

**РОССИЙСКАЯ АКАДЕМИЯ НАУК  
ОРДЕНА ЛЕНИНА  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
имени М.В. Келдыша**

А.Н. Андрианов, С.Б. Базаров, А.Б. Бугеря,  
П.И. Колударов, И.М. Набоко

**Применение отладчика параллельных программ  
при решении задачи о фокусировке ударных  
и взрывных волн на многопроцессорных ЭВМ**

Москва, 2004 г.

Андрианов А.Н., Базаров С.Б., Бугеря А.Б., Колударов П.И., Набоко И.М.

Применение отладчика параллельных программ при решении задачи о фокусировке ударных и взрывных волн на многопроцессорных ЭВМ

### **Аннотация**

В данной работе описана работа с отладчиком параллельных программ на языке NORMA (с внешними модулями на Фортране). Приведены результаты численного моделирования задачи о фокусировке волн.

Andrianov A.N., Bazarov S.B., Bugerya A.B., Koludarov P.I., Naboko I.M.

Application of parallel programs debugger for multiCPU modeling of shock and blast waves focusing.

### **Abstract**

In this paper using of debugger for parallel programs on NORMA language (with external Fortran modules) is considered. The results of multiCPU modeling of shock and blast waves focusing are given.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных Исследований (проект 04-01-00197)

## Содержание.

Общие сведения.....	3
Последовательность шагов для запуска программы под отладчиком .....	4
1.  Генерация дополнительного кода.....	4
2.  Компиляция сгенерированного кода .....	4
Windows.....	4
UNIX.....	5
3.  Линковка с Библиотекой связи с монитором.....	5
Windows.....	5
UNIX.....	6
4.  Запуск отладчика.....	6
Локальная отладка.....	7
Удаленная отладка.....	7
Запуск и опции монитора.....	8
Постановка задачи.....	9
Математическая модель и метод численного решения.....	9
Результаты моделирования.....	10
Заключение.....	11
Литература.....	12

Появление различных архитектур ЭВМ, позволяющих выполнять параллельные вычисления, дает возможность проводить математическое моделирование различных физических процессов с высоким уровнем точности. Но при этом возникает проблема отладки программного обеспечения, являющаяся гораздо более трудной задачей, чем процесс отладки на компьютерах традиционной однопроцессорной архитектуры. В данной работе описана работа отладчика параллельных программ на языке НОРМА (с внешними модулями на Фортране). Приводится решение полученной на кластерной архитектуре (с использованием MPI-библиотеки) задачи о фокусировке волн, которое было получено с его использованием.

### *Общие сведения*

Отладчик программ на языке НОРМА позволяет отлаживать в терминах исходной программы на языке НОРМА исполняемую программу. На данный момент поддерживается только отладка генерированных Фортран программ. Отладчик состоит из следующих составных частей:

1. Пользовательский интерфейс (User Interface, UI), являющийся частью Интегрированной Среды Разработки программ на языке НОРМА (смотрите также руководство по Интегрированной Среде Разработки). Выполняется на компьютере пользователя, демонстрирует ход выполнения отлаживаемой программы и предоставляет удобный интерфейс ко всем функциям отладчика. Всю информацию для визуализации процесса отладки UI получает от Монитора, ему же отправляет все команды и запросы пользователя, без какой-либо предварительной обработки.
2. Монитор. Основная программа, управляющая выполнением отлаживаемой программы. Все данные (информация о выполняемых разделах и операторах, переменные, их типы, размеры и распределение и т.д.) получает от параллельно выполняющихся модулей отлаживаемой программы через Библиотеку связи с монитором. Монитор синхронизирует параллельные отлаживаемые процессы, определяет факт начала и конца выполнения разделов и каждого оператора, на основании информации о точках останова и текущем режиме (по шагам, и т.п.) отладки принимает решение об останове или продолжении счета, информирует UI о происходящих событиях, выдает команды продолжения счета, запросы на

получения значений переменных конкретным процессам, обрабатывает запросы и команды пользователя, полученные от UI и т.п. Можно сказать, что Монитор является логическим и управляющим центром отладчика.

3. Библиотека связи с монитором. Находится в каждом выполняемом модуле отлаживаемой параллельной программы. Взаимодействует с Монитором, посылая ему информацию и ожидая ответных команд (таким образом приостанавливая выполнение модуля отлаживаемой параллельной программы). Вызывается из отлаживаемой программы через вызовы подпрограмм.
4. Код, дополнительно встраиваемый транслятором в генерируемую программу при задании режима трансляции "для отладки":

*Последовательность шагов для запуска программы под отладчиком*

### **1. Генерация дополнительного кода.**

При трансляции программы с языка FORTRAN должна быть задана опция компилятора 'debug' для генерации вызовов процедур отладчика. Например (Windows):

```
>normarc.exe test.hop mpi debug
```

