

в текущий архив переписуются модули OLM 101, OLM 102 и DIA 306, расположенные в архиве, начинающемся с 1140-й зоны диска 2145.

Если в текущем архиве уже есть модуль с указанным именем, то перепись данного модуля не производится и выдается диагностическое сообщение. Диагностическое сообщение не выдается только в случае, если тексты модулей идентичны.

## ЛИТЕРАТУРА

1. Карпов В. Я., Корягин Д. А., Самарский А. А. Принципы разработки пакетов прикладных программ для задач математической физики.— ЖВМиМФ, 1978, т. 18, № 2, с. 458—467.
2. Roberts K. V. An introduction to the OLYMPUS system.— Computer physics communications, 1974, vol. 7, p. 237—247.
3. Горбунов-Посадов М. М., Корягин Д. А., Красотченко В. В. Реализация системного наполнения пакета прикладных программ САФРА. (Версия 2.0): Препр. ИПМ АН СССР № 186. М., 1979. 26 с.
4. Усов С. А. Диалоговый монитор ДИМОН. М.: Наука, 1979. 128 с.
5. Петренко А. К. Диалоговый монитор ДИМОН. Новые возможности пакетной обработки в ДИМОНе: Препр. ИПМ АН СССР № 139. М., 1978. 48 с.
6. Бокова И. Д., Зельдинова С. А., Зуев В. И. и др. Операционная система ДИСПАК для БЭСМ-6. Пользователю: Препр. ИПМ АН СССР № 1. М., 1973. 60 с.
7. Мазный Г. Л. Программирование на БЭСМ-6 в системе Дубна. М.: Наука, 1978. 272 с.
8. Баяковский Ю. М., Мишакова С. Т. Автоматизированная система подготовки публикаций и документов (АСПИД): Препр. ИПМ АН СССР № 19. М., 1977. 25 с.
9. Алферова М. И., Горбунов-Посадов М. М., Матеев М. П., Нефедова Р. А. Средства документирования в пакете Сафра: Препр. ИПМ АН СССР № 27. М., 1981. 13 с.

УДК 533 : 536

## ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РАСЧЕТА УРАВНЕНИЙ ОДНОМЕРНОЙ ГАЗОВОЙ ДИНАМИКИ С ТЕПЛОПРОВОДНОСТЬЮ

С. А. Гайфулин, А. В. Захаров, Н. В. Змитренко, В. Я. Карпов,  
А. П. Михайлов, Т. В. Мищенко, А. А. Самарский

### 1. Предварительные замечания

В работе описывается программа FLORA, предназначенная для расчета уравнений одномерной газовой динамики с теплопроводностью.

Программа основана на теоретических и расчетных работах по численному решению задач газовой динамики, уравнений диффузии, переноса и др. Этот опыт отражен в методиках и программах (см., например, [1—3], эффективно использовавшихся

при численном моделировании процессов в различных физических задачах.

В связи с совершенствованием организации работ в этой области и созданием пакета прикладных программ САФРА [4] возник вопрос о том, чтобы функциональное наполнение пакета соответствовало лучшим методикам, разработанным при решении вышеперечисленных задач.

В программе FLORA состояние среды описывается тремя основными независимыми переменными  $\rho$ ,  $u$ ,  $T$ , которые представляют собой плотность, скорость и температуру соответственно. Каждая переменная является функцией времени и одной пространственной координаты  $r$ . Эта координата является линейной координатой в плоской геометрии и радиусом в цилиндрической или сферической геометриях. Дифференциальные уравнения, описывающие изменения параметров среды, приведены в п. 3. Там же дан краткий вывод разностных уравнений и приведен метод их решения.

Использованная в программе разностная аппроксимация дифференциальных уравнений, а также методы решения разностных уравнений основаны на принципах, изложенных в [1—3].

Структура программы и общие правила ее написания базируются на системе OLYMPUS [5—7]. Передача информации между подпрограммами ведется, насколько это возможно, через блоки общей памяти. Все подпрограммы разделены на классы, а блоки общей памяти на группы, согласно классификации [7]. Каждой программной единице, присваивается номер, указывающий класс, которому она принадлежит, а также ее место в данном классе. Например, номер <1.5> указывает на пятую подпрограмму из первого класса (класса пролога). Аналогичное соглашение имеется и для общих блоков. Так блок <C1.2> есть второй COMMON блок из первой группы (общие параметры системы). В данной работе отчасти использовались соглашения, принятые в программе MEDUSA [8]. В п. 4 подробно описывается организация программы.

Программа FLORA является частью функционального наполнения пакета прикладных программ САФРА [4] и существенно использует предоставляемые пакетом возможности сборки.

При работе по составлению программы FLORA в обсуждениях участвовали П. П. Волосевич, Г. В. Данилова, Н. Н. Калиткин, С. П. Курдюмов, Ю. П. Попов, Л. С. Царева. Всем им авторы приносят искреннюю благодарность.

## 2. Основные сведения о программе

**Название программы FLORA.**

**ЭВМ, для которой предназначена программа БЭСМ-6 и машины серии ЕС.**

**Физическая постановка задачи.** Решаются уравнения одномерной газовой динамики с теплопроводностью в плоской, цилиндрической или сферической геометриях. Уравнения решаются в лагранжевых координатах. Предусмотрена возможность учета граничных условий различных типов.

**Метод решения.** Уравнения в частных производных решаются разностными методами. Используется полностью консервативная разностная схема.

**Ограничения сложности задачи.** Ограничения сложности связаны только с ограничением ресурсов вычислительной машины. Память машины БЭСМ-6 позволяет использовать сетку порядка 200 точек.

**Время счета.** При сетке 100 точек характерное время расчета одного временного слоя порядка 1 с.

**Особенности программы.** Программа является частью функционального наполнения пакета прикладных программ САФРА и существенно использует предоставленные пакетом услуги. Программа написана в рамках системы OLYMPUS. Она может быть легко модифицирована для учета специальных объемных источников энергии, уравнения состояния и т. д.

**Вспомогательные программы.** Программа использует подпрограммы системы OLYMPUS.

**Состояние программы.** Программа успешно эксплуатируется.

**Требования к вычислительной машине.** Программа рассчитана на ЭВМ, имеющую объем оперативной памяти и быстродействие порядка машины БЭСМ-6.

**Язык программирования.** Программа написана на языке Фортран-Дубна. Однако в программе нигде не использовались операторы указанной версии языка, которые не входят в языки Фортран-IV и Фортран-Стандарт.

**Операционная система.** Программа не предъявляет специальных требований к операционной системе.

### 3. Основные сведения о задаче

**3.1. Математическая постановка задачи.** В программе FLORA вещество в механическом отношении рассматривается как идеальная (невязкая) сжимаемая жидкость, состояние которой зависит от одной пространственной координаты. Эта координата совпадает с радиусом в цилиндрической и сферической геометриях и является линейной координатой в плоской геометрии. Тип геометрии выделяется признаком:  $g=1$  — плоская геометрия, область — прямоугольник единичного поперечного сечения;  $g=2$  — цилиндрическая геометрия, область — сектор цилиндра единичной высоты с углом в один радиан;  $g=3$  — сферическая геометрия, область — сектор шара с углом в один стерадиан.

Движение вещества описывается уравнением Эйлера, кото-

рое в массовых лагранжевых координатах  $(m, t)$  имеет вид

$$\frac{\partial u}{\partial t} = -r^{g-1} \frac{\partial p}{\partial m}. \quad (3.1)$$

Здесь  $u(m, t)$  — скорость вещества, определяющая изменение эйлеровой пространственной координаты  $r(m, t)$

$$dr/dt = u(m, t); \quad (3.2)$$

$\rho(m, t)$  — плотность вещества;  $p(m, t)$  — гидродинамическое давление.

