МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное учреждение

высшего профессионального образования

«Пермский государственный национальный исследовательский университет»

Механико-математический факультет

Кафедра механики сплошных сред и вычислительных технологий

ВИЛДАНОВ Владислав Рафисович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

**УДК 519.95**

**Численное моделирование установившегося течения**

**газа в соплах**

Выпускная работа бакалавра.

**Научный руководитель:**

ст. преп. кафедры МССиВТ

**Сергеев Олег Борисович**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

 (Подпись)

Пермь 2012**Аннотация**

В данной работе численно решается задача стационарного течения идеального совершенного газа в пространственных соплах. Для получения стационарного решения используется метод установления по времени. Решение нестационарной задачи осуществляется методом С.К. Годунова. Составлен программный комплекс в среде программирования Delphi XE2.

**Содержание**

Введение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4
Обозначения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6
1.   Постановка задачи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7
2.   Система уравнений нестационарногопотока совершенного газа\_\_\_\_\_\_\_8
     2.1.   Система уравнений\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8
     2.2.   Обезразмеривание уравнений\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10
3.   Дискретизация расчётной области\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_12
     3.1.   Построение разностной сетки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_12
     3.2.   Определение геометрических параметров ячейки\_\_\_\_\_\_\_\_\_\_\_\_\_\_14
4.   Разностная схема С.К.Годунова\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17
     4.1.   Вывод разностных уравнений для ячейки расчётной области\_\_\_\_17
     4.2.   Определение параметров течения на гранях ячейки\_\_\_\_\_\_\_\_\_\_\_\_21
     4.3.   Выбор шага по времени\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_25
5.   Начальное приближение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_26
6.   Результаты расчётов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_28
Заключение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_30
Список литературы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_31
Приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_32
      Приложение 1. Описание программного комплекса\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_33
      Приложение 2. Код программного комплекса\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_35
**Введение**

Решение задач о расчёте смешанных (до- и сверхзвуковых) течений газа состоит в интегрировании трёхмерной нелинейной системы дифференциальных уравнений в частных производных эллиптико-гиперболического типа. К настоящему времени опубликовано большое количество работ, посвящённых численному анализу обтекания тел с отошедшей ударной волной и численному исследованию течений в соплах. Наряду с методом интегральных соотношений [1], и методом прямых [2], которые базируются на рассмотрении стационарных систем уравнений, для анализа течения около тупых тел и в соплах использовался метод установления [3,4]. При этом решение стационарной задачи как нестационарной позволяет облегчить получение решения, так как это связано с переходом от системы дифференциальных уравнений смешанного типа к системе гиперболического типа. Другими словами, для решения системы смешанного типа надо выделить участки, где уравнения эллиптического типа, а где гиперболического. Интегрирование таких систем принципиально различно. При методе установления оба этих участка интегрируются одним способом.

Среди разработанных к данному моменту времени численных методов расчёта нестационарных течений газа схема С.К. Годунова является одной из наиболее распространённых. Для расчёта плоских и осесимметричных течений её успешно использовали авторы [4], а для решения пространственных задач – авторы [5], где схема С.К. Годунова применяется для расчёта пространственного течения газа в поворотном управляющем сопле и примыкающей к нему камере. Широкое и весьма успешное применение этой разностной схемы обусловлено хорошей адаптируемостью разрабатываемых на её основе алгоритмов к особенностям рассчитываемых течений. При разработке своего метода [6] С.К. Годунов сформулировал важное требование, которому должна удовлетворять схема, предназначенная для численного построения разрывных решений уравнений Эйлера. Это требование монотонности схемы, обеспечивающие отсутствие осцилляций численного решения в окрестности разрыва.

С.К. Годунов установил, что среди схем второго порядка аппроксимации невозможно найти однородную двухслойную линейную разностную схему, отвечающей условию монотонности. Оказалось, что среди монотонных схем первого порядка аппроксимации потоков через границы ячеек задачу о распаде произвольного разрыва, даёт наименьшую аппроксимирующую погрешность, то есть является лучшей.

Требование монотонности схемы является весьма существенным и необходимым для решения задач газовой динамики. Прочие ограничения, такие как линейность, однородность и т.д., были удобны автору для удобства анализа, так что вне рамок указанных ограничений могут существовать разностные схемы более высокого порядка аппроксимации, превосходящие схему С.К. Годунова по точности или при заданной точности по быстродействию.

Известно, что консервативные разностные схемы на грубых сетках дают существенно бóльшую точность, чем схемы того же порядка аппроксимации, но не обладающие свойством консервативности.

С.К. Годуновым установлено, что в цилиндрической системе координат его разностная схема не обладает свойством консервативности, в отличие от схемы в декартовой системе отчёта.

Именно этим обусловлен выбор метода С.К. Годунова для решения задачи о течении газа в Сопле в декартовой системе координат.

**Обозначения**

Для данной задачи введём следующие обозначения:

*  – прямоугольная декартовая система координат;
* – цилиндрическая система координат;
* – вектор скорости частиц газа;
* –компоненты вектора скорости в декартовой системе координат;
*  – плотность газа;
* – давление в газе;
* –абсолютная температура газа;
* – внутренняя энергия единицы массы;
* – полная энергия единицы объёма;
* – показатель адиабаты;
* – скорость звука.

**1.\_Постановка задачи**

Рассматривается стационарное смешанное (до- и сверхзвуковое) течение идеального совершенного газа в пространственном сопле (рис.1.1). Начало координат помещаем во входное сечение сопла. Дозвуковой части отвечает сужающаяся часть, сверхзвуковой – расширяющаяся. 

Рис.1.1. Геометрия канала.

Стационарная система уравнений потока газа имеет вид:

 (1.1)

где

 

Граничные условия:

 (1.2)

 (1.3)

Требуется найти параметры газа в каждом сечении сопла.

**2.Система уравнений нестационарного
потока совершенного газа**

**2.1Система уравнений**

Так как для решения поставленной задачи используется метод установления, то выпишем уравнения для нестационарного течения газа в отсутствии массовых сил:

 (2.1)

Учитывая связь между полной и внутренней энергиями

.

уравнение полной энергии:

. (2.2)

Система уравнений (2.1) состоит из 6 уравнений и 6 неизвестных, то есть получилась замкнутая система уравнений.

Учитывая (2.2), выпишем (2.1) в координатной форме:

**** (2.3)

где ,  - параметры в камере сгорания, а  - нормальная к поверхности канала составляющая вектора скорости.

Граничные условия (1.2), (1.3) и начальные условия:

 (2.5)

где , ,  - начальное приближение, определяемое в главе 5.

**2.2. Обезразмеривание уравнений нестационарного потока газа**

Выразим размерные величины через безразмерные:



где величины с чертами – безразмерные, а со звёздочками – размерные характерные величины.

За базисные размерные величины возьмём и – плотность и давление в камере сгорания, а также – радиус критического сечения сопла.

Зависимые размерные величины выражаются следующим образом:

. (2.6)

Уравнение полной энергии примет вид

.

Учитывая (2.6), имеем:

.

Тогда из последнего соотношения получаем:

 (2.7)

Система уравнений потока газа (2.3) примет вид:



Учитывая (2.6) и (2.7), получим безразмерную систему неустановившегося потока совершенного идеального газа:

**** (2.8)

Не теряя общности, в дальнейшем безразмерные величины будем обозначать без черточек.

Введём вспомогательные векторы-столбцы:

****.(2.9)

Учитывая (2.8) и (2.9), имеем:

. (2.10)

Это уравнение – компактная запись безразмерной системы уравнений нестационарного потока совершенного газа в прямоугольной декартовой системе координат.

**3.Дискретизация расчётной области**

**3.1. Построение разностнойсетки**

Пусть начало координат находится во входном сечении сопла.Разбивка расчётной области в данной работе производится в цилиндрической системе координат, связанной с декартовой системой координат следующими соотношениями:

 (3.1)

По оси $z$ расчётная область разбивается на $K$ равных отрезков (рис.2.1a). Получится сетка.В каждом поперечном сечении делается разбивка по окружности и по радиусу (рис.2.1 б). Индекс при говорит о том, что в каждом поперечном сечении будет свой радиус.

**

Рис.2.1. Дискретизация области.

После произведённой разбивки получатся ячейки. Данные ячейки нумеруются следующим образом: вдоль оси *z*числами , вдоль радиуса числами , и по окружности числами , а соответствующие грани числами . Поверхностям, разделяющим ячейки, приписываются один целый и два дробных индекса, причём их численное значение и порядок определяются положением рассматриваемых участков границ. Например, на границах, перпендикулярных оси *z*, первые индексы всегда целые, а остальные два всегда дробные.

Координаты узла в декартовой системе координат:

 (3.2)

**3.2. Определение геометрических параметров ячейки**

Объём ячейки считается следующим образом:

. (3.3)

Площади боковых граней ячейки вычисляются следующим образом:

(3.4)

Нормаль к граням  направим в положительном направлении оси *z*, а касательные в положительные направления осей *x*и *y*. Таким образом для грани  имеем:

 (3.5)

Нормаль к грани  направлена в направлении от оси канала из геометрического центра грани:

 (3.6)

где

 (3.7)

Нормаль к грани  направим в сторону увеличения угла :

 (3.8)

Для осесимметричных задач все геометрические параметры ячейки можно переписать в упрощённом виде, если ввести параметр  – радиус в сечении .

Тогда:

 (3.9)

Подставим (3.9) в (3.3) и получим выражение для вычисления объёма ячейки:

 (3.10)

Формулы для вычисления площадей граней (3.4) с учётом (3.9) примут вид:

 (3.11)

Так как

, , , (3.12)

то соотношения (3.6) и (3.7) можно переписать в следующем виде

 (3.13)

 (3.14)

Для грани , учитывая (3.12), соотношения (3.8) можно переписать следующим образом

 (3.15)

Координаты каждого узла в декартовой системе координат, учитывая (3.10) и (3.13) выражаются следующим образом

 (3.16)

Проекции грани на координатные плоскости:

(3.17)

Проекции грани на координатные плоскости:

(3.18)

Проекции грани на координатные плоскости:

(3.18)

**4. Разностная схема С.К. Годунова**

**4.1. Вывод разностных уравненийдля ячейки**

**расчетной области**

Интегральные законы сохранения, отвечающие уравнениям (2.10), запишем в виде:

. (4.1)

где Ω – произвольный замкнутый объём, *S*– ограничивающая его замкнутая поверхность.

