлера. Однако практическое применение этого метода было тогда весьма ограничено из-за огромного объема ручных вычислений, связанных с размерностью получаемых систем алгебраических уравнений, на решение которых требовались годы. В настоящее время, с появлением быстродействующих компьютеров, ситуация в корне изменилась. Этот метод стал удобен для практического использования и является одним из наиболее эффективных при решении различных задач математической физики. [30]».

Основная идея *метода конечных разностей (метода сеток)* для приближенного численного решения краевой задачи для двумерного дифференциального уравнения в частных производных состоит в том, что

- 1) на плоскости в области A , в которой ищется решение, строится *сеточная область* A_s (рисунок 3.3), состоящая из одинаковых ячеек размером s (s – шаг сетки) и являющаяся приближением данной области A ;
- 2) заданное дифференциальное уравнение в частных производных заменяется в узлах сетки A_s соответствующим конечно-разностным уравнением;
- 3) с учетом граничных условий устанавливаются значения искомого решения в граничных узлах области A_s .

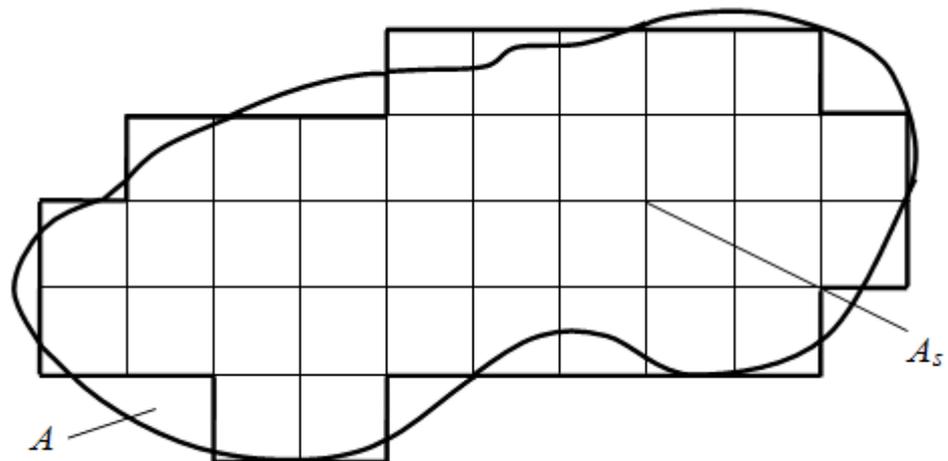


Рисунок 3.3 – Построение сеточной области

«Решая полученную систему конечно-разностных алгебраических уравнений, получим значения искомой функции в узлах сетки A_s , т.е. приближенное численное решение краевой задачи. Выбор сеточной области A_s зависит от конкретной задачи, но всегда надо стремиться к тому, чтобы контур сеточной области A_s наилучшим образом аппроксимировал контур области A . [30]».

Величина шага по времени определяется из условий устойчивости Куранта-Фридрихса-Леви.

Алгоритм вычислений параметров задачи на каждом временном слое складывается из следующей последовательности действий (Рисунок 3.4):

- 1) определяется из условий устойчивости величина шага по времени;
- 2) проводится расчет параметров течения среды во внутренних ячейках расчетной области 1;
- 3) проводится расчет движения поршня в зависимости от величины t ;
- 4) проводится расчет параметров течения среды в ячейках, прилегающей к подвижной правой границе расчетной области 1;
- 5) проверка на вскрытие входного отверстия в регулирующем объеме;
- 6) если вскрылось входное отверстие в регулирующей объем, проводится расчет параметров течения газа в регулирующем объеме;
- 7) если вскрылись выходные отверстия регулирующего объема, проводится расчет параметров течения газа в области 2 с учетом постепенного инициирования химической реакции на поверхности частиц k -фазы;
- 8) если произошло инициирование химической реакции на поверхности всех частиц k -фазы, то выполняются вместо двух предыдущих пунктов три следующих;
- 9) проводится расчет параметров течения среды во внутренних ячейках расчетной области 2;
- 10) проводится расчет движения снаряда в зависимости от величины времени t ;
- 11) проводится расчет параметров течения среды в ячейке, прилегающей к подвижной границе расчетной области 2;
- 12) если выполнилось условие вывода информации, то проводится вывод результатов расчета на этом временном слое.

Получается решение на каждом временном слое через интервал времени τ . Циклическое повторение счета шага по времени в соответствии с вышеописанной последовательностью действий дает информацию о нестационарном развитии процесса во времени.

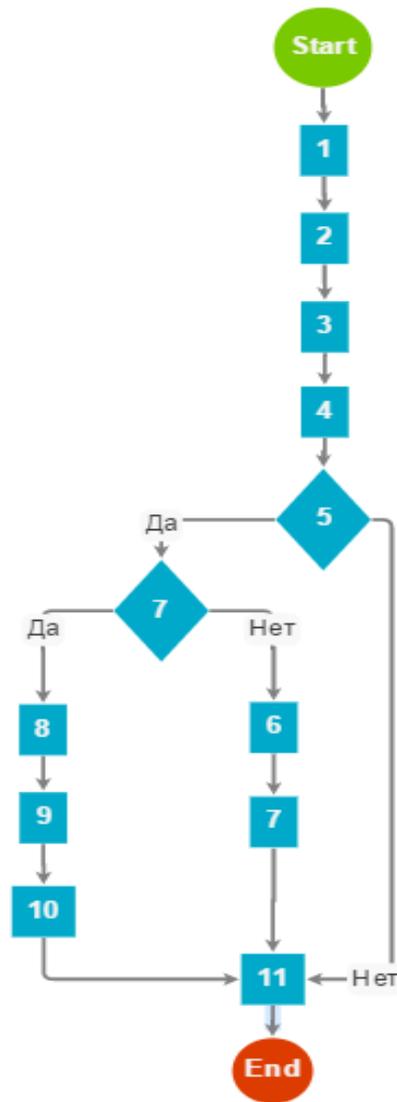


Рисунок 3.4 – Блок-схема алгоритма расчета параметров на временном слое

3.2.2 Подбор входных параметров с использованием алгоритмов распараллеливания

При нахождении наилучших параметров системы для снижения времени вычисления применялось распараллеливание алгоритма [26, 28, 39, 40, 19].

Распараллеливание в компьютерной модели используется для ускорения нахождения наилучших параметров системы, за счет массового запуска расчетов в параллельных потоках с заданным приращением для параметров.

В программе присутствуют 2 режима расчетов:

1. Единичный расчет;

2. Многопоточковый расчет.

В режиме единичного вычисления пользователь может вручную произвести 10 вычислений прибавляя, например, к весу 1 миллиграмм, что повлечет временные затраты на заполнение параметров и на единичные вычисления. В режиме многопоточности пользователь может воспользоваться механизмом массового подбора. Например, задав приращение в 1 миллиграмм к весу и запустить многопоточность. Это сократит временные затраты на подбор входных параметров.

Алгоритм работы режима многопоточности заключается в создании параллельных потоков с приращениями параметров, пока количество вычислений системы не достигнет заданного значения итераций. Потоки создаются исходя из возможностей процессора. За все время многопоточного режима, процессор выполняет максимальное количество потоков, как только один заканчивает вычисления, программа создает новый и запускает его параллельно остальным. Все потоки закончившие расчеты высвобождают память, при этом сохраняют выходные данные.

3.3 Разработка программного приложения

Программный продукт был реализован на языке программирования C#, в среде Microsoft Visual Studio 2010 [35].

C# (произносится "Си-шарп") является языком программирования, который разработан для создания множества приложений, работающих в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую C-подобным языкам [37].

Visual C# — это реализация языка C# корпорацией Майкрософт. Поддержка Visual C# в Visual Studio обеспечивается с помощью полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим

службам операционной системы и к другим полезным, хорошо спроектированным классам, что существенно ускоряет цикл разработки [37].

Microsoft Visual Studio — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Эти инструменты предназначены для максимально эффективной совместной работы; все они доступны в интегрированной среде разработки (IDE) Visual Studio [35].

Обоснование выбора Microsoft Visual Studio [35] заключается в следующих пунктах:

- Мощная платформа для разработки мультимедийных приложений;
- Высокая точность расчетов;
- Высокая скорость обработки данных;
- Доступность использования;
- Огромно количество сторонних библиотек;
- Наличие функций распараллеливания алгоритмов;

3.3.1 Структура головного модуля

В теле основной программы описаны методы для создания новых экземпляров класса CalculateDD (описание в пункте 3.3.5), заполнения его входных параметров и обработки полученных от него массивов данных. При автоматическом подборе входных параметров (для нахождения наилучших выходных данных) каждый экземпляр класса CalculateDD создается в отдельном потоке с использованием технологий параллельного вычисления [41, 40, 39, 34], что ускоряет время нахождения наилучших входных и следовательно выходных параметров системы. Также при автоматическом подборе в экземпляр класса передаются не только входные данные, но и массив приращений (введенные пользователем) для входных данных, за счет чего и реализуется подбор.

Контроль корректности заполнения входной информации требует особого внимания, так как ошибки при вводе могут привести к аварийным ситуациям (критическим ошибкам) при счете, хотя обработчик ошибок их уловит и в терминале отразит статус вычисления «неудачно».

3.3.2 Описание переменных головного модуля

Int maxThread – максимальное число потоков за подбор, начальное значение maxThread = 100;

Int threadCount – число потоков в данном сеансе, начальное значение threadCount = 1;

Int calcCount – число вычислений проделанных в данном сеансе, начальное значение calcCount = 0;

DataTable DataTableinitialData, outputData, terminalData – таблицы данных для отображения входных, выходных и данных терминала;

CalculateDD currentCalculateDD – текущий выбранный класс CalculateDD;

ThreadCalculate[] arrayThreadCalculateDD– массив хранящий экземпляры класса ThreadCalculate.

3.3.3 Описание методов головного модуля

Метод «calculateThread»

Метод предназначен для создания новых экземпляров класса CalculateDD и исполнения их в отдельных параллельных потоках. Метод отслеживает выполнения всех созданных им потоков и отражает информацию о статусе их выполнения в терминале.

Если расчет проводится в режиме единичного вычисления, то создается один поток. После окончания выполнения поток уничтожается а выходные данные сохраняются.

Входные параметры:

Int index – индекс последнего запущенного потока в текущем сеансе.

Int iterator – итератор для приращений параметров.

Object lockOut – объект определяющий доступность терминала для записи в него результатов вычислений потоков. Если один поток заблокировал терминал, следующие закончившие вычисления потоки ждут своей очереди на запись данных в терминал

Метод «setCurrentCalculate»

Метод предназначен для выбора класса из созданных экземпляров класса CalculateDD и вывода его выходных данных в таблицы initialData и outputData. Выбор экземпляра осуществляется во вкладке «Терминал».

Входные параметры:

Int index – индекс класса в массиве классов CalculateDD.

Метод «dataGridToHTML»

Метод предназначен для сохранения данных в HTML формат отображаемых в элементах dataGrid на главной форме, данными для преобразования служат таблицы initialData и outputData.

Входные параметры:

DataTable DataTableinitialData, outputData, terminalData – таблицы данных для отображения входных, выходных и данных терминала;

Метод «saveProjectToFile»

Метод предназначен для сохранения текущего проекта в файл. Сохраняются данные всех вкладок. При открытии проекта восстанавливаются значения терминала, и можно обратиться к любому экземпляру вычислений.

Метод «saveXmlPicToFile»

Метод предназначен для сохранения данных таблиц initialData, outputData, terminalData в XML формат если открыты соответствующие вкладки, если открыта вкладка «Графики» то последует сохранение изображения в файл.

Входные параметры:

DataTable DataTableinitialData, outputData, terminalData – таблицы данных для отображения входных, выходных и данных терминала;

Boolean BooleanthisPic – булева переменная, если истина то сохраняем XML, иначе сохраняем изображение.

Метод «openXmlFromFile»

Метод предназначен для загрузки данных из XML файла в таблицы initialData, outputData, terminalData.

Метод «openProjectFromFile»

Метод предназначен для загрузки проекта из файла.

Метод «chartBuildF»

Метод предназначен для построения графиков на основе данных текущего экземпляра класса CalculateDD.

Входные параметры:

CalculateDD CurrentCalculateDD – текущий экземпляр класса.

3.3.4 Структура класса ThreadCalculate

Класс хранит информацию о всех проведенных вычислениях.

Переменные класса:

CalculateDD _Calculate – экземпляр класса вычислений;

Thread _Thread – поток в котором проводятся вычисления (создается и освобождается);

double _time – время затраченной потоком на вычисления;

int _check – отображает состояние вычислений (вычисляет, закончил, ошибка);

int _index – номер экземпляра класса;

3.3.5 Структура класса CalculateDD

Класс основных расчетов CalculateDD состоит из головного метода startCalculate() и множества подметодов, из которых можно выделить три основных:

- timeStep(...);
- calcParamsOfGas(...);
- output(...).

В головном методе проводится заполнение входной информации, вывод исходной информации для визуального контроля её корректности, подготовка необходимых исходных данных и, если проводится расчет с записью промежуточных данных в массивы.

Все исходные данные заполняются в системе единиц ДЕЦИМЕТР-КИЛОГРАММ-СЕКУНДА. Все координаты при вводе геометрии канала отсчитываются от левой границы канала.

В головном методе также организован счет шага решения задачи с обращением к необходимым для этого методам. Далее будут описаны основные методы и переменные входящие в состав класса CalculateDD.

Метод «timeStep»

Метод предназначен для расчета параметров течения среды во внутренних ячейках расчетной области, расчета движения поршня и расчета параметров течения среды в ячейках, прилегающей к правой подвижной границе расчетной области.

Входные параметры:

R - массив плотности газовой фазы;

P - массив давления газа;

U - массив скорости газа;

W - массив скорости частиц;

FI - массив пористости;

Z - массив относительной толщины прореагировавшего свода частицы;

S - массив площадей поперечного сечения канала в центрах ячеек;

SV - массив площадей поперечного сечения канала на границах ячеек;

COVOLUME - коволюм;

POLYTROP - показатель политропы;

τ - величина шага по пространству;

AKP, ALP, AMP - коэффициенты формы частицы;

EB - начальная толщина свода частицы;

AFL - коэффициент в законе скорости массоприхода от химической реакции;

W0 - начальный объём частицы;

S02 - начальная площадь поверхности частицы;

QP - тепловой эффект от химической реакции;

XS - координата положения левой границы поршня;

US - скорость поршня на n -том временном слое;

USN- скорость поршня на $n+1$ -ом временном слое;

ACSYS - ускорение системы координат; в области 1 ACSYS = 0;

APCH1 - знаменатель в условии устойчивости по правым частям;

N - количество ячеек в расчетной области;

KMOV_2- признак движения частиц к-фазы в области 2;

KMOV - признак движения частиц к-фазы в области 1;

P2 - давление на поршень справа;

QFORCER - масса поршня.

Расчет параметров течения среды и движения поршней проводится по математической модели и методике, изложенной в разделе 2. Подпрограмма рассчитывает параметры течения среды в инерциальной системе координат для области 1 (при ACSYS = 0) и в неинерциальной системе координат для области 2 (при ACSYS=0).

Метод «output»

Метод предназначен для вывода получаемой в ходе расчетов информации.

Входные данные:

RGC - газовая постоянная для газа области 1;

RGC2 - газовая постоянная для газа области 2;

COVOLUME - коволюм газа области 1;

COVOLUME_2 - коволюм газа области 2;

TIM - время;

τ - шаг по пространству в области 1;

H2 - шаг по пространству в области 2;
 POLYTROP - показатель политропы газа области 1;
 QFORCER - масса поршня;
 POLYTROP_2 - показатель политропы газа области 2;
 RP - плотность к-фазы;
 S - массив площадей поперечного сечения канала в области 1;
 S2 - массив площадей поперечного сечения канала в области 2;
 US - скорость поршня в инерциальной системе координат;
 US2 - скорость снаряда в подвижной системе координат;
 XPL - координата положения поршня;
 XS2 - координата положения снаряда;
 N - количество ячеек в расчетной области 1;
 N2 - количество ячеек в расчетной области 2;
 NDTV - количество ячеек на длине входного отверстия в регулирующий объём;
 NBUX - количество выходных отверстий из регулирующего объёма в область 2.

Метод «calcParamsOfGas»

Метод предназначен для расчета параметров газа, получившихся после распада произвольного разрыва в параметрах на скачке сечения справа и слева от скачка.

Входные параметры:

SK - перепад площади канала на скачке сечения, $SK \leq I$;

POLYTROP, UM, PM, RM, A1 - показатель политропы, скорость, давление, плотность и коволюм газа в узком сечении канала;

POLYTROP, UB, PB, RB, A2 - показатель политропы, скорость, давление, плотность и коволюм газа в широком сечении канала;

Выходные параметры:

BU, BP, BR - скорость, давление, плотность газа на скачке сечения в широкой частиц канала после распада разрыва;

AMU, AMP, AMR - скорость, давление, плотность газа на скачке в узком сечении канала после распада разрыва;

НВ, АВ - показатель политропы и коволюм на скачке после распада разрыва.

Метод «fluxReactingParticles»

Метод предназначен для расчета параметров течения реагирующих частиц на границах расчетных ячеек после распада разрыва в параметрах в соседних ячейках.

Входные параметры:

RKL, WL - объёмная плотность и скорость реагирующих частиц в левой ячейке; RKP, WP - объёмная плотность и скорость реагирующих частиц в правой ячейке;

Выходные параметры:

RRK, WW - объёмная плотность и скорость реагирующих частиц после распада разрыва на границе ячеек;

3.3.5.1 Описание переменных класса

LENSCALE - масштаб длины;

PM - масштаб давления;

CM - масштаб скорости;

RM - масштаб плотности;

SM - масштаб площади;

BPM - масштаб времени.