Массовая координата  $m$  связана с эйлеровой пространственной координатой  $r$  соотношением

$$dm = r^{g-1} \rho dr. \quad (3.3)$$

Предположим, что значения массовой координаты лежат в неизменяемых со временем пределах  $0 \leq m \leq M$ . Пределы изменения пространственной координаты  $r$  зависят от времени, а именно  $r_{\text{лев}}(t) = r(0, t) \leq r \leq r(M, t) = r_{\text{прав}}(t)$ .

На границах области  $m=0$ ,  $r=r_{\text{лев}}$  и  $m=M$ ,  $r=r_{\text{прав}}$  для уравнения (3.1) будем рассматривать условия двух типов: а) задана граничная скорость  $u(0, t) = u_{\text{лев}}(t)$ ,  $u(M, t) = u_{\text{прав}}(t)$ ; б) задано граничное давление  $p(0, t) = p_{\text{лев}}(t)$ ,  $p(M, t) = p_{\text{прав}}(t)$ , причем типы граничных условий слева и справа независимы.

В начальный момент времени для уравнений (3.1), (3.2) должно быть задано распределение пространственной координаты  $r(m, 0)$  (или, что эквивалентно вследствие (3.3), распределение  $\rho(m, 0)$  или  $\rho(r, 0)$ ) и начальной скорости  $u(m, 0)$ .

Гидродинамическое давление  $p$ , входящее в уравнение (3.1), определяется плотностью вещества и его температурой. Распределение в пространстве и изменение со временем температуры в рассматриваемой области описываются уравнением энергии

$$\frac{\partial E}{\partial t} = -p \frac{\partial}{\partial t} \left( \frac{1}{\rho} \right) - \frac{\partial W}{\partial m} + G. \quad (3.4)$$

Здесь  $E(m, t)$  внутренняя энергия вещества, отнесенная к единице массы;  $W(m, t)$  — поток тепла через поперечное сечение области;  $G(m, t)$  — массовая плотность дополнительных источников энергии.

Поток тепла  $W$  связан с градиентом температуры

$$W = -\kappa \rho r^{2(g-1)} \frac{\partial T}{\partial m}, \quad (3.5)$$

где  $\kappa$  — коэффициент теплопроводности.

Для уравнения энергии (3.4) будем рассматривать граничные условия двух типов: а) задана граничная температура  $T(0, t) = T_{\text{лев}}(t)$ ,  $T(M, t) = T_{\text{прав}}(t)$ ; б) задан поток тепла через границу  $W(0, t) = W_{\text{лев}}(t)$ ,  $W(M, t) = W_{\text{прав}}(t)$ , причем типы граничных условий слева и справа независимы.

В начальный момент времени  $t=0$  в области должно быть задано распределение температуры  $T(m, 0)$ .

Уравнения (3.1)—(3.5) совместно с уравнением состояния и выражением для коэффициента теплопроводности образуют замкнутую систему.

**Уравнение состояния.** В стандартной версии программы предполагается, что среда представляет собой идеальный газ, давление и внутренняя энергия которого вычисляются через плотность и температуру по формулам

$$p = R\rho T, \quad E = \frac{R}{\gamma - 1} T, \quad (3.7)$$

где  $R$  и  $\gamma$  — задаваемые пользователем постоянные.

**Коэффициент теплопроводности.** В стандартной версии программы предполагается, что коэффициент теплопроводности  $\kappa$  связан с температурой и плотностью соотношением вида

$$\kappa = \kappa_0 T^a \rho^b, \quad (3.8)$$

где  $\kappa_0$ ,  $a$ ,  $b$  — задаваемые пользователем постоянные.

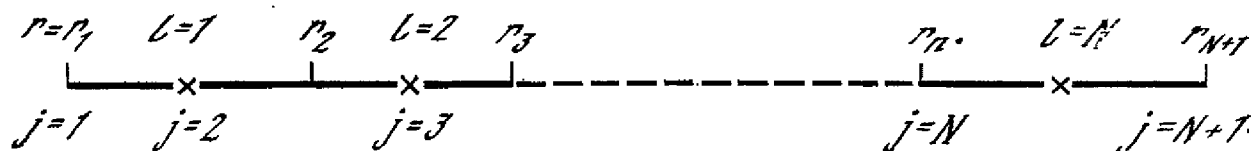
**Дополнительные источники энергии.** В стандартной версии программы предполагается, что дополнительные источники энергии имеют массовую плотность вида

$$G = G_0 T^c \rho^d, \quad (3.9)$$

где  $G_0$ ,  $c$ ,  $d$  — задаваемые пользователем постоянные. Кроме этого, в программу можно включить источники произвольного вида.

**3.2. Система разностных уравнений.** Дифференциальные уравнения энергии и движения совместно с уравнением состояния полностью описывают математическую модель задачи. Для численного решения этих уравнений воспользуемся разностными методами и введем сетки как по времени, так и по пространству. При этом процедура решения уравнений сводится к последовательному нахождению решения на каждом временном слое.

Номер временного слоя будем отмечать верхним индексом  $n$ , так что  $t^n$  — значение времени на  $n$ -м слое. Шаг по времени будем обозначать  $\Delta t^n = t^{n+1} - t^n$  или кратко  $\Delta t$ .



Область изменения координаты  $r(m)$ , простирающаяся от точки  $r=r_{\text{лев}} (m=0)$  до точки  $r=r_{\text{прав}} (m=M)$ , делится на  $N$  интервалов (ячеек), как показано на рисунке. Центры ячеек отмечены на рис. 1 крестиком. Они нумеруются индексом  $l$ , пробегающим значения от 1 до  $N$ . Границы ячеек нумеруются индексом  $j$ , который изменяется от 1 до  $N+1$ .

Давление  $p$ , плотность  $\rho$ , температура  $T$ , энергия  $E$ , источники энергии  $G$  считаются определенными в центрах ячеек и отмечаются индексом  $l$ . Скорость  $u$ , пространственная координата  $r$ , поток тепла  $W$  относятся к границам ячеек и отмечаются индексом  $j$ .

Каждая ячейка содержит не изменяющуюся со временем массу вещества  $m_l$ , так что плотность вещества в ячейке  $l$  равна

$$\rho_l = \frac{gm_l}{r_{j+1}^g - r_j^g} \quad (1 \leq j \leq N, 1 \leq l \leq N \quad (l = j)). \quad (3.10)$$

Уравнение Эйлера (3.1) во внутренних узлах сетки аппроксимируем разностным уравнением следующим образом:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -R_j \frac{2}{m_l + m_{l-1}} \{p_l^{n+1} + q_l^{n+1} - p_{l-1}^{n+1} - q_{l-1}^{n+1}\}. \quad (3.11)$$

Здесь  $R_j$  — коэффициент, являющийся разностной аппроксимацией коэффициента  $r^{g-1}$  в (3.1):

$$R_j = \begin{cases} 1 & \text{при } g = 1; \\ (r_j^{n+1} + r_j^n)/2 & \text{при } g = 2; \\ \{(r_j^{n+1})^2 + r_j^{n+1}r_j^n + (r_j^n)^2\}/3 & \text{при } g = 3, \end{cases} \quad (3.12)$$

а  $q_l$  — давление искусственной вязкости, вводимое в разностную схему для сквозного счета ударных волн. Будем учитывать его в виде комбинации линейной и квадратичной вязкостей.

