

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Математический факультет

Кафедра вычислительной математики

## ДИПЛОМНАЯ РАБОТА

«Дифракция волн над подвижным и неподвижным дном»

студента 5 курса

Долгова Дмитрия Андреевича

Специальность 010501 – «Прикладная математика и информатика»

Научный руководитель

д.ф-м.н., профессор

Захаров Ю.Н.

Работа допущена к защите:

” \_\_\_\_ ” 2012г.

Зав. кафедрой

д.ф.-м.н., профессор

Ю.Н. Захаров

Работа защищена:

” \_\_\_\_ ” 2012г.

с оценкой \_\_\_\_\_

Председатель ГАК

\_\_\_\_\_

Члены ГАК:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Кемерово 2012

# Оглавление

|  |           |
|--|-----------|
| <b>Введение</b>  | <b>3</b>  |
| <b>Глава 1. Метод решения</b>                                      | <b>5</b>  |
| § 1.1. Постановка задачи . . . . .                                 | 5         |
| § 1.2. Аппроксимация уравнений . . . . .                           | 7         |
| § 1.3. Выбор граничных условий . . . . .                           | 9         |
| 1.3.1. Виды неотражающих краевых условий . . . . .                 | 9         |
| 1.3.2. Замыкание системы уравнений . . . . .                       | 9         |
| § 1.4. Алгоритм решения . . . . .                                  | 14        |
| § 1.5. Описание двухслойной схемы неполной аппроксимации . . . . . | 15        |
| 1.5.1. Случай, когда $\alpha_{n+1} = \text{const}$ . . . . .       | 15        |
| 1.5.2. Случай, когда $\alpha_{n+1}$ является матрицей . . . . .    | 17        |
| 1.5.3. Метод минимальных невязок . . . . .                         | 18        |
| <b>Глава 2. Результаты расчетов</b>                                | <b>19</b> |
| § 2.1. Расчет со сложной начальной формой волны . . . . .          | 20        |
| § 2.2. Расчет движения над сложным дном . . . . .                  | 24        |
| § 2.3. Сравнение линейных и нелинейных уравнений . . . . .         | 29        |
| § 2.4. Эффект возникновения волны . . . . .                        | 37        |
| 2.4.1. Зависимость от параметров . . . . .                         | 41        |

|  |           |
|--|-----------|
| 2.4.2. Расчет с начальных данных . . . . . | 43        |
| § 2.5. Выводы . . . . .                    | 45        |
| <b>Глава 3. Приложения</b>                 | <b>46</b> |
| § 3.1. Программные средства . . . . .      | 46        |
| 3.1.1. Решенные задачи . . . . .           | 47        |
| 3.1.2. Сервер вычислений . . . . .         | 52        |
| § 3.2 Список иллюстраций . . . . .         | 55        |
| § 3.3 Список таблиц . . . . .              | 56        |
| <b>Литература</b>                          | <b>57</b> |

# Введение

Работа посвящена решению задачи о движении волн на поверхности идеальной несжимаемой жидкости в однородном поле сил тяжести над подвижным и неподвижным дном различной формы. Влияние изменения дна на поведение поверхностных волн до сих пор недостаточно изучено, хотя его понимание является во многих случаях важным. Для практики необходимо знать параметры, ответственные за механизмы возникновения и роста волн, законы их распространения, взаимодействия и действия на твердые тела.

Т.к. проведение натурных и лабораторных экспериментов в этой области почти невозможно, основным способом изучения данной проблемы является математическое моделирование, которое имеет ряд значимых преимуществ перед натурным экспериментом. В настоящее время содержание математического моделирования, его возможности и актуальность создания математических моделей претерпели крупные изменения. Это связано не только с быстрым развитием средств вычислительной техники, но и с совершенствованием существующих и разработкой новых численных методов реализации сложных математических моделей.

Существует несколько моделей, описывающих движение поверхностных волн разной степени сложности и адекватности реальному процессу их распространения. Но при использовании любой из них для описания движения волн в безграничной области существует проблема переноса краевых условий из бесконечности на границу расчетной области.

