

РАЗРАБОТКА ПАКЕТА ПРИКЛАДНЫХ ПРОГРАММ “AKORD” ДЛЯ РЕШЕНИЯ ЗАДАЧ СО СВОБОДНЫМИ ГРАНИЦАМИ

К. Е. АФАНАСЬЕВ, Г. Г. КОРОТКОВ, Р. Р. ДОЛАЕВ
Кемеровский государственный университет, Россия
e-mail: keafa@kemgu.kemerovo.su

The principles of construction and realization of applied software package for the computation of problems with free boundaries by a boundary integral method are briefly considered. Model problem exposition in terms of the developed software package is described. The example of free surface evolution problem numerical modeling is considered. The free surface has a semicircular cavity at the initial instant and there is a plate (disk) under the one in liquid.

Пакеты прикладных программ (ППП) в практике проведения сложных инженерных и научных расчетов используются достаточно давно. Опыт применения ППП изложен в ряде отечественных работ, например: [8] (пакет FEMINA), [10] (пакет Минкур). Среди зарубежных ППП популярными являются пакеты “Algor” (www.algor.com) и “Ansys” (www.ansys.com) — для метода конечных элементов, пакет “Beasy” (www.beasy.com) — для метода граничных элементов. Все перечисленные пакеты имеют свою определенную специфику и не применимы для задач гидродинамики идеальной несжимаемой жидкости со свободными границами.

В центре новых информационных технологий Кемеровского государственного университета разрабатывается ППП “Akord”, предназначенный для исследования плоских и осесимметричных нестационарных задач идеальной несжимаемой жидкости со свободными границами. Результаты расчетов течений жидкости со свободными поверхностями находят многочисленные технические приложения, прежде всего в тех областях, где вязкостью можно пренебречь. Примерами таких течений являются: нестационарное движение уединенной волны по поверхности бассейна, имеющего дно произвольной геометрии, выход уединенных волн на мелководье, горизонтальное и вертикальное движение тел в жидкости с образованием волн, эволюция газовой полости в жидкости и др.

1. Базовая модель и алгоритм решения задачи идеальной несжимаемой жидкости

В основу пакета положена программная реализация метода граничных элементов (МГЭ), получившего в последнее время широкое развитие. Привлекательность данного метода прежде всего в том, что дискретизации подвергается лишь граница области. Для реализации такой возможности в МГЭ требуется переход от исходной краевой задачи для дифференциальных уравнений, описывающих некоторый процесс, к соотношениям, связывающим неизвестные функции на границе области. Эти соотношения, по существу, представляют собой граничные интегральные уравнения. В результате пространственная размерность задачи снижается на единицу и понижаются требования к гладкости функции.

Для осесимметричного случая применяется МГЭ [2], основанный на использовании третьей формулы Грина [6], связывающей искомую функцию дифференциальной задачи и ее нормальную производную. Для плоских задач наиболее эффективным является метод комплексных граничных элементов (МКГЭ), основанный на использовании интегральной формулы Коши [3] для функции комплексного потенциала.

В постановочной части плоская и осесимметричная задачи являются однотипными, поэтому ниже изложена постановка и методика решения плоской задачи на основе интегральной формулы Коши.

1.1. Постановка задачи

В расчетной области течения D , ограниченной свободной поверхностью Γ_1 и твердыми стенками Γ_2 (рис. 1, a), решается уравнение Лапласа

$$\Delta w(z) = 0, \quad z = x + iy \in D \quad (1)$$

для функции комплексного потенциала $w(z) = \varphi(x, y) + i\psi(x, y)$, где $\varphi(x, y)$ — функция потенциала скорости, $\psi(x, y)$ — функция тока, удовлетворяющие условиям Коши — Римана. На твердых границах выполняется условие непротекания

$$\psi = 0, \quad z \in \Gamma_2. \quad (2)$$

На свободной границе выполняются кинематическое и динамическое условия

$$\frac{dz}{dt} = \frac{\partial \varphi}{\partial x} + i \frac{\partial \varphi}{\partial y}, \quad z \in \Gamma_1, \quad (3)$$

$$\frac{d\varphi}{dt} - \frac{1}{2} |\nabla \varphi|^2 + gy = 0, \quad z \in \Gamma_1, \quad (4)$$

где g — ускорение свободного падения.

Для решения краевой задачи (1)–(4) необходимо также задать начальное положение свободной границы и начальное распределение потенциала на ней.