При  $\rho_l^n \geq \rho_l^{n+1}$  и при  $u_{j+1}^{n+1} \geq u_j^{n+1}$  вязкость равна нулю, а в противном случае вычисляется по формуле

$$q_l^{n+1} = \rho_l^{n+1} \{\mu_1 a_l \Delta u_l + \mu_2 (\Delta u_l)^2\}. \quad (3.13)$$

Здесь  $\mu_1$  и  $\mu_2$  — коэффициенты линейной и квадратичной вязкостей, которые могут быть заданы пользователем;  $a_l = (\rho_l^{n+1}/\rho_l^{n+1})^{1/2}$  — оценка скорости звука;

$$\Delta u_l = \frac{m_l}{r_l^{g-1} \Delta t} \left( \frac{1}{\rho_l^{n+1}} - \frac{1}{\rho_l^n} \right), \quad (r_l = 0,5 (r_j^n + r_{j+1}^n)) -$$

аппроксимация разности скоростей границ ячейки.

В граничных точках  $j=1, N+1$  в случае заданной скорости уравнение движения (3.11) решать не нужно, поскольку

$$u_1^{n+1} = u(0, t^{n+1}); \quad u_{N+1}^{n+1} = u(M, t^{n+1}),$$

а в случае заданного давления полагаем

$$\frac{u_1^{n+1} - u_1^n}{\Delta t} = -R_1 \frac{2}{m_1} \{p_1^{n+1} + q_1^{n+1} - p_{\text{лев}}(t^{n+1})\};$$

$$\frac{u_{N+1}^{n+1} - u_{N+1}^{n+1}}{\Delta t} = -R_{N+1} \frac{2}{m_N} \{p_{\text{прав}}(t^{n+1}) - p_N^{n+1} - q_N^{n+1}\}. \quad (3.14)$$

Уравнение траекторий (3.2) аппроксимируем соотношением

$$\frac{r_j^{n+1} - r_j^n}{\Delta t} = \frac{1}{2} \{u_j^{n+1} + u_j^n\}, \quad (1 \leq j \leq N+1). \quad (3.15)$$

Для решения системы разностных уравнений (3.10—(3.15) на нулевом слое при  $n=0$  должны быть определены: массы ячеек  $m_l$ ,  $1 \leq l \leq N$ ; границы ячеек  $r_j$ ,  $1 \leq j \leq N+1$ ; скорости границ  $u_j$ ,  $1 \leq j \leq N+1$ ; плотности в ячейках  $\rho_l$ ,  $1 \leq l \leq N$ . Величины  $m_l$ ,  $r_j$  и  $\rho_l$  не независимы, а связаны соотношением (3.10).

Разностную аппроксимацию уравнения энергии (3.4) запишем в виде

$$E_l^{n+1} - E_l^n = - (p_l^{n+1} + q_l^{n+1}) \left( \frac{1}{\rho_l^{n+1}} - \frac{1}{\rho_l^n} \right) - \frac{\Delta t}{m_l} (W_{j+1}^{n+1} - W_j^{n+1}) + \Delta t G_l^{n+1} \quad (1 \leq l \leq N). \quad (3.16)$$

Связь (3.5) потока тепла с градиентом температуры аппроксимируем выражением

$$W_j = -2(\rho\kappa)_j r_j^{2(g-1)} \frac{T_l^{n+1} - T_{l-1}^{n+1}}{m_l + m_{l-1}} \quad (j = 2, 3, \dots, N, l = j). \quad (3.17)$$

Значения коэффициентов  $(\rho\kappa)_j$  на границах ячеек вычисляются следующим образом. Введем температуру границы ячейки

$$T_j = 0,5(T_{l-1} + T_l), \quad (2 \leq j \leq N, l = j)$$

и коэффициенты  $\rho\kappa$ , вычисленные по значениям плотностей в ячейках  $l-1$  и  $l$ , но по граничной температуре  $T_j$

$$\rho_l \kappa_l^+ = \rho_l \kappa_l(\rho_l, T_{j+1}) \quad \text{и} \quad \rho_l \kappa_l^- = \rho_l \kappa_l(\rho_l, T_j).$$

Тогда

$$(\rho\kappa)_j = \frac{\rho_l \rho_{l-1} \kappa_{l-1}^+ \kappa_l^- (m_l + m_{l-1})}{m_l \rho_{l-1} \kappa_{l-1}^+ + m_{l-1} \rho_l \kappa_l^-}.$$

В граничных точках  $j=1, N+1$  потоки тепла либо заданы:

$$W_1^{n+1} = W_{\text{лев}}(t^{n+1}); \quad W_{N+1}^{n+1} = W_{\text{прав}}(t^{n+1}),$$

либо в случае заданных температур  $T_{\text{лев}}(t)$ ,  $T_{\text{прав}}(t)$  на границах вычисляются по формулам

$$W_1^{n+1} = -2(\rho\kappa)_1 r_1^{2(g-1)} \frac{T_1^{n+1} - T_{\text{лев}}(t^{n+1})}{m_1},$$

$$W_{N+1}^{n+1} = -2(\rho\kappa)_{N+1} r_{N+1}^{2(g-1)} \frac{T_{\text{прав}}(t^{n+1}) - T_N^{n+1}}{m_N}, \quad (3.18)$$

где  $\rho x$  вычисляются по плотности первой (последней) ячейки, но по граничной температуре. Соотношения (3.18) представляют собой соотношение (3.17), записанное на полуинтервалах ( $j=1, l=1$ ) и ( $l=N, j=N+1$ ).

Если ввести обозначения

$$a_j = \frac{2(\rho x)_j r_l^{2(g-1)}}{m_l + m_{l+1}}, \quad (j = 2, 3, \dots, N);$$

$$a_1 = \frac{2(\rho x)_1 r_1^{2(g-1)}}{m_1}; \quad a_{N+1} = \frac{2(\rho x)_{N+1} r_{N+1}^{2(g-1)}}{m_N}, \quad (3.19)$$

где все величины вычисляются на  $(n+1)$ -м временном слое, то во внутренних ячейках уравнение энергии запишется в виде

$$\frac{\Delta t}{m_l} \{a_{j+1} T_{l+1}^{n+1} - (a_j + a_{j+1}) T_l^{n+1} + a_j T_{l-1}^{n+1}\} -$$

$$- E_l^{n+1} + \Delta A_l^{n+1} + E_l^n + \Delta t G_l^{n+1} = 0, \quad (l = 2, 3, \dots, N-1), \quad (3.20)$$

$$\Delta A_l^{n+1} = - (p_l^{n+1} + q_l^{n+1}) \left( \frac{1}{\rho_l^{n+1}} - \frac{1}{\rho_l^n} \right); \quad (3.21)$$

последнее выражение представляет собой работу сил давления. На левом конце при  $l=1$  в случае заданной граничной температуры уравнение энергии имеет вид

$$\frac{\Delta t}{m_1} \{a_2 T_2^{n+1} - (a_1 + a_2) T_1^{n+1} + a_1 T_{\text{лев}}(t^{n+1})\} -$$

$$- E_1^{n+1} + E_1^n + \Delta A_1^{n+1} + \Delta t G_1^{n+1} = 0, \quad (3.22)$$

а в случае заданного потока тепла

$$\frac{\Delta t}{m_1} \{a_2 (T_2^{n+1} - T_1^{n+1}) + W_{\text{лев}}(t^{n+1})\} -$$

$$- E_1^{n+1} + E_1^n + \Delta A_1^{n+1} + \Delta t G_1^{n+1} = 0. \quad (3.23)$$

Аналогичные выражения имеют место на правом конце при  $l=N$ .

Для решения системы разностных уравнений энергии в начальный момент времени должна быть задана температура  $T_l^0, 1 \leq l \leq N$ .