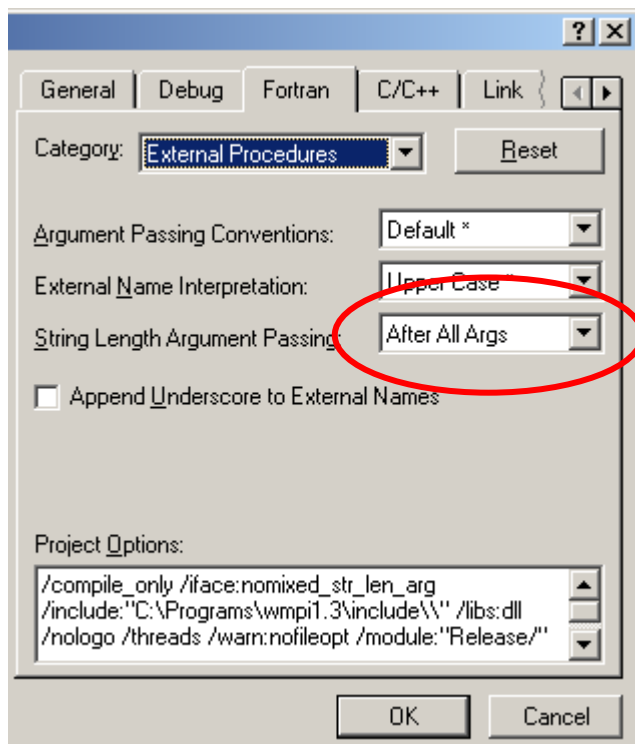
Программа, сгенерированная с данной опцией может выполняться только под отладчиком.

### **2. Компиляция сгенерированного кода**

#### *Windows*

1. Компилятор Фортрана должен при передаче строк в качестве фактических параметров передавать неявный параметр "длина строки" после всех аргументов.

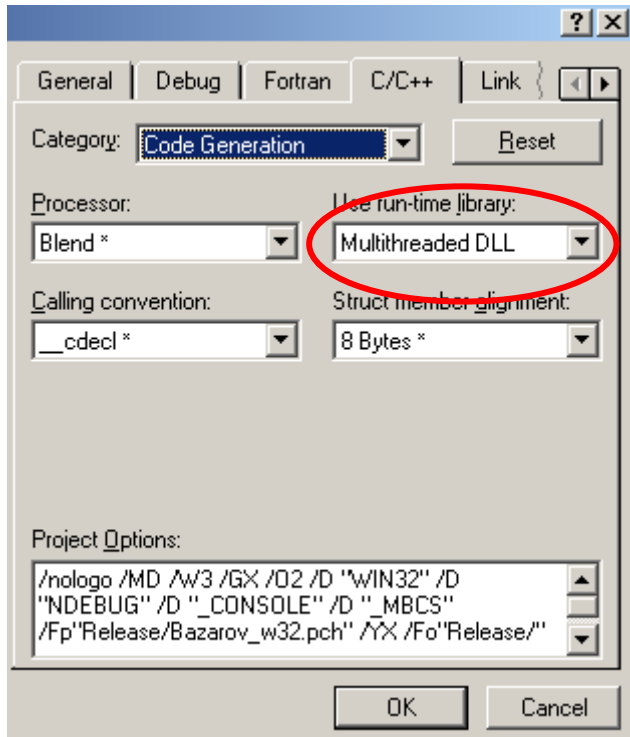
При использовании Compaq Visual Fortran v 6.1 поставьте в Project Settings ->



Fortran “Category:” в “External Procedures” и затем “String Length Argument Passing” в “After All Args”.

## 2. Использовать Run-Time Library надо Multithreaded DLL.

При использовании Compaq Visual Fortran v 6.1 поставьте в Project Settings -> C/C++ “Category:” в “Code Generation” и затем “Use run-time Library” – в “Multithreaded DLL”.



## UNIX

Компилятор Фортрана должен при передаче строк в качестве фактических параметров передавать неявный параметр “длина строки” после всех аргументов.

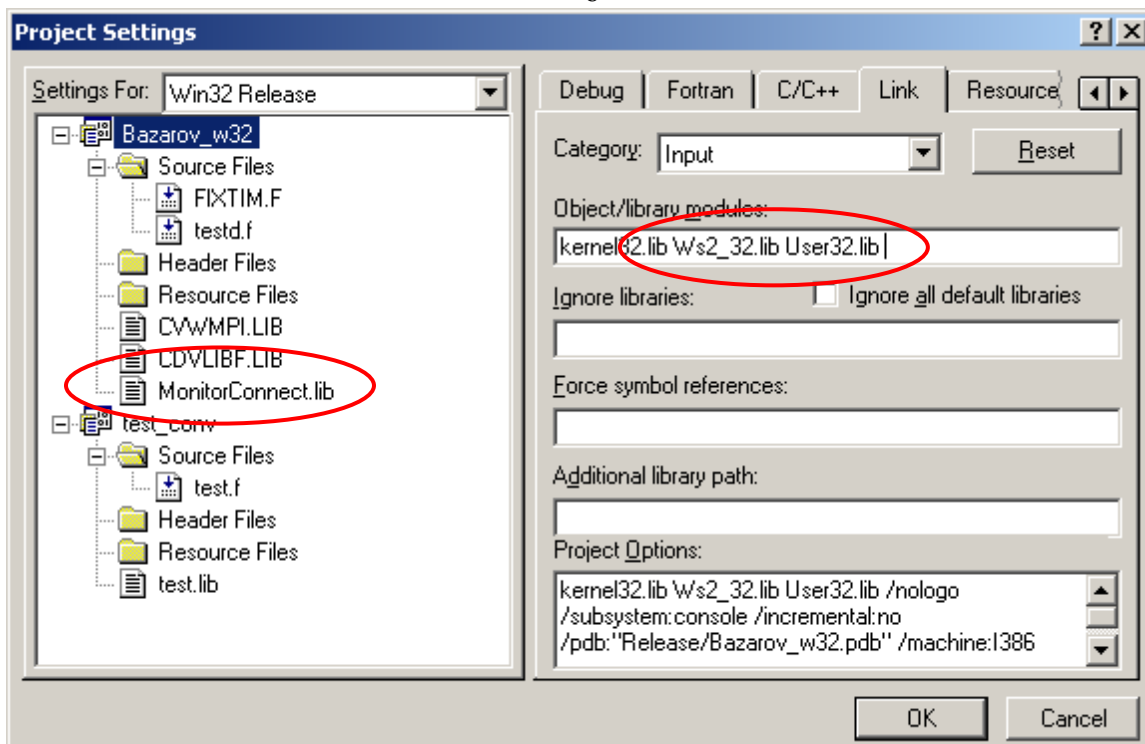
Компилятор g77 v 2.95.3 делает это по умолчанию.