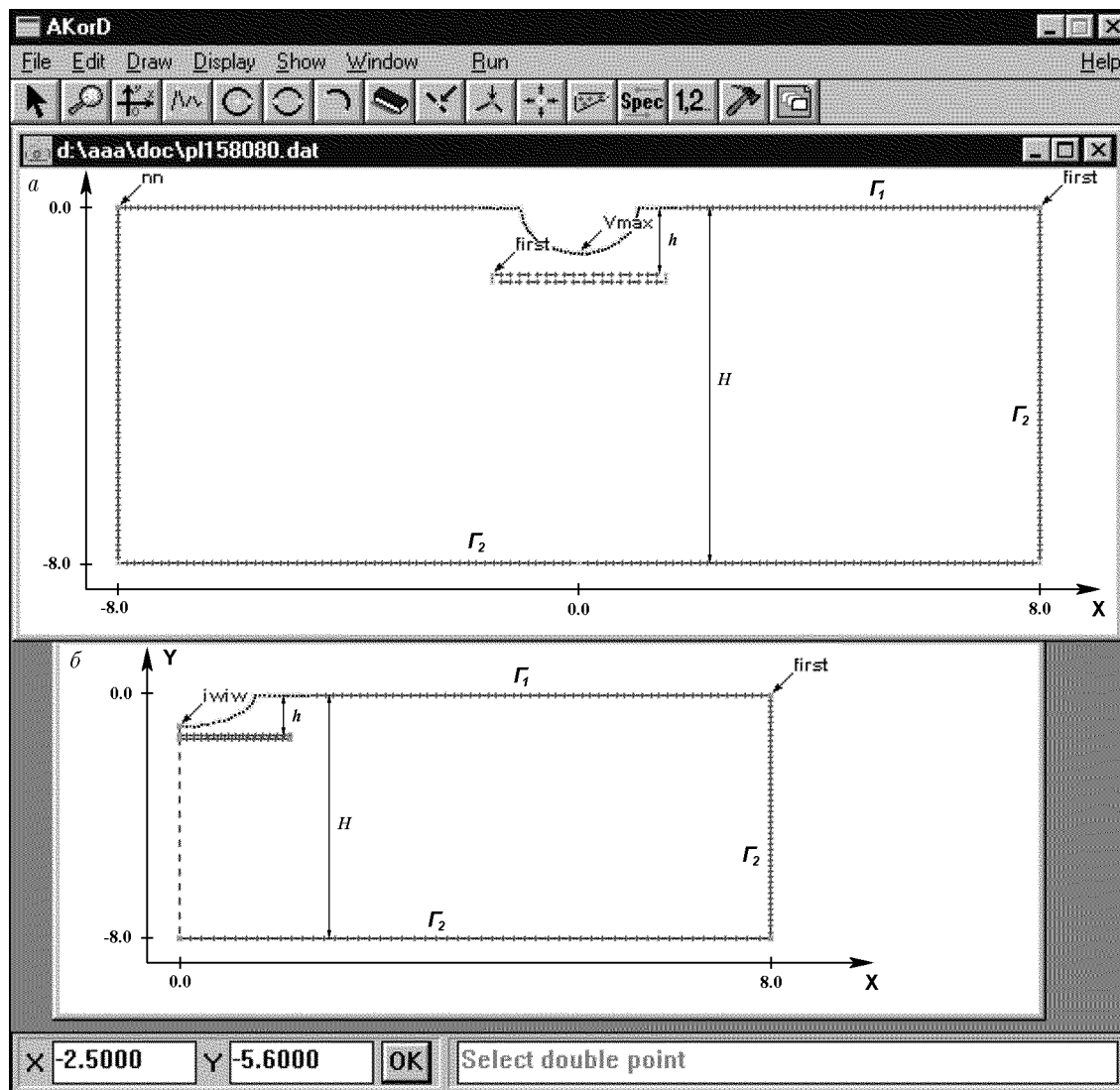


Рис. 1. Рабочий экран препроцессора и начальные геометрии для задач о схлопывании полукруговой выемки на свободной поверхности.
a — плоский случай, *b* — осесимметричный случай.

1.2. Алгоритм движения по времени

Краевая задача (1) – (4) является нелинейной в силу нелинейности динамического условия (4) и заранее неизвестного положения свободной границы. Кроме того, она является нестационарной, хотя и отличается от традиционных задач математической физики, в которых время явно входит в исходное дифференциальное уравнение. Здесь же явную зависимость от времени имеют лишь краевые условия (3) и (4), представляющие собой обыкновенные дифференциальные уравнения первого порядка, для решения которых применяется метод Эйлера с автоматическим выбором шага по времени.

Пусть в некоторый момент времени t_0 заданы положение свободной границы Γ_1 и распределение потенциала $\varphi^0(x, y)$ на ней. Для нахождения функции тока $\psi^0(x, y)$ и вектора скорости $\nabla\varphi$ на Γ_1 необходимо решить уравнение Лапласа (1) с условием $\varphi^0(x, y)$ на Γ_1 и условием (2) на Γ_2 . Новое положение свободной границы и распределение потенциала на ней для момента времени $t_0 + \tau$ можно вычислить, используя условия (3) и (4), дискретный

аналог которых расписывается по явной схеме Эйлера следующим образом:

$$z^{k+1} = z^k + (\nabla\varphi)^k \tau, \quad (5)$$

$$\varphi^{k+1} = \varphi^k + (0,5 |(\nabla\varphi)^k|^2 - gy^k) \tau, \quad k = 0, 1, 2, \dots, \quad (6)$$

где z^k, y^k, φ^k — значения функций на k -м шаге по времени. В результате получается смешанная краевая задача для уравнения Лапласа (1) с граничными условиями (2) и (6), но уже для момента времени $t_0 + \tau$. Повторное ее решение и использование граничных условий позволяет определить положение свободной границы и распределение потенциала на ней для момента времени $t_0 + 2\tau$ и т. д. Впервые такой алгоритм был описан в работе [5].

Величина τ , определяющая шаг по времени, находится автоматически из дополнительных условий:

1) любая частица жидкости за временной шаг не может переместиться на расстояние больше заданного;

2) не допускается самопересечение границы области.

Более подробно алгоритм движения по времени и выбор шага изложен в работе [1].

1.3. Метод граничных элементов

Свести краевую задачу для дифференциального уравнения эллиптического типа к граничному интегральному уравнению можно различными способами. Ниже описан метод, основанный на применении интегральной формулы Коши.