Разностная схема, при помощи которой на известном временном слое $t$ определяются параметры течения газа на следующем временном слое, получается интегрированием от до уравнений (4.1) по времени к элементарным многогранникам-ячейкам сетки (рис.3.1).

**

Рис.3.1. Ячейка сетки

Разбивка области осуществлена в цилиндрической системе координат. Ячейку между ** и**поверхностями разбиения по продольной оси*z*, ** и**поверхностями разбиения по радиусу *r*,** и**поверхностями разбиенияпо угловой координате будем обозначать индексами. Границы ячейки между **и**,** и**поверхностями имеют индексыи; между** и**,** и**поверхностями имеют индексы и; между** и**,** и**поверхностями имеют индексы и. Так как в цилиндрической системе координат возрастанию радиальной*r*и угловойкоординат не всегда соответствует возрастанию декартовых координат *x*и*y*, при дальнейшем изложении для параметров на гранях ячейки удобнее отказаться от индексных обозначений.

Пусть и площади передней и задней грани ячейки соответственно (рис.3). То есть, это грани исоответственно.и – площади проекций этих граней на плоскость, и  – площади проекций этих граней на плоскость*xz*, $S\_{1}^{z}$и $S\_{2}^{z}$ – площади проекций этих граней на плоскость *xy*.Пусть также и площади верхней и нижней грани ячейки, изображённой на рис.3 (то есть, это грани исоответственно), а и, и,и–площади проекций этих граней соответственно на координатные плоскости*yz*, *xz*, *xy*. Пусть и  – площади правой и левой граней той же ячейки (то есть, это гранейи), а и, и,и –площади проекций этих граней соответственно на координатные плоскости*yz*, *xz*, *xy*.

Проиллюстрируем вывод разностных уравнений на примере уравнения неразрывности. Первое уравнение системы (4.1) имеет вид

 (4.2)

и его первый член, записанный в конечных разностях, имеет следующий вид



Нижние индексы относятся к параметрам на временном слое*t*, а верхние индексы – на временном слое , –объём рассматриваемой ячейки сетки, –шаг по времени.

Второе слагаемое уравнения (4.2) аппроксимируется следующим образом

,

где *R*и*U* – осреднённые по соответствующим граням ячейки значения и*u*.Аналогично запишем оставшиеся члены уравнения (4.2)

,

где*V*и*W*–осреднённые по соответствующим граням значения компонент вектора скорости *v*и*w*. Подставив записанные выражения аппроксимации в уравнение (4.2), получим

, (4.3)

где

 (4.4)

По аналогии получаем разностные соотношения для остальных уравнений системы (2.6)

, (4.5)

**, (4.6)

**, (4.7)

 (4.8)

где

 (4.9)

 (4.10)

 (4.11)

 (4.12)

где через *P*обозначено среднее по соответствующей грани значение давления*p*.

**4.2.Определение параметров течения на гранях ячейки**

Для определения параметров *R* ,*P* ,*U* ,*V* ,*W*на границах ячейки, которые входят в полученные выше разностные уравнения, рассматривается распад произвольного одномерного разрыва.

Рассмотрим случай, когда граница ячейки является внутренней на примере грани . В начальный момент времени давление, плотность и компоненты вектора скорости в ячейках по обе стороны грани равны , , , , и , , , ,  соответственно. Аналогично для других граней.

Ориентация грани в пространстве известна, а это значит, что можно определить нормальную и тангенсальную составляющую вектора скорости по обе стороны грани. Обозначим их соответственно через , ,  и , , .

Для двух газов с параметрами , ,  и , ,  соответственно решим задачу произвольного разрыва.

В результате распада возникают три волны, одна из которых представляет собой контактную поверхность, а две другие могут быть либо ударной волной, либо волной разрежения. Давление и нормальная компонента скорости на контактном разрыве вычисляются по формулам:

 (4.13)

В случае, когда в рассчитываемой области течения отсутствуют сильные разрывы параметров, для определения ** и **можно пользоваться приближёнными формулами

. (4.14)

В областях со значительными градиентами параметров (скачки уплотнения, сильные волны разрежения) для определения , ** и **следует использовать итерационные формулы

 (4.15)

 (4.16)

После того, как итерации сойдутся,  вычисляется при помощи формулы (4.13).

Плотность по обе стороны от контактного разрыва определяется по приближённым формулам

 (4.17)

 (4.18)

Три образующиеся волны представляются разрывами, которые распространяются со скоростями

 (4.19)

делят пространство на 4 области. Значения величин на грани выбираются в зависимости от того, в какую из областей с течением времени попадает граница. Параметры на грани обозначим как , , , , 

Таким образом:

если , 

 (4.20)

если , 

 (4.21)

если , 

 (4.22)

если , 

 (4.23)

если граница попадает в левый веер волн разрежения (то есть на границе выполняется , где )

 (4.24)

если граница попадает в правый веер волн разрежения (то есть на границе выполняется , где )

 (4.25)

Параметры *U*, *V*, *W*на граниопределяются из, , 

В случае, когда грань ячейки *k*, *l*, *m* (параметры , и) совпадает с твёрдой стенкой, добавляется фиктивная ячейка с параметрами , и . Далее решается задача распада произвольного разрыва, рассмотренная выше.

Если грань ячейки лежит в плоскости входного сечения сопла, то добавляются фиктивные ячейки с параметрами ,  и . Далее решается задача распада произвольного разрыва.

**4.3. Выбор шага по времени**

Исследование условия устойчивости рассмотренной разностной схемы проведено С.К. Годуновым и имеет вид

, (3.49)

где , , – интервалы времени, за которые волны, образующиеся при распаде разрыва, достигают противоположных ячеек по осям*x*, *y*, *z*. Исходя из этого условия, можно выписать выражения, определяющие , , 

 (3.50)

где , , – размеры расчётной ячейки, –скорость звука в данной ячейке

Для каждого временного слоя выбирается свой шаг по времени.

Условие устойчивости (3.49) является необходимым и достаточным.

**5. Определение начального приближения**

Для нахождения начальных функций (2.5) будем решать квазиодномерную задачу стационарного потока газа в трубе переменного сечения:

.

Для данной задачи при помощи интеграла Бернулли



получаются газодинамические функции плотности

 (5.1)

и давления

, (5.2)

где



–число Маха.

Так как мы решаем задачу в безразмерном виде, то газодинамические функции (5.1) и (5.2) – это безразмерные плотность и давление соответственно.

Нахождение значений безразмерных скорости, давления и плотности сводится к уравнению

,

где F – текущая площадь сечения, а  – площадь критического сечения. Учитывая то, что задача решается в безразмерном виде, то получим следующее соотношение, для определения числа Маха в сечениях

, (5.3)

где *r*– радиус текущего сечения.

Решать уравнение будем для сечений . Для безразмерной задачи соотношение примет вид

. (5.4)

Решая данное уравнение относительнометодом Нелдера-Мида, находим, и .Тогда для ячеек  имеем

 (5.5)

**6. Результаты расчётов**

Для реализации описанного выше метода был разработан комплекс программ на языке Pascal в среде программирования Delphi XE2. Описание программного комплекса приведено в приложении 1, а тексты программ в приложении 2. Проводились многочисленные расчёты для сопел различной конфигурации с различными входными данными. На рис 6.1, 6.2, 6.3 приведены результаты расчётов для следующих входных данных

* длина сужающейся части м;
* длина расширяющейся части м;
* радиус входного сечения сопла м;
* радиус критического сечения сопла м;
* радиус выходного сечения сопла м;
* давление в камере сгорания Па;
* температура в камере сгорания К;
* показатель адиабаты ;
* газовая постоянная ;
* разбивка по *z* сужающейся части сопла ;
* разбивка по *z* расширяющейся части сопла ;
* разбивка по радиусу ;
* разбивка по угловой координате .

Стационарное решение считается найденным, если невязка



где *ε* – малая величина. В расчётах принималось *.* В результате расчёта стационарное решение было достигнуто на 245 шаге. На рис. 6.1 показана зависимость невязки от времени.



Рис 6.1 Зависимость невязки от времени.



Рис 6.2. Зависимость числа Маха от координаты *z*.

Рис 6.3. Зависимость давления от координаты *z*.

**Заключение**

В ходе данной работы был изучен метод С.К.Годунова для расчёта течения газа. Разработан программный комплекс в среде программирования DelphiXE2. Программный комплекс был отлажен на симметричном сопле.

Результаты данной работы будут использованы для решении задачи о стационарном потоке газа в существенно несиметричных каналах.

**Список литературы**

1. Минайлос А.Н. О расчёте течения у затупленного тела вращения под углом атаки в сверхзвуковом потоке газа. - М.: ЖВММФ. – 1964. - т.4
№ 1. - с. 171-177.
2. Теленин Г.Ф., Тиняков Г.П. Метод расчёта пространственного обтекания тела с отошедшей волной. – М.: ДАНСССР. -т.154 №5
с. 1056-1058
3. Иванов М.Я. К решению двумерных и пространственных задач обтекания тел околозвуковым потоком. -М.: ЖВММФ. – 1975. –т.15 №5. –с. 1222-1240
4. КрайкоА.Н., ТилляеваН.И. ЩербаковС.А.Метод расчёта течений идеального газа в плоских и осесимметричных соплах с изломами контура. -М.: ЖВММФ. – 1986. – т.26 №11. –с. 1679–1694
5. ИвановМ.Я., РылькоО.А., Расчёт трансзвукового течения в пространственных соплах. –М.: ЖВММФ. – 1972. – т.12 №5.**–**
с.1280–1291
6. ГодуновС.К. Разностный метод численного расчета разрывных решений уравнений гидродинамики. -Матем. сб. – 1959. – т.47(89) №3. –
с. 271–306
7. Атанов Г.А. Газовая динамика. - Киев: Выща школа, 1991.
8. Чёрный Г.Г. Газовая динамика. – М.: Наука,1988.