## 3. Линковка с Библиотекой связи с монитором.

### Windows

Библиотека связи с монитором *MonitorConnect.lib* должна быть прилинкована к выполняемой программе, т.е. включена в проект или указана в дополнительных библиотеках, требуемых для сборки. Также необходимы системные библиотеки *Ws2\_32.lib* *User32.lib*.

При использовании Compaq Visual Fortran v 6.1 поставьте в Project Settings -> Link “Category:” в “Input” и затем в “Object/Library modules:” укажите все необходимые библиотеки.



## UNIX

Библиотека связи с монитором *libndebug.a* должна быть прилинкована к выполняемой программе, т.е. включена в проект или указана в дополнительных библиотеках, требуемых для сборки. Сделать это можно просто указав ее в команде сборки, например:

```
mpif77 -o mpi_go summa.f libndebug.a
```

## 4. Запуск отладчика

Для старта отладчика вначале запускаются Интерфейс Пользователя в составе Интегрированной Среды (смотрите описание работы с Интегрированной Средой Разработки программ на языке НОРМА) и монитор. Они устанавливают соединение друг с другом, монитор создает файл с информацией для отлаживаемой программы, и затем запускается отлаживаемая программа и соединяется с монитором.

Все взаимодействия (между Библиотекой связи с монитором и Монитором, между Монитором и Пользовательским интерфейсом) реализованы с помощью семейства протоколов TCP/IP, используя любой свободный TCP/IP порт. Сделаем небольшое отступление о том, как организуется взаимодействие по протоколу TCP/IP и о сложностях, возникающих из-за “закрытости” портов по соображениям безопасности.

Один из взаимодействующих процессов выступает в роли “Сервера”, другой – “Клиента”. Сервер занимает определенный TCP/IP порт (если он уже не занят на данный момент никаким другим приложением) на компьютере, на котором он был запущен, и начинает его “слушать”. Клиент для установления соединения обращается к компьютеру, где запущен Сервер, по тому TCP/IP порту, который слушает Сервер (т.е. Клиент должен знать TCP/IP адрес и номер порта Сервера). Сервер, получив

запрос на соединение, устанавливает его. После этого Клиент и Сервер могут начать обмениваться сообщениями.

Существует достаточно большой набор так называемых “стандартных” протоколов, построенных на основе TCP/IP и занимающих заранее известные TCP/IP порты. Так, например, сервер, обеспечивающий передачу данных по протоколу FTP, слушает и устанавливает соединения по TCP/IP порту 21, HTTP сервер использует TCP/IP порт 80 и т.д. Остальные программы, к которым относится и наш отладчик, могут использовать любые другие свободные порты. Но здесь могут возникать сложности, связанные с установленными на компьютере, где выполняется отлаживаемая программа, и в сети, обеспечивающей доступ к этому компьютеру, средств обеспечения компьютерной безопасности. Чаще всего системные администраторы в настройках брандмауэров запрещают использование всех TCP/IP портов, кроме тех, которые необходимы установленному в сети программному обеспечению.

Для наибольшей гибкости и успешного преодоления сложностей в процессе коммуникации, связанных с настройками безопасности, и Монитор, и Пользовательский интерфейс могут выступать как в качестве Клиента, так и в качестве Сервера при установке соединения между собой. Библиотека связи с монитором выступает всегда в роли Клиента, а Монитор – в роли сервера при установке соединения между ними, т.к. в данном случае обратное распределение ролей бесполезно. Если существует возможность установить соединение от Монитора-Клиента к отлаживаемой задаче-Серверу, то и обратную связь можно будет также установить.

По-другому обстоит дело с Пользовательским интерфейсом, выполняющемся на компьютере пользователя. Зачастую для доступа к удаленному компьютеру пользователь может использовать узкий набор стандартных интерфейсов (скажем, SSH, HTTP) и все, использование остальных TCP/IP портов закрыто по соображениям безопасности. Запустить Монитор на удаленном компьютере с использованием какого-то стандартного порта нельзя, т.к. там, на удаленном компьютере, все стандартные TCP/IP порты уже заняты. Предлагаемый в нашей схеме выход – использовать стандартные TCP/IP порты на своем, локальном компьютере. Таким образом, запустив Пользовательский интерфейс в роли сервера, обслуживающего, скажем, 80-й TCP/IP порт (на котором должен быть HTTP сервер), можно соединиться с ним Монитором-Клиентом, т.к. системы безопасности удаленного компьютера будут считать, что используется стандартный протокол.