Пусть для области D с кусочно-гладкой замкнутой границей $\Gamma = \Gamma_1 \cup \Gamma_2$ (см. рис. 1, а) справедлива интегральная формула Коши, которую можно записать с помощью предельных формул Сохоцкого в виде

$$w(z_0) = \frac{1}{\varepsilon(z_0)i} \int_C \frac{w(z)}{z - z_0} dz, \quad (7)$$

где $\varepsilon(z_0) = 2\pi$ для внутренней точки, $\varepsilon(z_0) = \pi$ для точки на гладкой границе Γ , $\varepsilon(z_0) = \beta$ для угловой точки границы Γ (β — угол при вершине). Положительное направление обхода контура Γ берется таким образом, чтобы область D оставалась слева.

Учитывая, что на свободной поверхности известна действительная часть $\varphi(x, y)$ функции $w(z)$, а мнимая часть $\psi(x, y)$ известна на твердых стенках, мы имеем смешанную краевую задачу для функции $w(z)$. Численное решение этой задачи можно получить, разбив контур Γ на N линейных элементов Γ_j узлами z_j ($j = \overline{1, N}$). После указанного разбиения и линейной аппроксимации функции $w(z)$ на границе интеграл Коши можно вычислить аналитически в смысле главного значения при $z \rightarrow z_j$ [3]. В результате получим систему $N \times N$ линейных алгебраических уравнений

$$\mathbf{B}\mathbf{X} = \mathbf{F}, \quad (8)$$

где \mathbf{B} — полнозаполненная матрица, \mathbf{X} — вектор неизвестных для определения $\operatorname{Re} w$ на Γ_2 и $\operatorname{Im} w$ на Γ_1 , \mathbf{F} — вектор правой части, N — количество узлов, на которое разбита граница области. Для решения системы (8) применяется метод Гаусса с выбором ведущего элемента.

Для поиска решения в любой внутренней точке области используются уже известные значения функций на границе.

2. Структура пакета

При проведении численных расчетов трудности возникают не только на стадии собственно вычислений, но и при подготовке файла начальных данных и при обработке файла результатов расчета. Поэтому традиционно ППП состоят из трех основных частей: препроцессора, решателя и постпроцессора. В пакете “Akord” каждая из этих частей является независимой и может быть использована как отдельная программа. Связь между программами осуществляется с помощью стандартизированных потоков данных (файлов). Диаграмма работы пользователя с пакетом “Akord” изображена на рис. 2.

Препроцессор и постпроцессор написаны в программной среде Delphi 3.0 для операционной системы Windows 95, что делает описываемые программы легкими в освоении и позволяет пользоваться широким спектром возможностей, предоставляемых операционной системой. Решатель написан на языке FORTRAN, что дает возможность легко перенести программу практически на любую платформу. Предусмотрена также возможность запуска решателя из ППП “Akord” на удаленном высокопроизводительном вычислительном сервере.

2.1. Препроцессор

При проведении численных расчетов, особенно в областях со сложной геометрией, возникает проблема достоверного ввода информации геометрического и физического свойства. Для решения этой проблемы создаются специализированные программы, называемые препроцессорами.

На рис. 1 показано окно препроцессора для плоских и осесимметричных задач, а также приведен пример разбиения границы области D на граничные элементы для задач об эволюции полукруговой и полусферической выемки.

Препроцессор обладает развитым пользовательским интерфейсом, который позволяет существенно облегчить процесс ввода начальных данных задачи. Он дает возможность визуального ввода и редактирования начальной геометрии области, а также обладает набором диалоговых окон для задания свойств граничных элементов.

Оболочка препроцессора построена по принципу многодокументного интерфейса, что дает возможность одновременного редактирования нескольких файлов данных. Набор кнопок на панели инструментов соответствует различным функциям препроцессора и обеспечивает быстрый вызов этих функций.

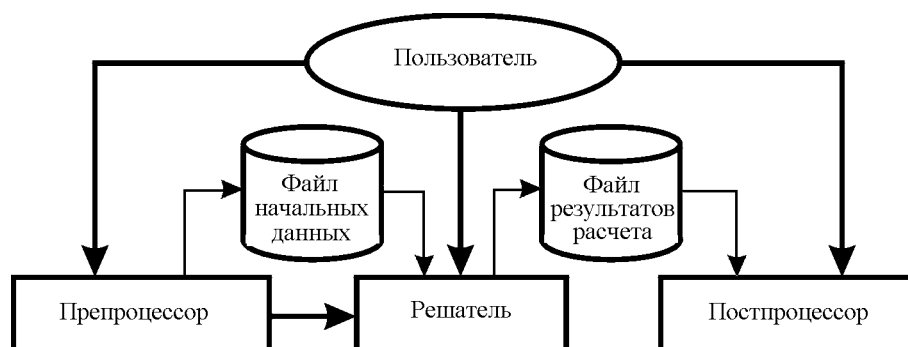


Рис. 2. Диаграмма работы пользователя и информационные потоки в пакете “Akord”.

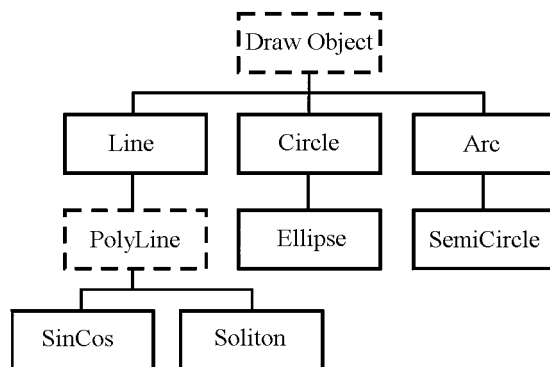


Рис. 3. Структура графических объектов.