R - массив плотности газа в области 1;

P - массив давления в области 1;

V - массив скорости газа в области 1;

W - массив скорости частиц в области 1;

FI - массив пористости в области 1;

Z - массив относительной толщины прореагировавшего свода частицы в области 1;

R2 - массив плотности газа в области 2;

- P2 - массив давления в области 2;
U2 - массив скорости газа в области 2;
W2 - массив скорости частиц в области 2;
FI2 - массив пористости в области 2;
Z2 - массив относительно глубины прореагировавшего свода частицы в области 2;
- PR2 - массив давления в регулирующем объёме;
RR2 - массив плотности газа в регулирующем объёме;
UR2 - массив скорости газа в регулирующем объёме;
SR2 - массив площади поперечного сечения регулирующего объёма;
EIR2 - массив пористости в регулирующем объёме, = I;
TR2 - массив температуры газа в регулирующем объёме;
T - массив распределений температуры зёрен к-фазы в области 2;
PI - число π ;
RP - плотность конденсированной фазы частиц;
BG - вязкость газа;
FIPR - предельная пористость;
RKPR - предельная объёмная плотность;
COVOLUME - коволюм газа в области 1;
POLYTROP - показатель политропы газа в области 1;
AKP2, ALP2, AMP2 - коэффициенты формы зерна в области 2;
COVOLUME_2 - коволюм газа в области 2;
POLYTROP_2 - показатель политропы в области 2;
RGC2 - газовая постоянная в области 2;
W02 - начальный объём зерна;
RGC - газовая постоянная в области 1;
ALG - коэффициент теплопроводности газа;
S022 - начальная площадь поверхности зерна;
EPSO1, EPSO2 - коэффициенты черноты газа и к-фазы;
STEFBOLTZ - постоянная Стефана-Больцмана;

ALKF - коэффициент теплопроводности к-фазы;
CAP - коэффициент температуропроводности к-фазы;
QZ - предэкспонент в законе Аррениуса;
EA - энергия активации химической реакции;
CP - удельная теплоёмкость к-фазы;
RP2 - плотность вещества к-фазы;
AFLMP, ANU - коэффициенты в законе скорости химического превращения при малых давлениях;
ACSYS - ускорение системы координат (x' , t);
EB2 - толщина свода зерна;
QP2 - тепловой эффект от химической реакции в области 2;
TZ - критическая температура возбуждения химической реакции на поверхности зерна;
TIME, TIM - безразмерное и размерное время;
SBXR - площадь входного отверстия в регулирующий объём;
SKBUX2 - относительная вскрытая часть боковой поверхности регулирующего объёма;
NBUX - количество выходных отверстий регулирующего объёма;
NOTV - количество расчетных ячеек по длине входного отверстия в регулирующий объём.
SEAND - площадь поперечного сечения в последней расчетной ячейке;
QPP - тепловой эффект от химической реакции в области 1;
QPP2 - тепловой эффект от химической реакции в области 2;
QS - масса снаряда;
DXFORCER - толщина поршня;
TH - величина шага по времени;
HRO - величина шага по пространству в регулирующем объёме;
H2 - величина шага по пространству в области 2 на этапе инициирования;
AMGG, AEGG - масса и энергия газа, втекшего в регулирующий объём и область 2 на этапе инициирования;

P_{MAX} - максимальное давление в области 1;

P_{MAXSB} - максимальное давление на дно поршня;

P_{2MAX} - максимальное давление в области 2;

NR - количество расчетных ячеек в регулирующем объеме;

KOTKR1 - признак вскрытия входного отверстия в регулирующей области;

3.3.5.2 Описание входной информации

1) N, N₂

N - начальное количество ячеек в области 1;

Задается из расчета, чтобы на длине всего канала было не более 190 ячеек. Минимальное необходимое количество ячеек в области 1 определяется путем решения серии задач с одинаковыми начальными условиями и различными N с последующим анализом сходимости и точности расчета при увеличении шага по пространству (при уменьшении N) - начальное количество ячеек в области 2 при расчете по системе уравнений с начальными условиями (после этапа инициирования).

N₂ задается из расчета, чтобы шаг по пространству в области 2 был близок по величине к шагу в области 1;

2) X₀, X₁, X₂, X₃

X₀ - начало отсчета координат, дм;

X₁ - координата окончания первого участка канала, дм;

X₂ - координата окончания второго участка канала, дм;

X₃ - координата окончания третьего участка канала, дм;

3) X_{PL}, X_{PP}, X_S, LENS_{SCALE}, CLENS_{SCALE}

X_{PL} - координата левой границы поршня, дм;

X_{PP} - координата правой границы поршня, дм;

X_S - координата левой границы снаряда, дм;

X_S задается из условия, чтобы сохранить величину свободного объема между поршнями, если между поршнями есть центральное тело. В этом случае $X_S = X_{PP} + W_{св} / \pi D^2 / 4$

где $W_{св}$ - измеренная величина объёма области 2, в которой помещаются зерна к-фазы, $дм^3$;

D - диаметр цилиндрического канала в месте расположения поршней, $дм$;

LENSCALE - длина области 1, $дм$;

CLENSCALE - общая длина канала, $дм$;

Задаваемые координаты показаны на рисунке 2.2.

5) D_0, D_1, D_2, D_3, D

D_0 - диаметр начала канала, $дм$;

D_1 - диаметр в конце первого участка канала, $дм$;

D_2 - диаметр в конце второго участка канала, $дм$;

D_3 - диаметр в конце третьего участка канала, $дм$;

D - диаметр цилиндрического участка канала, $дм$;

6) $NTS, B_2, C_2, АКР, ALP, AMP$

B_2 - начальная толщина свода частицы в области 1, $дм$;

B_2 - начальный диаметр частицы в области 1, $дм$;

C_2 - начальная высота частицы в области 1, $дм$;

$АКР, ALP, AMP$ - коэффициенты формы частицы в области 1.

7) $NPOR, DPOR$

$NPOR$ - количество каналов в частице области 1;

$DPOR$ - начальный диаметр канала в частице области 1, $дм$.

8) $AFL, FP, CV, TPG, COVOLUME, POLYTROP$

AFL - коэффициент в законе скорости массоперехода от химической реакции ($U=a \cdot p$) в области 1, $дм/(с \cdot дн/дм^2)$;

FP - сила вещества к-фазы в области 1, $дн \cdot дм/кг$;

CV - теплоёмкость газа при постоянном объёме в области 1, $дм^2/(с^2 \cdot К)$;

TPG - коэффициент теплопроводности газа в области 1, $дн/(с \cdot К)$;

$COVOLUME$ - коволюм газа в области 1, $дм^3/кг$;

$POLYTROP$ - показатель политропы газа в области 1;

9) $W_1, W_2, QFORCER, QS$

W_1 - масса к-фазы в области 1, $кг$;

W2 - масса к-фазы в области 2, кг;

QFORCER - масса поршня, кг;

QS - масса снаряда; кг;

10) PF, RP, RP2

PF - начальное давление в области 1, дн/(дм²);

RP - плотность к-фазы в области 1, кг/(дм³);

RP2 - плотность к-фазы в области 2, кг/(дм³);

11) EB2, B22, C22, AKP2, ALP2, AMP2

EB2 - начальная толщина свода частицы в области 2, дм;

B22 - начальный диаметр частицы в области 2, дм;

C22 - начальная высота частицы в области 2, дм;

AKP2, ALP2, AMP2 - коэффициенты формы частицы области 2.

12) NPOR2, DPOR2

NPOR2 - количество каналов в частице области 2;

DPOR2 - начальный диаметр канала в частице области 2, дм;

13) AFL2, FP2, CV2, TPG2, COVOLUME_2, POLYTROP_2

AFL2 - коэффициент в законе скорости массоприхода от химической реакции ($U' = a'p'$) в области 2, дм/(с*дн/ дм²);

FP2 - сила вещества к-фазы в области 2, дн*дм/кг;

CV2 - теплоёмкость газа при постоянном объёме в области 2, дм²/(с²*К);

TPG2 - коэффициент теплопроводности газа в области 2, дн/(с К)

COVOLUME_2 - коволюм газа в области 2, дм³/кг;

POLYTROP_2 - показатель политропы газа в области 2.

14) NR2, N2, NOTV

NR2 - количество расчетных ячеек в регулирующем объёме, $20 \geq NRO \geq 6$;

N2 - количество расчетных ячеек в области 2 при расчете инициирования; $N2 \leq 80$;

NOTV - количество расчетных ячеек по длине входного отверстия в регулирующей объём;

15) DBX, ALR2, DR2, ALO2, DO2

DBX- диаметр входного отверстия в регулирующий объём, дм;

ALR2 - длина регулирующего объёма, дм;

DR2 - диаметр регулирующего объёма, дм;

DO2- диаметр области 2, дм;

16) NBUX, DBUX

NBUX - количество выходных отверстий из регулирующего объёма;

DBUX - диаметр выходного отверстия из регулирующего объёма, дм;

17) AFLMP, ANU, ALG, ALKF, EPSO1, EPSO2

AFLMP - коэффициент в законе скорости массоприхода от химической реакции ($U' = a' p' v'$) при малых давлениях $\text{дм}/(\text{с} * (\text{дн}/\text{дм}^2)^{\text{ANU}})$;

ANU - коэффициент в законе скорости массоприхода от химической реакции ($U' = a' p' v'$) при малых давлениях;

ALG - коэффициент теплопроводности газа в области 2 на этапе инициирования, $\text{дн}/(\text{с} * \text{K})$;

ALKF - коэффициент теплопроводности к-фазы в области 2 на этапе инициирования, $\text{дн}/(\text{с} * \text{K})$;

EPSO1, EPSO2 - коэффициенты черноты газа и к-фазы;

18) CP, EA, QZ, TN, PF2, TZ

CP- удельная теплоёмкость вещества к-фазы, $\text{дм}^2/(\text{с} * 2 * \text{K})$;

EA - энергия активации химической реакции, $\text{дн} * \text{дм}/\text{кг}$;

QZ - предэкспонент в законе Аррениуса, $10E+12 * \text{дн} * \text{дм}/(\text{кг} * \text{с})$;

TN - начальная температура в области 2, K;

PF2 - давление форсирования выходных отверстий регулирующего объёма, $\text{дн}/\text{дм}^2$;

TZ - критическая температура возбуждения химической реакции на поверхности зерна, K;

19) PF1

PF1- давление форсирования входного в регулирующий объём отверстия, $\text{дн}/(\text{дм}^2)$.

3.3.5.3 Описание выходной информации

В начале работы программы после ввода исходных данных проводится вывод всех исходных данных задачи с соответствующими комментариями и указанием соответствующей размерности. Затем проводится вывод значений масштабов задачи.

Далее выводится информация, на каждом слое из массивов:

AMS - суммарная масса в областях 1 и 2, кг;

AES - суммарная энергия смеси газа и частиц конденсированной фазы в областях 1 и 2, дн/дм;

ВРЕМЯ - текущее время с начала процесса, сек;

X1 - координата левой границы поршня, дм;

X2 - координата левой границы снаряда, дм;

P1 - давление в канале (в его начале) и на поршень справа, Мпа;

PN - давление на поршень слева, Мпа;

P - давление в области 1, Мпа;

P2 - давление в области 2, Мпа;

R - плотность газа в области 1, кг/дм³;

R2 - плотность газа в области 2, кг/дм³;

U - скорость газа в области 1, м/с;

U2 - скорость газа в области 2, м/с;

W - скорость частиц в области 1, м/с;

W2 - скорость частиц в области 2, м/с;

FI - пористость в области 1;

FI2 - пористость в области 2;

Z - относительная толщина прореагировавшего свода частиц в области 1;

Z2 - относительная толщина прореагировавшего свода частиц в области 2.

После вскрытия входного отверстия в регулирующей объём добавляется вывод распределений параметров течения в регулирующем объёме и области 2 на этапе расчета иницирования. Их вывод происходит до момента

инициирования реакции на всех частицах, а затем они не выводятся. Вывод этих данных проводится в следующей последовательности:

PR2 - давление в регулирующем объеме, МПа;

P2 - давление в области 2, МПа;

RR2 - плотность газа в регулирующем объеме, кг/дм³;

R2 - плотность газа в области 2, кг/дм³;

TR2 - температура газа в регулирующем объеме, К;

T2 - температура газа в области 2, К;

UR2 - скорость газа в регулирующем объеме, м/с;

U2 - скорость газа в области 2, м/с;

FIR - пористость в регулирующем объеме;

F2 - пористость в области 2.

После завершения расчетов выводятся:

PMAX - максимальное давление в области 1 за рассчитываемый промежуток времени, МПа;

PMAXSB - максимальное давление на дно сборки, определяемое для момента времени вскрытия регулирующего отверстия, МПа;

P2MAX - максимальное давление в области 2 за рассчитанный промежуток времени, МПа;

3.4 Интерфейс программы

Интерфейс состоит из главного меню, элемента TabControl в котором находятся 5 вкладок, и кнопок управления.

Вкладки элемента TabControl:

1 вкладка – Терминал. Предназначена для вывода списка проделанных вычислений текущего сеанса.

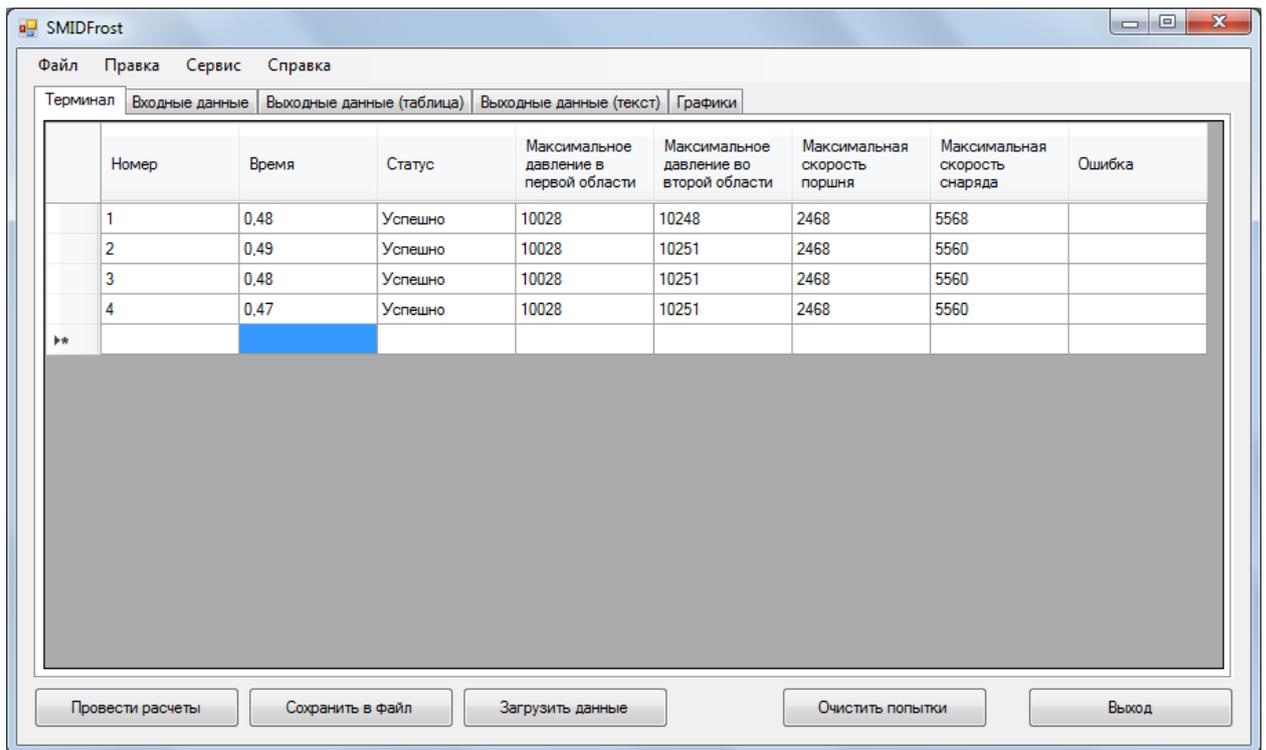


Рисунок 3.5 – Экранная форма - вкладка Терминал.

2 вкладка – *Входные данные*. Предназначена для ввода входных параметров системы. Также имеет меню настройки и запуска автоматического подбора.

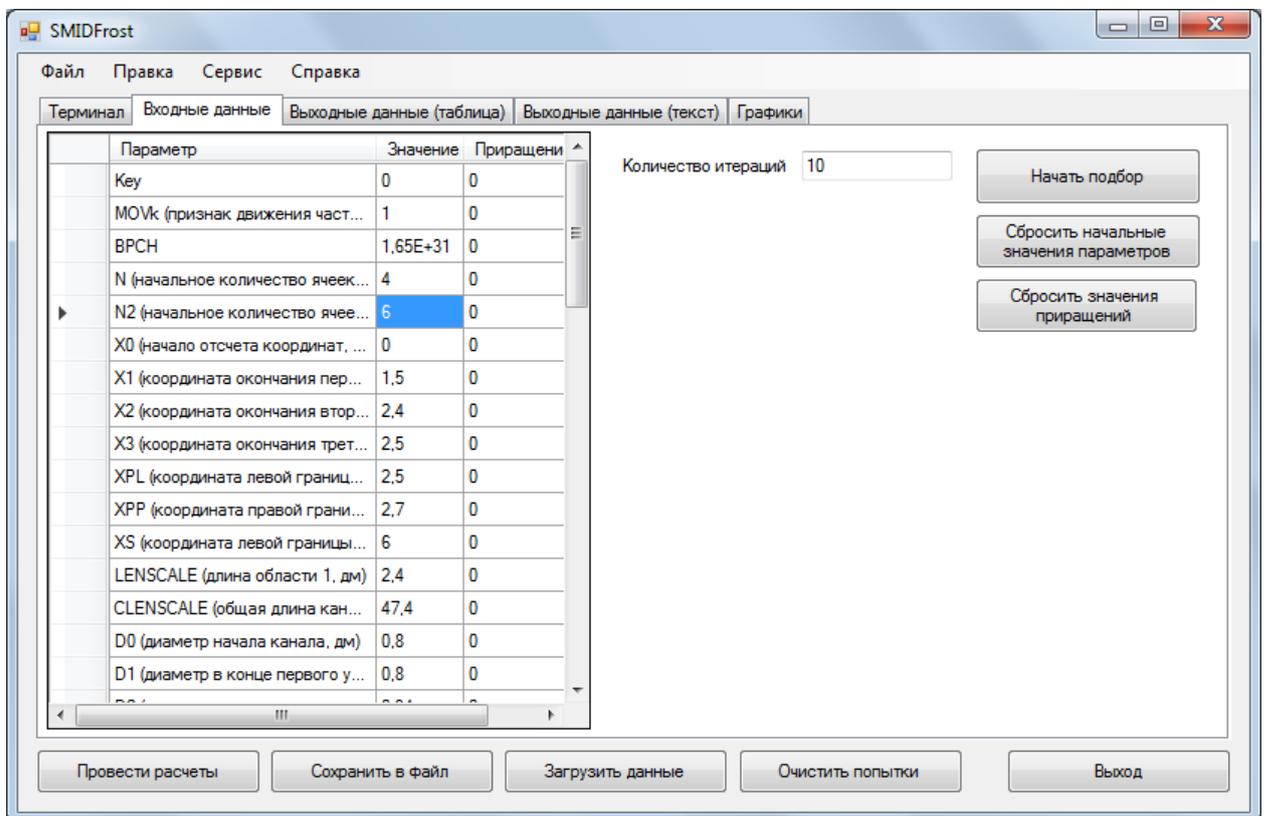


Рисунок 3.6 – Экранная форма - вкладка Входные данные.

3 вкладка – Выходные данные (таблица). Выводит выходные данные текущего экземпляра класса вычислений в виде таблицы.