### *Локальная отладка*

В режиме локальной отладки (смотрите описание работы с Интегрированной Средой Разработки программ на языке NORMA) выберите только порт для установления соединения (1500 по умолчанию) и укажите файл с выполняемой программой. Начните отладку в Интегрированной Среде, и монитор и выполняемая программа запустятся автоматически.

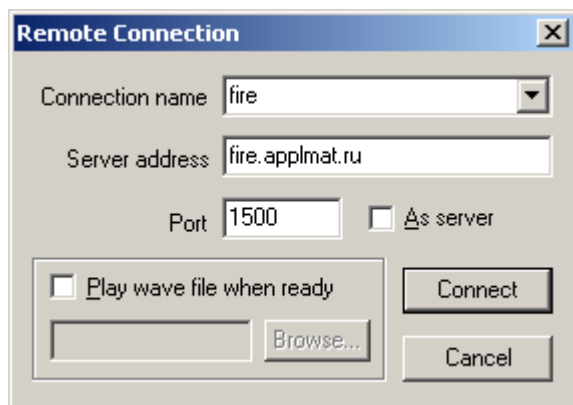
### *Удаленная отладка*

В режиме удаленной отладки запустите вначале либо монитор (смотрите ниже описание опций монитора), либо начните отладку в Интегрированной Среде – первым то, что должно выступать сервером при установке соединения. Затем запустите тот компонент, который должен выступать в роли клиента с указанием ему па-

раметров (адрес и порт) сервера. Например, если монитор запущен как сервер на удаленном компьютере и сообщил следующее:

```
Monitor for Norma debugging started
Server fire.applmat.ru ready on port 1500
Wait for the UI connect...
```

То диалог запуска удаленной отладки в Интегрированной Среде должен выглядеть следующим образом:



Затем, после установления соединения между монитором и Интерфейсом пользователя, монитор создает файл с информацией для отлаживаемой программы `NDmonpar.inf`. Этот файл должен находиться в директории, откуда запускается отлаживаемая программа. Она читает этот файл и устанавливает соединение с монитором. После этого начинается отладка.

### Запуск и опции монитора

Монитор представляет собой один исполняемый файл `Monitor.exe` под Windows и `monitor` под UNIX. Формат командной строки следующий:

```
monitor <arguments for mpi_run> [options for monitor]
```

Если в командной строке есть хоть один неизвестный монитору аргумент и не заданы опции монитору ни `-remote`, ни `-exec` (см. ниже их описание), то монитор вызывает на выполнение `'mpi_run'` и передает ему все нераспознанные аргументы.

Монитор имеет следующие опции:

`-remote`

Не вызывать `'mpi_run'`.

`-exec <programname>`

Вызывать не `'mpi_run'`, а указанную программу.

`-server <host>[:<port>]`

Выступить в роли клиента, присоединиться к указанному серверу по указанному порту. Если эта опция не задана, монитор будет выступать в роли сервера, занимая первый свободный порт начиная с 1500.



-wd <path>

Сменить текущую директорию на указанную.

Таким образом, типичный запуск монитора-сервера под UNIX выглядит следующим образом:

monitor -nr <кол-во процессов> <файл с исполняемой программой>

Поскольку численное решение задач газовой динамики предъявляет высокие требования к объему памяти и к быстродействию, был создан комплекс программ для моделирования широкого класса двух- и трехмерных задач на различных параллельных архитектурах с помощью языка НОРМА<sup>1</sup>. Использование вышеописанного отладчика позволило резко повысить как скорость разработки, так и качество соответствующего программного обеспечения. Ниже приводятся полученные на кластерной архитектуре результаты численного решения задачи о фокусировке волн.

#### *Постановка задачи.*

Рассматривается вхождение ударной или взрывной волны в полости различного вида. На рис.1 приведен один из рассматриваемых вариантов постановки задачи – движение плоской ударной волны заданной интенсивности и ее взаимодействие с полостью. Приводимые результаты расчетов вхождения волн в конус оригинальны в том отношении, что впервые проводилось моделирование кумуляции неплоских (сферических) волн.

#### *Математическая модель и метод численного решения.*

Рассматривается течение невязкого сжимаемого идеального нетеплопроводного газа. Описание такого течения проводится в переменных Эйлера. Двумерные нестационарные уравнения берутся в виде следующей системы :

$$\begin{aligned} \frac{\partial}{\partial t}(\rho y^\delta) + \frac{\partial}{\partial x}(\rho u y^\delta) + \frac{\partial}{\partial y}(\rho v y^\delta) &= 0, \\ \frac{\partial}{\partial t}(\rho u y^\delta) + \frac{\partial}{\partial x}[(\rho u^2 + p)y^\delta] + \frac{\partial}{\partial y}(\rho u v y^\delta) &= 0, \\ \frac{\partial}{\partial t}(\rho v y^\delta) + \frac{\partial}{\partial x}(\rho u v y^\delta) + \frac{\partial}{\partial y}[(\rho v^2 + p)y^\delta] &= \delta p, \\ \frac{\partial}{\partial t}(\rho e y^\delta) + \frac{\partial}{\partial x}[(\rho e + p)u y^\delta] + \frac{\partial}{\partial y}[(\rho e + p)v y^\delta] &= 0. \end{aligned}$$