Для построения расчетной области и задания граничных условий на ней используются визуальные средства графического конструирования. Начальная геометрия расчетной области задается при помощи встроенных графических объектов. Структура объектов приведена на рис. 3. Все графические объекты являются потомками базового класса $\langle DrawObject \rangle$, в котором определены функции и процедуры, характерные для всех объектов: отрисовка объекта, выделение объекта, перемещение объекта, разбиение объекта на граничные элементы, описание граничных условий на объекте, удаление объекта и т. д. Объект типа $\langle PolyLine \rangle$ является базовым для сложных графических объектов, задаваемых, например, формулами $a * \sin(c * x + b)$. Другим примером сложного графического объекта является объект типа $\langle Soliton \rangle$, применяемый при моделировании волновых задач.

В препроцессор также встроены дополнительные возможности, порожденные спецификой применяемого численного метода. Одна из таких возможностей — это задание двойных узлов в точках смены типа граничных условий [2]. Кроме того, имеется возможность присваивать точкам области уникальные имена, которые в дальнейшем используются решателем и постпроцессором.

Результатом работы препроцессора является текстовый файл, в котором данные по логическим признакам сгруппированы в следующие блоки.

[MAIN] — блок общих параметров: тип задачи (плоская или осесимметричная), общее количество узлов, количество внутренних узлов, количество подобластей;

[TIME] — блок временных параметров: время инициализации, минимальный и максимальный шаги по времени, максимально разрешенное время расчета;

[DOMAIN] — блок параметров, характеризующих область: ограниченная или безграничная, замкнутая или нет, направление нормали (вовнутрь или наружу);

[PHYSICAL] — блок физических параметров: максимальная погрешность расчета, поверхностное давление, плотность жидкости, коэффициент поверхностного натяжения, гравитация (есть или нет);

[NODES] — блок параметров, описывающих свойства граничных узлов: номер, координаты, тип граничного условия, начальное значение функции в узле;

[INTERNAL] — блок параметров, описывающих внутренние узлы области (если есть): номер, координаты;

[REGION] — блок параметров, описывающих границы подобластей: номер подобласти, номер первого узла подобласти, количество узлов в подобласти.

2.2. Решатель

Решатель предназначен для выполнения расчета по заданному файлу данных. В зависимости от значения параметров в файле данных происходит выбор соответствующих подпрограмм. Общий алгоритм типичного расчета выглядит следующим образом:

- INPUT — ввод начальных данных;
- CALC — проведение расчета (содержит цикл по времени);
 - OUTPUT — печать текущих результатов;
 - METHOD — приведение дискретного аналога интегральной формулы Коши к системе линейных алгебраических уравнений;
 - SOLVE — решение системы линейных уравнений методом Гауса с выбором ведущего элемента;
 - VELOXY — вычисление скоростей;
 - TIMEST — выбор шага по времени;
 - ALER — нахождение нового положения свободной границы и распределения потенциала на ней по формулам (5),(6);
 - ENER — вычисление кинетической и потенциальной энергий;
 - INTER — вычисление значения функции во внутренних точках области.

Результатом работы решателя является текстовый файл, данные в котором сгруппированы по блокам, каждый из которых имеет название, заключенное в квадратные скобки.

Существуют следующие виды блоков:

[ITERATION] — блок текущего момента времени (временного слоя).

Сразу после названия блока стоит обязательный параметр — время текущей итерации и необязательные параметры, характерные для области в целом (кинетическая и потенциальная энергии, циркуляция и т. п.). Для приведенного ниже примера время $t = .746800\text{E}+00$, потенциальная энергия $E_p = .627912\text{E}+00$, кинетическая энергия $E_k = .181619\text{E}+00$.

Кроме того, в этом блоке в каждой присутствующей строке содержатся значения координат одного узла, а также значения дополнительных параметров, заданных в этом узле. В описываемом примере первые два столбца определяют X -, Y -координаты, в третьем столбце приведено значение потенциала, в четвертом — значение функции тока, в пятом — значение X -компоненты вектора скорости, в шестом — значение Y -компоненты вектора скорости. Причем обязательными параметрами являются только значения координат узлов.

Пример:

```
[ITERATION] .746800E+00 .627912E+00 .181619E+00
             .399000E+00 -.363000E+00 .652470E+00 .116476E+01 -.112377E+01 .147556E+01
             .361000E+00 -.352000E+00 .711826E+00 .127641E+01 -.106145E+01 .167999E+01
```

...

[END]

[PARAM_DEF] — блок описания имен и типов параметров. Всем входящим в блок [Iteration] параметрам присваивается имя (NAME), по которому производится их идентификация. В частности, для приведенного выше примера этот блок имеет следующий вид:

```
[PARAM_DEF]
  NAME = 'Время'; SCOPE = Global
```

```

NAME = 'Потенциальная энергия'; SCOPE = Global
NAME = 'Кинетическая энергия'; SCOPE = Global
NAME = 'X'; SCOPE = Local
NAME = 'Y'; SCOPE = Local
NAME = 'Потенциал'; SCOPE = Local
NAME = 'Функция тока'; SCOPE = Local
NAME = 'Скорость по X'; SCOPE = Local
NAME = 'Скорость по Y'; SCOPE = Local

```

[END]

Кроме того, блок [PARAM_DEF] содержит переменную SCOPE, которая принимает значение Global, если параметр задан для области в целом, или Local, если параметр задан в каждом узле области.