Номер слоя	Время (сек)	U1 (скорость газа в первой области)	U2 (скорость газа во второй области)	X1 (путь поршня)	X2 (путь снаряда)	P1 (давление в первой области)	P2 (давление во второй области)	P1N	P2N	INDDIA	AMS	AES
1	3.952763781515...	0.236780546719...	0.236780546719...	2.500011783111...	6.000011783111...	21.75314354588	4.9	21.75849301020...	4.9	1.017098259360...	1.185644259490...	533839713.3i
2	1.990553956303...	0.762355579210...	0.762355579210...	2.500061503970...	6.000061503970...	48.52210556755...	4.9	48.51149052568...	4.9	3.274722283446...	1.185644252441...	533839709.5i
3	2.985830934454...	1.933490316175...	1.933490316175...	2.500195659638...	6.000195659638...	109.2560994206...	4.9	109.1717955396...	4.9	8.305368250559...	1.185644160973...	533839658.8i
4	3.981107912606...	4.565792584165...	4.565792584165...	2.500519088970...	6.000519088970...	251.5282886912...	4.9	250.9629973589...	4.9	0.000196125051...	1.185644160266...	533839658.4i
5	4.953712539365...	10.46289291277...	10.46289291277...	2.501249939451...	6.001249939451...	601.4429474048...	4.9	597.7037976189...	4.9	0.000445436844...	1.185644157341...	533839657.0i
6	5.814910234500...	22.82370631647...	22.82370631647...	2.502683252085...	6.002683252085...	1490.969949147...	4.9	1465.377217401...	4.9	0.000980399457...	1.185644138081...	533839649.4i
7	6.506932939799...	46.87889788041...	46.87889788041...	2.505095041321...	6.005095041321...	3899.602201624...	4.9	3706.375373636...	4.9	0.002013697750...	1.185644133967...	533839708.5i
8	7.001323129552...	89.42503412843...	89.42503412843...	2.508464407661...	6.008464407661...	10028.29949142...	4.9	8316.199705107...	4.9	0.003841280367...	1.216476047009...	547722533.8i
9	7.331223610748...	151.63892756979...	151.63892756979...	2.5124407635098...	6.0124407635098...	10023.57448801...	4.9	8521.746646971...	4.9	0.006513697658...	1.216476044105...	547722531.7i
10	7.661160267517...	217.84975617661...	217.84975617661...	2.518536156561...	6.018536156561...	9991.602164551...	4.9	8729.129704695...	4.9	0.009537804552...	1.216476043307...	547722534.5i
11	7.991371790425...	285.6409564627...	285.6409564627...	2.526849078310...	6.026849078310...	9879.950221723...	4.9	8797.173088777...	4.9	0.012269796807...	1.216476043332...	547722539.8i
12	8.318258271494...	352.0656300832...	352.0656300832...	2.537271961412...	6.037271961412...	9706.469698759...	4.9	8727.508014693...	4.9	0.015123089480...	1.216476048061...	547722553.6i
13	8.641766468119...	416.1602533470...	416.1602533470...	2.5496983299198...	6.0496983299198...	9521.792587331...	4.9	8586.301438221...	4.9	0.017876294111...	1.216476047087...	547722559.9i
14	8.962760345968...	478.0521295337...	478.0521295337...	2.564050164939...	6.064050164939...	9352.097630043...	4.9	8422.724179328...	4.9	0.020534879050...	1.2164760578515...	547722584.4i
15	9.281446315372...	537.9717257469...	537.9717257469...	2.580239792302...	6.080239792302...	9197.369229086...	4.9	8248.374885480...	4.9	0.023108744084...	1.216476060581...	547722598.0i
16	9.598127629504...	595.9983526031...	595.9983526031...	2.598195149032...	6.098195149032...	9046.917288602...	4.9	8058.716660948...	4.9	0.025601296027...	1.216476071626...	547722624.2i
17	9.913316002807...	652.0944373202...	652.0944373202...	2.617864365842...	6.117864365842...	8889.724859421...	4.9	7852.703115979...	4.9	0.028010920927...	1.216476080554...	547722647.9i
18	0.000102275266...	706.2232083157...	706.2232083157...	2.639204256932...	6.139204256932...	8718.888618942...	4.9	7635.943815990...	4.9	0.030336039250...	1.216476090317...	547722673.1i
19	0.000105411290...	758.4050875535...	758.4050875535...	2.662169803397...	6.162169803397...	8532.635267611...	4.9	7416.133049969...	4.9	0.032577528227...	1.216476092599...	547722688.3i
20	0.000108543348...	808.7093970374...	808.7093970374...	2.686711272830...	6.186711272830...	8333.288344433...	4.9	7198.884447961...	4.9	0.034738365606...	1.216476097688...	547722707.0i
21	0.000111672613...	857.2235501465...	857.2235501465...	2.712777000773...	6.212777000773...	8125.266152944...	4.9	6986.578732345...	4.9	0.036822306258...	1.216476104149...	547722729.0i
22	0.000114800008...	904.0282722016...	904.0282722016...	2.740317652144...	6.240317652144...	7913.282105759...	4.9	6779.179699630...	4.9	0.038832817763...	1.216476110934...	547722750.9i
23	0.000117926664...	949.1868672374...	949.1868672374...	2.769289262994...	6.269289262994...	7701.187379239...	4.9	6575.493368072...	4.9	0.040772619366...	1.216476120177...	547722776.0i
24	0.000121053348...	992.7449696510...	992.7449696510...	2.799654352696...	6.299654352696...	7491.6011310228...	4.9	6374.314656686...	4.9	0.042643671307...	1.216476119845...	547722788.9i

Рисунок 3.7 – Экранная форма - вкладка Выходные данные.

4 вкладка - Выходные данные (текст). Выводит выходные данные текущего экземпляра класса вычислений в виде текста.

5 вкладка - Графики. Выводит графики исходя из выходных данных текущего экземпляра класса вычислений.

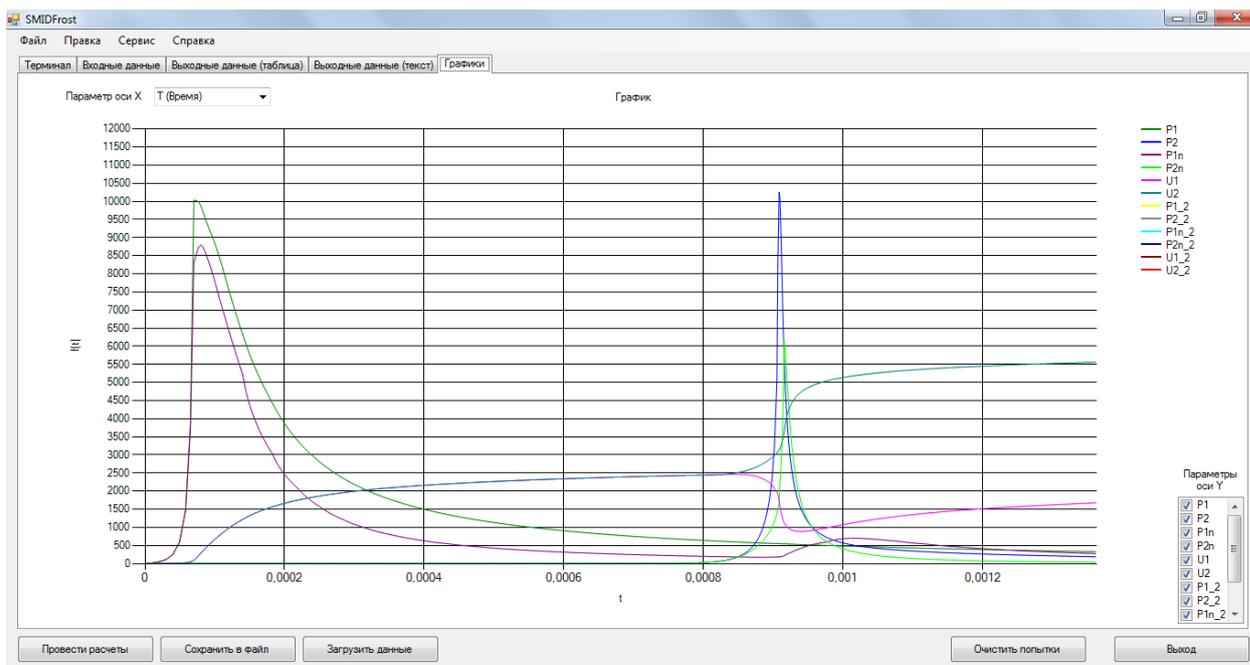


Рисунок 3.8 – Экранная форма - вкладка Графики (Время по оси X).

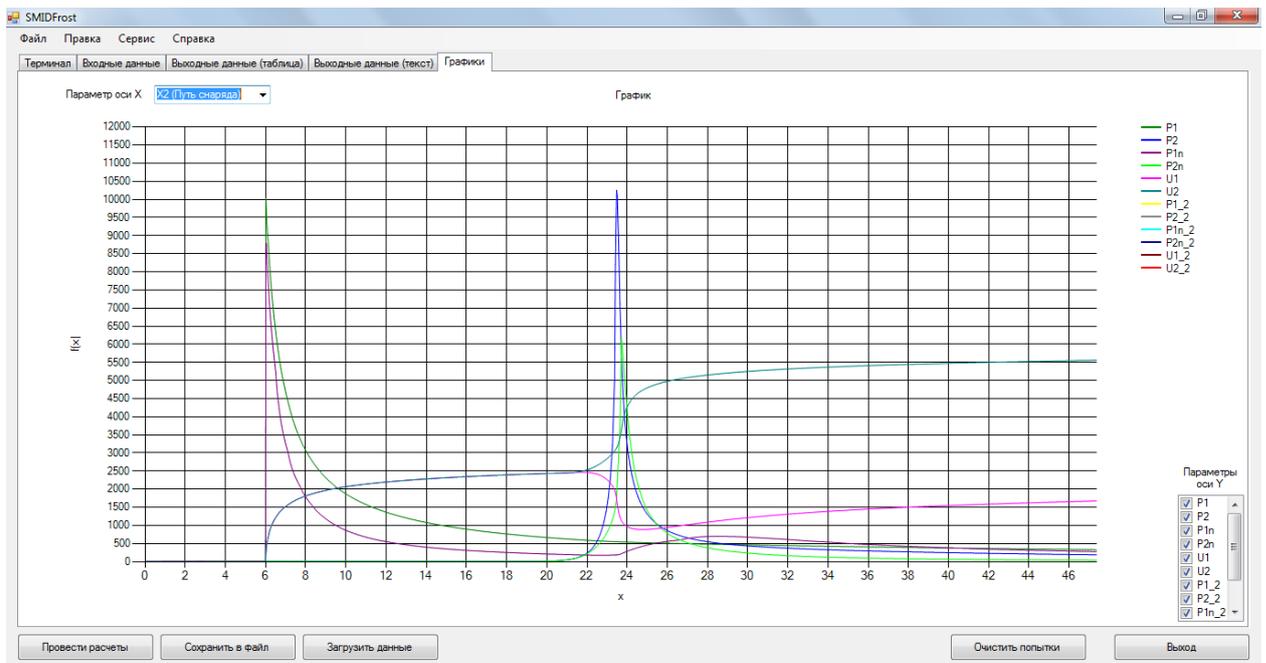


Рисунок 3.9 – Экранная форма - вкладка Графики (Путь снаряда по оси X).

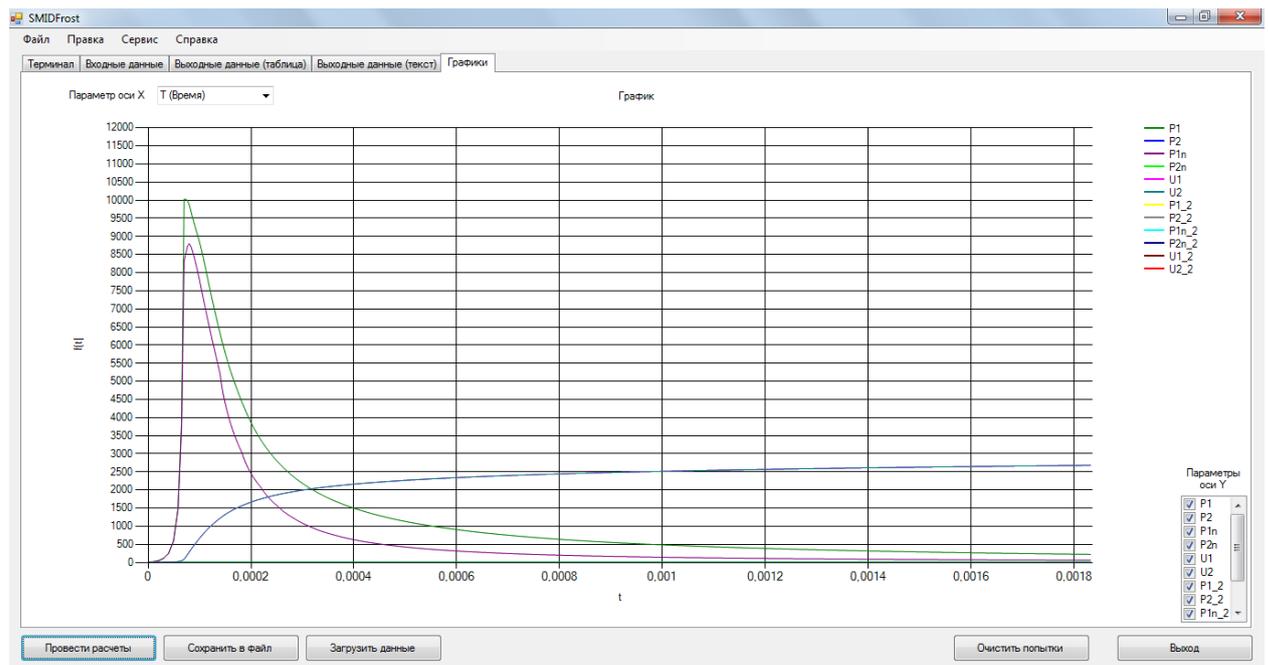


Рисунок 3.10 – Экранная форма - вкладка Графики (Моделирование одной области).

Кнопка «Провести расчеты» - инициализирует входные данные и запускает метод «calculateThread».

Кнопка «Сохранить в файл» - сохраняет данные в файл.

Кнопка «Загрузить в файл» - загружает данные из файла.

Кнопка «Выход» - закрывает приложение.

3.4.1 Главное меню программы

Главное меню программы содержит все функции программы, которые не были вынесены в основную рабочую область. Главное меню состоит из следующих разделов:

1. Файл;
2. Правка;
3. Сервис;
4. Справка.

Вкладка «Файл» служит для вызова системных функций:

- сохранение данных;
- загрузка данных;
- вывод на печать;
- выход (закрытие приложения).

Вкладка «Правка» служит для вызова стандартных системных функций форматирования:

- изменение шрифта;
- изменение цвета;
- копирование/вырезать;
- вставка;
- отмена;
- и др.

Вкладка «Сервис» служит для вызова дополнительных функций программы, основными из них являются:

- установка максимального количества одновременных потоков;
- установка максимального времени расчета для потока (в случае заикливания при возможном некорректном вводе входного параметра или при слишком слабых характеристиках компьютера);

Вкладка «Справка» служит для получения помощи в работе с программой и просмотра сведений о программе и авторе.

Справка по работе с программой открывается отдельным файлом и имеет формат help-файла (справки) в формате CHM для .NET (C#).

Справка включает следующие основные разделы включающие:

- Информация о программе;
 - a. Назначение;
 - b. Описание структуры;
 - c. Описание интерфейса;
- Описание класса вычислений;
 - a. структура;
 - b. методы;
 - c. переменные.
- Краткое описание физико-математической модели;
- Инструкции по параметризации входных данных;
- Описание режимов расчета;
- Пример использования программы;
- Пример заполнения входных данных и моделирования;
- И др.

ГЛАВА 4. РЕЗУЛЬТАТЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

В этой главе даётся оценка эффективности алгоритма разработанного программного продукта при распараллеливании процесса вычислений компьютерной модели, и приводятся результаты численного моделирования внутрикамерных процессов при срабатывании артиллерийского орудия.

Основные результаты работы, представленные в данной главе, отражены в [46, 47, 52, 53].

4.1 Результаты расчета системы малого калибра

Баллистика системы калибра [52] 34 мм СПЗ рассчитывалась при следующих параметрах системы. Масса основного заряда составляла 0,9 кг, присоединённого заряда 0,27 кг; массы поршня 65 г и метаемого элемента 39 г, соответственно. Длина канала системы составляла 4,1 м. Время задержки зажигания выбиралось равным 0,75 мс. Импульсы порохов зарядов подбирались под заданное максимальное давление в обеих областях $P = 1000$ МПа. При этих ограничениях максимальных давлений была получена расчётная скорость 5,55 б/р метаемого элемента массой $q = 39$ г.

Результаты расчета представлены на рисунке 4.1. Здесь представлены зависимости параметров процесса метания схемы СПЗ от времени.

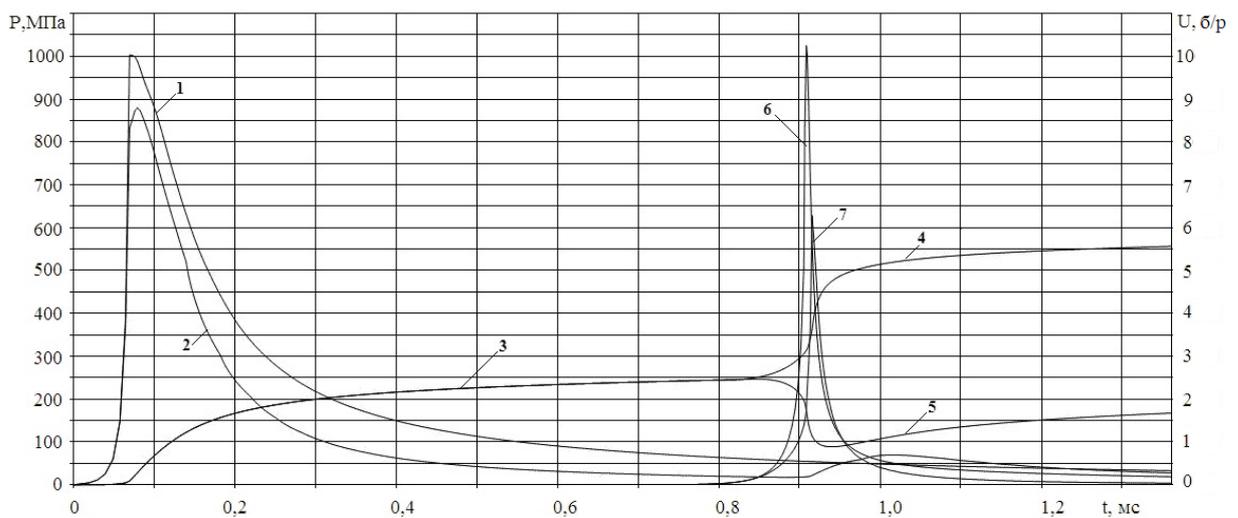


Рисунок 4.1 – Зависимости параметров в процессе метания по схеме СПЗ от времени: 1, 2 – давления на дно канала и на поршень слева, соответственно; 3 –

скорость сборки присоединённого заряда; 4 – скорость метаемого элемента; 5 – скорость поршня; 6,7 – давление на правую границу поршня и давление на метаемый элемент, соответственно

Из графиков видно, что после зажигания присоединённого заряда сборка с присоединённым зарядом разделяется, метаемый элемент резко ускоряется, а поршень значительно тормозится от скорости примерно 2,5 б/р до скорости примерно 0,9 б/р, передавая свою энергию метаемому элементу. За счёт перераспределения энергии порохового газа давление на метаемом элементе становится выше, чем на поршне, благодаря чему реализуется более эффективный метод ускорения метаемого элемента.