**Сохранение полной энергии.** Можно показать [1], что из разностных уравнений движения и энергии вытекает закон сохранения полной энергии. Введем уравнение кинетической энергии ячейки в  $n$ -м слое

$$(E_{\text{kin}})_l^n = \frac{1}{2} m_l \frac{(u_{j+1}^n)^2 + (u_j^n)^2}{2} \quad (3.24)$$



и внутренней энергии ячейки в  $n$ -м слое

$$(E_{\text{int}}^n = m_l E_l^n. \quad (3.25)$$

Из разностных уравнений движения и энергии вытекает соотношение

$$\begin{aligned} (E_{\text{kin}} + E_{\text{int}})_{l}^{n+1} - (E_{\text{kin}} + E_{\text{int}})_{l}^n = \\ = \Delta t \left\{ -R_{j+1} \frac{1}{2} (u_{j+1}^{n+1} + u_{j+1}^n) P_{j+1}^{n+1} + R_j \frac{1}{2} (u_j^{n+1} + u_j^n) P_j^{n+1} \right\} - \\ - \Delta t \{W_{j+1}^{n+1} - W_j^{n+1}\} + \Delta t m_l G_l^{n+1}, \end{aligned}$$

где

$$P_j^{n+1} = \frac{m_{l-1} (p_l^{n+1} + q_l^{n+1}) + m_l (p_{l-1}^{n+1} + q_{l-1}^{n+1})}{m_{l-1} + m_l}.$$

Введем уравнения кинетической энергии всей системы на  $n$ -м слое

$$E_{\text{kin}}^n = \sum_{l=1}^N (E_{\text{kin}})_l^n, \quad (3.26)$$

внутренней энергии всей системы на  $n$ -м слое,

$$E_{\text{int}}^n = \sum_{l=1}^N (E_{\text{int}})_l^n, \quad (3.27)$$

полной энергии всей системы на  $n$ -м слое

$$E_{\text{tot}}^n = E_{\text{kin}}^n + E_{\text{int}}^n,$$

работы над системой внешних сил при переходе от  $n$ -го временного слоя к  $(n+1)$ -му

$$\begin{aligned} \Delta A = \rho_{\text{лев}}(t^{n+1}) \frac{1}{g} \{(rg)_1^{n+1} - (rg)_1^n\} - \\ - \rho_{\text{прав}}(t^{n+1}) \frac{1}{g} \{(rg)_{N+1}^{n+1} - (rg)_{N+1}^n\} = \\ = \rho_{\text{лев}}(t^{n+1}) R_1 \frac{\Delta t}{2} (u_1^{n+1} + u_1^n) - \\ - \rho_{\text{прав}}(t^{n+1}) R_{n+1} \frac{\Delta t}{2} (u_{N+1}^{n+1} + u_{N+1}^n). \end{aligned} \quad (3.28)$$

Тогда в результате тождественных преобразований из уравнений (3.10) — (3.16) следует соотношение

$$\begin{aligned} E_{\text{tot}}^{n+1} - E_{\text{tot}}^n = \Delta A + \Delta t \{W_{\text{лев}}(t^{n+1}) - \\ - W_{\text{прав}}(t^{n+1})\} + \Delta t \sum_{l=1}^N m_l G_l^{n+1}, \end{aligned} \quad (3.29)$$

т. е. изменение полной энергии системы на одном временном ша-

ге связано с работой внешних сил, притоком тепла через внешние границы и с объемными источниками энергии.

**3.3. Решение разностных уравнений.** Разностные уравнения движения и энергии совместно с уравнением состояния представляют собой систему нелинейных алгебраических уравнений, для решения которых используется следующий итерационный метод: а) задаем приближенные значения скорости и температуры на  $(n+1)$ -м временном слое (подпрограмма <2.3> STIT, описание см. в п. 4); б) при фиксированной температуре решаем уравнение движения и находим следующее приближенное значение скорости на  $(n+1)$ -м слое (подпрограмма <2.2> MOTION, описание см. в п. 4); в) при фиксированной скорости решаем уравнение энергии и находим новое приближенное значение температуры на  $(n+1)$ -м слое (подпрограмма <2.20> ENERGY, описание см. в п. 4); г) проверяем выполнение закона сохранения энергии (подпрограмма <2.32> CONSRV). Если он выполнен с требуемой точностью, то считаем, что значения функций на момент времени  $t^{n+1}$  найдены. В противном случае возвращаемся к выполнению п. б) и снова повторяем описанный процесс.

Этот алгоритм решения уравнений зафиксирован в схеме счета программы, и в частности в подпрограмме управления расчетом временного шага <2.1> STEPON.

**3.4. Решение уравнений движения.** Система нелинейных уравнений (3.10) — (3.15) решается с помощью итераций. Отметим, что использованный ниже метод решения несколько отличается от метода Ньютона, описанного в [1].

Обозначая верхним индексом номер итерации, запишем для каждой функции в этих уравнениях формулу перехода от  $s$ -й к  $(s+1)$ -й итерации:

$$f^{s+1} = f^s + \delta f^s,$$

где  $\delta f$  — изменение функции. Частично опуская индексы, будем писать.

$$r_j^{s+1} = r_j^{n+1} + \delta r_j = r_j^{n+1} + \frac{\partial r_j}{\partial u_j} \delta u_j;$$

$$u_j^{s+1} = u_j^{n+1} + \delta u_j;$$

$$p_l^{s+1} = p_l^{n+1} + \delta p_l = p_l^{n+1} + \frac{\partial p_l}{\partial u_j} \delta u_j + \frac{\partial p_l}{\partial u_{j+1}} \delta u_{j+1};$$

$$q_l^{s+1} = q_l^{n+1} + \delta q_l = q_l^{n+1} + \frac{\partial q_l}{\partial u_j} \delta u_j + \frac{\partial q_l}{\partial u_{j+1}} \delta u_{j+1};$$

$$R_j^{s+1} = R_j + \delta R_j = R_j + \frac{\partial R_j}{\partial u_j} \delta u_j.$$

Здесь  $r_j^{n+1}$ ,  $u_j^{n+1}$  и т. д. обозначают значения функций на  $s$ -й ите-

рации. Подставляя в соотношение (3.15) значения функций на  $(s+1)$ -й итерации и считая, что на  $s$ -й итерации оно удовлетворяется, получим

$$\delta r_j = \frac{\Delta t}{2} \delta u_j. \quad (3.30)$$

Подставляя значения функций на  $(s+1)$ -й итерации в уравнение (3.11), пренебрегая квадратами изменений функций и вводя обозначения

$$\begin{aligned} F_j &= -\frac{m_{l-1} + m_l}{2\Delta t} \{u_j^{n+1} - u_j^n\} - R_j \{p_l^{n+1} + q_l^{n+1} - p_{l-1}^{n+1} - q_{l-1}^{n+1}\}; \\ C_j &= \frac{m_l + m_{l-1}}{2\Delta t} + \frac{\partial R_j}{\partial u_j} \{p_l^{n+1} + q_l^{n+1} - p_{l-1}^{n+1} - q_{l-1}^{n+1}\} + \\ &+ R_j \left\{ \frac{\partial p_l}{\partial u_j} + \frac{\partial q_l}{\partial u_j} - \frac{\partial p_{l-1}}{\partial u_j} - \frac{\partial q_{l-1}}{\partial u_j} \right\}; \\ A_j &= R_j \left\{ \frac{\partial p_{l-1}}{\partial u_{j-1}} + \frac{\partial q_{l-1}}{\partial u_{j-1}} \right\}, \\ B_j &= -R_j \left\{ \frac{\partial p_l}{\partial u_{j+1}} + \frac{\partial q_l}{\partial u_{j+1}} \right\}, \end{aligned} \quad (3.31)$$

получим, что уравнение для изменения скорости при переходе от  $s$ -й к  $(s+1)$ -й итерации имеет вид

$$A_j \delta u_{j-1} - C_j \delta u_j + B_j \delta u_{j+1} = -F_j, \quad (j = 2, 3, \dots, N). \quad (3.32)$$

Разностное уравнение (3.32) можно записать и в граничных точках  $j=1$  и  $j=N+1$ , положив  $A_1 = B_{N+1} = 0$ . В случае, если задана граничная скорость на левой (правой) границе, должно выполняться условие  $\delta u_1 = 0$  ( $\delta u_{N+1} = 0$ ). Поэтому полагаем  $B_1 = 0$ ,  $C_1 = 1$ ,  $F_1 = 0$  ( $A_{N+1} = 0$ ,  $C_{N+1} = 1$ ,  $F_{N+1} = 0$ ).