[REGIONS] — блок описания границ подобластей: номер подобласти, номер первого узла подобласти, количество узлов в подобласти.

Блоки [PARAM_DEF] и [REGIONS] расположены в самом начале файла результатов.

2.3. Постпроцессор

Данные, получаемые в результате расчетов, передаются постпроцессору, который выполняет их дальнейшую обработку. Постпроцессор позволяет представить геометрические и физические свойства изучаемого процесса в графическом виде и проводить анализ результатов расчета. Поскольку задачи со свободными границами в большинстве случаев являются нестационарными, возможность просмотра результатов расчета в динамике — важная особенность постпроцессора.

Принципы программной реализации постпроцессора предусматривают возможность разработки дополнительных программных компонентов (модулей), позволяющих использовать существующие алгоритмы визуализации при появлении новых типов результирующих данных. Модулем называется часть программы, предназначенная для решения какого-либо выделенного класса задач. На рис. 4 приведена структура постпроцессора и схема взаимодействия модулей различного типа.

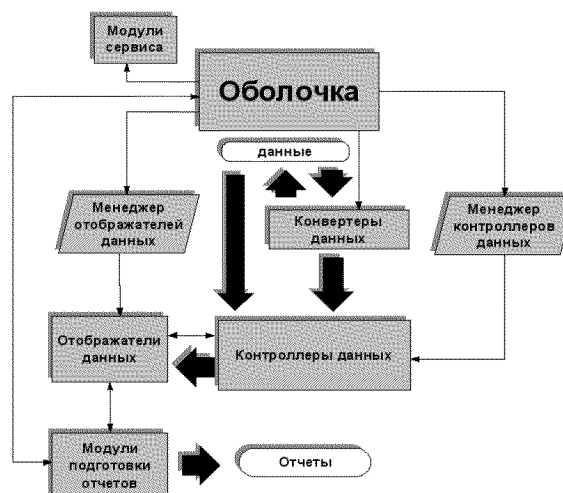


Рис. 4. Структура постпроцессора.

Существуют следующие основные типы модулей.

КОНВЕРТЕРЫ ДАННЫХ используются для преобразования данных из одного фор-

мата в другой. Реализованы три вида конвертеров:

KONVERT_TO_EXT — преобразование в расширенный формат. Этот конвертер создает таблицу распределения данных в файле. В таблице содержатся смещения каждого блока типа [Iteration] относительно начала файла, что позволяет организовать быстрый доступ к отдельным блокам данных.

KONVERT_TO_BIN — преобразование в двоичный формат. Этот конвертер преобразует расширенный формат файла данных в двоичный, в результате чего данные занимают меньший объем, кроме того, существенно возрастает скорость их обработки.

KONVERT_TO_TEXT — преобразование двоичного файла в текстовый.

КОНТРОЛЛЕРЫ ДАННЫХ предназначены для считывания результатов расчета и предоставления данных другим модулям. Реализованы следующие виды контроллеров.

CONTROL_PARAM — контроллер изменения параметра. Этот модуль осуществляет выборку данных из каждого временного слоя для указанного глобального параметра. В результате выдается изменение выбранного параметра с течением времени.

CONTROL_LAYER — контроллер временного слоя. Этот модуль вычисляет временной интервал для существующих временных слоев в файле данных и предоставляет их для любого значения времени из этого интервала. Если для запрашиваемого времени данных не существует, то производится их интерполяция по близлежащим временным слоям.

CONTROL_INTER_PNT — контроллер внутренних точек. Является производным от контроллера временного слоя. С его помощью осуществляется построение четырехугольной сетки по заданной границе расчетной области. Сетка строится либо в автоматическом (области выпуклости определяются автоматически), либо в полуавтоматическом (области выпуклости задаются вручную) режиме. Значения функции во внутренних узлах сетки вычисляются с помощью обобщенной формулы Коши (плоский случай) или при помощи третьей формулы Грина (осесимметричный случай).

ОТОБРАЖАТЕЛИ ДАННЫХ отвечают за представление результатов расчета в графическом виде. Каждый из отображателей генерирует свое рабочее окно. Реализованы следующие виды отображателей:

WIEV_STATIC_DAT — отображатель статических данных. Предназначен для отображения статических данных типа набора точек. Данные для этого отображателя предоставляются контроллером изменения параметра.

WIEV_BOUND — отображатель границы. Предназначен для отображения динамических данных типа набора точек. Исходные данные для этого отображателя предоставляются контроллером временного слоя. Этот отображатель позволяет отследить изменение с течением времени границы расчетной области. Он также позволяет выводить скорости и ускорения точек на границе области в виде стрелок различной длины, задающих направление и длину векторов. Предусмотрена возможность просмотра отдельных частей границы в увеличенном масштабе. Для этого необходимо выделить увеличиваемую область (фрейм) и задать закон ее движения. Отображатель границы генерирует два окна, в одном из которых выводится полная геометрия области, а в другом все выделенные фреймы.

WIEV_GRID — отображатель сетки, вызывает контроллер внутренних точек и отображает результат его работы.

WIEV_VECTOR_FIELD — отображатель векторного поля. Предназначен для отображения статических векторных полей. Исходные данные для этого отображателя предоставляются контроллером внутренних точек.