Здесь  $t$  – время,  $\delta=0$  для плоского,  $\delta=1$  для осесимметричного течения,  $x, y$  – пространственные координаты (декартовы для плоского, осевая и радиальная соответственно для осесимметричного течения),  $u, v$  – компоненты вектора скорости  $\mathbf{w}=(u, v)$  по  $x$  и  $y$  соответственно,  $\rho$  – плотность,  $p$  – давление,  $e$  – удельная полная энергия. Система замыкается уравнением состояния в виде

$$p = (\gamma - 1)\rho \left( e - \frac{u^2 + v^2}{2} \right), \text{ где } \gamma \text{ – показатель адиабаты газа.}$$

Численное решение проводится методом С.К.Годунова<sup>2</sup>.

В расчётах размеры сетки всегда ограничены, поэтому на её границе ставятся условия гладкого продолжения течения, заключающиеся в приравнивании нулю производных искомых функций по нормали к внешним границам. Это означает рас-

чѐт в граничных ячейках распада разрыва с фиктивной ячейкой с такими же параметрами. Расчѐт в ячейках, прилегающих к линиям симметрии или неподвижным жѐстким стенкам, заключается в расчѐте распада разрыва с фиктивной ячейкой, в которой давление, плотность и продольная скорость такие же, а скорость по нормали к внешней границе имеет противоположный знак.

Заметим, что разностная схема исследована на устойчивость для линейной системы с постоянными коэффициентами и линеаризованным уравнением состояния. При проведении расчѐта шаг по времени выбирался равным

$$\Delta t = K \min(h_x, h_y) / \max_{i,j}(D_L, D_R), \quad i = 1, 2, \dots, N_x, j = 1, 2, \dots, N_y, \quad 0 < K < 1$$

где  $h_x, h_y$  – шаги сетки по соответствующим пространственным координатам,  $D_L, D_R$  – скорости соответственно крайних левой и правой волн, образующихся при распаде разрыва на границе между расчетными ячейками,  $K$  – коэффициент запаса. Смысл этого коэффициента следующий: его увеличение улучшает “вид” получаемого численного решения (например, увеличивается крутизна профиля ударной волны). Но при приближении его к единице появляются неустойчивости численного решения, делающие невозможным дальнейший расчет.

#### *Результаты моделирования.*

Получаемые в расчетах числовые данные дают представление о развитии течения при их интерпретации в виде: изолиний (например давления: вхождение взрывной волны в конус – рис. 2), распределения параметра течения вдоль координаты  $x$  (давление на рис. 3), изменения параметров в какой-либо фиксированной точке по времени (датчик в вершине конуса – рис. 4).

В качестве характеристики кумуляции в полости принимается значение степени фокусировки, определяемое как отношение максимального значения параметра, полученного внутри объема в процессе расчета, к значению, которое получилось бы при нормальном отражении от плоской стенки ударной волны с параметрами, равными локальным значениям их на входе в полость. Так, например, для волны, у которой при входе в полость давление на фронте равно  $P_{вх}=1.8$ , соответствующая волна с плоским фронтом имеет давление отражения, равное  $P_{торм}=3.1$ .

Параметрические расчеты показали изменение показателя фокусировки в зависимости от угла раствора. На рис.5 (вход плоской ударной волны с числом Маха 1.647 в конусовидную полость) показан характер изменения степени фокусировки по давлению и температуре в зависимости от угла полураствора ( $\alpha/2$  – рис.1). Следует отметить, что положения максимумов этих двух параметров отличаются. А изменение степени фокусировки по температуре носит более плавный характер.

Использование многопроцессорных машин позволило провести большое количество расчетов, в которых изменялся только коэффициент запаса  $K$  в условии устойчивости разностной схемы. На рис.6 приведены соответствующие результаты – зависимости степени фокусировки в вершине конусовидной полости от угла ее полураствора при различных коэффициентах запаса (число Маха ударной волны 2.4, показатель адиабаты газа  $\gamma=1.4$ ).

Полученные результаты показывают, что при изменении  $K$  максимумы степени фокусировки в вершине конуса реализуются при различных углах раствора конуса (80 градусов при  $K=0.5$  и 82 градус при  $K=0.6$  и  $0.7$ ). При этом чем ближе к этим значениям углов, тем сильнее различаются степени фокусировки. Так, при 64-градусном растворе конуса различие степени фокусировки при  $K=0.5$  и  $K=0.7$  составляет 8%, а для 82 градусов оно достигает 16%. Заметим также, что если при уг-

лах 60÷66 градусов ещё был возможен расчет с  $K = 0.8$ , то при дальнейшем увеличении угла такое значение коэффициента уже не позволяло вести расчет (т.е. нарушалась устойчивость разностной схемы).

В свете этого представляет интерес провести расчет и произвести сравнение с решением аналогичной задачей о фокусировке волны, полученной другим расчетным методом. На рис.7 приведена постановка такой задачи: движение ударной волны заданной интенсивности в плоском канале и ее взаимодействие с фокусирующей полостью, которой заканчивается канал. Задача в такой постановке рассмотрена в <sup>3</sup>. В этой работе фокусирующая полость заканчивается плоской стенкой, имеющей высоту  $H$ , не равную нулю. Нами же рассматривался и вариант «чисто» клиновидной полости.