**Приложение**

**Приложение 1**

Описание программного комплекса

* **Functional**(**Mach\_P**,**F**,**prmtr**) – вычисляетзначениеквадрата разности правой и левой частей уравнения (5.4) для текущего поперечного сечения, а также добавляет штрафную функцию к этому значению, если нарушены ограничения на число Маха;
* **Reflection**(**Mach\_Max**,**Mach\_C**) – отражаетвершинусимплекса с максимальным значением функционала относительно центра масс симплекса;
* **Expansion**(**Mach\_C**,**Mach\_r**) – отражает с расширениемвершинусимплекса, отражённую функцией **Reflection**, относительно центра масс симплекса;
* **Reduction**(**Mach\_C**,**Mach\_r**) – отражаетсосжатиемвершину симплекса, отражённую функцией **Reflection**, относительно центра масс симплекса;
* **Gomotetia**(**Simplex**,**NMin**,**Mach\_C**) – процедура, котораясжимает симплекс относительно своего центра масс;
* **Nelder\_Mead**(**Simplex**,**F**,**prmtr**) – процедура, котораярешает уравнение (5.4) методом Нелдера-Мида для конкретного поперечного сечения сопла;
* **Radius** – процедура, которая вычисляет радиус поперечного сечения для всех сечений ;
* **Cell\_Geometry**– вычисляетобъём и координаты вершин ячеек расчётной области;
* **NumberPoint** – процедура, в которой происходит нумерация узлов сетки;
* **Pressure\_Density**(**Mach\_P**,**Press**,**rho**) – вычислениедавленияиплотности для текущего значения числа Маха при помощи газодинамических функций;
* **Find\_Tau** – нахождение временного шага, при котором схема будет устойчива;
* **First\_Guess** – вычисление начального приближения (решение квазиодномерной задачи);
* **Riemann\_Problem**(**p\_1**,**p\_2**,**rh\_1**,**rh\_2**,**un\_1**,**un\_2**,**ut1\_1**,**ut1\_2**,**ut2\_1**, **ut2\_2**,**p\_g**,**rh\_g**,**Un\_g**,**Ut1\_g**,**Ut2\_g**) – процедура, решающая задачу о распаде произвольного разрыва;
* **Norma**–проверка на точность решения;
* **Print\_Results** – процедура, делающая вывод распределения давления и числа Маха вдоль оси и стенок канала в текстовые файлы;
* **Godunov** – процедура, в которой реализован метод С.К.Годунова
* **TForm1.Start\_ButtonClick**(**Sender**) – процедура, отвечающаяза интерфейс пользователя.

**Приложение 2**

Кодпрограммногокомплекса

unitVlad;

interface

uses

 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

 System.Classes, Vcl.Graphics,Vcl.Controls, Vcl.Forms, Vcl.Dialogs,

 Vcl.StdCtrls, Vcl.ComCtrls;

const

 alpha = 1; //коэффициент отражения

 beta = 2; //коэффициент растяжения

sigma =0.5; //коэффициент сжатия

eps = 1e-6; //погрешность

NN = 100; //максимальная разбивка по какой-либо координате

type

 TForm1 = class(TForm)

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Label6: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 Label9: TLabel;

 Label11: TLabel;

 Label10: TLabel;

 Label12: TLabel;

 Label13: TLabel;

 Label14: TLabel;

 Label15: TLabel;

 Edit\_L1: TEdit;

 Edit\_L2: TEdit;

 Edit\_R1: TEdit;

 Edit\_Rc: TEdit;

 Edit\_R2: TEdit;

 Edit\_P0: TEdit;

 Edit\_rho0: TEdit;

 Edit\_gam: TEdit;

 Edit\_KK1: TEdit;

 Edit\_KK2: TEdit;

 Edit\_LL: TEdit;

 Edit\_MM: TEdit;

 Start\_Button: TButton;

 Label16: TLabel;

 Edit\_R\_gaz: TEdit;

 Label17: TLabel;

 procedure Start\_ButtonClick(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

 TSimplex = array [1..2] of extended; //тип симплекса

 TCell = array [0..5\*NN,0..NN,0..NN] of extended; //тип параметров в ячейках

 TPoint= array [0..5\*NN+1,0..NN+1,0..NN] of extended; //тип узла

 TRad = array [0..5\*NN+1] of extended; //тип радиуса

 TCoef = array [1..6] of Extended; //тип коэффициентов A,B,C,D,E,F

var

 Form1: TForm1;

 p0,rho0,L1,L2,R1,Rc,R2, //размерные величины

 L1\_,L2\_,R1\_,R2\_, //обезразмеренные геометрические параметры

gam, //показатель адиабаты

U\_dim,T\_dim,P\_dim, //размерные коэффициенты

temp\_1\_functional,temp\_2\_functional:extended; //для нахождения чисел Маха

 R\_gaz,Temperature:extended; //универсальная газовая постоянная

KK1,KK2,LL,MM:integer; //параметры разбивки

hz1,hz2,htheta:extended; //шаги разбивки

P,Rh,U,V,W,Volume,Mach, //параметры в ячейках(текущее прибл)

P\_prev,Rh\_prev,U\_prev,V\_prev,W\_prev:TCell;//параметры в ячейках(пред прибл)

X,Y,Z:Array[1..10000] of extended; //Координаты ухлов

 XX,YY,ZZ:TPoint;

 Rad:TRad;

 kkk:LongInt;

 N\_P,N\_El:Integer;

 NumP:array[1..10000,1..3] of integer;

 Iter:Integer;

implementation

{$R \*.dfm}

function Functional(Mach\_P,F:extended;prmtr:Integer):extended;

 var temp:extended;

 begin

 temp:=Abs(temp\_2\_functional\*Mach\_P/Exp(Ln(1+0.5\*(gam-1)\*

 sqr(Mach\_P))\*temp\_1\_functional)-1/F);

 if prmtr=1 then

 if (Mach\_P>1)or(Mach\_P<0) then temp:=temp+1000

 else

 if (Mach\_P<1)or(Mach\_P<0) then temp:=temp+1000;

 Functional:=temp;

 end;

function Reflection(Mach\_Max,Mach\_C:extended):extended;

 begin

 Reflection:= (1+alpha)\*Mach\_C-alpha\*Mach\_Max;

 end;

function Expansion(Mach\_C,Mach\_r:extended):extended;

 begin

 Expansion:= Mach\_C + beta\*(Mach\_r - Mach\_C);

 end;

function Reduction(Mach\_C,Mach\_r:extended):extended;

 begin

 Reduction:= Mach\_C + sigma\*(Mach\_r - Mach\_C);

 end;

procedure Gomotetia(var Simplex:TSimplex;NMin:integer;Mach\_C:extended);

var i:integer;

 begin

 for i:= 1 to 2 do

 if i <> NMin then

 Simplex[i]:= Mach\_C + sigma\*(Simplex[i] - Mach\_C);

 end;

function Nelder\_Mead(Simplex:TSimplex;F:extended;prmtr:integer):extended;

var FMin\_Mach,FMax\_Mach,Temp,FMach\_r,FMach\_s,FMach\_e,

 Min\_Mach,Max\_Mach,Mach\_s,Mach\_r,Mach\_c,Mach\_e:extended;

 Max\_Ind,Min\_Ind,Iter:integer;

 temp\_check:boolean;

 begin

 Iter:=0;

 repeat

 Iter:=Iter+1;

 FMax\_Mach:=Functional(Simplex[1],F,prmtr);

 Max\_Mach:=Simplex[1];

 Max\_Ind:=1;

 FMin\_Mach:=Functional(Simplex[2],F,prmtr);

 Min\_Mach:=Simplex[1];

 Min\_Ind:=1;

 Temp:=Functional(Simplex[2],F,prmtr);

 if Temp>FMax\_Mach then

 begin

 FMax\_Mach:=Temp;

 Max\_Mach:=Simplex[2];

 Max\_Ind:=2;

 end

 else

 begin

 FMin\_Mach:=Temp;

 Min\_Mach:=Simplex[2];

 Min\_Ind:=2;

 end;

 Mach\_C:=Min\_Mach;

 Mach\_r:=Reflection(Max\_Mach,Mach\_C);

 FMach\_r:=Functional(Mach\_r,F,prmtr);

 if FMach\_r<FMin\_Mach then

 begin

 Mach\_e:=Expansion(Mach\_C,Mach\_r);

 FMach\_e:=Functional(Mach\_e,F,prmtr);

 if FMach\_r>FMach\_e then Simplex[Max\_Ind]:=Mach\_e

 else Simplex[Max\_Ind]:=Mach\_r;

 end

 else

 begin

 Mach\_s:=Reduction(Mach\_C,Mach\_r);

 FMach\_s:=Functional(Mach\_s,F,prmtr);

 if FMach\_s<FMax\_Mach then Simplex[Max\_Ind]:=Mach\_s

 else Gomotetia(Simplex,Min\_Ind,Mach\_C);

 end;

 if prmtr=1 then temp\_check:=Simplex[Max\_Ind]<1

 else temp\_check:=Simplex[Max\_Ind]>1;

 until ((Abs(Simplex[1]-Simplex[2])<=eps)and temp\_check)or(Iter>=2000);

 Nelder\_Mead:=Simplex[Max\_Ind];

 end;

Procedure Radius;

var k:integer;

 begin

 for k := 0 to KK1+KK2 do

 if k<KK1 then Rad[k]:=R1\_-(R1\_-1)/L1\_\*k\*hz1

 else Rad[k]:=1+(R2\_-1)/L2\_\*(k-KK1)\*hz2;

 end;

procedure Cell\_Geometry;

var k,l,m,Num\_P:integer;

 temp:Extended;

 check1,check2:extended;

 t1,t2:text;

 begin

 assign(t1,'Протокол.объём ячеек.txt');

 rewrite(t1);

 assign(t2,'Протокол.координаты узлов.txt');

 rewrite(t2);

 check1:=0;

 for k := 0 to KK1+KK2-1 do

 for l := 0 to LL-1 do

 begin

 Temp:=Pi\*(2\*l+1)/(3\*sqr(LL)\*MM)\*(sqr(Rad[k])