WIEV_LEVEL_LINES — отображатель линий уровня: отвечает за отрисовку линий уровня по данным, полученным от контроллера внутренних точек.

МОДУЛИ ПОДГОТОВКИ ОТЧЕТОВ предназначены для подготовки, сохранения и распечатки графических изображений.

REPORT_GRAPH — графический отчет. Графические объекты для этого модуля предоставляются отображателями данных, после чего появляется возможность их редактирования, а именно: выбирать различный вид осей координат и добавлять необходимые сопутствующие надписи. Характерной особенностью модуля подготовки графических отчетов является то, что все объекты задаются в векторной форме, позволяющей формировать графические изображения с любым необходимым разрешением. Графические изображения можно сохранять в форматах *.bmp*, *.psx*, *.wmf*.

МОДУЛИ СЕРВИСА выполняют различные вспомогательные функции.

Интерактивная графическая среда программы (рис. 5) содержит стандартные элементы управления, характерные для многодокументного интерфейса, такие как главное меню

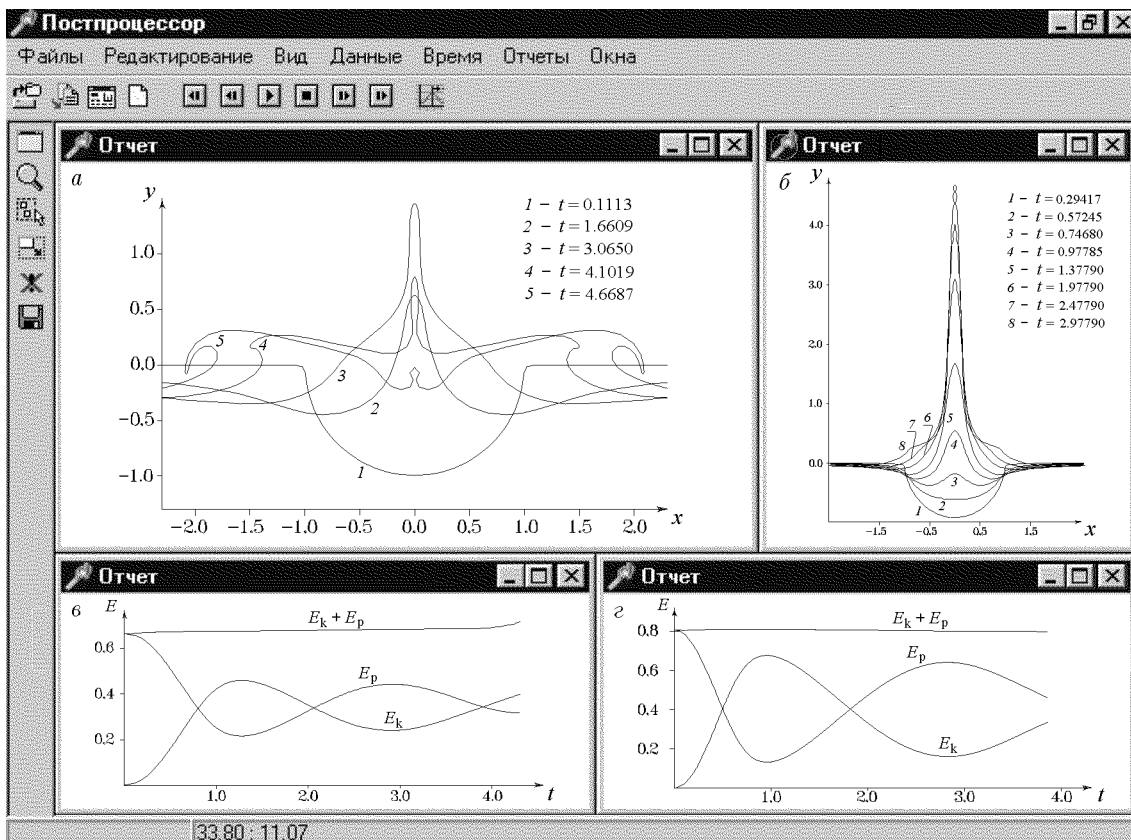


Рис. 5. Эволюция свободной поверхности и изменение энергетических характеристик с течением времени для плоского (а, б) и осесимметричного (в, г) случаев.

E_k — кинетическая, E_p — потенциальная энергия.

программы, панели инструментов и строка состояния, рабочие и диалоговые окна программы. Горизонтальная панель инструментов отвечает за вызов функций постпроцессора, характерных для документа в целом (открытие файла данных, изменение текущего момента времени, и т. д.), вертикальная панель отвечает за вызов локальных функций активного рабочего окна и изменяется при переключении окон.

3. Применение ППП

В качестве примера использования ППП “Akord” рассмотрим задачу об эволюции полукруговой выемки на свободной поверхности, которая впервые была поставлена М. А. Лаврентьевым [5]. Данный класс задач интересен тем, что в процессе схлопывания образуется кумулятивная струя жидкости, обладающая высокой скоростью и сложной геометрией, вследствие чего данное явление становится трудным для математического моделирования. В идеальной жидкости задача решалась многими авторами (см., например, [4], [9]). Практически во всех работах проводится анализ формы свободной поверхности без проведения аналогий между плоской и осесимметричной задачами.

В настоящей работе исследуется влияние дна и близко расположенного снизу препятствия в виде пластины (диска) на скорость кумулятивной струи и на характер возникающих в жидкости течений. Задача решается в плоской и осесимметричной постановках.