Определение максимального значения давления проводилось как по всему расчетному полю, так и по значениям, зафиксированным датчиками, расположенными с некоторым шагом вдоль стенки полости. Результаты сведены в следующую таблицу:

Полуугол	H/R	Максимум по полю давления	Максимум по датчикам на стенке полости	Другой численный метод
30	0	4.184	5.792	
30	0.2	3.789	4.016	4.7
30	0.3	3.152	3.177	4.5
45	0	4.963	5.555	
45	0.1	3.996	3.8	7.1
45	0.3	2.486	2.513	2.7
60	0	2.33	2.333	
60	0.3	2.698	1.656	2.0

В случае, когда степень фокусировки в 3-й колонке больше, чем в 4-й, означает, что максимальное значение давления реализовывалось не на стенке полости (т.е. в том числе и не в её вершине). Как видно, может наблюдаться как близость результатов, полученных различными расчетными методами, так и их расхождение даже в рамках одного метода (что показывают и вышеприведенный рис.6).

#### *Заключение.*

Разработанное программное обеспечение позволяет в сжатые сроки проводить массовые параметрические расчеты, моделирующие нестационарные газодинамические течения.

Показано, что любой конкретный расчет может считаться достоверным только тогда, когда его результаты близки к результатам физического эксперимента. Данный расчет моделировал физический эксперимент <sup>4</sup>, выполненный для условий кумуляции взрывных волн в конусе.

*Литература.*

<sup>1</sup> А.Н.Андрианов, А.Б.Бугеря, К.Н.Ефимкин, И.Б.Задыхайло. НОРМА. Описание языка. Рабочий стандарт. Препринт ИМП им.М.В.Келдыша. №120. 1995.

<sup>2</sup> Годунов С.К., Забродин А.В., Иванов М.Я. и др. Численное решение многомерных задач газовой динамики. М. Наука. 1976.

<sup>3</sup> О.А.Васильев. Численное моделирование процессов фокусировки ударных волн и детонации в химически неравновесных газах. Дисс. к.ф.-м.н. МАИ. Москва. 1995.

<sup>4</sup> С.Б.Базаров, П.А.Гусев, И.М.Набоко, В.А.Петухов, О.И.Солнцев. Кумуляция взрывных волн в конусе в физическом и вычислительном экспериментах. Научные труды института теплофизики экстремальных состояний ОИВТ РАН. М. 2003.

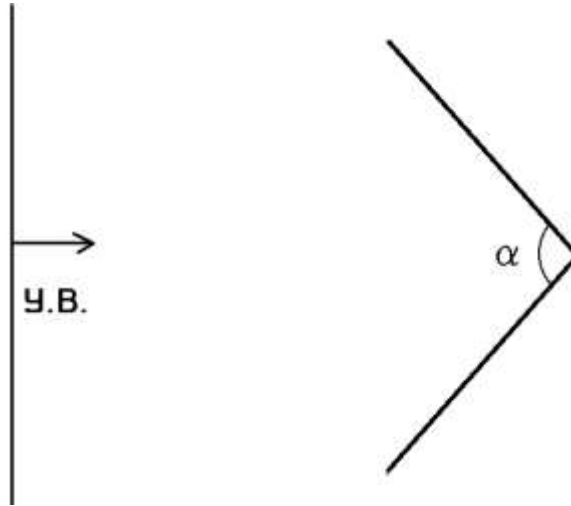


Рис.1

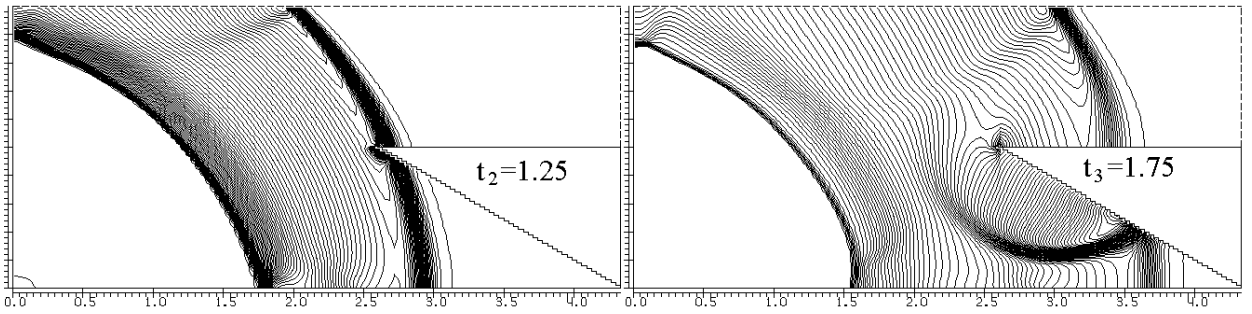


Рис. 2.