При заданном давлении в результате выполнения описанных выше преобразований применительно к уравнениям (3.14) получим на левой границе

$$\begin{aligned} F_1 &= -\frac{m_1}{2\Delta t} \{u_1^{n+1} - u_1^n\} - R_1 \{p_1^{n+1} + q_1^{n+1} - p_{\text{лев}}(t^{n+1})\}; \\ C_1 &= \frac{m_1}{2\Delta t} + \frac{\partial R_1}{\partial u_1} \{p_1^{n+1} + q_1^{n+1} - p_{\text{лев}}(t^{n+1})\} + R_1 \left\{ \frac{\partial p_1}{\partial u_1} + \frac{\partial q_1}{\partial u_1} \right\}; \\ B_1 &= -R_1 \left\{ \frac{\partial p_1}{\partial u_2} + \frac{\partial q_1}{\partial u_2} \right\}. \end{aligned} \quad (3.33)$$

Аналогичные формулы получаем и для правой границы:

$$F_{N+1} = -\frac{m_N}{2\Delta t} \{u_{N+1}^{n+1} - u_{N+1}^n\} - R_{N+1} \{p_{\text{прав}}(t^{n+1}) - p_N^{n+1} - q_N^{n+1}\};$$

$$\begin{aligned}
C_{N+1} &= \frac{m_N}{2\Delta t} + \frac{\partial R_{N+1}}{\partial u_{N+1}} \{p_{\text{прав}}(t^{n+1}) - p_N^{n+1} - q_N^{n+1}\} - \\
&- R_{N+1} \left\{ \frac{\partial p_N}{\partial u_{N+1}} + \frac{\partial q_N}{\partial u_{N+1}} \right\}; \\
A_{N+1} &= R_{N+1} \left\{ \frac{\partial p_N}{\partial u_N} + \frac{\partial q_N}{\partial u_N} \right\}.
\end{aligned} \tag{3.34}$$

Система разностных уравнений (3.32) решается методом прогонки. Двигаясь слева направо вычисляем значения прогоночных коэффициентов:

$$\begin{aligned}
\alpha_1 &= B_1/C_1; \quad \beta_1 = E_1/C_1; \quad \alpha_j = B_j/(C_j - A_j\alpha_{j-1}); \\
\beta_j &= (F_j - A_j\beta_{j-1})/(C_j - A_j\alpha_{j-1}), \quad (j = 2, 3, \dots, N+1).
\end{aligned}$$

По значениям прогоночных коэффициентов обратной прогонкой вычисляем значения функции:

$$y_{N+1} = \beta_{N+1}; \quad y_j = \alpha_j y_{j+1} + \beta_j, \quad (j = N, N-1, \dots, 1, (y_j = \delta u_j)).$$

Для получения коэффициентов разностного уравнения нужно вычислить значения производных по скорости.

Производную давления по скорости можно свести к производной давления по плотности, которая определяется из уравнения состояния:

$$\frac{\partial p_l}{\partial u_j} = -\rho_l^2 \left( \frac{\partial p}{\partial \rho} \right)_l \frac{\partial}{\partial u_j} \left( \frac{1}{\rho_l} \right).$$

Воспользовавшись соотношениями (3.10) и (3.15), получим

$$\frac{\partial p_l}{\partial u_j} = \rho_l^2 \left( \frac{\partial p}{\partial \rho} \right)_l \frac{r_j^{g-1}}{m_l} \frac{\Delta t}{2}.$$

Аналогично

$$\frac{\partial p_l}{\partial u_{j+1}} = -\rho_l^2 \left( \frac{\partial p}{\partial \rho} \right)_l \frac{r_{j+1}^{g-1}}{m_l} \frac{\Delta t}{2}.$$

Производную давления искусственной вязкости по скорости точно так же можно свести к производной по удельному объему  $V_l = 1/\rho_l$ , которая определяется видом искусственной вязкости:

$$\frac{\partial q_l}{\partial u_j} = - \left( \frac{\partial q}{\partial V} \right)_l \frac{r_j^{g-1}}{m_l} \frac{\Delta t}{2}, \quad \frac{\partial q_l}{\partial u_{j+1}} = \left( \frac{\partial q}{\partial V} \right)_l \frac{r_{j+1}^{g-1}}{m_l} \frac{\Delta t}{2}.$$

Производная  $\partial q/\partial V$  равна нулю, если вязкость полагается равной нулю, а в противном случае согласно (3.13)

$$\left( \frac{\partial q}{\partial V} \right)_l = -\rho_l^{n+1} q_l^{n+1} + \rho_l^{n+1} \{ \mu_1 a_l + 2\mu_2 \Delta \bar{u} \} \frac{m_l}{r_l^{g-1} \Delta t}.$$

Производная для коэффициента  $R_j$  получается из формул (3.12) и (3.15):

$$\frac{\partial R_j}{\partial u_j} = \begin{cases} 0 & \text{при } g = 1; \\ \Delta t/4 & \text{при } g = 2; \\ (\Delta t/6)(2r_j^{n+1} + r_j^n) & \text{при } g = 3. \end{cases}$$

Уравнения считаются решенными, если приращения скорости на итерации удовлетворяют условиям

$$|\delta u_j| < \varepsilon_M |u_j| + u_{\min}, \quad (j = 1, 2, \dots, N + 1),$$

где  $\varepsilon_M$  и  $u_{\min}$  — постоянные, которые могут быть заданы пользователем.

**3.5. Решение уравнений энергии.** Система уравнений энергии (3.20), (3.22), (3.23) решается итерациями. Проведя преобразования, аналогичные описанным выше для уравнений движения, и вводя обозначения

$$\begin{aligned} F_l &= \frac{\Delta t}{m_l} \{a_{j+1} T_{l+1}^{n+1} - (a_j + a_{j+1}) T_l^{n+1} + a_j T_{l-1}^{n+1}\} - \\ &\quad - E_l^{n+1} + E_l^n + \Delta A_l^{n+1} + \Delta t G_l^{n+1}; \\ C_l &= \left\{ a_j + a_{j+1} - \frac{\partial a_{j+1}}{\partial T_l} (T_{l+1}^{n+1} - T_l^{n+1}) + \frac{\partial a_j}{\partial T_l} (T_l^{n+1} - T_{l-1}^{n+1}) \right\} \frac{\Delta t}{m_l} + \\ &\quad + \left( \frac{\partial E}{\partial T} \right)_l - \left( \frac{\partial \Delta A}{\partial T} \right)_l - \left( \frac{\partial G}{\partial T} \right)_l \Delta t; \end{aligned} \quad (3.35)$$

$$A_l = \frac{\Delta t}{m_l} \left\{ a_j - \frac{\partial a_j}{\partial T_{l-1}} (T_l^{n+1} - T_{l-1}^{n+1}) \right\};$$

$$B_l = \frac{\Delta t}{m_l} \left\{ a_{j+1} + \frac{\partial a_{j+1}}{\partial T_{l+1}} (T_{l+1}^{n+1} - T_l^{n+1}) \right\},$$