Повышение расчётной начальной скорости метаемого элемента при СПЗ по сравнению со скоростью при классической схеме метания составило примерно 25%.

Таким образом, расчетным путем показано, что на системах малого калибра при применении схемы СПЗ скорости метания могут быть значительно повышены. Таким образом, расчетным путем показано, что на системах малого калибра при применении схемы СПЗ скорости метания могут быть значительно повышены.

4.2 Согласование расчётных данных с экспериментом

Специально для согласования модели с экспериментом были взяты результаты реального испытания проведенным в Нижнем Новгороде на баллистической системе калибра 100 мм.

Для определения давления в реальном эксперименте датчики давления располагались на расстоянии 0,47 м и 1,7 м от дна канала.

На рисунке 4.2 приведены зависимости давлений в сечении $x=0,47$ м зарегистрированных датчиком и определённых теоретически по используемой гетерогенной модели сплошной среды.

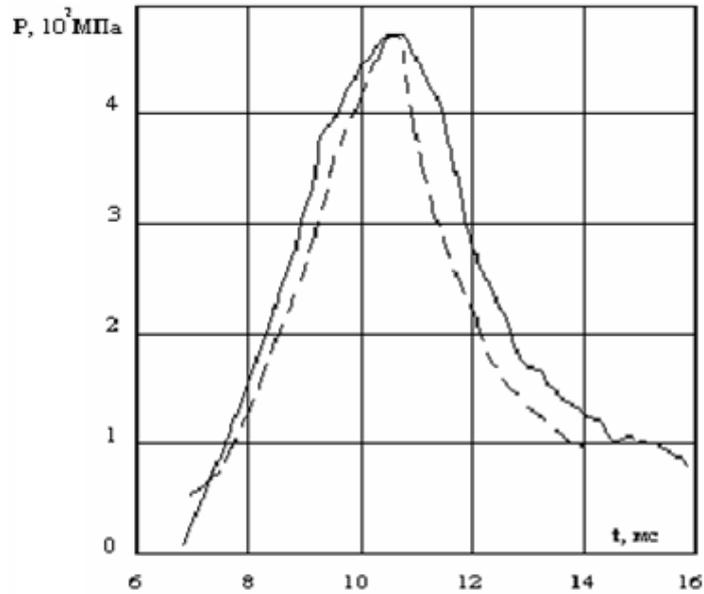


Рисунок 4.2 – Зависимость давления от времени на 1-м датчике:
 — зависимость давления от времени записанная датчиком 1, $X=0,47$ м;
 ---- расчётная зависимость полученная в программе.

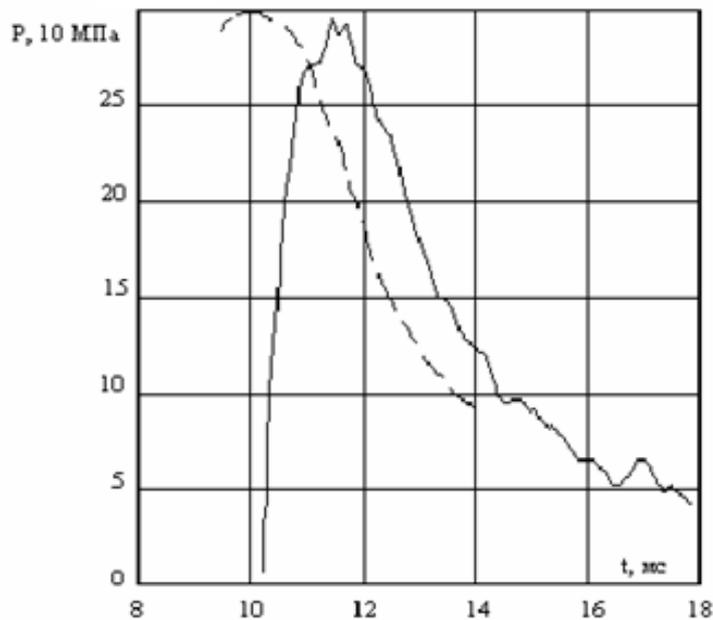


Рисунок 4.3 – Зависимость давления от времени на 2-м датчике:
 — зависимость давления от времени записанная датчиком 2, $X=1,7$ м;
 ---- расчётная зависимость полученная в программе.

В результате расчетов компьютерной модели с входными параметрами реального эксперимента баллистической системы калибра 100мм получено, что

разница между расчётной и экспериментально полученной дульными скоростями составила 5%.

Можно сделать вывод, что результаты компьютерного моделирования двухфазной двухскоростной среды с присоединенными зарядами для расчета параметров баллистических систем с помощью разработанной программы в рамках диссертационного исследования, согласуются с результатами реальных экспериментов на артиллерийской системе.

4.3 Оценка эффективности алгоритма распараллеливания

Основное назначение распараллеливания – выполнить несколько расчетов с разными входными параметрами (приращениями к основному параметру) параллельно друг другу, что повысит скорость получения результатов [14, 19, 39, 41].

Были проведены расчеты в режимах единичного и многопоточного вычисления на компьютерах с разными характеристиками. На каждом компьютере было совершено 10 запусков единичного расчета и 1 запуск в режиме многопоточности с 10 итерациями. Время на ввод значений пользователем в режиме единичных расчетов опущено. На каждом компьютере данные действия проводились 5-10 раз для количественного анализа. В результате количественного анализа расхождения между попытками не сильно отличались, поэтому была взята случайная попытка и проведен качественный анализ результатов между двумя режимами расчетов.

Таблица 4.1 – Сравнение времени вычислений на компьютерах с разными характеристиками

Поток	t1, с	t2, с	t3, с	t4, с	t5, с	t6, с	t7, с	t8, с	t9, с	t10, с
p1	1,3	-	0,8	1,9	0,7	1,17	0,49	1,68	0,39	0,83
p2	1,34	-	0,75	1,81	0,74	1,43	0,5	1,25	0,35	0,83
p3	1,35	-	0,9	1,88	0,65	1,47	0,52	1,78	0,36	0,92
p4	1,32	-	0,81	1,96	0,72	0,84	0,49	1,51	0,36	1,26
p5	1,36	-	0,78	1,94	0,66	0,98	0,51	1,64	0,34	1,25
p6	1,36	-	0,78	0,91	0,71	1,06	0,51	1,44	0,33	0,71

p7	1,36	-	0,77	1	0,73	1,08	0,51	1,81	0,32	0,72
p8	1,32	-	0,78	1,28	0,64	1,3	0,48	1,32	0,35	0,65
p9	1,32	-	0,9	1,31	0,65	0,75	0,47	1,37	0,37	0,46
p10	1,37	-	0,7	1,32	0,73	0,78	0,5	1,36	0,33	0,47
sum(t)	13,4	-	7,97	15,3 1	6,93	10,86	4,98	15,1 6	3,5	8,1
fact(t)	13,4	-	7,97	4,92	6,93	4,01	4,98	3,28	3,5	2,57
Характеристики компьютера	Процессор Intel Pentium 4 3.0GHz (1 поток), 2GB ОЗУ	Процессор Intel Core i3 2x1.8GHz (4 потока), 4GB ОЗУ	Процессор Intel Core i5 2x2.5GHz (4 потока), 8GB ОЗУ	Процессор Intel Core i5 4x3.3GHz (4 потока), 16GB ОЗУ	Процессор Intel Core i7 4x3.4GHz (8 потоков), 32GB ОЗУ					

где,

t_1, t_3, t_5, t_7, t_9 – Время выполнения единичного вычисления;

$t_2, t_4, t_6, t_8, t_{10}$ – Время выполнения в режиме многопоточности;

p_1-p_{10} – Количество потоков;

$sum(t)$ – Суммарное время выполнения потоков;

$fact(t)$ – Фактическое время выполнения потоков с учетом распараллеливания.

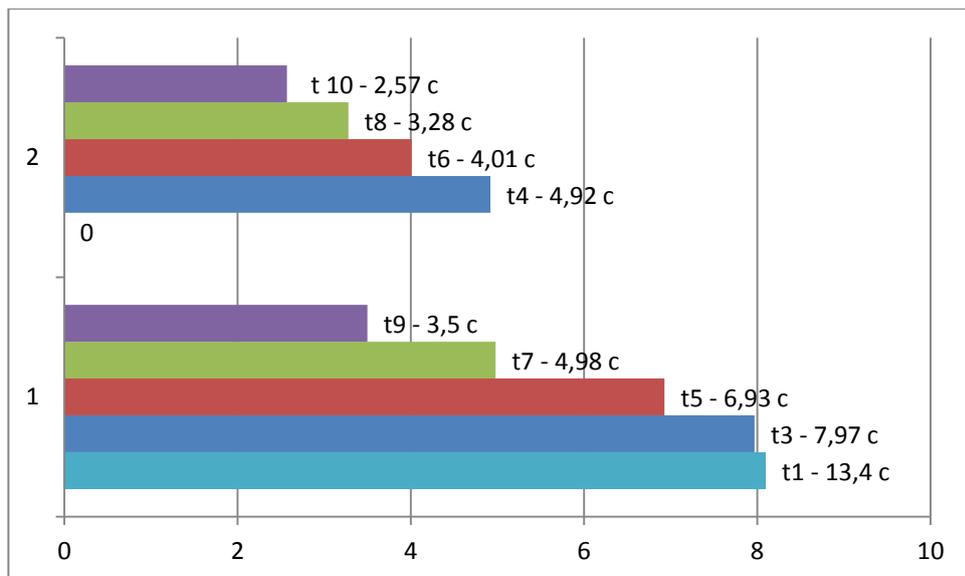


Рисунок 4.4 – Диаграмма соотношений времени выполнения вычислений в разных режимах: ось X – время вычислений; ось Y (1 – режим единичного вычисления, 2 – режим многопоточности).

Исходя из графического отображения на рисунке 4.4 результатов сопоставления единичного и многопоточного режимов вычисления, можно сделать вывод, что в многопоточном режиме время нахождения наилучших выходных параметров системы уменьшается примерно на 30% за счет распараллеливания.

Если учесть что пользователь в режиме единичного вычисления тратит время на ввод или изменение входных параметров больше чем делятся среднестатистические расчеты, когда в режиме многопоточности он лишь единожды задает приращение, после чего система сама подбирает значения, можно сделать выводы что, используя режим многопоточного подбора параметров, пользователь экономит минимум 130% времени.

ЗАКЛЮЧЕНИЕ

1. Исследованы высокоэнергетические процессы во внутренней баллистике на основе подхода механики гетерогенных сред [9, 10, 11, 31];
2. Рассмотрена двухфазная двухскоростная математическая модель баллистической системы [9, 10, 11] присоединенным зарядом в качестве объекта компьютерного моделирования [46, 47, 48];
3. Описана физико-математическая модель [17, 27] на основе современного подхода механики гетерогенных сред [9, 10, 11];
4. Выбран метод численного решения систем уравнений в частных производных [12, 25] с соответствующими начальными и граничными условиями с учетом особенностей схемы метания с присоединенным зарядом;
5. На основе выбранного численного метода и физико-математической модели была составлена алгоритмическая модель [12, 17, 25, 27] высокоэнергетических внутрикамерных процессов в системах ствольного метания с присоединенным зарядом;
6. Полученная алгоритмическая модель была преобразована в компьютерную модель [8, 12, 14, 35, 36] посредством языка программирования C# с применением алгоритмов распараллеливания [16, 22, 26, 28, 29, 34, 39, 40, 41] для ускорения подбора входных параметров расчетной системы;
7. Произведено параметрическое исследование и моделирование баллистических систем 34мм и 100мм с использованием разработанной компьютерной модели;
8. В результате моделирования получено согласование результатов серии экспериментов проведенных в Нижнем Новгороде с результатами расчетов разработанной компьютерной модели. При подборе реального максимального давления в системе получена расчетная скорость метаемого элемента с точностью до 5%.

С научной точки зрения, показано, как применение компьютерных моделей высокоэнергетических процессов (в данном контексте высокоэнергетических процессов во внутренней баллистике) помогает при

параметрическом исследовании значительно улучшать энергетические характеристики баллистических систем.

Образовательным аспектом исследовательской работы является то, что разработанная программа, поможет студентам быстрее освоить навыки математического и компьютерного моделирования сложных многопараметрических систем и детально усвоить физику протекающих при этом высокоэнергетических процессов, что, несомненно, поможет им ориентироваться в дальнейшей профессиональной деятельности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Нормативно-правовые акты

1. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание документа.
2. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления
3. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов.
4. ГОСТ Р 7.05-2008 Библиографическая ссылка.
5. ГОСТ 9327-60 Бумага и изделия из бумаги. Потребительские форматы.
6. ГОСТ 2.105-95 Единая система конструкторской документации. Общие требования к текстовым документам.
7. ГОСТ 6.38-90 Унифицированные системы документации. Система организационно-распорядительной документации. Требования к оформлению документов.

Монографии

8. Годунов, С.К. Численное решение многомерных задач газовой динамики [Текст]: монография / С.К. Годунов [и др.] — М.: Наука, 1976.—400с.
9. Ныгматулин, Р.И. Основы механики гетерогенных сред [Текст]: монография/ Р.И. Ныгматулин. — М.: Наука, 1978.— 336 с.
10. Русяк, И.Г., Ушаков, В.М. Внутрикамерные гетерогенные процессы в ствольных системах [Текст]: монография / И.Г. Русяк, В.М. Ушаков. — Екатеринбург: УрО РАН, 2001.— 259 с.
11. Хоменко, Ю. П., Ищенко, А.Н., Касимов, В.З. Внутренняя баллистика. Математическое моделирование внутрибаллистических процессов в ствольных системах: монография / Ю. П. Хоменко, А.Н. Ищенко, В.З. Касимов. — Новосибирск: Издательство СО РАН, 1999. — 256 с.

Учебники и учебные пособия

12. Бахвалов, Н.С. Численные методы [Текст]: учебное пособие / Н.С. Бахвалов. - М.: Наука, 2006. 631 с.

13. Бетехтин, С.А., Веницкий, А.М., Горохов, М.С. и др. Газодинамические основы внутренней баллистики [Текст]: учебное пособие / С.А. Бетехтин, А.М. Веницкий, М.С. Горохов и др. — М.: Оборонгиз, 1957. — 219 с.
14. Боев, В.Д. Компьютерное моделирование. Пособие для курсового и дипломного проектирования [Текст]: учебное пособие / В.Д. Боев, Д.И. Кирик, Р.П. Сыпченко. — М.: Военная Академия связи, Санкт-Петербург, 2011. — 348 с.
15. Бураков, П.В., Косовцева, Т.Р. Информатика. Алгоритмы и программирование. [Текст]: учебное пособие / П.В. Бураков, Т.Р. Косовцева. — Санкт-Петербург: НИУ ИТМО, 2014. — 83 с.
16. Воеводин, В. В. Параллельные вычисления: учебное пособие / В.В. Воеводин, Вл.В. Воеводин — Санкт-Петербург: БХВ-Петербург, 2002. — 608 с.
17. Вороненко, Б.А. Введение в математическое моделирование [Текст]: учебно-методическое пособие / Б.А. Вороненко и др. — СПб.: НИУ ИТМО; ИХиБТ, 2014. — 44 с.
18. Горохов, М.С., Липанов, А.М., Русяк, И.Г. Основы современной теории внутренней баллистики орудий [Текст]: учебник / М.С. Горохов, А.М. Липанов, И.Г. Русяк. — М.: ЦНИИИТИ и ТЭИ, 1988.
19. Гофф, М. К. Сетевые распределенные вычисления: достижения и проблемы: учебник / М.К. Гофф. — М.: КУДИЦ—Образ, 2005. — 320 с.
20. Гринь, А.Г. Вероятность и статистика [Электронный ресурс]: учебное пособие / А.Г. Гринь.— Омск: Омский государственный университет, 2013.— 304 с.
21. Давыдов, Ю.М., Косолапов, Е.А. Численное моделирование двухфазных течений в соплах методом крупных частиц [Текст]: учебное пособие / Ю.М. Давыдов, Е.А. Косолапов. — М.: НАПН, 1998. — 86 с.
22. Дубаков, А.А. Сетевое программирование [Текст]: учебное пособие / А.А. Дубаков. — Санкт-Петербург: НИУ ИТМО, 2013. — 248 с.

23. Душин, С.Е., Красов, А.В., Литвинов, Ю.В. Моделирование систем и комплексов [Текст]: учебное пособие / С.Е. Душин, А.В. Красов, Ю.В. Литвинов. — Санкт-Петербург: СПбГУ ИТМО, 2011. — 178 с.
24. Захаренков, В.Ф. Внутренняя баллистика и автоматизация проектирования артиллерийских орудий [Текст]: учебник / В.Ф. Захаренков. — СПб: Балт. гос. техн. ун-т., 2010. — 276 с.
25. Карпов, В.Е. Численные методы, алгоритмы и программы. Введение в распараллеливание [Текст]: учебное пособие для вузов / В.Е. Карпов, А.И. Лобанов. — М.: Издат-во Москов. физ.-технич. института, 2014. — 190 с.
26. Качко, Е.Г. Параллельное программирование [Текст]: учебное пособие / Е.Г. Качко. — Харьков: Форт, 2011. — 528 с.
27. Колесов, Ю.Б. Компонентные технологии математического моделирования [Текст]: учебное пособие / Ю.Б. Колесов, Ю.Б. Сениченков. — Санкт-Петербург: Издательство Политехнического университета, 2013. — 233 с.
28. Косяков, М.С. Введение в распределенные вычисления [Текст] / учебное пособие / М.С. Косяков. — Санкт-Петербург: НИУ ИТМО, 2014. — 155 с.
29. Лупин, С.А., Посыпкин, М.А. Технологии параллельного программирования [Текст]: учебник / С.А. Лупин, М.А. Посыпкин. М.: ИД «ФОРУМ»: ИНФРА-М, 2011. — 208 с.
30. Мареев, В.В. Основы методов конечных разностей [Текст]: учебное пособие / В.В. Мареев, Е.Н. Станкова. — СПб.: Издательство С. — Петербургского университета, 2012. — 64 с.
31. Ныгматулин, Р. И. Динамика многофазных сред [Текст]: учебник / Р.И. Ныгматулин. — М.: Наука, 1987. — Часть I. — 464 с. Часть II. — 360 с.
32. Петров, В.Ю. Информатика. Алгоритмизация и программирование. [Текст]: учебное пособие / В.Ю. Петров. — Часть 1 — Санкт-Петербург: Университет ИТМО, 2015. — 91 с.
33. Радаев, А.В. Основы огневой подготовки: учебное пособие / А.В. Радаев — ГМА им. адм. С.О.Макарова, — Санкт-Петербург, 2010. — 62 стр

34. Радченко, Г.И. Распределенные вычислительные системы [Текст]: учебное пособие / Г.И. Радченко. — Челябинск: Фотохудожник, 2012. — 184 с.
35. Рендольф, Н. Microsoft Visual Studio 2010 для профессионалов [Текст]: учебник / Н. Рендольф, Д. Гарднер, М. Минутилло, К. Андерсон. — М.: MicrosoftPress, 2011. — 1184 с.
36. Сизова, Т.М. Статистика [Текст]: учебное пособие / Т.М. Сизова. — Санкт-Петербург: НИУ ИТМО, 2013. — 176 с.
37. Скит, Джон. С# для профессионалов: тонкости программирования [Текст]: учебник / Джон Скит. — 3-е издание, новый перевод. — М.: «Вильямс», 2014. — 608 с.
38. Тупицына, А.И. Методы компьютерного моделирования физических процессов и сложных систем [Текст]: учебное пособие / А.И. Тупицына. — СПб: Университет ИТМО, 2014. — 48 с.
39. Туральчук, К.А. Параллельное программирование с помощью языка С# [Текст]: учебное пособие / К.А. Туральчук. — 2-е издание. — М.: НОУ «Интуит», 2016. — 190 с.
40. Цимбал, А.А. Параллельные алгоритмы. Разработка и реализация [Текст]: учебное пособие / А.А. Цимбал. — М.: Бином «ИНТУИТ», 2014. — 344 с.
41. Эндрюс, Г.Р. Основы многопоточного, параллельного и распределенного программирования [Текст]: учебник / Г.Р. Эндрюс. М.: «Вильямс», 2003. — 512 с.