 +sqr(Rad[k+1])+Rad[k+1]\*Rad[k]);

 if k<KK1 then Temp:=Temp\*hz1

 else Temp:=Temp\*hz2;

 for m := 0 to MM-1 do

 begin

 check1:=check1+temp;

 Volume[k,l,m]:=Temp;

 end;

 WriteLn(t1,k,'.5;',l,'.5 - ',Temp);

 end;

 Num\_P:=0;

 for k := 0 to KK1+KK2 do

 for l := 0 to LL do

 for m := 0 to MM-1 do

 begin

 Num\_P:=Num\_P+1;

 NumP[Num\_P,1]:=k;

 NumP[Num\_P,2]:=l;

 NumP[Num\_P,3]:=m;

 XX[k,l,m]:=Rad[k]\*l\*cos(2\*Pi\*m/MM)/LL;

 YY[k,l,m]:=Rad[k]\*l\*sin(2\*Pi\*m/MM)/LL;

 X[Num\_P]:=XX[k,l,m]\*Rc;

 Y[Num\_P]:=YY[k,l,m]\*Rc;

 if k<=KK1 then

 begin

 ZZ[k,l,m]:=k\*hz1;

 Z[Num\_P]:=k\*hz1\*Rc

 end

 else

 begin

 ZZ[k,l,m]:=L1\_+hz2\*(k-KK1);

 Z[Num\_P]:=(L1\_+hz2\*(k-KK1))\*Rc;

 end;

 Writeln(t2,'(',k,';',l,';',m,') - x = ',X[Num\_P],

 ' y = ',Y[Num\_P],' z = ',Z[Num\_P]);

 end;

 check2:=Pi\*((Sqr(R1\_)+R1\_+1)\*L1\_+(Sqr(R2\_)+R2\_+1)\*L2\_)/3;

 writeln(t1,check1,' ',check2,' ',Num\_P);

 closefile(t1);

 closefile(t2);

 end;

Procedure Pressure\_Density(Mach\_P:extended;var Press,rho:extended);

 begin

 rho:=Exp((-1/(gam-1))\*Ln(1+0.5\*(gam-1)\*sqr(Mach\_P)));

 Press:=Exp(gam\*Ln(rho));

 end;

function Find\_Tau:Extended;

var k,l,m:integer;

 hr,htheta,cos\_th,sin\_th,th:Extended;

 U\_r,U\_Th,a\_cell:Extended;

 temp\_tau,Tau\_r,Tau\_th,Tau\_z,Tau\_0:Extended;

 begin

 for k := 0 to KK1+KK2-1 do

 for L := 0 to LL-1 do

 begin

 hr:=(Rad[k]+Rad[k+1])/(2\*LL);

 for m := 0 to MM-1 do

 begin

 htheta:=2\*Pi/MM\*(Rad[k]+Rad[k+1]);

 th:=Pi\*(2\*m+1)/MM;

 cos\_th:=Cos(th);

 sin\_th:=Sin(th);

 U\_r:=U[k,l,m]\*Sin\_th+V[k,l,m]\*Cos\_th;

 U\_th:=U[k,l,m]\*Cos\_th-V[k,l,m]\*Sin\_th;

 a\_cell:=sqrt(gam\*P[k,l,m]/Rh[k,l,m]);

 Tau\_r:=hr/(Abs(U\_r)+a\_cell);

 Tau\_th:=htheta/(Abs(U\_r)+a\_cell);

 if k<KK1 then Tau\_z:=hz1/(a\_cell+Abs(W[k,l,m]))

 else Tau\_z:=hz2/(a\_cell+Abs(W[k,l,m]));

 temp\_tau:=tau\_r\*tau\_th\*Tau\_z/(tau\_r\*tau\_th+tau\_r\*tau\_z+tau\_z\*tau\_th);

 if (k=0)and(l=0)and(m=0)

 then

 Tau\_0:=Temp\_tau

 else

 if Temp\_Tau<Tau\_0 then Tau\_0:=temp\_tau;

 end;

 end;

 Find\_Tau:=0.001\*tau\_0;

 end;

procedure First\_Guess;

var k,l,m:integer;

 Simplex:TSimplex;

 r\_,F,p\_temp,rh\_Temp,W\_Temp,Mach\_Temp:extended;

 ttt:text;

begin

 assign(ttt,'Протокол.первое приближение.txt');

 rewrite(ttt);

 for k:=0 to KK1+KK2-1 do

 begin

 if k<KK1 then

 begin

 r\_:=R1\_-(R1\_-1)/L1\_\*(k+0.5)\*hz1;

 if k=0 then

 begin

 Simplex[1]:=0;

 Simplex[2]:=0.1;

 end

 else

 begin

 Simplex[1]:=Mach[k-1,1,1];

 Simplex[2]:=1.05\*Simplex[1];

 end;

 end

 else

 begin

 r\_:=1+(R2\_-1)/L2\_\*(k-KK1+0.5)\*hz2;

 if k=KK1 then

 begin

 Simplex[1]:=1.01;

 Simplex[2]:=1.11;

 end

 else

 begin

 Simplex[1]:=Mach[k-1,1,1];

 Simplex[2]:=1.05\*Simplex[1];

 end;

 end;

 Mach\_Temp:=Nelder\_Mead(Simplex,sqr(r\_),1);

 Pressure\_Density(Mach\_Temp,P\_Temp,Rh\_Temp);

 W\_temp:=Mach\_Temp\*sqrt(gam\*P\_temp/rh\_temp);

 for l := 0 to LL-1 do

 for m := 0 to MM-1 do

 begin

 Mach[k,l,m]:=Mach\_temp;

 P[k,l,m]:=P\_Temp;

 Rh[k,l,m]:=Rh\_Temp;

 U[k,l,m]:=0;

 V[k,l,m]:=0;

 W[k,l,m]:=W\_Temp;

 end;

 WriteLn(ttt,k,'.5 - ',W\_Temp,' - ',P[k,1,1],' - ',Rh[k,1,1],' - ',Mach[k,1,1]);

 end;

 writeln(ttt,'Tau\_0 = ',find\_tau);

 closeFile(ttt);

end;

procedure Riemann\_Problem(p\_1,p\_2,rh\_1,rh\_2,un\_1,un\_2,

 ut1\_1,ut1\_2,ut2\_1,ut2\_2:extended;

 var p\_g,rh\_g,Un\_g,Ut1\_g,Ut2\_g:extended);

var a,b,t,c1,c2,s,rr1,rr2,rr3,rr4,rr5,rr6,rr7,

 rem,d1,d11,d2,d22,r1,r2,r3,a1,pkr,ukr,ro3,ro4:extended;

 i:integer;

begin

 c1:=sqrt((gam\*p\_1)/rh\_1);

 c2:=sqrt((gam\*p\_2)/rh\_2);

 rem:=1;

 if (un\_1-un\_2)<(-2\*(c1+c2)/(gam-1))

 then

 begin

 pkr:=0;

 r1:=0;

 r2:=0;

 ukr:=0;

 d1:=un\_1-c1;

 d11:=un\_1+2\*c1/(gam-1);

 d2:=un\_2+c2;

 d22:=un\_2-2\*c2/(gam-1);

 end

 else

 begin

 if abs(p\_1-p\_2)<eps

 then

 begin

 if un\_1<=un\_2

 then

 begin

 rr1:=sqrt(rh\_1);

 rr2:=sqrt(rh\_2);

 rr3:=rr1+rr2;

 ukr:=(uN\_1\*rr1+un\_2\*rr2)/rr3;

 rr4:=2\*gam/(gam-1);

 pkr:=p\_1\*exp(rr4\*ln(1-(gam-1)\*(ukr-un\_1)/(2\*c1)));

 rr6:=1/gam;

 rr5:=exp(rr6\*ln(pkr/p\_1));

 r1:=rh\_1\*rr5;

 r2:=rh\_2\*rr5;

 d1:=un\_1-c1;

 d11:=ukr-sqrt(gam\*pkr/r1);

 d2:=un\_2+c2;

 d22:=ukr+sqrt(gam\*pkr/r2);

 end

 else

 begin

 rr1:=sqrt(rh\_1);

 rr2:=sqrt(rh\_2);

 rr3:=rr1+rr2;

 ukr:=(un\_1\*rr1+un\_2\*rr2)/rr3;

 rr5:=sqr(un\_1-ukr);

 rr4:=(gam+1)\*rh\_1\*rr5/4;

 pkr:=p\_1+rr4+sqrt(rr4\*rr4+gam\*p\_1\*rh\_1\*rr5);

 rr6:=((gam+1)\*pkr+(gam-1)\*p\_1)/((gam+1)\*p\_1+(gam-1)\*pkr);

 r1:=rh\_1\*rr6;

 r2:=rh\_2\*rr6;

 rr7:=((gam+1)\*pkr+(gam-1)\*p\_1)/2;

 d1:=un\_1-sqrt(rr7/rh\_1);

 d11:=d1;

 d2:=un\_2+sqrt(rr7/rh\_2);

 d22:=d2;

 end

 end

 else

 begin

 t:=1;i:=1;

 s:=(p\_1+p\_2)/2;

 while (t>eps)and(rem>0) do

 begin

 if s>p\_1

 then

 a:=sqrt(rh\_1\*(((gam+1)/2)\*s+((gam-1)/2)\*p\_1))

 else

 begin

 a:=((gam-1)/(2\*gam))\*rh\_1\*c1;

 if abs(s/p\_1-1)>0.0001 then

 a:=a\*((1-s/p\_1)/(1-exp(((gam-1)/(2\*gam))\*ln(s/p\_1))));

 end;

 if s>p\_2

 then

 b:=sqrt(rh\_2\*(((gam+1)/2)\*s+((gam-1)/2)\*p\_2))

 else

 begin

 b:=((gam-1)/(2\*gam))\*rh\_2\*c2;

 if abs(s/p\_2-1)>0.0001 then

 b:=b\*((1-s/p\_2)/(1-exp(((gam-1)/(2\*gam))\*ln(s/p\_2))));

 end;