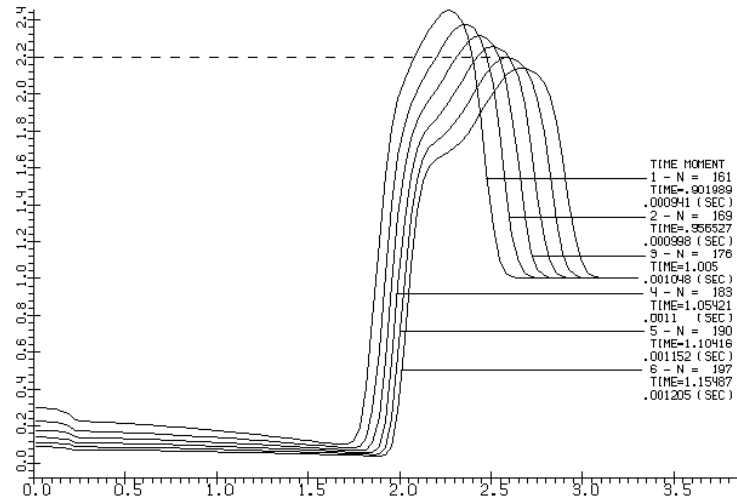


Рис. 3.

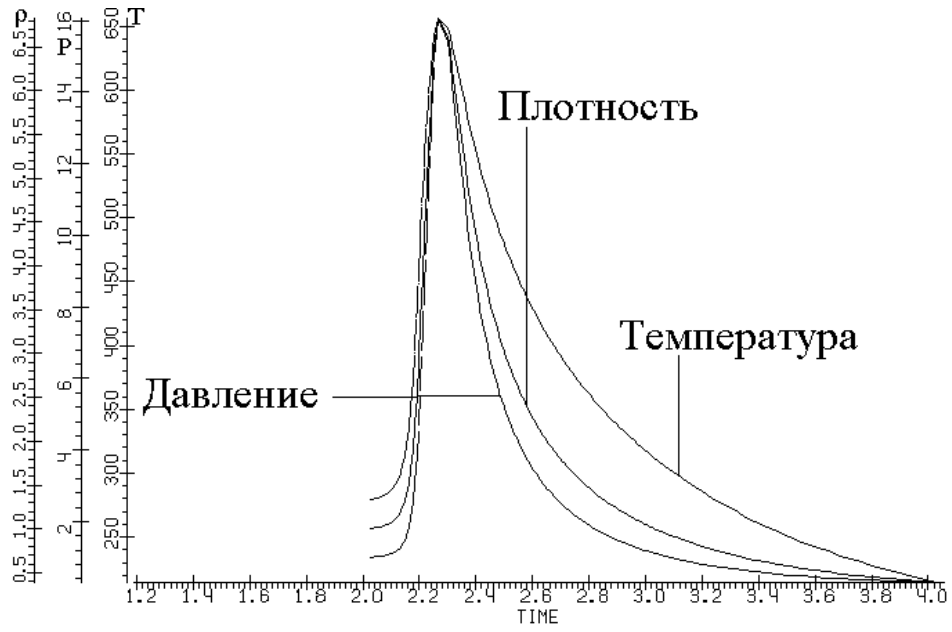


Рис. 4.