Периодические издания

42. Басевич, В. Я., Иванов, В. С., Семенов, И. В., Фролов, С. М., Фролов, Ф. С. Математическое моделирование вспышки при выстреле [Текст] / В.Я. Басевич, В.С. Иванов, И.В. Семенов, С.М. Фролов, Ф.С. Фролов // Горение и взрыв, Т.6. № 6, 2013. — С. 112 -117.
43. Быков, Н.В., Владимиров В.С., Зеленцов В.В. Инженерная методика расчета внутренней баллистики систем высокоскоростного метания [Текст] / Н.В. Быков, В.С. Владимиров, В.В. Зеленцов // Оборонная техника, №8, 2011. - С. 3-9

44. Егоров, М.Ю., Парфёнов, А.Ю., Егоров, Д.М. Численное исследование динамики внутрикамерных процессов при срабатывании артиллерийского выстрела [Текст] / М.Ю. Егоров, А.Ю. Парфёнов, Д.М. Егоров // Вестник ПНИПУ. Аэрокосмическая техника, № 32, 2012. - С. 50-66.
45. Егоров, М. Ю., Парфёнов, А. Ю. Численное моделирование процесса срабатывания дульного тормоза артиллерийского орудия [Текст] / М.Ю. Егоров, А.Ю. Парфёнов // Вестник Ижевского государственного технического университета, №4, 2013. - С. 155-158.
46. Сафронов, А.И. Внутренняя баллистика ствольной системы с присоединенной камерой подгона [Текст] / А.И.Сафронов, А.Ю. Крайнов // Вестник ТГПУ, Вып.6 (43), 2004. – С. 67-70.
47. Сафронов, А.И. Использование нетрадиционной схемы метания в баллистических схемах [Текст] / А.И. Сафронов, Л.В. Комаровский // Международная конференция по математике и механике: Избранные доклады / Под общей редакцией Н.Р. Щербакова.— Томск, 2003. – С.182-187.
48. Семенов, И. В., Меньшов, И. С., Подложнюк, А. Д., Ахмедьянов, И. Ф. Численное моделирование теплового состояния ствола артиллерийского орудия при выстреле [Текст] / И.В. Семенов, И.С. Меньшов, А.Д. Подложнюк, И.Ф. Ахмедьянов // Горение и взрыв, Т. 7. № 7, 2014. – С. 374-379.
49. Семенов, И.В., Меньшов, И.С., Уткин, П.С., Ахмедьянов, И.Ф, Пасынков, П.А., Попов, А.А. Многомерное численное моделирование связанных задач внутренней и промежуточной баллистики [Текст] / И.В. Семенов, И.С. Меньшов, П.С. Уткин, И.Ф. Ахмедьянов, П.А. Пасынков, А.А. Попов // Известия ВУЗов. Физика, №6/3, 2013. – С. 58-60.
50. Семенов, И. В., Меньшов, И. С., Уткин, П. С., Ахмедьянов, И. Ф. БАРС-1МП – программный комплекс для численного исследования внутрибаллистических процессов на многопроцессорных ЭВМ [Текст] / И.В. Семенов, И.С. Меньшов, П.С. Уткин, И.Ф. Ахмедьянов // Известия ВУЗов. Физика, №6/3, 2013. – С. 61-63.

51. Быков, Н. В., Владимиров, В. С., Зеленцов, В. В. Численное моделирование внутренней баллистики цилиндроконических стволов с использованием пластических снарядов, 2012 // Наука и образование: электронное научно-техническое издание. № 3. С. 41. [Электронный ресурс]: <http://technomag.edu.ru/doc/310721.html>
52. Сафронов, А. И., Морозов, А. А., Азовский, В. М. Анализ и баллистическое проектирование системы малого калибра с присоединённым зарядом, 2016 // ИННОВАЦИИ. НАУКА. ОБРАЗОВАНИЕ: электронное научно-техническое издание. № 4. [Электронный ресурс]: <http://ojs.innovjourn.ru/index.php/innov/article/view/28>
53. Парфенов, А.А. Численное моделирование динамики внутрикамерных процессов при срабатывании артиллерийского орудия: диссертация кандидата технических наук: 05.13.18 / А.А. Парфенов; [Место защиты: Пермский национальный исследовательский политехнический университет]. – Пермь, 2014. – 148 с. [Электронный ресурс]: http://pstu.ru/files/file/adm/dissertacii/parfenov/parfenov_a_yu_dissertatsiya.pdf

Литература на иностранном языке

54. Baker E., Templeton D. Ballistics 2011: 26th International Symposium — DEStech Publications, 2011. – 2162 p.
55. Griffiths I. Programming C# 5.0: Building Windows 8, Web, and Desktop Applications for the .NET 4.5 Framework – O'Reilly Media. – 1 edition, 2012. –886 p.
56. Herlihy M., Shavit N. The Art of Multiprocessor Programming. – Revised 1st Edition. — Morgan Kaufmann, 2012. – 537 p.
57. Ingalls J. M. Interior Ballistics– Nabu Press. – Primary Source Edition, 2014. – 258 p.
58. Pelland P., Haines K. Moving to Microsoft Visual Studio 2010 – Microsoft Press, 2011. –336 p.

ПРИЛОЖЕНИЕ 1. Программный код класса CalculateDD

// Класс вычислений CalculateDD предназначен для расчета параметров внутренней баллистики

```
public class CalculateDD
```

```
{
```

```
    // CONST
```

```
    const int N_max = 192 + 1;
    const int N2_max = 192 + 1;
    const int Layers_max = 5000;
    const double PI = 3.14159265;
    const double BG = 0.00000518;
    const double FIPR = 0.4;
    //const double precision = 0.2;
```

```
    // BOOLEAN
```

```
    public Boolean errorCalc = false;
```

```
    // STRING
```

```
    public String OutString = "";
```

```
    // INTEGER
```

```
    int I, KK, NDS, NN;
    public int layersCount, N, N2;
```

```
    // DOUBLE
```

```
    double STEFBOLTZ,
           TIME,
           TIM,
           US,
           US2,
           PIND,
           APCH1,
           APCH12,
           ACSYS,
           RKPR,
           KINDI,
           KINDII;
```

```
    // DOUBLE заполняются из файла или вручную
```

```
    public double precision, MOVk, BPCH,
                X0, X1, X2, X3,
                XPL, XPP, XS, LENSACLE, CLENSCALE,
                D0, D1, D2, D3, D,
                NTS, ND1, NC1, AKP, ALP, AMP,
                NPOR, DPOR,
                AFL, FP, CV, TPG,  $\alpha$ _COVOLUME, POLYTROP,
                W1, W2, QPPP, QS,
                PFF, RP,
                NTS2, ND2, NC2, AKP2, ALP2, AMP2,
                NPOR2, DPOR2,
                AFL2, FP2, CV2, TPG2,  $\alpha$ _COVOLUME_2, POLYTROP_2,
                PFF2, XIGN;
```

```
    // DOUBLE
```

```
    double RGC, RGC2, QP, QP2,
           PM, CM, RM, SM, BPM, DELTA, WKM,
           SEND, QPP, QPP2, DXP, TH,
           G1, G2, G3, G4, G5, G6,
```

```

    XX1, XX2, DI1, DI2;

double QFORCER, W0, W02, S02, S022, KPER, KMOV, IIGN,  $\tau$ , H2, NLD,
    AES0, PFNA, W22, DELTA2,
    PSI0, SIG0, Z0, TG,
    XSO, WKMSCH,
    UMAX, UMAX1, THPCH, UMAX2, UMAX21, TH2, TH2PCH,
    UBB, PBB, RBB, UMM, PMM, RMM, EKB, ALB,
    DRR, USN, US2N;

double UPOUT, USOUT, XPLOUT, XSOUT, DXN, DXON, AMS, AES,
    EKG1, EBG1, EKK1, EXK1, EKG2, EBG2, EKK2, EXK2,
    ESQ1, ESB, EKP, ESQ1SB, ESQ1P, ESQ2, ESN, ESQ2SN, AESD, PGLOBAL;

double RKP, RRK1, WW1, ARK1, AWRK1, AR1, AUR1, AER1, ZN, PST, WST, FIST,
    ARK2, AWRK2, AR2, AUR2, AER2, AR3, AUR3, AER3,
    DWSDX, PSIZ, ANI, SIGZ, GM, RE,
    RKL, RRK2, WW2, SK, UR, PR, RR, UL, PL, RL,
    CX, DP, ATR, TTR, Q, FMASG, FIMPS, FENS, FMASK,
    APCH, RSFI, RUSFI, RESFI, RKSFI, FIN, RN, UN, EN, PN, DPDX,
    KIN, RKWSFI, WN, ZN1,
    USI, PMSB, XSN, DX,
    WNEND, SNEND;

// ARRAY
double[] R = new double[N_max + 1],
    P = new double[N_max + 1],
    U = new double[N_max + 1],
    W = new double[N_max + 1],
    FI = new double[N_max + 1],
    Z = new double[N_max + 1],
    _R2 = new double[N_max + 1],
    _P2 = new double[N_max + 1],
    _U2 = new double[N_max + 1],
    _W2 = new double[N_max + 1],
    _FI2 = new double[N_max + 1],
    _Z2 = new double[N_max + 1],
    S = new double[N_max + 1],
    SV = new double[N_max + 1],
    _S2 = new double[N_max + 1];

public double[] outP = new double[Layers_max],
    outP2 = new double[Layers_max],
    outPN = new double[Layers_max],
    outU = new double[Layers_max],
    outU2 = new double[Layers_max],
    outT = new double[Layers_max],
    outP2N = new double[Layers_max],
    outX = new double[Layers_max],
    outX2 = new double[Layers_max],
    outAMS = new double[Layers_max],
    outAES = new double[Layers_max],
    outAESD = new double[Layers_max],
    outINDDIA = new double[Layers_max];

double Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, P1, P2, P3, P4, P5,
    P6, P7, P8, P9, P10, PSH, G7, G8, G9, GR,
    U1, U2, U3, U4, U5, U6, R1, R2, R3, R4, R5, R6, A7, A8;

/// <summary>
/// Конструктор класса
/// </summary>

```

```

public CalculateDD(double precision, double MOVk, double BPCH, int N, int N2,
    double X0, double X1, double X2, double X3,
    double XPL, double XPP, double XS, double LENSSCALE, double CLENSCALE,
    double D0, double D1, double D2, double D3, double D,
    double NTS, double ND1, double NC1, double AKP, double ALP, double AMP,
    double NPOR, double DPOR, double AFL, double FP, double CV, double TPG,
    double  $\alpha$ _COVOLUME, double POLYTROP, double W1, double W2, double QPPP,
double QS,
    double PFF, double RP, double NTS2, double ND2, double NC2, double
AKP2,
    double ALP2, double AMP2, double NPOR2, double DPOR2,
    double AFL2, double FP2, double CV2, double TPG2, double  $\alpha$ _COVOLUME_2,
double POLYTROP_2,
    double PFF2, double XIGN)
{
    this.precision = precision;
    this.MOVk = MOVk;
    this.BPCH = BPCH;
    this.N = N;
    this.N2 = N2;
    this.X0 = X0;
    this.X1 = X1;
    this.X2 = X2;
    this.X3 = X3;
    this.XPL = XPL;
    this.XPP = XPP;
    this.XS = XS;
    this.LENSCALE = LENSSCALE;
    this.CLENSCALE = CLENSCALE;
    this.D0 = D0;
    this.D1 = D1;
    this.D2 = D2;
    this.D3 = D3;
    this.D = D;
    this.NTS = NTS;
    this.ND1 = ND1;
    this.NC1 = NC1;
    this.AKP = AKP;
    this.ALP = ALP;
    this.AMP = AMP;
    this.NPOR = NPOR;
    this.DPOR = DPOR;
    this.AFL = AFL;
    this.FP = FP;
    this.CV = CV;
    this.TPG = TPG;
    this. $\alpha$ _COVOLUME =  $\alpha$ _COVOLUME;
    this.POLYTROP = POLYTROP;
    this.W1 = W1;
    this.W2 = W2;
    this.QPPP = QPPP;
    this.QS = QS;
    this.PFF = PFF;
    this.RP = RP;
    this.NTS2 = NTS2;
    this.ND2 = ND2;
    this.NC2 = NC2;
    this.AKP2 = AKP2;
    this.ALP2 = ALP2;
    this.AMP2 = AMP2;
    this.NPOR2 = NPOR2;
    this.DPOR2 = DPOR2;
}

```

```

this.AFL2 = AFL2;
this.FP2 = FP2;
this.CV2 = CV2;
this.TPG2 = TPG2;
this.alpha_COVOLUME_2 = alpha_COVOLUME_2;
this.POLYTROP_2 = POLYTROP_2;
this.PFF2 = PFF2;
this.XIGN = XIGN;

//Входные параметры для отладки
//precision = 0; MOVk = 1; BPCN = 16500;
//N = 11; N2 = 3;
//X0 = 0; X1 = 5.64; X2 = 6.04; X3 = 8.91;
//XPL = 8.9100; XPP = 9.1300; XS = 11.03; LENSSCALE = 8.9100; CLENSCALE =
63;
//D0 = 1.2419; D1 = 1.2419; D2 = 1; D3 = 1; D = 1;
//NTS = 0.00725; ND1 = 0.051; NC1 = 4.2; AKP = 1; ALP = 0; AMP = 0;
//NPOR = 1; DPOR = 0.022;
//AFL = 0.000000061; FP = 99960000; CV = 136000; TPG = 1.17; alpha_COVOLUME =
1.005; POLYTROP = 1.222;
//W1 = 6.5; W2 = 0.8; QPPP = 1.67; QS = 4.55;
//PFF = 980000; RP = 1.6;
//NTS2 = 0.0035; ND2 = 0.023333; NC2 = 0.06; AKP2 = 1.1167; ALP2 = -
0.10448; AMP2 = 0;
//NPOR2 = 1; DPOR2 = 0.009333;
//AFL2 = 0.00000010017; FP2 = 99960000; CV2 = 136000; TPG2 = 1.17;
alpha_COVOLUME_2 = 1.005; POLYTROP_2 = 1.222;
//PFF2 = 490000; XIGN = 30;
}

//-----
void D31(double U, double P, double R, int IZN, double SK, double H2, double
A2)
{
Y1 = R * U / SK;
if (Math.Abs(U) >= 0.000101) goto g30;
U3 = U;
P3 = P;
R3 = R;
GR = 0.0;
goto g31;
g30:
PSH = P;
Y2 = (R * U * U + P) / SK - PSH * (1.0 / SK - 1.0);
Y3 = H2 * P / R - P * A2 + G4 * U * U / 2.0;
Y4 = H2 * Y2 / Y1;
Y5 = 2.0 * G3 * (A2 * Y2 + Y3);
Y6 = Y4 + A2 * Y1;
Y4 = Y6 * Y6 - Y5;
if (Y4 >= 0) GR = 1;
if (Y4 < 0) GR = -1;
U3 = (Y6 - IZN * Math.Sqrt(Math.Abs(Y4))) / G3;
R3 = Y1 / U3;
P3 = Y2 - Y1 * U3;
g31:
return;
}

//-----
void D4(double U, double P, double R, int IZN, double SK, double H1, double A1)

```

```

{
    if (Math.Abs(U) >= 0.000101) goto g40;
    U4 = U;
    P4 = P;
    R4 = R;
    goto g41;
g40:
    Y1 = U * R * SK;
    Y2 = (R * U * U + P) * SK + 1.0 * ((1 - IZN) * (1 - SK) / 2.0);
    Y3 = H1 * P / R - P * A1 + G2 * U * U / 2.0;
    Y4 = H1 * Y2 / Y1 + Y1 * A1;
    Y5 = 1.0 - (1.0 + IZN) * (1.0 - SK) / 2.0;
    Y6 = (Y3 * Y5 + Y2 * A1) * (4 * H1 - 2.0 * Y5 * G2);
    U4 = (Y4 + IZN * Math.Sqrt(Math.Abs(Y4 * Y4 - Y6))) / (2.0 * H1 - Y5 * G2);
    R4 = Y1 / U4;
    P4 = (Y2 - Y1 * U4) / Y5;
g41:
    return;
}

//-----
-----

void D5(double U, double P, double R, double A, double τ)
{
    Y1 = τ - 1.0;
    Y3 = τ + 1.0;
    Y2 = Y1 / Y3;
    P5 = Y2 * P * ((U * U * R * 2.0 * (1.0 - A * R) / (P * Y1)) - 1.0);
    R5 = 1.0 / ((1.0 / R - A) * (P + Y2 * P5) / (Y2 * P + P5) + A);
    U5 = U * R / R5;
    return;
}

double R1P1(double RM, double PM, double P1local, double A1, double G5)
{
    return 1.0 / ((1.0 / RM - A1) * (G5 * P1local + PM) / (G5 * PM + P1local) +
A1);
}

double U1P1(double RM, double PM, double R1local, double P1local, double UM)
{
    return UM + Math.Sqrt((R1local - RM) * (P1local - PM) / (R1local * RM));
}

double R1PM(double RM, double PM, double P1local, double A1, double H1)
{
    return 1.0 / (Math.Pow(PM / P1local, 1.0 / H1) * (1.0 / RM - A1) + A1);
}

double U1PM(double RM, double PM, double R1local, double P1local, double A1,
double H1, double G2, double UM)
{
    return UM + (2.0 / G2) * (Math.Sqrt((H1 * P1local / R1local) * (1.0 - A1 *
R1local)) - Math.Sqrt((H1 * PM / RM) * (1.0 - A1 * RM)));
}

double R2P2(double RB, double PB, double P2local, double A2, double G6)
{
    return 1.0 / ((1.0 / RB - A2) * (PB + G6 * P2local) / (G6 * PB + P2local) +
A2);
}