получим следующее разностное уравнение для изменения температуры ячеек  $\delta T_l$  при переходе от  $s$ -й к  $s+1$ -й итерации:

$$A_l \delta T_{l-1} - C_l \delta T_l + B_l \delta T_{l+1} = -F_l, \quad (l = 2, 3, \dots, N - 1). \quad (3.36)$$

При  $l=1$  полагаем  $A_1=0$ . При заданном потоке тепла из уравнения (3.23) получаем:

$$\begin{aligned} F_1 &= \frac{\Delta t}{m_1} \{W_1 + a_2 (T_2^{n+1} - T_1^{n+1})\} - E_1^{n+1} + E_1^n + \Delta A_1^{n+1} + \Delta t G_1^{n+1}; \\ B_1 &= \frac{\Delta t}{m_1} \left\{ a_2 + \frac{\partial a_2}{\partial T_2} (T_2^{n+1} - T_1^{n+1}) \right\}; \end{aligned} \quad (3.37)$$

$$\begin{aligned} C_1 &= \frac{\Delta t}{m_1} \left\{ a_2 - \frac{\partial a_2}{\partial T_1} (T_2^{n+1} - T_1^{n+1}) \right\} - \\ &\quad - \left( \frac{\partial \Delta A}{\partial T} \right)_1 - \left( \frac{\partial G}{\partial T} \right)_1 \Delta t + \left( \frac{\partial E}{\partial T} \right)_1; \quad W_1 = W_{\text{лев}}(t^{n+1}). \end{aligned}$$

При заданной температуре и  $l=1$  для коэффициентов  $B_1, C_1, F_1$  можно применить формулы (3.35), если в них принять  $T_0 = T_{\text{лев}}(t^{n+1})$ . При  $l=N$  полагаем  $B_N=0$ . При заданном потоке тепла

$$\begin{aligned}
 F_N &= \frac{\Delta t}{m_N} \{ -a_N (T_N^{n+1} - T_{N-1}^{n+1}) - W_{N+1} \} - E_N^{n+1} + E_N^n + \\
 &\quad + \Delta A_N^{n+1} + \Delta t G_N^{n+1}, \\
 A_N &= \frac{\Delta t}{m_N} \left\{ -\frac{\partial a_N}{\partial T_{N-1}} (T_N^{n+1} - T_{N-1}^{n+1}) + a_N \right\}, \\
 C_N &= \frac{\Delta t}{m_N} \left\{ \frac{\partial a_N}{\partial T_N} (T_N^{n+1} - T_{N-1}^{n+1}) + a_N \right\} - \\
 &\quad - \left( \frac{\partial \Delta A}{\partial T} \right)_N - \left( \frac{\partial G}{\partial T} \right)_N \Delta t + \left( \frac{\partial E}{\partial T} \right)_N; \quad W_{N+1} = W_{\text{прав}}(t^{n+1}).
 \end{aligned} \tag{3.38}$$

В случае заданной температуры для коэффициентов  $A_N, C_N, F_N$  будут применимы формулы (3.35), если в них принять  $T_{N+1} = T_{\text{прав}}(t^{n+1})$ .

Система разностных уравнений (3.36) решается методом прогонки.

Для вычисления коэффициентов разностного уравнения нужно вычислить значения производных по температуре. Производная внутренней энергии по температуре определяется уравнением состояния (3.7). Производная работы по температуре согласно (3.21) выражается через производную давления.

$$\left( \frac{\partial \Delta A}{\partial T} \right)_l = - \left( \frac{\partial p}{\partial T} \right)_l \left\{ \frac{1}{\rho_l^{n+1}} - \frac{1}{\rho_l^n} \right\},$$

которая также определяется уравнением состояния. Производные коэффициентов  $a_l$  по температуре очевидным образом выражаются через производные коэффициента теплопроводности.

Уравнения энергии считаются решенными, если приращения температуры на итерации удовлетворяют условиям:

$$|\delta T_l| < \varepsilon_E |T_l| + T_{\min}, \quad (l=1, 2, \dots, N),$$

где  $\varepsilon_E$  и  $T_{\min}$  — постоянные.

**3.6. Случай отсутствия теплопроводности.** Система разностных уравнений (3.36) упрощается в случае отсутствия теплопроводности. При этом  $A_l=B_l=0$ ,  $F_l = -E_l^{n+1} + E_l^n + \Delta A_l^{n+1} + G_l^{n+1} \Delta t$ , а изменение температуры при переходе от  $s$ -й к  $(s+1)$ -й итерации равно  $\delta T_l = F_l / C_l$ , так что трехточечное разностное уравнение (3.36) решать не нужно.

#### 4. Краткое описание программы

Программа FLORA написана на языке Фортран-Дубна [9—10] и предназначена для машины БЭСМ-6. Однако в программе нигде не использовались операторы указанной версии языка, которые не входят в языки Фортран-IV и Фортран-Стандарт.

Программа FLORA написана в рамках системы OLYMPUS [5—7] и существенно использует предоставляемые системой возможности. В программе нет основной программы и подпрограмм класса управления расчетом, в качестве которых используются соответствующие подпрограммы пакета OLYMPUS. Все переменные, необходимые для работы программы, хранятся в оперативной памяти машины и группируются в блоки общей памяти (см. табл. 1). Вывод информации на ленту ведется только для записи результатов расчета.

Каждой программной единице присваивается номер, указывающий класс, которому она принадлежит, а также ее место в данном классе. Например, запись <1.5> указывает на пятую подпрограмму из первого класса (класса пролога). Аналогичное соглашение имеется и для общих блоков. Так, блок <C1.2> есть второй COMMON-блок из первой группы (общие параметры системы).

**Класс 0. Управление расчетом.** Этот класс включает основную программную единицу MAIN и четыре подпрограммы: BASIC, MODIFY, CONTROL, EXPERT. Назначение этих подпрограмм определено системой OLYMPUS. В подпрограмме MODIFY реализован ввод основных переменных системы OLYMPUS.

Таблица 1

Список блоков общей памяти программы FLORA

Имя блока	Номер	Архивное имя	Назначение
Группа 2. Физические переменные			
COMHYD	2.1	FLOC21	Гидродинамические величины
COMTH	2.2	FLOC22	Термодинамические величины
COMEN	2.6	FLOC26	Энергии
COMCON	2.7	FLOC27	Параметры физического управления
Группа 3. Параметры расчета			
COMNC	3.1	FLOC31	Параметры численного управления
COMNUM	3.2	FLOC32	Сетка и численные методы
Группа 4. Управление			
COMADM	4.1	FLOC41	Административные переменные
Группа 5. Ввод-вывод			
COMOUT	5.1	FLOC51	Параметры управления вводом — выводом

**Класс 1. Пролог.** В этот класс входят подпрограммы, имена которых перечислены в табл. 2. Функциональные имена подпрограмм и их функциональное назначение совпадают с именами и назначениями, описанными в OLYMPUS.

**Класс 2. Вычисления.** В этот класс входят подпрограммы, перечисленные в табл. 2.

**<2.1> STEPON** управляет расчетом значений функций на  $(n+1)$ -м временном слое по заданным значениям на  $n$ -м слое. Как уже было описано в разд. 3, система уравнений движения и энергии решается итерациями. Счетчиком числа итераций этих двух уравнений, итераций внешнего цикла, служит переменная NIT.