Цель расчетов — сравнение плоской и осесимметричной задач по скорости кумулятивной струи и по энергетическим характеристикам с целью нахождения аналогий и возможного обобщения на задачи, не имеющие осевой симметрии. Тогда по расчету плоской задачи можно будет судить о соответствующей трехмерной задаче.

3.1. Задание начальной геометрии

Рассматривается задача о формировании кумулятивной струи из полукруговой (полусферической) выемки, расположенной на свободной поверхности Γ_1 области D , ограниченной твердыми стенками Γ_2 (см. рис. 1). Начало системы координат помещалось в центр круга (шара), нижняя часть которого образует выемку. Для удобства расчетов задача приводилась к безразмерному виду. В качестве характерных размерных величин брались начальный радиус выемки R и ускорение свободного падения g . Глубина бассейна H изменялась в пределах $R < H < 10R$. Ширина бассейна равнялась $16R$. На глубине $R < h \leq H$ располагалась планка (диск), ширина которой L изменялась в пределах $0 < L < 6R$. Центр планки (диска) всегда находился в точке $x = 0$.

Для ввода начальной геометрии области D при помощи препроцессора необходимо выполнить следующую типичную последовательность действий:

- запустить препроцессор;
- при помощи пункта меню *File/New...* задать имя нового файла данных и выбрать тип задачи (плоская или осесимметричная);
- при помощи пункта меню *Display/Size...* задать физические размеры графического окна;
- при помощи графических объектов *Line* и *Semicircle* нарисовать границу области;
- задать тип граничного условия, а также разбиение каждого графического объекта на граничные элементы (пункт меню *Draw/Set Boundary Param...*);
- задать первую точку и направление обхода области (пункт меню *Draw/First Point*);

задать двойные точки (пункт меню *Draw/Double Point*);
 задать специальные точки (пункт меню *Draw/Special Point...*);
 сохранить файл данных (пункт меню *File/Save as...*).

Для плоской задачи количество точек на свободной поверхности равнялось 200, на дне 100, на левой и правой стенках 50. На планке разбиение вводилось таким образом, чтобы длина граничного элемента на планке и на твердых стенках совпадала. Разбиение для осесимметричной задачи соответствует разбиению для плоской задачи (длины соответствующих элементов равны).

3.2. Обработка результатов расчета

После подготовки файла начальных данных при помощи пункта меню *Run* дается команда на расчет задачи. Включается выбранный решатель, результатом работы которого является текстовый файл (его формат описан в п. 3.2).

Для того, чтобы в постпроцессоре получить изображения, аналогичные приведенным на рис. 5, нужно выполнить следующую последовательность действий:

- запустить постпроцессор;
- при помощи пункта меню *Файл/Конвертировать...* преобразовать текстовый файл в двоичный;
- загрузить преобразованный файл (пункт меню *Файл/Открыть...*);
- создать окно отображателя данных (пункт меню *Данные/Создать отображатель*);
- выбрать отображаемые данные (пункт меню *Данные/Добавить данные...*);
- задать шаг по времени (пункт меню *Время/Параметры времени...*);
- при помощи кнопок на главной панели инструментов просмотреть изменение выбранных данных с течением времени;
- создать окно отчета (пункт меню *Отчеты/Создать отчет*);
- при помощи пункта меню *Отчеты/Поместить в отчет* геометрию области для текущего момента времени из окна отображателя данных поместить в отчет.
- добавить сопутствующие надписи (пункт меню *Отчеты/Текст*);
- сохранить полученный рисунок как графический файл (пункт меню *Файл/Сохранить как...*);

3.3. Результаты расчета

На рис. 5, *a, б* приведены формы свободной поверхности в различные моменты времени (радиус выемки $R = 1$, глубина бассейна $H = 2$). В плоской задаче (см. рис. 5, *a*) кривая 3 соответствует максимальной амплитуде всплеска. Для подтверждения консервативности методов на рис. 5, *в, г* приведены энергетические характеристики задачи. Из рисунка видно, что полная энергия с достаточной точностью сохраняет свое значение.

Из опытов известно, что при уменьшении глубины бассейна амплитуда кумулятивной струи возрастает [7]. В подтверждение этого факта на рис. 6, *a* показана зависимость максимальной скорости кумулятивной струи от глубины бассейна для осесимметричного случая. Из рисунков видно, что при глубине, равной начальному диаметру выемки и больше, глубина практически не оказывает влияния на скорость кумулятивной струи.

Для плоской задачи кривая зависимости имеет аналогичный вид. На рис. 6, *б* показана относительная скорость струи $V = V_0/V_n$, где V_0 и V_n — максимальные скорости кумулятивной струи соответственно для осесимметричного и для плоского случаев.

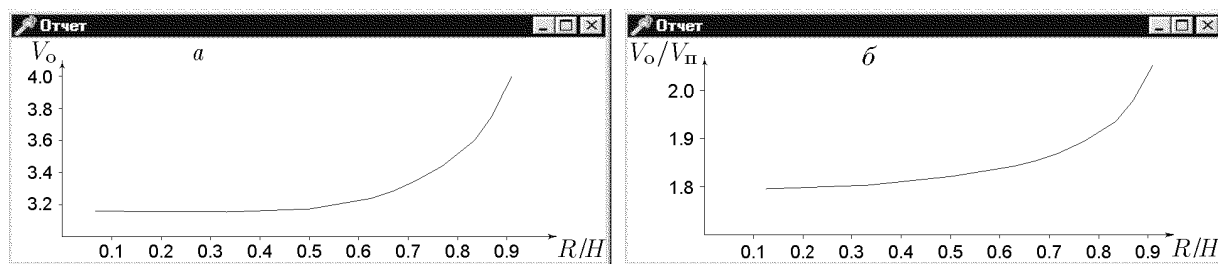


Рис. 6. Зависимость скорости кумулятивной струи от глубины бассейна для осесимметричного случая (а) и относительной скорости $V = V_o/V_{\Pi}$ (б).