```

```

double U2P2(double RB, double PB, double R2local, double P2local, double UB)
{
    return UB - Math.Sqrt((R2local - RB) * (P2local - PB) / (R2local * RB));
}

double R2PB(double RB, double PB, double P2local, double A2, double H2)
{
    return 1.0 / (Math.Pow(PB / P2local, 1.0 / H2) * (1.0 / RB - A2) + A2);
}

double U2PB(double RB, double PB, double R2local, double P2local, double A2,
double H2, double G4, double UB)
{
    return UB - (2.0 / G4) * (Math.Sqrt(H2 * P2local * (1.0 / R2local - A2)) -
Math.Sqrt(H2 * PB * (1.0 / RB - A2)));
}

/// <summary>
/// Метод расчета распада произвольного разрыва на скачке сечения
/// </summary>
/// <param name="SK"></param>
/// <param name="H1"></param>
/// <param name="UM"></param>
/// <param name="PM"></param>
/// <param name="RM"></param>
/// <param name="A1"></param>
/// <param name="H2"></param>
/// <param name="UB"></param>
/// <param name="PB"></param>
/// <param name="RB"></param>
/// <param name="A2"></param>
/// <param name="BU"></param>
/// <param name="BP"></param>
/// <param name="BR"></param>
/// <param name="AMU"></param>
/// <param name="AMP"></param>
/// <param name="AMR"></param>
/// <param name="HB"></param>
/// <param name="AB"></param>
void calcParamsOfGas(double SK, double H1, double UM, ref double PM, ref double
RM, double A1,
    double H2, double UB, ref double PB, ref double RB, double A2, ref
double BU,
    ref double BP, ref double BR, ref double AMU, ref double AMP, ref
double AMR, ref double HB, ref double AB)
{
    if ((PM < 0) || (RM < 0) || (PB < 0) || (RB < 0)) goto g101;
g102:
    P2 = (PB + PM) * 0.0001001;
    P1 = (PB + PM) * 0.0001001;
    if (P2 <= PB) goto g66;
    R2 = R2P2(RB, PB, P2, A2, G6);
    U2 = U2P2(RB, PB, R2, P2, UB);
    goto g67;
g66:
    R2 = R2PB(RB, PB, P2, A2, H2);
    U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g67:
    if (P1 <= PM) goto g50;
    R1 = R1P1(RM, PM, P1, A1, G5);
    U1 = U1P1(RM, PM, R1, P1, UM);
    goto g51;
}

```

```

g50:
  R1 = R1PM(RM, PM, P1, A1, H1);
  U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g51:
  if (U1 > U2) goto g103;
  G7 = (PB + PM) * 0.000001;
  P2 = G7;
g104:
  P2 = P2 + PB;
  if (P2 <= PB) goto g68;
  R2 = R2P2(RB, PB, P2, A2, G6);
  U2 = U2P2(RB, PB, R2, P2, UB);
  goto g69;
g68:
  R2 = R2PB(RB, PB, P2, A2, H2);
  U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g69:
  if (U2 > 0) goto g104;
  Y7 = P2 - PB;
  Y8 = P2;
g105:
  P2 = (Y7 + Y8) / 2.0;
  if (P2 <= PB) goto g70;
  R2 = R2P2(RB, PB, P2, A2, G6);
  U2 = U2P2(RB, PB, R2, P2, UB);
  goto g71;
g70:
  R2 = R2PB(RB, PB, P2, A2, H2);
  U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g71:
  if (U2 > 0) goto g21;
  Y8 = P2;
  goto g22;
g21:
  Y7 = P2;
g22:
  if (Math.Abs(Y8 - Y7) > G7) goto g105;
  P10 = (Y7 + Y8) / 2.0;
  if (SK < 0.00001) goto g106;
  P1 = P10;
  if (P1 <= PM) goto g52;
  R1 = R1P1(RM, PM, P1, A1, G5);
  U1 = U1P1(RM, PM, R1, P1, UM);
  goto g53;
g52:
  R1 = R1PM(RM, PM, P1, A1, H1);
  U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g53:
  if (U1 < 0) goto g100;
  HB = H2;
  AB = A2;
  A7 = G7;
  A8 = P10;
g107:
  P2 = (A7 + A8) / 2.0;
  if (P2 <= PB) goto g72;
  R2 = R2P2(RB, PB, P2, A2, G6);
  U2 = U2P2(RB, PB, R2, P2, UB);
  goto g73;
g72:
  R2 = R2PB(RB, PB, P2, A2, H2);
  U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);

```

```

g73:
  if (U2 * U2 <= H2 * P2 / (R2 * (1 - A2 * R2))) goto g201;
  A7 = P2;
  goto g107;
g201:
  D31(U2, P2, R2, 1, SK, H2, A2);
  if (Math.Abs(A8 - A7) < G7) goto g108;
  if ((GR < 0) || (U3 * U3 > H2 * P3 / (R3 * (1 - A2 * R3)))) goto g23;
  A8 = P2;
  goto g18;
g23:
  A7 = P2;
g18:
  goto g107;
g108:
  if (UB < Math.Sqrt(H2 * PB / (RB * (1 - A2 * RB)))) goto g109;
  D5(UB, PB, RB, A2, H2);
  if (P2 > P5) goto g109;
  D31(U5, P5, R5, 1, SK, H2, A2);
  P1 = P3;
  if (P1 <= PM) goto g54;
  R1 = R1P1(RM, PM, P1, A1, G5);
  U1 = U1P1(RM, PM, R1, P1, UM);
  goto g55;
g54:
  R1 = R1PM(RM, PM, P1, A1, H1);
  U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g55:
  if (U1 >= U3) goto g110;
  Y7 = P5;
  goto g111;
g110:
  D31(UB, PB, RB, -1, SK, H2, A2);
  BU = UB;
  BP = PB;
  BR = RB;
  AMU = U3;
  AMP = P3;
  AMR = R3;
  goto g1000;
g109:
  P1 = P3;
  if (P1 <= PM) goto g56;
  R1 = R1P1(RM, PM, P1, A1, G5);
  U1 = U1P1(RM, PM, R1, P1, UM);
  goto g57;
g56:
  R1 = R1PM(RM, PM, P1, A1, H1);
  U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g57:
  if (U1 >= U3) goto g112;
  Y7 = P2;
  goto g111;
g112:
  BU = U2;
  BP = P2;
  BR = R2;
  AMU = U3;
  AMP = P3;
  AMR = R3;
  goto g1000;
g111:

```

```

Y8 = P10;
g113:
P2 = (Y7 + Y8) / 2.0;
if (P2 <= PB) goto g74;
R2 = R2P2(RB, PB, P2, A2, G6);
U2 = U2P2(RB, PB, R2, P2, UB);
goto g75;
g74:
R2 = R2PB(RB, PB, P2, A2, H2);
U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g75:
D31(U2, P2, R2, 1, SK, H2, A2);
if (Math.Abs(Y8 - Y7) < G7) goto g112;
P1 = P3;
if (P1 <= PM) goto g58;
R1 = R1P1(RM, PM, P1, A1, G5);
U1 = U1P1(RM, PM, R1, P1, UM);
goto g59;
g58:
R1 = R1PM(RM, PM, P1, A1, H1);
U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g59:
if (U1 > U3) goto g24;
Y7 = P2;
goto g17;
g24:
Y8 = P2;
g17:
goto g113;
g106:
HB = H2;
AB = A2;
BU = U2;
BP = P2;
BR = R2;
AMU = UM;
AMP = PM;
AMR = RM;
goto g1000;
g103:
AMU = 0;
AMP = P1;
AMR = R1;
BU = 0;
BP = P2;
BR = R2;
HB = H2;
AB = A2;
goto g1000;
g101:
PM = Math.Abs(PM);
RM = Math.Abs(RM);
PB = Math.Abs(PB);
RB = Math.Abs(RB);
goto g102;
g100:
AB = A1;
HB = H1;
P1 = G7;
g114:
P1 = P1 + PM;
if (P1 <= PM) goto g60;

```

```

R1 = R1P1(RM, PM, P1, A1, G5);
U1 = U1P1(RM, PM, R1, P1, UM);
goto g61;
g60:
R1 = R1PM(RM, PM, P1, A1, H1);
U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g61:
if (U1 < 0) goto g114;
P10 = P1;
P9 = P1;
Y7 = G7;
Y8 = P1;
g115:
P1 = (Y7 + Y8) / 2.0;
if (P1 <= PM) goto g62;
R1 = R1P1(RM, PM, P1, A1, G5);
U1 = U1P1(RM, PM, R1, P1, UM);
goto g63;
g62:
R1 = R1PM(RM, PM, P1, A1, H1);
U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g63:
if (Math.Abs(Y8 - Y7) < G7) goto g116;
if (U1 > -Math.Sqrt(H1 * P1 / (R1 * (1 - A1 * R1)))) goto g20;
Y7 = P1;
goto g19;
g20:
Y8 = P1;
g19:
goto g115;
g116:
if (UM > -Math.Sqrt(H1 * PM / RM)) goto g117;
D5(UM, PM, RM, A1, H1);
if (P5 > P1) goto g118;
g117:
D4(U1, P1, R1, 1, SK, H1, A1);
P2 = P4;
if (P2 <= PB) goto g76;
R2 = R2P2(RB, PB, P2, A2, G6);
U2 = U2P2(RB, PB, R2, P2, UB);
goto g77;
g76:
R2 = R2PB(RB, PB, P2, A2, H2);
U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g77:
if (U2 > U4) goto g119;
D4(U1, P1, R1, -1, SK, H1, A1);
D5(U4, P4, R4, A1, H1);
P2 = P5;
if (P2 <= PB) goto g78;
R2 = R2P2(RB, PB, P2, A2, G6);
U2 = U2P2(RB, PB, R2, P2, UB);
goto g79;
g78:
R2 = R2PB(RB, PB, P2, A2, H2);
U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g79:
AMU = U1;
AMP = P1;
AMR = R1;
if (U2 > U5) goto g120;
g121:

```

```

    BU = U4;
    BP = P4;
    BR = R4;
    goto g1000;
g118:
    D4(U5, P5, R5, +1, SK, H1, A1);
    P2 = P4;
    if (P2 <= PB) goto g80;
    R2 = R2P2(RB, PB, P2, A2, G6);
    U2 = U2P2(RB, PB, R2, P2, UB);
    goto g81;
g80:
    R2 = R2PB(RB, PB, P2, A2, H2);
    U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g81:
    if (U2 <= U4) goto g25;
    P1 = P5;
    goto g119;
g25:
    D4(UM, PM, RM, -1, SK, H1, A1);
    D5(U4, P4, R4, A1, H1);
    P2 = P5;
    if (P2 <= PB) goto g82;
    R2 = R2P2(RB, PB, P2, A2, G6);
    U2 = U2P2(RB, PB, R2, P2, UB);
    goto g83;
g82:
    R2 = R2PB(RB, PB, P2, A2, H2);
    U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g83:
    AMU = UM;
    AMP = PM;
    AMR = RM;
    if (U2 > U5) goto g120;
g122:
    BU = U4;
    BP = P4;
    BR = R4;
    goto g1000;
g119:
    Y7 = P1;
    Y8 = P10;
g123:
    P1 = (Y7 + Y8) / 2.0;
    if (P1 <= PM) goto g64;
    R1 = R1P1(RM, PM, P1, A1, G5);
    U1 = U1P1(RM, PM, R1, P1, UM);
    goto g65;
g64:
    R1 = R1PM(RM, PM, P1, A1, H1);
    U1 = U1PM(RM, PM, R1, P1, A1, H1, G2, UM);
g65:
    if (U1 <= 0) goto g26;
    Y8 = P1;
    goto g123;
g26:
    D4(U1, P1, R1, 1, SK, H1, A1);
    if (Math.Abs(Y8 - Y7) < G7) goto g124;
    P2 = P4;
    if (P2 <= PB) goto g84;
    R2 = R2P2(RB, PB, P2, A2, G6);
    U2 = U2P2(RB, PB, R2, P2, UB);

```

```

    goto g85;
g84:
    R2 = R2PB(RB, PB, P2, A2, H2);
    U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g85:

    if (U2 > U4) goto g27;
    Y8 = P1;
    goto g28;
g27:
    Y7 = P1;
g28:
    goto g123;
g124:
    AMU = U1;
    AMP = P1;
    AMR = R1;
    goto g122;
g120:
    Y7 = SK;
    Y8 = 1.0;
g125:
    Y9 = (Y7 + Y8) / 2.0;
    D4(AMU, AMP, AMR, -1, SK / Y9, H1, A1);
    D5(U4, P4, R4, A1, H1);
    D4(U5, P5, R5, 1, Y9, H1, A1);
    if (Math.Abs(Y8 - Y7) < 0.00001) goto g126;
    P2 = P4;
    if (P2 <= PB) goto g86;
    R2 = R2P2(RB, PB, P2, A2, G6);
    U2 = U2P2(RB, PB, R2, P2, UB);
    goto g87;
g86:
    R2 = R2PB(RB, PB, P2, A2, H2);
    U2 = U2PB(RB, PB, R2, P2, A2, H2, G4, UB);
g87:
    if (U2 > U4) goto g29;
    Y7 = Y9;
    goto g30;
g29:
    Y8 = Y9;
g30:
    goto g125;
g126:
    BU = U4;
    BP = P4;
    BR = R4;
g1000:
    return;
}

```

```

/// <summary>
/// Расчет распада произвольного разрыва для среды не имеющей собственного
давления

```

```

/// </summary>
/// <param name="RKL"></param>
/// <param name="WL"></param>
/// <param name="RKP"></param>
/// <param name="WP"></param>
/// <param name="RRK"></param>
/// <param name="WW"></param>

```

```

void RPRK(double RKL, double WL, double RKP, double WP, ref double RRK, ref
double WW)
{

    if ((RKL <= 0) && (RKP <= 0)) goto g15;
    if ((WL >= 0.0) && (WP >= 0.0)) goto g10;
    if ((WL <= 0.0) && (WP <= 0.0)) goto g11;
    if ((WL < 0.0) && (WP > 0.0)) goto g12;
    if ((WL > 0.0) && (WP < 0.0)) goto g13;
g10:
    RRK = RKL;
    WW = WL;
    goto g16;
g11:
    RRK = RKP;
    WW = WP;
    goto g16;
g12:
    RRK = 0.0;
    WW = 0.0;
    goto g16;
g13:
    DRR = (WP * Math.Sqrt(RKP) + WL * Math.Sqrt(RKL)) / (Math.Sqrt(RKP) +
Math.Sqrt(RKL));
    if (Math.Abs(DRR) < 1E-09) goto g14;
    if (DRR > 0.0) goto g10;
    goto g11;
g14:
    RRK = (RKL + RKP) / 2.0;
    WW = 0.0;
    goto g16;
g15:
    RRK = 0.0;
    WW = 0.0;
g16:
    return;
}

/// <summary>
/// Метод счета шага течения смеси во времени
/// </summary>
/// <param name="R"></param>
/// <param name="P"></param>
/// <param name="U"></param>
/// <param name="W"></param>
/// <param name="FI"></param>
/// <param name="Z"></param>
/// <param name="S"></param>
/// <param name="SV"></param>
/// <param name="α_COVOLUME"></param>
/// <param name="POLYTROP"></param>
/// <param name="τ"></param>
/// <param name="AKP"></param>
/// <param name="ALP"></param>
/// <param name="AMP"></param>
/// <param name="NTS"></param>
/// <param name="AFL"></param>
/// <param name="W0"></param>
/// <param name="S02"></param>
/// <param name="QP"></param>
/// <param name="XS"></param>

```

```

/// <param name="US"></param>
/// <param name="USN"></param>
/// <param name="ACSYS"></param>
/// <param name="APCH1"></param>
/// <param name="N"></param>
/// <param name="KMOV"></param>
/// <param name="MOVk"></param>
/// <param name="P2local"></param>
/// <param name="QFORCER"></param>
void timeStep(ref double[] R, ref double[] P, ref double[] U, ref double[] W,
ref double[] FI, ref double[] Z, ref double[] S, ref double[] SV,
double α_COVOLUME, double POLYTROP, double τ, double AKP, double ALP,
double AMP,
double NTS, double AFL, double W0, double S02, double QP, ref double
XS,
double US, ref double USN, double ACSYS, double APCH1, ref int N,
double KMOV,
double MOVk, double P2local, double QFORCER)
{
    G1 = POLYTROP + 1.0;
    G2 = POLYTROP - 1.0;
    G3 = POLYTROP + 1.0;
    G4 = POLYTROP - 1.0;
    G5 = G2 / G1;
    G6 = G4 / G3;

    RKP = RP / RM * (1.0 - FI[1]);
    RPRK(RKP, -W[1], RKP, W[1], ref RRK1, ref WW1);
    calcParamsOfGas(1.0, POLYTROP, U[1], ref P[1], ref R[1], α_COVOLUME,
POLYTROP, -U[1], ref P[1], ref R[1], α_COVOLUME, ref UBB, ref PBB, ref RBB, ref UMM,
ref PMM, ref RMM, ref EKB, ref ALB);

    // Расчет потоков

    ARK1 = SV[1] * RRK1 * WW1;
    AWRK1 = SV[1] * RRK1 * WW1 * WW1;
    AR1 = S[1] * UMM * RMM * FI[1];
    AUR1 = S[1] * FI[1] * (RMM * UMM * UMM + PMM);
    AER1 = S[1] * FI[1] * UMM * (RMM * PMM * (1.0 / RMM - α_COVOLUME) /
(POLYTROP - 1.0) + RMM * UMM * UMM / 2.0 + PMM);
    ZN = Z[1] - TH / 2.0 / τ * (W[1] - Math.Abs(W[1])) * (Z[2] - Z[1]) + AFL *
P[1] * PM / NTS * TH * BPM;
    if (ZN > 0.985) ZN = 1.0;
    PST = P[1];
    WST = W[1];
    FIST = FI[1];

    // Расчет параметров во внутренних ячейка

    for (I = 2; I <= N; I++)
    {
        RKL = RP / RM * (1.0 - FI[I - 1]);
        RKP = RP / RM * (1.0 - FI[I]);
        RPRK(RKL, W[I - 1], RKP, W[I], ref RRK2, ref WW2);
        if ((RKL > RKPR) || (RKP > RKPR)) WW2 = 0.0;
        if (S[I] * FI[I] <= S[I - 1] * FI[I - 1]) goto g204;
        SK = S[I - 1] * FI[I - 1] / S[I] / FI[I];
        calcParamsOfGas(SK, POLYTROP, -U[I - 1], ref P[I - 1], ref R[I - 1],
α_COVOLUME, POLYTROP, -U[I], ref P[I], ref R[I], α_COVOLUME, ref UR, ref PR, ref RR,
ref UL, ref PL, ref RL, ref EKB, ref ALB);
        UR = -UR;
        UL = -UL;
    }
}