 pkr:=(b\*p\_1+a\*p\_2+a\*b\*(un\_1-un\_2))/(a+b);

 t:=abs(1-pkr/s);

 s:=pkr;

 i:=i+1;

 if i>900 then rem:=-1;

 end;

 ukr:=(a\*un\_1+b\*un\_2+p\_1-p\_2)/(a+b);

 rr1:=(gam+1)\*pkr+(gam-1)\*p\_1;

 rr2:=(gam+1)\*p\_1+(gam-1)\*pkr;

 rr3:=(gam+1)\*pkr+(gam-1)\*p\_2;

 rr4:=(gam+1)\*p\_2+(gam-1)\*pkr;

 if s>p\_1

 then

 begin

 r1:=rh\_1\*rr1/rr2;

 d1:=un\_1-sqrt(rr1/2/rh\_1);

 d11:=d1;

 end

 else

 begin

 r1:=rh\_1\*exp((1/gam)\*ln(s/p\_1));

 d1:=un\_1-c1;

 d11:=ukr-sqrt(gam\*pkr/r1);

 end;

 if s>p\_2

 then

 begin

 r2:=rh\_2\*rr3/rr4;

 d2:=un\_2+sqrt(rr3/2/rh\_2);

 d22:=d2;

 end

 else

 begin

 r2:=rh\_2\*exp((1/gam)\*ln(s/p\_2));

 d2:=un\_2+c2;

 d22:=ukr+sqrt(gam\*pkr/r2);

 end;

 end;

 end;

ro3:=r1;

ro4:=r2;

if d1>0 then

 begin

 Un\_g:=un\_1;

 P\_g:=p\_1;

 Rh\_g:=rh\_1;

 Ut1\_g:=ut1\_1;

 Ut2\_g:=ut2\_1;

 end;

if (d1<=0)and(d11>0) then

 begin

 a1:=sqrt(gam\*p\_1/rh\_1);

 r2:=2/(gam-1);

 r3:=gam\*r2;

 r1:=2/(gam+1)+(gam-1)/(gam+1)\*un\_1/a1;

 Un\_g:=(gam-1)/(gam+1)\*un\_1+2/(gam+1)\*a1;

 P\_g:=p\_1\*exp(r3\*ln(r1));

 Rh\_g:=rh\_1\*exp(r2\*ln(r1));

 Ut1\_g:=Ut1\_1;

 Ut2\_g:=Ut2\_1;

 end;

if (d11<=0)and(ukr>=0) then

 begin

 Un\_g:=ukr;

 P\_g:=pkr;

 Rh\_g:=ro3;

 Ut1\_g:=Ut1\_1;

 Ut2\_g:=Ut2\_1;

 end;

if (ukr<0)and(d22>0) then

 begin

 Un\_g:=ukr;

 P\_g:=pkr;

 Rh\_g:=ro4;

 Ut1\_g:=Ut1\_2;

 Ut2\_g:=Ut2\_2;

 end;

if (d22<=0)and(d2>0) then

 begin

 a1:=sqrt(gam\*p\_2/rh\_2);

 r2:=2/(gam-1);

 r3:=gam\*r2;

 r1:=2/(gam+1)-(gam-1)/(gam+1)\*un\_2/a1;

 Un\_g:=(gam-1)/(gam+1)\*un\_2-2/(gam+1)\*a1;

 P\_g:=p\_2\*exp(r3\*ln(r1));

 Rh\_g:=rh\_2\*exp(r2\*ln(r1));

 Ut1\_g:=Ut1\_2;

 Ut2\_g:=Ut2\_2;

 end;

if d2<=0 then

begin

 Un\_g:=un\_2;

 P\_g:=p\_2;

 Rh\_g:=rh\_2;

 Ut1\_g:=Ut1\_1;

 Ut2\_g:=Ut2\_2;

end;

end;

functionNorma:boolean; //проверка окончания счёта

var k,l,m:integer;

 temp:extended;

 t:text;

 begin

 assign(t,'Ошибки.txt');

 append(t);

 temp:=1000;

 for k := 0 to KK1+KK2-1 do

 for l := 0 to ll-1 do

 for m := 0 to MM-1 do

 begin

 if temp>Abs(Rh[k,l,m]-Rh\_prev[k,l,m]) then

 temp:=Abs(Rh[k,l,m]-Rh\_prev[k,l,m]);

 end;

 writeln(t,',{',Iter,',',temp,'}');

 close(t);

 Norma:=temp<1e-6;

 end;

Function NumberPoint(k,l,m:integer):Integer;

var i:integer;

 begin

 for i := 1 to N\_P do

 if ((NumP[i,1]=k) and (NumP[i,2]=l) and (NumP[i,3]=m))

 then NumberPoint:=i;

 end;

{procedure Print\_Results;

var Num\_El,i,k,l,m,k1,l1,m1:integer;

 ftec,ff:text;

 Matrix\_El:array[1..10000,1..8] of integer;

 P\_el,Rh\_El,U\_El,V\_El,W\_El:array[1..10000] of extended;

 begin

 assign(ftec,'tecplot\_Vlad.dat');

 assign(ff,'vlad\_\_.txt');

 rewrite(ff);

 REWRITE(ftec);

 writeln(ftec,'TITLE="Canal of variable section: "');

 writeln(ftec,'Variables= X , Y, Z, P, Rho, U, V, W');

 writeln(ftec,'Zone N= ',N\_p:7,' E=',N\_el:7,' F=FEBLOCK, ET=BRICK');

 writeln(ftec,'DT=(SINGLE SINGLE SINGLE SINGLE SINGLE SINGLE SINGLE SINGLE)');

 writeln(ftec,'VARLOCATION=([4-8]=CELLCENTERED)');

 for i:=1 to N\_p do

 begin

 write(ftec,X[i]:14:6);

 if (frac(i/8)=0) or (i=N\_p) then writeln (ftec);

 end;

 for i:=1 to N\_p do

 begin

 write(ftec,Y[i]:14:6);

 if (frac(i/8)=0) or (i=N\_p)then writeln (ftec);

 end;

 for i:=1 to N\_p do

 begin

 write(ftec,Z[i]:14:6);

 if (frac(i/8)=0) or (i=N\_p)then writeln (ftec);

 end;

 Num\_El:=0;

 for k := 0 to KK1+KK2-1 do

 for l := 0 to LL-1 do

 for m := 0 to MM-1 do

 begin

 Num\_El:=Num\_El+1;

 if m=MM-1 then m1:=0

 else m1:=m+1;

 if l=0 then

 begin

 Matrix\_El[Num\_El,1]:=NumberPoint(k,0,0);

 Matrix\_El[Num\_El,2]:=Matrix\_El[Num\_El,1];

 Matrix\_El[Num\_El,3]:=NumberPoint(k+1,0,0);

 Matrix\_El[Num\_El,4]:=Matrix\_El[Num\_El,3];

 end

 else

 begin

 Matrix\_El[Num\_El,1]:=NumberPoint(k,l,m);

 Matrix\_El[Num\_El,2]:=NumberPoint(k,l,m1);

 Matrix\_El[Num\_El,3]:=NumberPoint(k+1,l,m1);

 Matrix\_El[Num\_El,4]:=NumberPoint(k+1,l,m);

 end;

 Matrix\_El[Num\_El,5]:=NumberPoint(k,l+1,m);

 Matrix\_El[Num\_El,6]:=NumberPoint(k,l+1,m1);

 Matrix\_El[Num\_El,7]:=NumberPoint(k+1,l+1,m1);

 Matrix\_El[Num\_El,8]:=NumberPoint(k+1,l+1,m);

 P\_El[Num\_El]:=P[k,l,m]\*p0;

 Rh\_El[Num\_El]:=Rh[k,l,m]\*rho0;

 U\_El[Num\_El]:=U[k,l,m]\*U\_dim;

 V\_El[Num\_El]:=V[k,l,m]\*U\_dim;

 W\_El[Num\_El]:=W[k,l,m]\*U\_dim;

 end;

 for i := 1 to Num\_El do

 begin

 write(ftec,P\_el[i]:14:6,' ');

 if (frac(i/8)=0) or (i=Num\_El) then writeln(ftec);

 end;

 for i := 1 to Num\_El do

 begin

 write(ftec,RH\_el[i]:14:6,' ');

 if (frac(i/8)=0) or (i=Num\_El) then writeln(ftec);

 end;

 for i := 1 to Num\_El do

 begin

 write(ftec,U\_el[i]:14:6,' ');

 if (frac(i/8)=0) or (i=Num\_El) then writeln(ftec);

 end;

 for i := 1 to Num\_El do

 begin

 write(ftec,V\_el[i]:14:6,' ');

 if (frac(i/8)=0) or (i=Num\_El) then writeln(ftec);

 end;

 for i := 1 to Num\_El do

 begin

 write(ftec,W\_el[i]:14:6);

 if (frac(i/8)=0) or (i=Num\_El) then writeln(ftec);

 end;

 for i := 1 to Num\_El do

 begin

 writeln(ftec,Matrix\_El[i,1]:6,Matrix\_El[i,2]:6,Matrix\_El[i,3]:6,

 Matrix\_El[i,4]:6,Matrix\_El[i,5]:6,Matrix\_El[i,6]:6,Matrix\_El[i,7]:6,

 Matrix\_El[i,8]:6);

 end;

 writeln(ff,Num\_El);

 close(ff);