R — начальный радиус выемки на свободной поверхности, H — глубина бассейна.

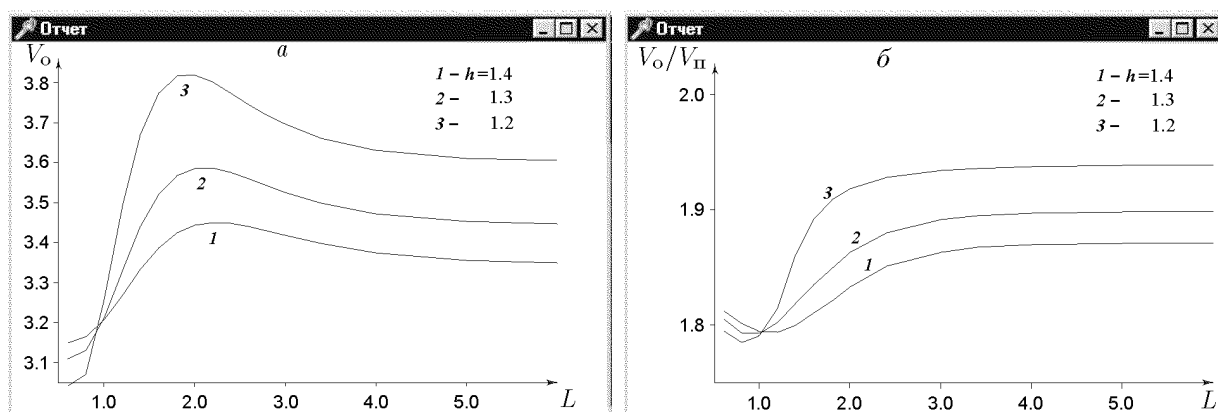


Рис. 7. Зависимость скорости V кумулятивной струи от ширины планки L для осесимметричного случая (а) и относительной скорости $V = V_o/V_{\Pi}$ (б).

h — глубина залегания планки.

На рис. 7, а показана зависимость максимальной скорости кумулятивной струи от диаметра диска для осесимметричного случая. Кривая 1 соответствует глубине размещения диска $1 - h = 1.4$, 2 — $h = 1.3$, 3 — $h = 1.2$. Видно, что максимальную скорость кумулятивная струя имеет, когда диаметр диска близок к начальному диаметру выемки на свободной поверхности. При увеличении диаметра диска скорость кумулятивной струи падает и выходит на уровень, соответствующий скорости вблизи дна при отсутствии диска (см. рис. 6, а). Для плоской задачи график имеет аналогичный вид. На рис. 7, б показана относительная скорость струи $V = V_o/V_{\Pi}$.

Из графиков относительных скоростей хорошо видно, что между физическими процессами, возникающими при эволюции выемки в плоском и осесимметричном случаях, существует подобие. В дальнейшем это позволит оценить результаты трехмерных задач по анализу плоских, но не имеющих осевой симметрии.

Подводя итоги, отметим, что использование пакета прикладных программ для решения волновых задач гидродинамики в значительной степени облегчает процесс исследования на всех стадиях работы над задачей.

Список литературы

- [1] АФАНАСЬЕВ К. Е., САМОЙЛОВА Т. И. Техника использования метода граничных элементов в задачах со свободными границами. *Вычисл. технологии. ИВТ СО РАН*, Новосибирск, 4, №11, 1995, 19–37.
- [2] БРЕВБИЯ К., ТЕЛЛЕС Ж., ВРОУБЕЛ Л. *Методы граничных элементов*. Мир, М., 1987.
- [3] ГРОМАДКА Т., ЛЕЙ Ч. *Комплексный метод граничных элементов*. Мир, М., 1990.
- [4] КЕДРИНСКИЙ В. К. Поверхностные эффекты при подводном взрыве (обзор). *Журн. прикл. мех. и техн. физ.*, №5, 1978.
- [5] ЛАВРЕНТЬЕВ М. А., ШАБАТ Б. В. *Проблемы гидродинамики и их математических моделей*. Наука, М., 1977.
- [6] МАРЧУК Г. И., АГОШКОВ В. И. *Введение в проекционно-сеточные методы*. Наука, М., 1981.
- [7] МАЙЕР В. В. *Кумулятивный эффект в простых опытах*. Наука, М., 1989.
- [8] ПОЛЕЖАЕВ В. И. *Математическое моделирование конвекционного теплообмена на основе уравнений Навье – Стокса*. Наука, М., 1987.
- [9] ТЕРЕНТЬЕВ А. Г., АФАНАСЬЕВ К. Е. Численные методы в гидродинамике: *Учеб. пособие. Чуваш. ун-т. им. И. Н. Ульянова*, Чебоксары, 1987.
- [10] ШОКИН Ю. И., ЧУБАРОВ Л. Б., МАРЧУК А. Г., СИМОНОВ К. В. *Вычислительный эксперимент в проблеме цунами*. Наука, М., 1989.

Поступила в редакцию 30 ноября 1998 г.,
в переработанном виде 2 июня 1999 г.