1. Вызовом **<2.3> STIT** задаем приближенное значение скорости и температуры на  $(n+1)$ -м слое, устанавливаем нулем значение счетчика итераций.

2. Увеличиваем на единицу значение счетчика итераций.

3. Вызовом **<2.2> MOTION** решаем уравнение движения при фиксированной температуре. Подпрограмма MOTION вырабатывает значение логического ключа BREAK (прерывание). Если его значение — истина, то вызовом **<2.10> TIMSTP** пытаемся изменить значение шага по времени и возвращаемся на начало счета шага. Если значение ключа — ложь, то продолжаем расчет.

4. Вызовом **<2.20> ENERGY** решаем уравнение энергии при фиксированных гидродинамических величинах. Подпрограмма ENERGY вырабатывает значение логического ключа BREAK, обработка которого идет аналогично тому, как это описано в предыдущем пункте.

5. Вызовом **<2.9> CVERGE** проверяем выполнение закона сохранения энергии, т. е. сходимость итераций внешнего цикла. Подпрограмма вырабатывает значения двух логических ключей: NLGOON и BREAK. Если значение NLGOON — истина, то функции на  $(n+1)$ -м слое вычислены и можно оформлять переход к новому слою. Если значение NLGOON — ложь, то функции не найдены. В этом случае, если значение ключа BREAK — ложь, то имеет смысл продолжать итерации, и тогда возвращаемся к выполнению п. 2. Если значение ключа BREAK — истина, то итерации продолжать бессмысленно и надо попытаться выбрать шаг по времени, как это описано в п. 3.

6. Оформляем переход к новому временному слою. Для этого вызовом **<2.12> SHIFT** сдвигаем значения функций с  $(n+1)$ -го слоя на  $n$ -й, вызовом **<2.10> TIMSTP** выбираем новый шаг по времени.

На этом подпрограмма STEPON свою работу заканчивает.

**<2.2> MOTION** управляет расчетом уравнения движения. Как было описано в разд. 3, система уравнений движения решается итерациями. Счетчиком числа итераций служит переменная NITM.

1. Увеличиваем на единицу значение счетчика итераций.



Таблица 2

## Список программных единиц программы FLORA

Имя подпрограммы	Номер	Архивное имя	Назначение
Класс 0. Управление расчетом			
MODIFU	0.2	FLNOO2	Изменение управляющих параметров
Класс 1. Пролог			
LABRUN	1.1	FLN101	Маркировка расчета
PRESET	1,3	FLO103	Предварительное определение параметров расчета
DATA	1.4	FLN104	Определение данных расчета
AUXVAL	1.5	FLO105	Определение вспомогательных величин
INITAL	1.6	FLN106	Определение начальных физических условий
RESUME	1.7	FLO107	Возобновление расчета
Класс 2 Вычисления			
STEPON	2.1	FLO201	Шаг вычислений
MOTION	2.2	FLO202	Решение уравнений движения
STIT	2.3	FLO203	Начало итераций
PRESS	2.4	FLO204	Расчет гидродинамического давления
VISART	2.5	FLO205	Расчет давления искусственной вязкости
ABCFM	2.6	FLO206	Расчет коэффициентов разностного уравнения движения
GAUSS1	2.7	FLO207	Решение разностного уравнения методом прогонки
MOVEON	2.8	FLO208	Расчет гидродинамических переменных
CVERGE	2.9	FLO209	Исследование сходимости итераций
TIMSTP	2.10	FLO210	Выбор временного шага
EXAM	2.11	FLO211	Изучение текущего состояния системы
SHIFT	2.12	FLO212	Перепись переменных со слоя 2 на слой 1
BOUNDY	2.13	FLO213	Вычисление граничных условий
ENERGY	2.20	FLO220	Решение уравнения энергии
STATE	2.21	FLO221	Расчет уравнения состояния
SOURCE	2.24	FLO224	Расчет объемных источников энергии
HC DUCT	2.26	FLO226	Расчет коэффициента теплопроводности
ABCFE	2.27	FLO227	Расчет коэффициентов разностного уравнения энергии
FINDT	2.29	FLO229	Вычисление температуры на новом слое
CONSRV	2.32	FLO232	Вычисление отдельных членов закона сохранения энергии
Класс 3. Вывод			
OUTPUT	3.1	FLO301	Управление выводом
RECORD	3.2	FLO302	Чтение и запись результатов расчета
MPRINT	3.3	FLO303	Печать результатов расчета

Таблица 2 (окончание)

Имя подпрограммы	Номер	Архивное имя	Назначение
Класс 4. Эпилог			
TESEND	4.1	FLO401	Проверка на конец расчета
ENDRUN	4.2	FLO402	Прекращение расчета
Класс 5. Диагностика			
CLIST	5.2	FLO502	Печать COMMON переменных
ARRAYS	5.3	FLO503	Печать COMMON массивов

2. Вызовом <2.4> PRESS и <2.5> VISART вычисляем гидродинамическое давление, давление искусственной вязкости и их производные.

3. Вызовом <2.6> ABCFM вычисляем коэффициенты разностного уравнения (3.32) для изменения скорости.

4. Вызовом <2.7> GAUSS1 решаем разностное уравнение (3.32) и определяем изменения скоростей.

5. Вызовом <2.8> MOVEON по изменению скорости определяем значения скоростей, координат и плотностей на новой итерации.

6. Вызовом <2.11> EXAM проверяем разумность полученных значений функций (плотности и объемы ячеек должны быть положительны).

7. Вызовом <2.9> CVERGE проверяем, сошлись ли итерации.

Подпрограмма CVERGE вырабатывает значения двух логических ключей: NLGOON и BREAK. Если значение NLGOON — истина, то уравнения движения решены. Если значение NLGOON — ложь, то уравнения движения не решены. В этом случае, если значение ключа BREAK — ложь, то имеет смысл продолжать итерации, и тогда возвращаемся к выполнению п. 1. Если значение ключа BREAK — истина, то итерации продолжать бессмысленно и надо возвращаться в вызывающую программу.

<2.3> STIT оформляет начало итерационного процесса вычисления функций на новом временном слое. Значения температур и скорости на  $(n+1)$ -м слое полагаются равными значениям на  $n$ -м слое, а координаты и плотности рассчитываются по значениям скоростей.

<2.4> PRESS вычисляет давление во всех ячейках обращения к <2.21> STATE.

<2.5> VISART вычисляет давление искусственной вязкости согласно (3.13) и его производную по удельному объему.

<2.6> ABCFM вычисляет коэффициенты разностного уравнения (3.32) для приращения скорости согласно (3.31), (3.33), (3.34)...

<2.7> *GAUSS1* решает трехточечное разностное уравнение вида (3.32) методом прогонки.

<2.8> *MOVEON* вычисляет значение скоростей, координат и плотности на новой итерации согласно (3.30), (3.10).

<2.9> *CVERGE* проверяет выполнение условий сходимости итерационного процесса. Подпрограмма имеет один формальный параметр — целую переменную  $k$ , которая может принимать значения 1, 2, 3. При  $k=1$  исследуется сходимость итераций в уравнении движения, при  $k=2$  — в уравнении энергии, при  $k=3$  проверяется выполнение закона сохранения энергии.