 Close(ftec);

 end; }

procedure Print\_Results;

var k,l,m:integer;

 t1,t2,t3,t4:text;

 temp1,temp2:extended;

begin

assign(t1,'Давление вдоль оси канала.txt');

assign(t2,'Давление вдоль стенки канала.txt');

assign(t3,'Число Маха вдоль оси канала.txt');

assign(t4,'Число Маха вдоль стенки канала.txt');

rewrite(t1);

 rewrite(t2);

 rewrite(t3);

 rewrite(t4);

 for k := 0 to KK1+KK2-1 do

 begin

 writeln(t1,',{',(ZZ[k+1,0,0]+zz[k,0,0])/2\*Rc:14:6,',',P[k,0,0]\*P0:14:0,'}');

 writeln(t2,',{',(ZZ[k+1,0,0]+ZZ[k,0,0])/2\*Rc:14:6,',',P[k,LL-1,0]\*P0:14:0,'}');

 temp1:=sqrt(sqr(U[k,0,0])+sqr(V[k,0,0])+sqr(W[k,0,0]))/sqrt(gam\*P[k,0,0]/Rh[k,0,0]);

 temp2:=sqrt(sqr(U[k,LL-1,5])+sqr(V[k,LL-1,5])+sqr(W[k,LL-1,5]))/sqrt(gam\*P[k,LL-1,0]/Rh[k,LL-1,0]);

 writeln(t3,',{',(ZZ[k+1,0,0]+ZZ[k,0,0])/2\*RC:14:6,',',temp1:14:6,'}');

 writeln(t4,',{',(ZZ[k+1,0,0]+ZZ[k,0,0])/2\*Rc:14:6,',',temp2:14:6,'}');

 end;

 close(t1);

 close(t2);

 close(t3);

 close(t4);

end;

procedure Godunov;

var k,l,m:integer;

 nx1,ny1,nz1,

 nx3,ny3,nz3,

 nx4,ny4,nz4,

 nx5,ny5,nz5,

 nx6,ny6,nz6,

 t1x1,t1y1,t1z1,

 t1x3,t1y3,t1z3,

 t2x1,t2y1,t2z1,

 t2x3,t2y3,t2z3,

 t1x4,t1y4,t1z5,

 t1x5,t1y5,t1z4,

 t1x6,t1y6,t1z6,

 t2x4,t2y4,t2z5,

 t2x6,t2y5,t2z4,

 t2x5,t2y6,t2z6,

 a1,a2,a3,a4,a5,a6,

 b1,b2,b3,b4,b5,b6,

 c1,c2,c3,c4,c5,c6,

 d1,d2,d3,d4,d5,d6,

 f1,f2,f3,f4,f5,f6,

 h1,h2,h3,h4,h5,h6,

 p1,p2,p3,p4,p5,p6,

 rh1,rh2,rh3,rh4,rh5,rh6,

 u1,u2,u3,u4,u5,u6,

 v1,v2,v3,v4,v5,v6,

 w1,w2,w3,w4,w5,w6,

 s1x,s2x,s3x,s4x,s5x,s6x,

 s1y,s2y,s3y,s4y,s5y,s6y,

 s1z,s2z,s3z,s4z,s5z,s6z,

 s1,s2,s3,s4,s5,s6,

 un\_1,ut1\_1,ut2\_1,

 un\_2,ut1\_2,ut2\_2,

 un\_g,ut1\_g,ut2\_g,

 cos\_xi,sin\_xi,delta,tau,

 tau\_V,Mach\_temp,Energy,h\_z,

 p\_extra,rh\_extra:extended;

 t:text;

 name:string;

 begin

 Iter:=0;

 repeat

 Iter:=Iter+1;

 //name:='Протокол '+IntToStr(Iter)+'.txt';

 //assign(t,name);

 //rewrite(t);

 tau:=Find\_Tau;

 Rh\_prev:=Rh;

 P\_prev:=P;

 U\_prev:=U;

 V\_Prev:=V;

 W\_prev:=W;

 for k := 0 to KK1+KK2-1 do

 for l:= 0 to LL-1 do

 for m:= 0 to MM-1 do

 begin

 if k<KK1 then h\_z:=hz1

 else h\_z:=hz2;

 tau\_V:=tau/Volume[k,l,m];

 S1:=Pi\*(2\*l+1)\*sqr(Rad[k])/(sqr(LL)\*MM);

 S2:=Pi\*(2\*l+1)\*sqr(Rad[k+1])/(sqr(LL)\*MM);

 S3:=Pi\*l\*(Rad[k+1]+Rad[k])\*h\_z;

 S4:=Pi\*(l+1)\*(Rad[k+1]+Rad[k])\*h\_z;

 S5:=0.5\*Abs(Rad[k+1]-Rad[k])\*h\_z/LL;

 S6:=0.5\*Abs(Rad[k+1]-Rad[k])\*h\_z/LL;

 nx1:=0;

 ny1:=0;

 nz1:=1;

 t1x1:=0;

 t1y1:=1;

 t1z1:=0;

 t2x1:=1;

 t2y1:=0;

 t2z1:=0;

 s1x:=Abs(s1\*nx1);

 s1y:=Abs(s1\*ny1);

 s1z:=Abs(s1\*nz1);

 s2x:=Abs(s2\*nx1);

 s2y:=Abs(s2\*ny1);

 s2z:=Abs(s2\*nz1);

 s3x:=Abs(YY[k,l,m]-YY[k,l,m+1]+YY[k+1,l,m]-YY[k+1,l,m+1])\*h\_z;

 s3y:=Abs(XX[k,l,m]-XX[k,l,m+1]+XX[k+1,l,m]-XX[k+1,l,m+1])\*h\_z;

 s3z:=Pi\*sqr(l)/(sqr(LL)\*MM)\*Abs(sqr(Rad[k+1])-sqr(Rad[k]));

 s4x:=Abs(YY[k,l+1,m]-YY[k,l+1,m+1]+YY[k+1,l+1,m]-YY[k+1,l+1,m+1])\*h\_z;

 s4y:=Abs(XX[k,l+1,m]-XX[k,l+1,m+1]+XX[k+1,l+1,m]-XX[k+1,l+1,m+1])\*h\_z;

 s4z:=Pi\*sqr(l+1)/(sqr(LL)\*MM)\*Abs(sqr(Rad[k+1])-sqr(Rad[k]));

 un\_1:=nx1\*U\_prev[k,l,m]+ny1\*V\_prev[k,l,m]+nz1\*W\_prev[k,l,m];

 ut1\_1:=t1x1\*U\_prev[k,l,m]+t1y1\*V\_prev[k,l,m]+t1z1\*W\_prev[k,l,m];

 ut2\_1:=t2x1\*U\_prev[k,l,m]+t2y1\*V\_prev[k,l,m]+t2z1\*W\_prev[k,l,m];

 if k=0 then Riemann\_Problem(1,P\_prev[k,l,m],1,Rh\_prev[k,l,m],0,Un\_1,0,

 Ut1\_1,0,Ut2\_1,p1,Rh1,Un\_g,Ut1\_g,ut2\_g)

 else

 begin

 un\_2:=nx1\*U\_prev[k-1,l,m]+ny1\*V\_prev[k-1,l,m]+nz1\*W\_prev[k-1,l,m];

 ut1\_2:=t1x1\*U\_prev[k-1,l,m]+t1y1\*V\_prev[k-1,l,m]+t1z1\*W\_prev[k-1,l,m];

 ut2\_2:=t2x1\*U\_prev[k-1,l,m]+t2y1\*V\_prev[k-1,l,m]+t2z1\*W\_prev[k-1,l,m];

 Riemann\_Problem(P\_prev[k-1,l,m],P\_prev[k,l,m],Rh\_prev[k-1,l,m],

 Rh\_prev[k,l,m],un\_2,un\_1,ut1\_2,ut1\_1,ut2\_2,

 ut2\_1,p1,rh1,Un\_g,Ut1\_g,Ut2\_g);

 end;

 U1:=Un\_g\*nx1+Ut1\_g\*t1x1+Ut2\_g\*t2x1;

 V1:=Un\_g\*ny1+Ut1\_g\*t1y1+Ut2\_g\*t2y1;

 W1:=Un\_g\*nz1+Ut1\_g\*t1z1+Ut2\_g\*t2z1;

 if k=KK1+KK2-1 then

 begin

 un\_2:=nx1\*(2\*U\_prev[k,l,m]-U\_prev[k-1,l,m])

 +ny1\*(2\*V\_prev[k,l,m]-V\_prev[k-1,l,m])

 +nz1\*(2\*W\_prev[k,l,m]-W\_prev[k-1,l,m]);

 ut1\_2:=t1x1\*(2\*U\_prev[k,l,m]-U\_prev[k-1,l,m])

 +t1y1\*(2\*V\_prev[k,l,m]-V\_prev[k-1,l,m])

 +t1z1\*(2\*W\_prev[k,l,m]-W\_prev[k-1,l,m]);

 ut2\_2:=t2x1\*(2\*U\_prev[k,l,m]-U\_prev[k-1,l,m])

 +t2y1\*(2\*V\_prev[k,l,m]-V\_prev[k-1,l,m])

 +t2z1\*(2\*W\_prev[k,l,m]-W\_prev[k-1,l,m]);

 p\_extra:=2\*P\_prev[k,l,m]-P\_prev[k-1,l,m];

 rh\_extra:=2\*rh\_prev[k,l,m]-rh\_prev[k-1,l,m];

 Riemann\_Problem(P\_prev[k,l,m],P\_extra,

 Rh\_prev[k,l,m],Rh\_extra,

 Un\_1,Un\_2,Ut1\_1,Ut1\_2,ut2\_1,Ut2\_2,

 p2,rh2,Un\_g,Ut1\_g,Ut2\_g);

 end

 else

 begin

 un\_2:=nx1\*U\_prev[k+1,l,m]+ny1\*V\_prev[k+1,l,m]+nz1\*W\_prev[k+1,l,m];

 ut1\_2:=t1x1\*U\_prev[k+1,l,m]+t1y1\*V\_prev[k+1,l,m]+t1z1\*W\_prev[k+1,l,m];

 ut2\_2:=t2x1\*U\_prev[k+1,l,m]+t2y1\*V\_prev[k+1,l,m]+t2z1\*W\_prev[k+1,l,m];

 Riemann\_Problem(P\_prev[k,l,m],P\_prev[k+1,l,m],Rh\_prev[k,l,m],

 Rh\_prev[k+1,l,m],un\_1,un\_2,ut1\_1,ut1\_2,

 ut2\_1,ut2\_2,p2,rh2,Un\_g,Ut1\_g,Ut2\_g);

 end;

 U2:=Un\_g\*nx1+Ut1\_g\*t1x1+Ut2\_g\*t2x1;

 V2:=Un\_g\*ny1+Ut1\_g\*t1y1+Ut2\_g\*t2y1;