```

```

    goto g205;
g204:
    SK = S[I] * FI[I] / S[I - 1] / FI[I - 1];
    calcParamsOfGas(SK, POLYTROP, U[I], ref P[I], ref R[I], α_COVOLUME,
POLYTROP, U[I - 1], ref P[I - 1], ref R[I - 1], α_COVOLUME, ref UL, ref PL, ref RL, ref
UR, ref PR, ref RR, ref EKB, ref ALB);
g205:
    ARK2 = SV[I - 1] * RRK2 * WW2;
    AWRK2 = ARK2 * WW2;
    AR2 = S[I - 1] * UL * RL * FI[I - 1];
    AR3 = S[I] * UR * RR * FI[I];
    AUR2 = S[I - 1] * FI[I - 1] * (RL * UL * UL + PL);
    AUR3 = S[I] * FI[I] * (RR * UR * UR + PR);
    AER2 = S[I - 1] * FI[I - 1] * UL * (RL * PL * (1.0 / RL - α_COVOLUME) /
(POLYTROP - 1.0) + RL * UL * UL / 2.0 + PL);
    AER3 = S[I] * FI[I] * UR * (RR * PR * (1.0 / RR - α_COVOLUME) /
(POLYTROP - 1.0) + RR * UR * UR / 2.0 + PR);
    if (I == 2) DWSDX = ((1.0 - FI[I]) * W[I] * S[I] - (1.0 - FI[I - 1]) *
W[I - 1] * S[I - 1]) / τ;
    if (I != 2) DWSDX = ((1.0 - FI[I]) * W[I] * S[I] - (1.0 - FIST) * WST *
S[I - 2]) / 2.0 / τ;
    PSIZ = AKP * Z[I - 1] * (1.0 + ALP * Z[I - 1] + AMP * Z[I - 1] * Z[I -
1]);

    if (PSIZ > 1.0) PSIZ = 1.0;
    if (FI[I - 1] > 0.9995) goto g214;
    if (PSIZ == 1.0) goto g214;
    ANI = (1.0 - FI[I - 1]) / W0 / (1.0 - PSIZ);
    goto g215;
g214:
    ANI = 0.0;
g215:
    SIGZ = 1.0 + 2.0 * ALP * Z[I - 1] + 3.0 * AMP * Z[I - 1] * Z[I - 1];
    GM = S[I - 1] * ANI * S02 * SIGZ * RP * AFL * (P[I - 1] * PM) *
LENSCALE / RM / CM;
    RE = R[I - 1] * RM * Math.Abs(U[I - 1] - W[I - 1]) * CM * FI[I - 1] *
Math.Sqrt(S02) / BG;
    if (RE < 1.0) CX = 0.0;
    if (RE >= 3.0E+05) CX = 0.1;
    if ((RE >= 1.0) && (RE < 3.0E+05)) CX = 24.0 / RE + 0.48;
    DP = Math.Pow(W0 * (1.0 - PSIZ) * 6.0 / PI, 1.0 / 3.0);
    ATR = CX * R[I - 1] * ANI * S[I - 1] * PI * Math.Pow(DP, 2) / 4 *
Math.Pow(1.0 - PSIZ, 2.0 / 3.0) / 2.0 * LENSCALE;
    TTR = ATR * Math.Abs(U[I - 1] - W[I - 1]) * (U[I - 1] - W[I - 1]);
    Q = GM * (QP / CM / CM + W[I - 1] * W[I - 1] / 2.0);
    FMSG = GM;
    FIMPS = (GM * W[I - 1] - TTR - R[I - 1] * FI[I - 1] * S[I - 1] * ACSYS
* LENSCALE / CM / CM);
    FENS = -P[I - 1] * DWSDX + (Q - TTR * W[I - 1] - R[I - 1] * S[I - 1] *
FI[I - 1] * U[I - 1] * ACSYS * LENSCALE / Math.Pow(CM, 2));
    FMASK = -GM;
    APCH = Math.Abs((-ATR * Math.Abs(U[I - 1] - W[I - 1]))) / R[I - 1] / S[I
- 1] / FI[I - 1]);
    if (APCH1 < APCH) APCH1 = APCH;
    RSFI = R[I - 1] * S[I - 1] * FI[I - 1] - (AR2 - AR1) * TH / τ + FMSG *
TH;
    RUSFI = R[I - 1] * U[I - 1] * S[I - 1] * FI[I - 1] - (AUR2 - AUR1) * TH
/ τ + FIMPS * TH;
    RESFI = R[I - 1] * S[I - 1] * FI[I - 1] * (P[I - 1] * (1.0 / R[I - 1] -
α_COVOLUME) / (POLYTROP - 1.0) + U[I - 1] * U[I - 1] / 2.0) - (AER2 - AER1) * TH / τ +
FENS * TH;
    RKSFI = RP / RM * (1.0 - FI[I - 1]) * S[I - 1] - (ARK2 - ARK1) * TH / τ
+ FMASK * TH;

```

```

    FIN = 1.0 - RKSFI / RP / S[I - 1] * RM;
    if (FIN > 0.9995) FIN = 1.0;
    RN = RSFI / FIN / S[I - 1];
    UN = RUSFI / RSFI;
    EN = RESFI / RSFI;
    PN = (EN - UN * UN / 2.0) * (POLYTROP - 1.0) / (1.0 / RN -  $\alpha$ _COVOLUME);
    if (I == 2) DPDX = (P[I] - PST) /  $\tau$ ;
    if (I != 2) DPDX = (P[I] - PST) / 2.0 /  $\tau$ ;
    if (KMOV == 0) goto g206;
    if (FIN >= 0.9995) goto g206;
    if (FIN < FIPR) KIN = 0;
    if (FIN >= FIPR) KIN = 1;
    if (Math.Abs(ATR * (U[I - 1] - W[I - 1])) * Math.Abs(U[I - 1] - W[I - 1])) > RP / RM * (1.0 - FI[I - 1]) * S[I - 1] * ACSYS * BPM / CM) KIN = 1;
    RKWSFI = (RP / RM * (1.0 - FI[I - 1]) * S[I - 1] * W[I - 1] - (AWRK2 - AWRK1) * TH /  $\tau$  - (1.0 - FI[I - 1]) * S[I - 1] * DPDX * TH * KIN + ATR * (U[I - 1] - W[I - 1]) * Math.Abs(U[I - 1] - W[I - 1]) * TH * KIN - GM * W[I - 1] * TH - RP / RM * (1.0 - FI[I - 1]) * S[I - 1] * ACSYS * BPM / CM * TH * KIN) * MOVk;
    WN = RKWSFI / RKSFI;
    goto g207;
g206:
    WN = 0.0;
g207:
    if (I != N) ZN1 = Z[I] - TH / 2.0 /  $\tau$  * ((W[I] + Math.Abs(W[I])) * (Z[I] - Z[I - 1]) + (W[I] - Math.Abs(W[I])) * (Z[I + 1] - Z[I])) + AFL * P[I] * PM / NTS * TH * LENSACLE / CM;
    if (I == N) ZN1 = Z[I] - TH / 2.0 /  $\tau$  * (W[I] + Math.Abs(W[I])) * (Z[I] - Z[I - 1]) + AFL * P[I] * PM / NTS * TH * LENSACLE / CM;
    if (ZN1 > 0.985) ZN1 = 1.0;
    Z[I - 1] = ZN1;
    ZN = ZN1;
    R[I - 1] = RN;
    U[I - 1] = UN;
    WST = W[I - 1];
    W[I - 1] = WN;
    PST = P[I - 1];
    P[I - 1] = PN;
    FIST = FI[I - 1];
    FI[I - 1] = FIN;
    ARK1 = ARK2;
    AWRK1 = AWRK2;
    AR1 = AR3;
    AUR1 = AUR3;
    AER1 = AER3;
}

USI = U[N] - US;
calcParamsOfGas(1.0, POLYTROP, -USI, ref P[N], ref R[N],  $\alpha$ _COVOLUME, POLYTROP, USI, ref P[N], ref R[N],  $\alpha$ _COVOLUME, ref UBB, ref PBB, ref RBB, ref UMM, ref PMM, ref RMM, ref EKB, ref ALB);
PMSB = PMM * PM / 100000.0;
PIND = PIND + KINDI * PMSB * TH * BPM * QS / (QPPP + W2 + QS) + KINDII * PMSB * TH * BPM;
if (KMOV == 0) goto g219;
USN = US + (PMM - P2local) * SV[N] * BPM * PM * SM / CM / QFORCER * TH - ACSYS * TH * BPM / CM;
if (USN < 0.0) goto g219;
goto g220;
g219:
    USN = 0.0;
g220:
    XSN = XS + (US + USN) / 2 * TH;

```

```

DX = XS - τ * (N - 1);
DXN = XSN - τ * (N - 1);
XS = XSN;
UMM = (US + USN) / 2.0;
DWS DX = ((1.0 - FI[N]) * S[N] * W[N] - (1.0 - FIST) * S[N - 1] * WST) / ((τ
+ DX) / 2.0);
PSIZ = AKP * Z[N] * (1.0 + ALP * Z[N] + AMP * Z[N] * Z[N]);
if (PSIZ > 1.0) PSIZ = 1.0;
if (FI[N] > 0.9995) goto g216;
if (PSIZ == 1.0) goto g216;
ANI = (1.0 - FI[N]) / W0 / (1.0 - PSIZ);
goto g217;
g216:
ANI = 0.0;
g217:
SIGZ = 1.0 + 2.0 * ALP * Z[N] + 3.0 * AMP * Z[N] * Z[N];
GM = S[N] * ANI * S02 * SIGZ * RP * AFL * (P[N] * PM) * LENS SCALE / RM / CM;
RE = R[N] * RM * Math.Abs(U[N] - W[N]) * CM * FI[N] * Math.Sqrt(S02) / BG;
//if (RE < 0.0) printf("%lg", RE);
if (RE < 1.0) CX = 0.0;
if (RE >= 3.0E+05) CX = 0.1;
if ((RE >= 1.0) && (RE < 3.0E+05)) CX = 24.0 / RE + 0.48;
DP = Math.Pow(W0 * (1.0 - PSIZ) * 6.0 / PI, 1.0 / 3.0);
ATR = CX * R[N] * ANI * S[N] * PI * Math.Pow(DP, 2) / 4 * Math.Pow(1.0 -
PSIZ, 2.0 / 3.0) / 2.0 * LENS SCALE;
TTR = ATR * Math.Abs(U[N] - W[N]) * (U[N] - W[N]);
Q = GM * (QP / CM / CM + W[N] * W[N] / 2.0);
FMASG = GM;
FIMPS = (GM * W[N] - TTR - R[N] * FI[N] * S[N] * ACSYS * LENS SCALE / CM /
CM);
FENS = -P[N] * DWS DX + (Q - TTR * W[N] - R[N] * S[N] * FI[N] * U[N] * ACSYS
* LENS SCALE / Math.Pow(CM, 2));
FMASK = -GM;
APCH = Math.Abs((-ATR * Math.Abs(U[N] - W[N]))) / R[N] / S[N] / FI[N]);
RSFI = R[N] * SEND * FI[N] * DX / DXN - (0.0 - AR1) * TH / DXN + FMASG * TH
* (DX + DXN) / 2.0 / DXN;
RUSFI = R[N] * U[N] * SEND * FI[N] * DX / DXN - (PMM * FI[N] * SV[N] -
AUR1) * TH / DXN + FIMPS * TH * (DX + DXN) / 2.0 / DXN;
RESFI = R[N] * SEND * FI[N] * (P[N] * (1.0 / R[N] - α_COVOLUME) / (POLYTROP
- 1.0) + U[N] * U[N] / 2.0) * DX / DXN - (SV[N] * UMM * PMM - AER1) * TH / DXN + FENS *
TH * (DX + DXN) / 2.0 / DXN;
RKSFI = RP / RM * (1.0 - FI[N]) * SEND * DX / DXN - (0.0 - ARK1) * TH / DXN
+ FMASK * TH * (DX + DXN) / 2.0 / DXN;
WNEND = S[N] * τ + S[N + 1] * (DXN - τ);
SNEND = WNEND / DXN;
FIN = 1.0 - RKSFI / RP / SNEND * RM;
if (FIN > 0.9995) FIN = 1.0;
RN = RSFI / FIN / SNEND;
UN = RUSFI / RSFI;
EN = RESFI / RSFI;
PN = (EN - UN * UN / 2.0) * (POLYTROP - 1.0) / (1.0 / RN - α_COVOLUME);
DPDX = (P[N] - PST) / ((τ + DX) / 2.0);
if (KMOV == 0) goto g208;
if (FIN >= 0.9995) goto g208;
RKWSFI = (RP / RM * (1.0 - FI[N]) * SEND * W[N] * DX / DXN - (0.0 - AWRK1)
* TH / DXN - (1.0 - FI[N]) * SEND * DPDX * TH * (DX + DXN) / 2.0 / DXN + (ATR * (U[N] -
W[N]) * Math.Abs(U[N] - W[N]) * TH - GM * W[N] * TH - RP / RM * (1.0 - FI[N]) * S[N] *
ACSYS * BPM / CM * TH) * (DX + DXN) / 2.0 / DXN) * MOVk;
WN = RKWSFI / RKSFI;
goto g209;
g208:
WN = 0.0;

```

g209:

```

SEND = SSEND;
Z[N] = ZN;
R[N] = RN;
U[N] = UN;
W[N] = WN;
P[N] = PN;
FI[N] = FIN;

//Проверка выхода в следующую ячейку
NDS = (int)((DXN + 0.000001) / τ);
if (NDS == 1) goto g210;
NN = N + NDS - 1;
for (I = N; I <= NN; I++)
{
    P[I] = P[N];
    R[I] = R[N];
    U[I] = U[N];
    W[I] = W[N];
    FI[I] = FI[N];
    Z[I] = Z[N];
}
N = NN;
DXN = XS - τ * (N - 1);

```

g210:

```

return;
}

/// <summary>
/// Метод вывода информации
/// </summary>
/// <param name="RGC"></param>
/// <param name="RGC2"></param>
/// <param name="α_COVOLUME"></param>
/// <param name="α_COVOLUME_2"></param>
/// <param name="TIM"></param>
/// <param name="τ"></param>
/// <param name="H2"></param>
/// <param name="POLYTRROP"></param>
/// <param name="QFORCER"></param>
/// <param name="POLYTRROP_2"></param>
/// <param name="RP"></param>
/// <param name="S"></param>
/// <param name="S2"></param>
/// <param name="US"></param>
/// <param name="US2"></param>
/// <param name="XPL"></param>
/// <param name="XS2"></param>
/// <param name="N"></param>
/// <param name="N2"></param>
void OUTPUT(double RGC, double RGC2, double α_COVOLUME, double α_COVOLUME_2,
double TIM,
            double τ, double H2, double POLYTRROP, double QFORCER, double
POLYTRROP_2, double RP,
            double[] S, double[] S2, double US, double US2, double XPL,
            double XS2, int N, int N2)
{
    UPOUT = US * CM / 10.0;
    USOUT = (US + US2) * CM / 10.0;

```

```

XPLOUT = XPL * LENSACLE;
XSOUT = (XPL + DXP + XS2) * LENSACLE;
DXN = XPL - τ * (N - 1);
DXON = XS2 - H2 * (N2 - 1);
for (I = 1; I <= N; I++)
{
    P[I] = P[I] * PM / 100000.0;
    U[I] = U[I] * CM / 10.0;
    R[I] = R[I] * RM;
    W[I] = W[I] * CM / 10.0;
}
for (I = 1; I <= N2; I++)
{
    _P2[I] = _P2[I] * PM / 100000.0;
    _U2[I] = (_U2[I] + US) * CM / 10.0;
    _R2[I] = _R2[I] * RM;
    _W2[I] = (_W2[I] + US) * CM / 10.0;
}
KK = N - 1;
AMS = 0.0;
AES = 0.0;
for (I = 1; I <= KK; I++)
{
    AMS = AMS + R[I] * S[I] * SM * FI[I] * τ * LENSACLE + (1.0 - FI[I]) *
RP * S[I] * SM * τ * LENSACLE;
    AES = AES + (Math.Pow(U[I] * 10, 2) / 2.0 + 100000 * P[I] * (1.0 / R[I]
- α_COVOLUME / RM) / (POLYTROP - 1.0)) * R[I] * S[I] * SM * FI[I] * τ * LENSACLE + (1.0
- FI[I]) * RP * S[I] * Math.Pow(W[I] * 10.0, 2) / 2.0 * SM * τ * LENSACLE + (1.0 -
FI[I]) * RP * S[I] * SM * τ * LENSACLE * QPP;
}
AMS = AMS + (R[N] * FI[N] + (1.0 - FI[N]) * RP) * SEND * SM * DXN *
LENSACLE;
AES = AES + (Math.Pow(U[N] * 10, 2) / 2.0 + 100000 * P[N] * (1.0 / R[N] -
α_COVOLUME / RM) / (POLYTROP - 1.0)) * R[N] * SEND * SM * FI[N] * DXN * LENSACLE + (1.0
- FI[N]) * RP * SEND * Math.Pow(W[N] * 10.0, 2) / 2.0 * SM * DXN * LENSACLE + (1.0 -
FI[N]) * RP * SEND * SM * DXN * LENSACLE * QPP + QPPP * Math.Pow(UPOUT * 10.0, 2) /
2.0;

EKG1 = 0.0;
EBG1 = 0.0;
EKK1 = 0.0;
EXK1 = 0.0;
EKG2 = 0.0;
EBG2 = 0.0;
EKK2 = 0.0;
EXK2 = 0.0;

for (I = 1; I <= KK; I++)
{
    EKG1 = EKG1 + Math.Pow(U[I] * 10, 2) / 2.0 * R[I] * S[I] * SM * FI[I] *
τ * LENSACLE;
    EBG1 = EBG1 + 100000 * P[I] * (1.0 / R[I] - α_COVOLUME / RM) /
(POLYTROP - 1.0) * R[I] * S[I] * SM * FI[I] * τ * LENSACLE;
    EKK1 = EKK1 + (1.0 - FI[I]) * RP * S[I] * Math.Pow(W[I] * 10.0, 2) /
2.0 * SM * τ * LENSACLE;
    EXK1 = EXK1 + (1.0 - FI[I]) * RP * S[I] * SM * τ * LENSACLE * QPP;
}
EKG1 = EKG1 + Math.Pow(U[N] * 10, 2) / 2.0 * R[N] * SEND * SM * FI[N] * DXN
* LENSACLE;
EBG1 = EBG1 + 100000 * P[N] * (1.0 / R[N] - α_COVOLUME / RM) / (POLYTROP -
1.0) * R[N] * SEND * SM * FI[N] * DXN * LENSACLE;