<2.10> *TIMSTP* выбирает новый шаг по времени. Если входим в подпрограмму со значением ключа *BREAK* — истина, то это означает, что какой-то итерационный процесс не сходится. В этом случае шаг по времени уменьшаем в два раза. Если полученный таким образом шаг оказывается меньше минимального допустимого *DTMIN*, то выдаем диагностику вида

НОМЕР ШАГА... ШАГ ПО ВРЕМЕНИ... МЕНЬШЕ МИНИМАЛЬНОГО...

и уходим на окончание расчета. При значении ключа *BREAK* — ложь оставляем шаг постоянным, если число итераций хотя бы в одном из итерационных процессов оказалось больше половины максимального допустимого числа *NITMAX*. Если число итераций во всех процессах меньше или равно *NITMAX/2*, увеличиваем шаг по времени в полтора раза, если при этом он не станет больше максимального допустимого *DTMAX*.

<2.11> *EXAM* проверяет разумность результатов вычислений. Подпрограмма имеет один формальный параметр — целую переменную  $k$ . При  $k=1$  проверяется монотонность значений координат. При  $k=2$  проверяется положительность температуры. Если координаты не монотонны или температура отрицательна, выдается диагностика вида

ОТРИЦАТЕЛЬНЫЕ ОБЪЕМЫ или ТЕМПЕРАТУРА ОТРИЦАТЕЛЬНАЯ после чего печатается строка

*ICELL=... NSTEP=... DT=... TIME=...* и значение ключа *BREAK* полагается равным «истина».

<2.12> *SHIFT* переписывает значения переменных из массивов, относящихся к слою  $(n+1)$ -му, в массивы, относящиеся к слою  $n$ .

<2.13> *BOUNDY* рассчитывает граничные условия для уравнений движения и энергии. Подпрограмма имеет один формальный параметр — вещественную переменную *TNOW* — текущее значение времени. В теле подпрограммы имеется ссылка на модуль типа «вставка» с функциональным именем *BNDCON*. В стандартной версии программы эта вставка пустая и значения граничных величин не изменяются со временем.

<2.20> *ENERGY* управляет расчетом уравнения энергии. Как было описано в п. 3, уравнение энергии решается итерациями. Счетчиком числа итераций служит переменная *NITE*.

1. Увеличиваем на единицу значения счетчика итераций.

2. Вызовом <2.21> STATE вычисляем давление, энергию и их производные по температуре.

3. Вызовом <2.24> SOURCE вычисляем вклад объемных источников энергии.

4. Если имеется теплопроводность, вызовом <2.26> HCDUCT вычисляем значения коэффициентов теплопроводности.

5. Вызовом <2.27> ABCFE вычисляем коэффициенты разностного уравнения для изменения температуры на итерации.

6. Если есть теплопроводность, вызовом <2.7> GAUSS1 решаем разностное уравнение и определяем изменение температуры.

7. Вызовом <2.29> FINDT определяем температуру на новой итерации.

8. Вызовом <2.11> EXAM проверяем положительность полученных значений температуры.

9. Вызовом <2.9> CVERGE проверяем, сошлись ли итерации, и реагируем на значение ключей NLGOON и BREAK точно так же, как в подпрограмме <2.2> MOTION.

<2.21> STATE рассчитывает давление и внутреннюю энергию среды по уравнению состояния. Подпрограмма имеет один формальный параметр — целую переменную  $k$ . При  $k=2$  рассчитывается давление и его производная по плотности и температуре согласно (3.7). При  $k=1$ , кроме этого, рассчитываются внутренняя энергия и ее производная по температуре.

<2.24> SOURCE вычисляет вклад объемных источников энергии. В теле подпрограммы имеется ссылка на модуль типа «вставка» с функциональным именем ENSOUR. В стандартной версии программы эта вставка пустая и объемный источник энергии имеет массовую плотность вида (3.9).

<2.26> HCDUCT вычисляет коэффициент теплопроводности и его производную по температуре согласно (3.8).

<2.27> ABCFE вычисляет коэффициенты разностного уравнения для приращения температуры на итерации согласно (3.35), (3.37), (3.38).

<2.29> FINDT вычисляет температуру на слое  $(n+1)$ -м по ее приращению при наличии теплопроводности либо через коэффициенты разностного уравнения при ее отсутствии.

<2.32> CONSRV вычисляет отдельные члены, входящие в уравнение баланса энергии (3.29).

**Класс 3. Вывод.** К этому классу относятся подпрограммы <3.1> OUTPUT <3.2> RECORD, <3.3> MPRINT, обеспечивающие вывод информации на внешние устройства.

**Класс 4. Эпилог.** В этот класс входят подпрограммы <4.1> TESEND и <4.2> ENDRUN, назначение которых определяется системой OLYMPUS.

**Класс 5. Диагностика.** Подпрограммы <5.2> CLIST и <5.3> ARRAYS этого класса имеют назначение, определенное системой OLYMPUS.

## ЛИТЕРАТУРА

1. Самарский А. А., Попов Ю. П. Разностные методы решения задач газовой динамики. М.: Наука, 1980. 352 с.
2. Самарский А. А., Волосевич П. П., Волчинская М. И., Курдюмов С. П. Численные методы решения одномерных нестационарных задач магнитной гидродинамики. М.: ИПМ АН СССР, 1967. 72 с.
3. Самарский А. А., Волосевич П. П., Волчинская М. И., Курдюмов С. П. Метод конечных разностей для решения одномерных нестационарных задач магнитной гидродинамики.— ЖВМиМФ, 1968, т. 8, № 5, с. 1025—1038.
4. Горбунов-Посадов М. М., Карпов В. Я., Корягин Д. А. и др. Пакет прикладных программ САФРА. Системное наполнение: Препр. ИПМ АН СССР № 85. М., 1977. 27 с.
5. Roberts K. V. An introduction to the OLYMPUS sistem.— Computer phys. commun., 1974, N 7, p. 237—243.
6. Christiansen J. P., Roberts K. V. OLYMPUS. A standart control and utility package for inital-value FORTRAN programs.— Computer phys. commun., 1974, N 7, p. 245—270.
7. Гайфулин С. А., Карпов В. Я., Мищенко Т. В. САФРА. Функциональное наполнение. Система OLYMPUS: Препр. ИПМ АН СССР № 27. М., 1980. 16 с.
8. Christiansen J. P., Ashby D. E. T. F., Roberts K. V. MEDUSA. A onedimensional laser fusion code.— Computer phys. commun., 1974, N 7, p. 271—293.
9. Карпов В. Я. Алгоритмический язык Фортран. М.: Наука, 1976. 192 с.
10. Салтыков А. И., Макаренко Г. И. Программирование на языке Фортран. М.: Наука, 1976. 255 с.

УДК 532.516+519.63

### КОМПЛЕКС ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ГИДРОДИНАМИКИ, ТЕПЛО- И МАССООБМЕНА НА ОСНОВЕ НЕСТАЦИОНАРНЫХ УРАВНЕНИЙ НАВЬЕ — СТОКСА

А. В. Бунэ, В. Л. Грязнов, К. Г. Дубовик,  
В. И. Полежаев

Развитие методов численного моделирования гидродинамических процессов и все более широкое их использование при решении актуальных научно-технических задач требует создания специализированного математического обеспечения в виде пакетов и комплексов прикладных программ (КПП). При создании такого рода матобеспечения возникает ряд специфических проблем, рассматриваемых, например, в работах [1—3]. В настоящей статье описано программное обеспечение, разработанное в Институте проблем механики АН СССР для решения задач динамики несжимаемой вязкой жидкости на основе нестационарных уравнений Навье — Стокса, которое ориентировано на проведение научно-исследовательских работ пользователями различной квалификации. В связи с такой ориентацией было создано матобеспечение для анализа, обработки и представления результатов расчетов, а также программы, выполняющие