 W2:=Un\_g\*nz1+Ut1\_g\*t1z1+Ut2\_g\*t2z1;

 if l=0 then

 begin

 U3:=1;

 V3:=1;

 W3:=1;

 P3:=1;

 Rh3:=1;

 end

 else

 begin

 delta:=sqrt(sqr(h\_z)+sqr(l/LL\*(Rad[k+1]-Rad[k])));

 cos\_xi:=h\_z/delta;

 sin\_xi:=l\*(Rad[k+1]-Rad[k])/LL/delta;

 nx3:=cos\_xi\*cos(Pi\*(2\*m+1)/MM);

 ny3:=cos\_xi\*sin(Pi\*(2\*m+1)/MM);

 nz3:=Sin\_xi;

 t1x3:=-Sin(Pi\*(2\*m+1)/MM);

 t1y3:=Cos(Pi\*(2\*m+1)/MM);

 t1z3:=0;

 t2x3:=sin\_xi\*Cos(Pi\*(2\*m+1)/MM);

 t2y3:=sin\_xi\*Sin(Pi\*(2\*m+1)/MM);

 t2z3:=-cos\_xi;

 un\_1:=nx3\*U\_prev[k,l,m]+ny3\*V\_prev[k,l,m]+nz3\*W\_prev[k,l,m];

 ut1\_1:=t1x3\*U\_prev[k,l,m]+t1y3\*V\_prev[k,l,m]+t1z3\*W\_prev[k,l,m];

 ut2\_1:=t2x3\*U\_prev[k,l,m]+t2y3\*V\_prev[k,l,m]+t2z3\*W\_prev[k,l,m];

 un\_2:=nx3\*U\_prev[k,l-1,m]+ny3\*V\_prev[k,l-1,m]+nz3\*W\_prev[k,l-1,m];

 ut1\_2:=t1x3\*U\_prev[k,l-1,m]+t1y3\*V\_prev[k,l-1,m]+t1z3\*W\_prev[k,l-1,m];

 ut2\_2:=t2x3\*U\_prev[k,l-1,m]+t2y3\*V\_prev[k,l-1,m]+t2z3\*W\_prev[k,l-1,m];

 Riemann\_Problem(P\_prev[k,l-1,m],P\_prev[k,l,m],Rh\_prev[k,l-1,m],

 Rh\_prev[k,l,m],un\_2,un\_1,ut1\_2,ut1\_1,ut2\_2,

 ut2\_1,p3,rh3,Un\_g,Ut1\_g,Ut2\_g);

 U3:=Un\_g\*nx3+Ut1\_g\*t1x3+Ut2\_g\*t2x3;

 V3:=Un\_g\*ny3+Ut1\_g\*t1y3+Ut2\_g\*t2y3;

 W3:=Un\_g\*nz3+Ut1\_g\*t1z3+Ut2\_g\*t2z3;

 end;

 delta:=sqrt(sqr(h\_z)+sqr((l+1)/LL\*(Rad[k+1]-Rad[k])));

 cos\_xi:=h\_z/delta;

 sin\_xi:=(l+1)\*(Rad[k+1]-Rad[k])/LL/delta;

 nx4:=cos\_xi\*Cos(Pi\*(2\*m+1)/MM);

 ny4:=cos\_xi\*Sin(Pi\*(2\*m+1)/MM);

 nz4:=Sin\_xi;

 t1x4:=-sin(Pi\*(2\*m+1)/MM);

 t1y4:=cos(Pi\*(2\*m+1)/MM);

 t1z4:=0;

 t2x4:=sin\_xi\*cos(Pi\*(2\*m+1)/MM);

 t2y4:=sin\_xi\*sin(Pi\*(2\*m+1)/MM);

 t2z4:=-cos\_xi;

 un\_1:=nx4\*U\_prev[k,l,m]+ny4\*V\_prev[k,l,m]+nz4\*W\_prev[k,l,m];

 ut1\_1:=t1x4\*U\_prev[k,l,m]+t1y4\*V\_prev[k,l,m]+t1z4\*W\_prev[k,l,m];

 ut2\_1:=t2x4\*U\_prev[k,l,m]+t2y4\*V\_prev[k,l,m]+t2z4\*W\_prev[k,l,m];

 if l=LL-1 then Riemann\_Problem(P\_prev[k,l,m],P\_prev[k,l,m],

 Rh\_prev[k,l,m],Rh\_Prev[k,l,m],

 Un\_1,-Un\_1,Ut1\_1,Ut1\_1,ut2\_1,Ut2\_1,

 p4,rh4,Un\_g,Ut1\_g,Ut2\_g)

 else

 begin

 un\_2:=nx4\*U\_prev[k,l+1,m]+ny4\*V\_prev[k,l+1,m]+nz4\*W\_prev[k,l+1,m];

 ut1\_2:=t1x4\*U\_prev[k,l+1,m]+t1y4\*V\_prev[k,l+1,m]+t1z4\*W\_prev[k,l+1,m];

 ut2\_2:=t2x4\*U\_prev[k,l+1,m]+t2y4\*V\_prev[k,l+1,m]+t2z4\*W\_prev[k,l+1,m];

 Riemann\_Problem(P\_prev[k,l,m],P\_prev[k,l+1,m],Rh\_prev[k,l,m],

 Rh\_prev[k,l+1,m],un\_1,un\_2,ut1\_1,ut1\_2,

 ut2\_1,ut2\_2,p4,rh4,Un\_g,Ut1\_g,Ut2\_g);

 end;

 U4:=Un\_g\*nx4+Ut1\_g\*t1x4+Ut2\_g\*t2x4;

 V4:=Un\_g\*ny4+Ut1\_g\*t1y4+Ut2\_g\*t2y4;

 W4:=Un\_g\*nz4+Ut1\_g\*t1z4+Ut2\_g\*t2z4;

 if m=0 then

 begin

 nx5:=1;

 ny5:=0;

 nz5:=0;

 t1x5:=0;

 t1y5:=0;

 t1z5:=1;

 t1x5:=0;

 t1y5:=1;

 t1z5:=0;

 s5x:=Abs(s5\*nx5);

 s5y:=Abs(s5\*ny5);

 s5z:=Abs(s5\*nz5);

 un\_1:=nx5\*U\_prev[k,l,0]+ny5\*V\_prev[k,l,0]+nz5\*W\_prev[k,l,0];

 ut1\_1:=t1x5\*U\_prev[k,l,0]+t1y5\*V\_prev[k,l,0]+t1z5\*W\_prev[k,l,0];

 ut2\_1:=t2x5\*U\_prev[k,l,0]+t2y5\*V\_prev[k,l,0]+t2z5\*W\_prev[k,l,0];

 un\_2:=nx5\*U\_prev[k,l,MM-1]+ny5\*V\_prev[k,l,MM-1]+nz5\*W\_prev[k,l,MM-1];

 ut1\_2:=t1x5\*U\_prev[k,l,MM-1]+t1y5\*V\_prev[k,l,MM-1]+t1z5\*W\_prev[k,l,MM-1];

 ut2\_2:=t2x5\*U\_prev[k,l,MM-1]+t2y5\*V\_prev[k,l,MM-1]+t2z5\*W\_prev[k,l,MM-1];

 Riemann\_Problem(P\_prev[k,l,MM-1],P\_prev[k,l,0],Rh\_prev[k,l,MM-1],

 Rh\_prev[k,l,0],un\_2,un\_1,ut1\_2,

 ut1\_1,ut2\_2,ut2\_1,p5,rh5,Un\_g,Ut1\_g,Ut2\_g);

 end

 else

 begin

 nx5:=Cos(2\*Pi\*m/MM);

 ny5:=-Sin(2\*Pi\*m/MM);

 nz5:=0;

 t1x5:=0;

 t1y5:=0;

 t1z5:=1;

 t2x5:=Sin(2\*Pi\*m/MM);

 t2y5:=Cos(2\*Pi\*m/MM);

 t2z5:=0;

 s5x:=Abs(s5\*nx5);

 s5y:=Abs(s5\*ny5);

 s5z:=Abs(s5\*nz5);

 un\_1:=nx5\*U\_prev[k,l,m]+ny5\*V\_prev[k,l,m]+nz5\*W\_prev[k,l,m];

 ut1\_1:=t1x5\*U\_prev[k,l,m]+t1y5\*V\_prev[k,l,m]+t1z5\*W\_prev[k,l,m];

 ut2\_1:=t2x5\*U\_prev[k,l,m]+t2y5\*V\_prev[k,l,m]+t2z5\*W\_prev[k,l,m];

 un\_2:=nx5\*U\_prev[k,l,m-1]+ny5\*V\_prev[k,l,m-1]+nz5\*W\_prev[k,l,m-1];

 ut1\_2:=t1x5\*U\_prev[k,l,m-1]+t1y5\*V\_prev[k,l,m-1]+t1z5\*W\_prev[k,l,m-1];

 ut2\_2:=t2x5\*U\_prev[k,l,m-1]+t2y5\*V\_prev[k,l,m-1]+t2z5\*W\_prev[k,l,m-1];

 Riemann\_Problem(P\_prev[k,l,m-1],P\_prev[k,l,m],Rh\_prev[k,l,m-1],

 Rh\_prev[k,l,m],un\_2,un\_1,ut1\_2,ut1\_1,ut2\_2,

 ut2\_1,p5,rh5,Un\_g,Ut1\_g,Ut2\_g);

 end;

 U5:=Un\_g\*nx5+Ut1\_g\*t1x5+Ut2\_g\*t2x5;

 V5:=Un\_g\*ny5+Ut1\_g\*t1y5+Ut2\_g\*t2y5;