```

```

EKK1 = EKK1 + (1.0 - FI[N]) * RP * SEND * Math.Pow(W[N] * 10.0, 2) / 2.0 *
SM * DXN * LENSACLE;
EXK1 = EXK1 + (1.0 - FI[N]) * RP * SEND * SM * DXN * LENSACLE * QPP;

ESQ1 = EKG1 + EBG1 + EKK1 + EXK1;
ESB = QFORCER * Math.Pow(UPOUT * 10.0, 2) / 2.0;
EKP = QPPP * Math.Pow(UPOUT * 10.0, 2) / 2.0;
ESQ1SB = ESQ1 + ESB;
ESQ1P = ESQ1 + EKP;
KK = N2 - 1;
for (I = 1; I <= KK; I++)
{
    AMS = AMS + _R2[I] * S2[I] * SM * _FI2[I] * H2 * LENSACLE + (1.0 -
_FI2[I]) * RP * S2[I] * SM * H2 * LENSACLE;
    AES = AES + (Math.Pow(_U2[I] * 10, 2) / 2.0 + 100000 * _P2[I] * (1.0 /
_R2[I] - alpha_COVOLUME_2 / RM) / (POLYTROP_2 - 1.0)) * _R2[I] * S2[I] * SM * _FI2[I] * H2
* LENSACLE + (1.0 - _FI2[I]) * RP * S2[I] * Math.Pow(_W2[I] * 10.0, 2) / 2.0 * SM * H2
* LENSACLE + (1.0 - _FI2[I]) * RP * S2[I] * SM * H2 * LENSACLE * QPP2;
}
    AMS = AMS + (_R2[N2] * _FI2[N2] + (1.0 - _FI2[N2]) * RP) * S2[N2] * SM *
DXON * LENSACLE;
    AES = AES + (Math.Pow(_U2[N2] * 10, 2) / 2.0 + 100000 * _P2[N2] * (1.0 /
_R2[N2] - alpha_COVOLUME_2 / RM) / (POLYTROP_2 - 1.0)) * _R2[N2] * S2[N2] * SM * _FI2[N2] *
DXON * LENSACLE + (1.0 - _FI2[N2]) * RP * S2[N2] * Math.Pow(_W2[N2] * 10.0, 2) / 2.0 *
SM * DXON * LENSACLE + (1.0 - _FI2[N2]) * RP * S2[N2] * SM * DXON * LENSACLE * QPP2 +
QS * Math.Pow(USOUT * 10.0, 2) / 2.0;
    for (I = 1; I <= KK; I++)
    {
        EKG2 = EKG2 + Math.Pow(_U2[I] * 10, 2) / 2.0 * _R2[I] * S2[I] * SM *
_FI2[I] * H2 * LENSACLE;
        EBG2 = EBG2 + 100000 * _P2[I] * (1.0 / _R2[I] - alpha_COVOLUME_2 / RM) /
(POLYTROP_2 - 1.0) * _R2[I] * S2[I] * SM * _FI2[I] * H2 * LENSACLE;
        EKK2 = EKK2 + (1.0 - _FI2[I]) * RP * S2[I] * Math.Pow(_W2[I] * 10.0, 2)
/ 2.0 * SM * H2 * LENSACLE;
        EXK2 = EXK2 + (1.0 - _FI2[I]) * RP * S2[I] * SM * H2 * LENSACLE * QPP2;
    }
    EKG2 = EKG2 + Math.Pow(_U2[N2] * 10, 2) / 2.0 * _R2[N2] * S2[N2] * SM *
_FI2[N2] * DXON * LENSACLE;
    EBG2 = EBG2 + 100000 * _P2[N2] * (1.0 / _R2[N2] - alpha_COVOLUME_2 / RM) /
(POLYTROP_2 - 1.0) * _R2[N2] * S2[N2] * SM * _FI2[N2] * DXON * LENSACLE;
    EKK2 = EKK2 + (1.0 - _FI2[N2]) * RP * S2[N2] * Math.Pow(_W2[N2] * 10.0, 2)
/ 2.0 * SM * DXON * LENSACLE;
    EXK2 = EXK2 + (1.0 - _FI2[N2]) * RP * S2[N2] * SM * DXON * LENSACLE * QPP2;
    ESQ2 = EKG2 + EBG2 + EKK2 + EXK2;
    ESN = QS * Math.Pow(USOUT * 10.0, 2) / 2.0;
    ESQ2SN = ESQ2 + ESN;
    AESD = ESQ1P + ESQ2SN;

    OutString = OutString + "LAYER NUMBER = " + layersCount + "\r\n";
    OutString = OutString + "AMS = " + AMS + " " + AES + " AES = " + AESD +
"\r\n";
    OutString = OutString + "TIME = " + TIM + " SEC\r\n";
    OutString = OutString + "U = " + UPOUT + " " + USOUT + " X = " + XPLOUT +
" " + XSOUT + "\r\n";
    OutString = OutString + "P1 = " + P[1] + " " + _P2[1] + " PN = " + P[N] +
" " + _P2[N2] + "\r\n";
    OutString = OutString + "INDDIA = " + PIND + " MPA*C" + "\r\n";

    outAMS[layersCount] = AMS;
    outAES[layersCount] = AES;
    outAESD[layersCount] = AESD;
    outINDDIA[layersCount] = PIND;

```

```

outT[layersCount] = TIM;
outX[layersCount] = XPLOUT;
outX2[layersCount] = XSOUT;
outP[layersCount] = P[1];
outP2[layersCount] = _P2[1];
outPN[layersCount] = P[N];
outP2N[layersCount] = _P2[N2];
outU[layersCount] = UPOUT;
outU2[layersCount] = USOUT;

for (I = 1; I <= N; I = I + 500)
{
    OutString = OutString + "P = " + P[I] + "\r\n";
}

for (I = 1; I <= N2; I = I + 500)
{
    OutString = OutString + "P2 = " + _P2[I] + "\r\n";
}

for (I = 1; I <= N; I = I + 500)
{
    OutString = OutString + "FI = " + FI[I] + "\r\n";
}

for (I = 1; I <= N2; I = I + 500)
{
    OutString = OutString + "F2 = " + _FI2[I] + "\r\n";
}
OutString = OutString + "\r\n";

for (I = 1; I <= N; I++)
{
    P[I] = P[I] / PM * 100000.0;
    U[I] = U[I] / CM * 10.0;
    R[I] = R[I] / RM;
    W[I] = W[I] / CM * 10.0;
}
for (I = 1; I <= N2; I++)
{
    _P2[I] = _P2[I] / PM * 100000.0;
    _U2[I] = _U2[I] / CM * 10.0 - US;
    _R2[I] = _R2[I] / RM;
    _W2[I] = _W2[I] / CM * 10.0 - US;
}

return;
}

public int startCalculate()
{
    //Расчет начальных значений

    QFORCER = W2 + QPPP + QS;
    W0 = PI * (Math.Pow(ND1, 2) / 4 - NPOR * Math.Pow(DPOR, 2) / 4) * NC1;
    W02 = PI * (Math.Pow(ND2, 2) / 4 - NPOR2 * Math.Pow(DPOR2, 2) / 4) * NC2;
    S02 = PI * ((Math.Pow(ND1, 2) / 4 - NPOR * Math.Pow(DPOR, 2) / 4) * 2 +
(ND1 + NPOR * DPOR) * NC1);
    S022 = PI * ((Math.Pow(ND2, 2) / 4 - NPOR2 * Math.Pow(DPOR2, 2) / 4) * 2 +
(ND2 + NPOR2 * DPOR2) * NC2);
    KPER = 1;

```

```

KMOV = 0;
IIGN = 0;
τ = LENSACLE / N;
H2 = (XS - XPP) / N2;
NLD = (int)(CLENSCALE / τ) + 1;

RGC = CV * (POLYTROP - 1);
RGC2 = CV2 * (POLYTROP_2 - 1);
QP = FP / (POLYTROP - 1);
QP2 = FP2 / (POLYTROP_2 - 1);
QPP = QP;
QPP2 = QP2;

//Расчет масштабов задачи

WKM = PI * ((Math.Pow(D0, 2) + D1 * D0 + Math.Pow(D1, 2)) * (X1 - X0) +
(Math.Pow(D1, 2) + D1 * D2 + Math.Pow(D2, 2)) * (X2 - X1) + (Math.Pow(D2, 2) + D2 * D3
+ Math.Pow(D3, 2)) * (X3 - X2)) / 12.0;
SM = PI * Math.Pow(D, 2) / 4.0;
DELTA = W1 / WKM;
RM = DELTA;
PM = FP * RM;
CM = Math.Sqrt(PM / RM);
BPM = LENSACLE / CM;

AES0 = QP * W1 + QP2 * W2;
PFNA = FP * DELTA / (1 - α_COVOLUME * DELTA) / 100000;

// Установка начальных значений переменных

for (I = 1; I <= NLD; I++)
{
    XX1 = τ * (I - 1);
    XX2 = τ * I;
    if (τ * I < X1) goto g200;
    if (τ * I < X2) goto g201;
    if (τ * I <= X3 + 0.00001) goto g202;
    S[I] = PI * Math.Pow(D, 2) / 4.0 / SM;
    SV[I] = S[I];
    goto g212;
g200:
    DI1 = D0 - (D0 - D1) / (X1 - X0) * (XX1 - X0);
    DI2 = D0 - (D0 - D1) / (X1 - X0) * (XX2 - X0);
    S[I] = PI * (Math.Pow(DI1, 2) + DI1 * DI2 + Math.Pow(DI2, 2)) / 12 /
SM;

    SV[I] = PI * Math.Pow(DI2, 2) / 4.0 / SM;
    goto g212;
g201:
    DI2 = D1 - (D1 - D2) / (X2 - X1) * (XX2 - X1);
    S[I] = PI * (Math.Pow(DI1, 2) + DI1 * DI2 + Math.Pow(DI2, 2)) / 12.0 /
SM;

    if ((XX1 < X1) && (XX2 >= X1)) S[I] = PI * ((Math.Pow(DI1, 2) + DI1 *
D1 + Math.Pow(D1, 2)) / 12.0 / SM * (X1 - XX1) + (Math.Pow(D1, 2) + D1 * DI2 +
Math.Pow(DI1, 2)) / 12.0 / SM * (XX2 - X1)) / τ;
    SV[I] = PI * Math.Pow(DI2, 2) / 4.0 / SM;
    goto g212;
g202:
    DI2 = D2 - (D2 - D3) / (X3 - X2) * (XX2 - X2);
    S[I] = PI * (Math.Pow(DI1, 2) + DI1 * DI2 + Math.Pow(DI2, 2)) / 12.0 /
SM;

```

```

        if ((XX1 < X2) && (XX2 >= X2)) S[I] = PI * ((Math.Pow(DI1, 2) + DI1 *
D2 + Math.Pow(D2, 2)) / 12.0 / SM * (X2 - XX1) + (Math.Pow(D2, 2) + D2 * DI2 +
Math.Pow(DI2, 2)) / 12.0 / SM * (XX2 - X2)) / τ;
        SV[I] = PI * Math.Pow(DI2, 2) / 4.0 / SM;
g212:
        DI1 = DI2;
    }

    KK = N - 1;
    for (I = 1; I <= KK; I++)
    {
        S[I] = S[I] + (S[N] - 1.0) / (N - 1);
    }

    S[N] = 1;
    for (I = 1; I <= N_max; I++)
    {
        _S2[I] = 1;
    }

    SEND = S[N];
    WKMSCH = 0.0;
    for (I = 1; I <= N; I++)
    {
        WKMSCH = WKMSCH + S[I] * τ * SM;
    }

    PSI0 = (1.0 / DELTA - 1.0 / RP) / (FP / PFF + α_COVOLUME - 1.0 / RP);
    SIG0 = Math.Sqrt(1.0 + 4.0 * ALP / AKP * PSI0);
    Z0 = 2.0 * PSI0 / AKP / (1.0 + SIG0);
    TG = FP / RGC;
    for (I = 1; I <= N; I++)
    {
        P[I] = PFF / PM;
        R[I] = PFF / RGC / TG / RM;
        Z[I] = Z0;
        U[I] = 0.0;
        W[I] = 0.0;
        FI[I] = 1 - W1 / RP / WKM * (1 - AKP * Z0 * (1 + ALP * Z0 + AMP * Z0 *
Z0));
    }

    W22 = SM * (XS - XPP);

    DELTA2 = W2 / SM / (XS - XPP);

    PSI0 = (1.0 / DELTA2 - 1.0 / RP) / (FP2 / PFF2 + α_COVOLUME_2 - 1.0 / RP);
    SIG0 = Math.Sqrt(1.0 + 4.0 * ALP2 / AKP2 * PSI0);
    Z0 = 2.0 * PSI0 / AKP2 / (1.0 + SIG0);
    TG = FP2 / RGC2;

    for (I = 1; I <= N2; I++)
    {
        _P2[I] = PFF2 / PM;
        _R2[I] = PFF2 / RGC2 / TG / RM;
        _Z2[I] = Z0;
        _U2[I] = 0.0;
        _W2[I] = 0.0;
        _FI2[I] = 1.0 - W2 / RP / SM / (XS - XPP) * (1.0 - AKP2 * Z0 * (1.0 +
ALP2 * Z0 + AMP2 * Z0 * Z0));
    }

```

```
// Обезразмеривание некоторых параметров
```

```
τ = τ / LENSACLE;
H2 = H2 / LENSACLE;
CLENSACLE = CLENSACLE / LENSACLE;
α_COVOLUME = α_COVOLUME * RM;
α_COVOLUME_2 = α_COVOLUME_2 * RM;
XS = XS / LENSACLE;
XPL = XPL / LENSACLE;
XPP = XPP / LENSACLE;
XIGN = XIGN / LENSACLE;
DXP = XPP - XPL;
XSO = XS - XPP;
RKPR = RP / RM * (1.0 - FIPR);
```

```
// Устанавливаем начальные значения
```

```
layersCount = 0;
TIME = 0;
TIM = 0;
US = 0;
US2 = 0;
PIND = 0;
APCH1 = 0;
APCH12 = 0;
ACSYS = 0;
KINDI = 0;
KINDII = 0;
STEFBOLTZ = 0.0000000568;
```

```
// Расчет шага решения задачи по времени
```

```
g1000:
```

```
layersCount++;
UMAX = Math.Abs(U[1]) + Math.Sqrt(POLYTROP * P[1] / R[1]);
for (I = 2; I <= N; I++)
{
    UMAX1 = Math.Abs(U[I]) + Math.Sqrt(POLYTROP * P[I] / R[I]);
    if (UMAX1 > UMAX) UMAX = UMAX1;
}
TH = precision * τ / UMAX;
if (APCH1 == 0.0) goto g203;
THPCH = 0.8 / APCH1;
if (TH > THPCH) TH = THPCH;
APCH1 = 0.0;
```

```
g203:
```

```
UMAX2 = Math.Abs(_U2[1]) + Math.Sqrt(POLYTROP_2 * _P2[1] / _R2[1]);
for (I = 2; I <= N2; I++)
{
    UMAX21 = Math.Abs(_U2[I]) + Math.Sqrt(POLYTROP_2 * _P2[I] / _R2[I]);
    if (UMAX21 > UMAX2) UMAX2 = UMAX21;
}
TH2 = precision * H2 / UMAX2;
if (APCH12 == 0.0) goto g204;
TH2PCH = 0.8 / APCH12;
if (TH2 > TH2PCH) TH2 = TH2PCH;
APCH12 = 0.0;
```

```
g204:
```

```
if (TH > TH2) TH = TH2;
TIME = TIME + TH;
TIM = TIM + TH * BPM;
```

```

    if ((KMOV - 1) < 0) goto g233;
    if ((KMOV - 1) > 0) goto g234;
    if ((KMOV - 1) == 0) goto g234;
g234:
    G1 = POLYTROP_2 + 1.0;
    G2 = POLYTROP_2 - 1.0;
    G3 = POLYTROP_2 + 1.0;
    G4 = POLYTROP_2 - 1.0;
    G5 = G2 / G1;
    G6 = G4 / G3;
    calcParamsOfGas(1, POLYTROP_2, _U2[1], ref _P2[1], ref _R2[1],
    α_COVOLUME_2, POLYTROP_2, -_U2[1], ref _P2[1], ref _R2[1], α_COVOLUME_2, ref UBB, ref
    PBB, ref RBB, ref UMM, ref PMM, ref RMM, ref EKB, ref ALB);
    PGLOBAL = PMM;
    goto g235;
g233:
    PGLOBAL = 0.0;
g235:
    if (KMOV == 0) KINDI = 1;
    KINDII = 0;

    timeStep(ref R, ref P, ref U, ref W, ref FI, ref Z, ref S, ref SV,
    α_COVOLUME, POLYTROP, τ, AKP, ALP, AMP, NTS, AFL, W0, S02, QP, ref XPL, US, ref USN, 0,
    APCH1, ref N, 1, MOVk, PGLOBAL, QFORCER);

    ACSYS = (USN - US) / TH * CM / BPM;
    US = USN;

    if (XIGN >= XPL) goto g231;
    if (IIGN == 1) goto g236;
    IIGN = 1;
g236:
    if (KMOV == 1) goto g232;
    if (((P[N] - _P2[1]) / QPPP) > (_P2[N2] / QS)) goto g232;

    OUTPUT(RGC, RGC2, α_COVOLUME, α_COVOLUME_2, TIM, τ, H2, POLYTROP, QFORCER,
    POLYTROP_2, RP, S, _S2, US, US2, XPL, XS0, N, N2);
    if ((outP[layersCount] < 0) || (outP2[layersCount] < 0) ||
    (outU[layersCount] < 0) || (outU2[layersCount] < 0))
    {
        return 2;
    }
    KMOV = 1;
    QFORCER = QPPP;

g232:
    if (KMOV != 1) goto g243;
    QFORCER = QPPP;
    for (I = 1; I <= N2; I++)
    {
        if (_FI2[I] >= FIPR) goto g245;
        QFORCER = QFORCER + (1.0 - _FI2[I]) * RP * _S2[I] * SM * H2 * LENSACLE;
    }
g245:

g243:

    if (KMOV == 1) KINDII = 1;
    KINDI = 0;
    timeStep(ref _R2, ref _P2, ref _U2, ref _W2, ref _FI2, ref _Z2, ref _S2,
    ref _S2, α_COVOLUME_2, POLYTROP_2, H2, AKP2, ALP2, AMP2, NTS2, AFL2, W02, S022, QP2,
    ref XS0, US2, ref US2N, ACSYS, APCH12, ref N2, KMOV, 1, 0, QS);

```

```

        US2 = US2N;
g231:
        KPER = KPER + 1;
        if (0 > BPCH) goto g223;
        if (XPL + DXP + XSO > CLENSCALE) goto g222;
        if (KPER < 1) goto g211;
        OUTPUT(RGC, RGC2, α_COVOLUME, α_COVOLUME_2, TIM, τ, H2, POLYTROP, QFORCER,
POLYTROP_2, RP, S, _S2, US, US2, XPL, XSO, N, N2);
        if ((outP[layersCount] < 0) || (outP2[layersCount] < 0) ||
(outU[layersCount] < 0) || (outU2[layersCount] < 0))
        {
            return 2;
        }
g402:
        KPER = 1;
g211:
        goto g1000;
g222:
        OUTPUT(RGC, RGC2, α_COVOLUME, α_COVOLUME_2, TIM, τ, H2, POLYTROP, QFORCER,
POLYTROP_2, RP, S, _S2, US, US2, XPL, XSO, N, N2);
        if ((outP[layersCount] < 0) || (outP2[layersCount] < 0) || (outU[layersCount] <
0) || (outU2[layersCount] < 0))
        {
            return 2;
        }
        goto g224;
g223:
g224:
        N = 1;
        OutString = OutString + "P2MAX = " + outP2.Max() + " P2NMAX = " + outP2N.Max()
+ " P2NMAX = " + outP2N.Max();
        return 1;
    }
}

```