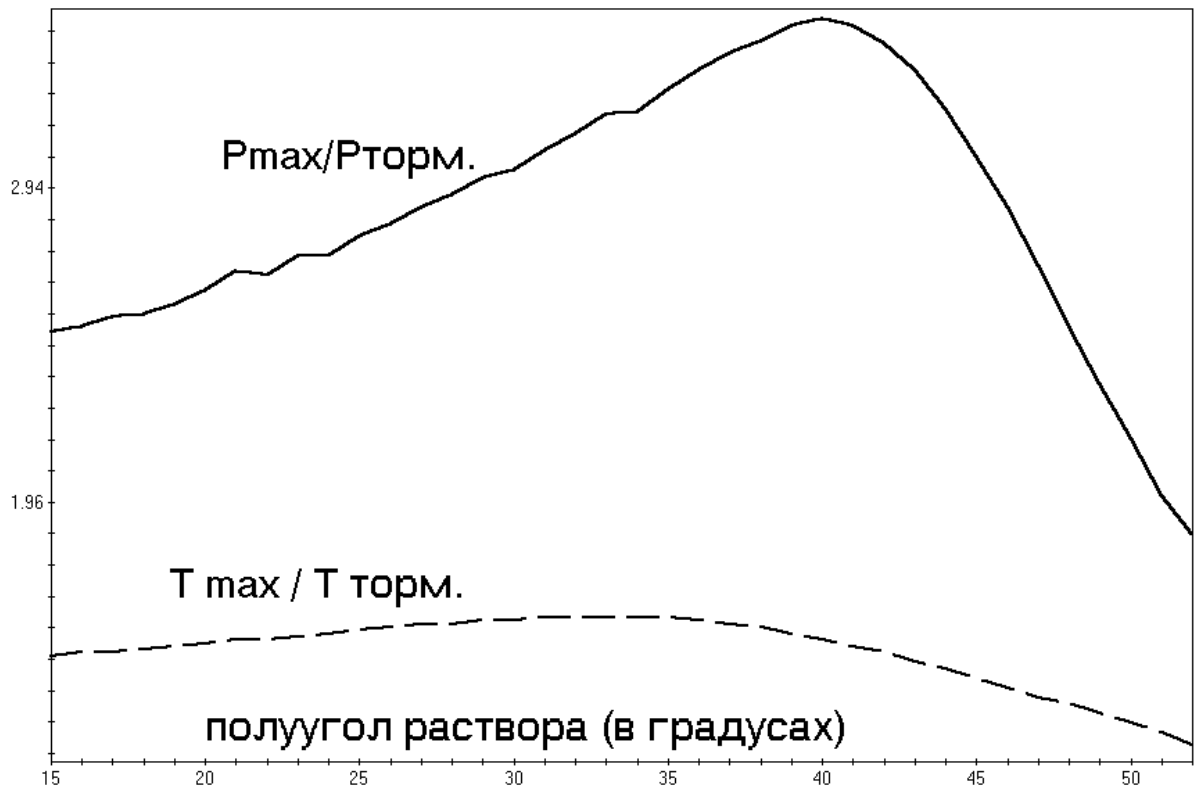


Рис. 5.

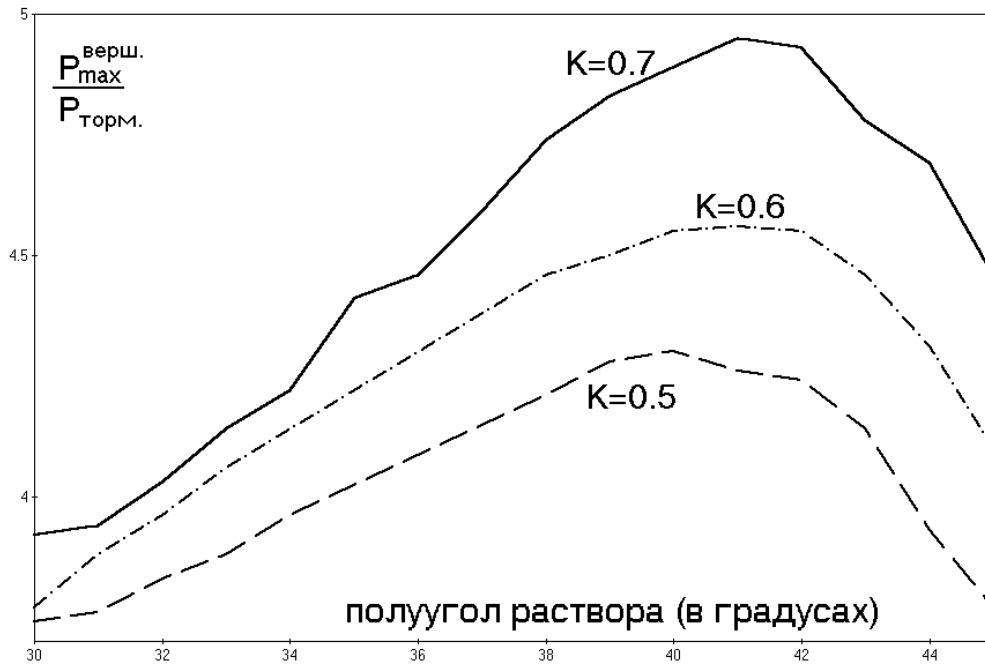


Рис. 6.

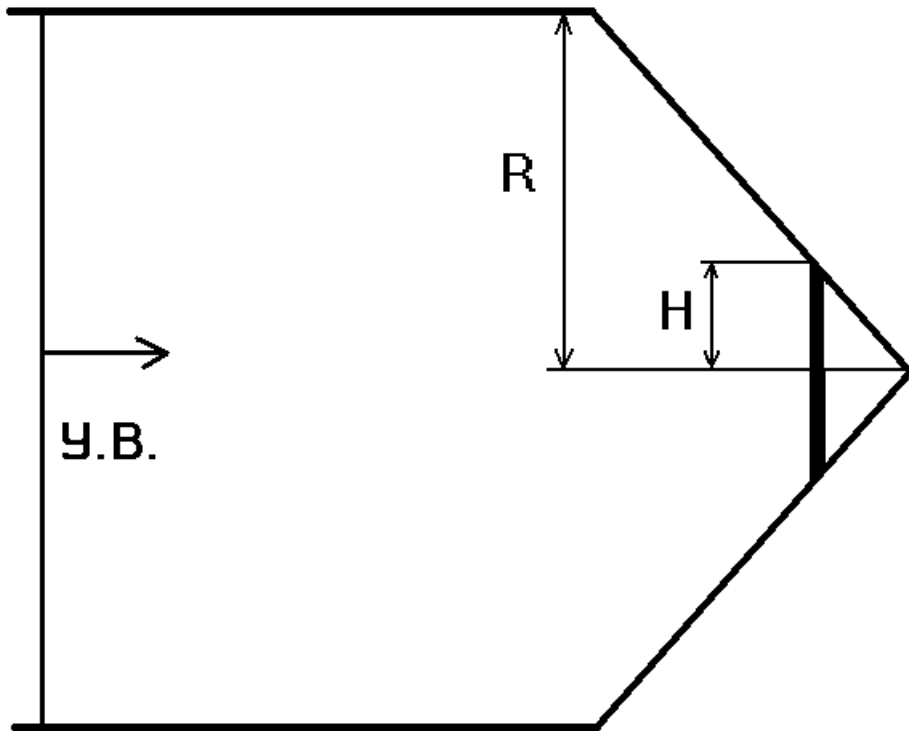


Рис.7