 W5:=Un\_g\*nz5+Ut1\_g\*t1z5+Ut2\_g\*t2z5;

 if m=MM-1 then

 begin

 nx6:=1;

 ny6:=0;

 nz6:=0;

 t1x6:=0;

 t1y6:=0;

 t1z6:=1;

 t2x6:=0;

 t2y6:=1;

 t2z6:=0;

 s6x:=Abs(s6\*nx6);

 s6y:=Abs(s6\*ny6);

 s6z:=Abs(s6\*nz6);

 un\_1:=nx6\*U\_prev[k,l,MM-1]+ny6\*V\_prev[k,l,MM-1]+nz6\*W\_prev[k,l,MM-1];

 ut1\_1:=t1x6\*U\_prev[k,l,MM-1]+t1y6\*V\_prev[k,l,MM-1]+t1z6\*W\_prev[k,l,MM-1];

 ut2\_1:=t2x6\*U\_prev[k,l,MM-1]+t2y6\*V\_prev[k,l,MM-1]+t2z6\*W\_prev[k,l,MM-1];

 un\_2:=nx6\*U\_prev[k,l,0]+ny6\*V\_prev[k,l,0]+nz6\*W\_prev[k,l,0];

 ut1\_2:=t1x6\*U\_prev[k,l,0]+t1y6\*V\_prev[k,l,0]+t1z6\*W\_prev[k,l,0];

 ut2\_2:=t2x6\*U[k,l,0]+t2y6\*V\_prev[k,l,0]+t2z6\*W\_prev[k,l,0];

 Riemann\_Problem(P\_prev[k,l,0],P\_prev[k,l,MM-1],Rh\_prev[k,l,0],

 Rh\_prev[k,l,MM-1],un\_1,un\_2,ut1\_1,ut1\_2,ut2\_1,

 ut2\_2,p6,rh6,Un\_g,Ut1\_g,Ut2\_g);

 end

 else

 begin

 nx6:=Cos(2\*Pi\*(m+1)/MM);

 ny6:=-Sin(2\*Pi\*(m+1)/MM);

 nz6:=0;

 t1x6:=0;

 t1y6:=0;

 t1z6:=1;

 t2x6:=Sin(2\*Pi\*(m+1)/MM);

 t2y6:=Cos(2\*Pi\*(m+1)/MM);

 t2z6:=0;

 s6x:=Abs(s6\*nx6);

 s6y:=Abs(s6\*ny6);

 s6z:=Abs(s6\*nz6);

 un\_1:=nx6\*U\_prev[k,l,m]+ny6\*V\_prev[k,l,m]+nz6\*W\_prev[k,l,m];

 ut1\_1:=t1x6\*U\_prev[k,l,m]+t1y6\*V\_prev[k,l,m]+t1z6\*W\_prev[k,l,m];

 ut2\_1:=t2x6\*U\_prev[k,l,m]+t2y6\*V\_prev[k,l,m]+t2z6\*W\_prev[k,l,m];

 un\_2:=nx6\*U\_prev[k,l,m+1]+ny6\*V\_prev[k,l,m+1]+nz6\*W\_prev[k,l,m+1];

 ut1\_2:=t1x6\*U\_prev[k,l,m+1]+t1y6\*V\_prev[k,l,m+1]+t1z6\*W\_prev[k,l,m+1];

 ut2\_2:=t2x6\*U\_prev[k,l,m+1]+t2y6\*V\_prev[k,l,m+1]+t2z6\*W\_prev[k,l,m+1];

 Riemann\_Problem(P\_prev[k,l,m],P\_prev[k,l,m+1],Rh\_prev[k,l,m],

 Rh\_prev[k,l,m+1],un\_1,un\_2,ut1\_1,ut1\_2,ut2\_1,

 ut2\_2,p6,rh6,Un\_g,Ut1\_g,Ut2\_g);

 end;

 U6:=Un\_g\*nx6+Ut1\_g\*t1x6+Ut2\_g\*t2x6;

 V6:=Un\_g\*ny6+Ut1\_g\*t1y6+Ut2\_g\*t2y6;

 W6:=Un\_g\*nz6+Ut1\_g\*t1z6+Ut2\_g\*t2z6;

 A1:=Rh1\*(U1\*S1x+V1\*S1y+W1\*S1z);

 A2:=Rh2\*(U2\*S2x+V2\*S2y+W2\*S2z);

 A3:=Rh3\*(U3\*S3x+V3\*S3y+W3\*S3z);

 A4:=Rh4\*(U4\*S4x+V4\*S4y+W4\*S4z);

 A5:=Rh5\*(U5\*S5x+V5\*S5y+W5\*S5z);

 A6:=Rh6\*(U6\*S6x+V6\*S6y+W6\*S6z);

 Rh[k,l,m]:=Rh\_prev[k,l,m]-Tau\_V\*(A2-A1+A4-A3+A6-A5);

 if Abs(Rh[k,l,m])<eps then Rh[k,l,m]:=0;

 B1:=A1\*U1\*P1\*S1x;

 B2:=A2\*U2\*P2\*S2x;

 B3:=A3\*U3\*P3\*S3x;

 B4:=A4\*U4\*P4\*S4x;

 B5:=A5\*U5\*P5\*S5x;

 B6:=A6\*U6\*P6\*S6x;

 U[k,l,m]:=(Rh\_prev[k,l,m]\*U\_prev[k,l,m]-Tau\_v\*

 (B2-B1+B4-B3+B6-B5))/Rh[k,l,m];

 if Abs(U[k,l,m])<eps then U[k,l,m]:=0;

 C1:=A1\*V1\*P1\*S1y;

 C2:=A2\*V2\*P2\*S2y;

 C3:=A3\*V3\*P3\*S3y;

 C4:=A4\*V4\*P4\*S4y;

 C5:=A5\*V5\*P5\*S5y;

 C6:=A6\*V6\*P6\*S6y;

 V[k,l,m]:=(Rh\_prev[k,l,m]\*V\_prev[k,l,m]-Tau\_v\*

 (C2-C1+C4-C3+C6-C5))/Rh[k,l,m];

 if Abs(V[k,l,m])<eps then V[k,l,m]:=0;

 D1:=A1\*w1\*P1\*S1z;

 D2:=A2\*w2\*P2\*S2z;

 D3:=A3\*w3\*P3\*S3z;

 D4:=A4\*w4\*P4\*S4z;

 D5:=A5\*w5\*P5\*S5z;

 D6:=A6\*w6\*P6\*S6z;

 W[k,l,m]:=(Rh\_prev[k,l,m]\*W\_prev[k,l,m]-Tau\_v\*

 (D2-D1+D4-D3+D6-D5))/Rh[k,l,m];

 if Abs(W[k,l,m])<eps then W[k,l,m]:=0;

 H1:=gam/(gam-1)\*P1/Rh1+0.5\*(sqr(U1)+sqr(V1)+sqr(W1));

 H2:=gam/(gam-1)\*P2/Rh2+0.5\*(sqr(U2)+sqr(V2)+sqr(W2));

 H3:=gam/(gam-1)\*P3/Rh3+0.5\*(sqr(U3)+sqr(V3)+sqr(W3));

 H4:=gam/(gam-1)\*P4/Rh4+0.5\*(sqr(U4)+sqr(V4)+sqr(W4));

 H5:=gam/(gam-1)\*P5/Rh5+0.5\*(sqr(U5)+sqr(V5)+sqr(W5));

 H6:=gam/(gam-1)\*P6/Rh6+0.5\*(sqr(U6)+sqr(V6)+sqr(W6));

 F1:=A1\*H1;

 F2:=A2\*H2;

 F3:=A3\*H3;

 F4:=A4\*H4;

 F5:=A5\*H5;

 F6:=A6\*H6;

 Energy:=(P\_prev[k,l,m]/(gam-1)+Rh\_prev[k,l,m]\*0.5\*

 (sqr(U\_prev[k,l,m])+sqr(V\_prev[k,l,m])+sqr(W\_prev[k,l,m])))

 -Tau\_V\*(F2-F1+F4-F3+F6-F5);

 P[k,l,m]:=(gam-1)\*(Energy-0.5\*Rh[k,l,m]\*

 (sqr(U[k,l,m])+sqr(V[k,l,m])+sqr(W[k,l,m])));

 if Abs(P[k,l,m])<eps then P[k,l,m]:=0;

 // Writeln(t,k,' ',l,' ',m,' ',P[k,l,m],' ',Rh[k,l,m],

 // ' ',U[k,l,m],' ',V[k,l,m],' ',W[k,l,m]);

 //if m=MM-1 then Writeln(t);

 end;

 //writeln(t,'Конец');

 //close(t);

 until Norma or (Iter>500);

 Print\_Results;

 end;

procedure TForm1.Start\_ButtonClick(Sender: TObject);

begin

 kkk:=0;

 //Входные параметры

 L1:=StrToFloat(Edit\_L1.text);

 L2:=StrToFloat(Edit\_L2.text);

 R1:=StrToFloat(Edit\_R1.text);

 Rc:=StrToFloat(Edit\_Rc.text);

 R2:=StrToFloat(Edit\_R2.text);

 p0:=StrToFloat(Edit\_p0.text);

 Temperature:=StrToFloat(Edit\_rho0.text);

 gam:=StrToFloat(Edit\_gam.text);

 KK1:=StrToInt(Edit\_KK1.text);

 KK2:=StrToInt(Edit\_KK2.text);

 LL:=StrToInt(Edit\_LL.text);

 MM:=StrToInt(Edit\_MM.text);

 R\_Gaz:=StrToFloat(Edit\_R\_gaz.text);

 rho0:=P0/(R\_Gaz\*Temperature);

 N\_P:=(KK1+KK2+1)\*(LL+1)\*(MM);

 N\_El:=(KK1+KK2)\*(LL)\*(MM);

//обезразмеривание геометрических параметров

L1\_:=L1/Rc;

L2\_:=L2/Rc;

 R1\_:=R1/Rc;

 R2\_:=R2/Rc;

 hz1:=L1\_/KK1;

 hz2:=L2\_/KK2;

//вычисление размерных коэффициентов для времени,скорости и давления

U\_dim:=sqrt(p0/rho0);

 T\_dim:=Rc/U\_dim;

//вычисление констант для определения чисел маха

temp\_1\_functional:=0.5\*(gam+1)/(gam-1);

 temp\_2\_functional:=Exp(Ln(0.5\*(gam+1))\*temp\_1\_functional);

 Radius; //вычисление радиусов сечений для каждого zk

Cell\_Geometry; //вычисление геометрических параметров ячеек

First\_Guess; //начальное приближение

Godunov;

end;

end.