В общем случае рассматриваемые процессы происходят на неограниченном участке пространства, но при моделировании необходимо поставить искусственные границы, чтобы предотвратить расчет бесконечного числа точек. При этом надо располагать границы достаточно далеко от исследуемого объекта, чтобы их влияние на интересующее нас ближнее поле было мало. Но такой подход применим не всегда.

Другой путь состоит в том, чтобы на близких искусственных границах ставить такие условия, которые неискажали бы поле внутри области, т.е. обеспечивали полное или частичное поглощение приходящих возмущений.

В данной работе рассматривается еще один способ ограничения расчет-

ной области. Исследуется влияние изменения формы дна на поведение поверхностных волн с учетом переноса краевых условий из бесконечности на границу расчетной области на примере модели мелкой воды. Эта модель с достаточной точностью описывает движение волн цунами, т.к. предполагает, что высота волны много меньше ее длины. Уравнения, описывающие данную модель, используются в линейной и нелинейной форме. Полученные расчеты анализируются и сравниваются между собой.

Раздел «Постановка задачи» содержит математическую постановку и графическое описание задачи. В разделе «Метод решения» содержится алгоритм проведения расчетов, а также кратко описывается метод неполной аппроксимации. Графические результаты расчетов с указанием расчетных параметров помещены в раздел «Результаты расчетов». В разделе «Программные средства» описывается техническая реализация расчетной программы и используемые сторонние средства.

# Глава 1. Метод решения

Глава содержит математическую постановку задачи и описание метода решения. Рассматриваются существующие способы постановки краевых условий и предлагается альтернативный способ замыкания системы уравнений. В конце дается краткое описание метода неполной аппроксимации минимальных невязок.

## § 1.1 Постановка задачи

Исследуется задача о движении волн на поверхности безграничной идеальной несжимаемой жидкости, которая находится в однородном поле сил тяжести. Течение является потенциальным, т.е. для вектора скорости  $\vec{u} = \{u, v, w\}$  существует  $\phi(x, y, z, t) : \vec{u} = \text{grad}\phi$ . Введём прямоугольную декартову систему координат так, что ось  $Oz$  направлена вертикально вверх, в сторону, противоположную направлению ускорения свободного падения, а оси  $Ox$  и  $Oy$  лежат на невозмущённой свободной поверхности (рис. 1.1). Жидкость заполняет область  $Q$ , ограниченную сверху свободной поверхностью  $z = \eta(x, y, t)$ , снизу — дном  $z = -\tilde{H}(x, y, t) = -H(x, y) + B(x, y, t)$ . Здесь  $H$  определяет неподвижную часть дна, а  $B$  — его динамическую составляющую.

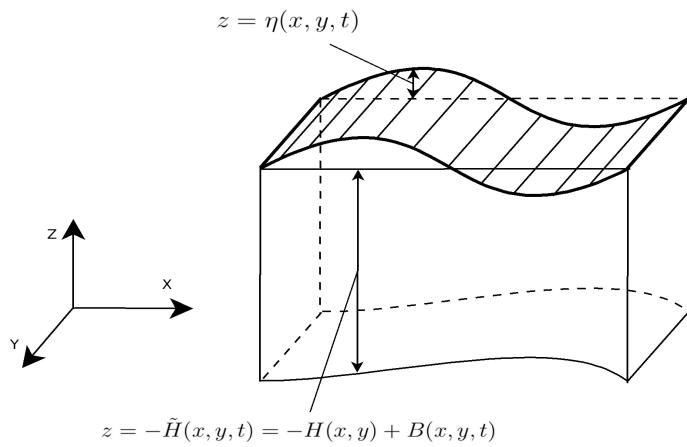


Рис. 1.1 Постановка задачи

В качестве физической модели, описывающей данную задачу, взяты уравнения теории мелкой воды (в линейной форме известные также как уравнения Сен-Венана), для которых существенными являются предположения о

малости вертикального ускорения частиц жидкости по отношению к ускорению свободного падения и о слабой зависимости горизонтальных скоростей от вертикальной координаты.

Ситуации, когда глубина акватории много меньше горизонтальных размеров, достаточно обычна, поэтому уравнения мелкой воды находят широкое применение. Часто они используются с учётом кориолисовых сил при моделировании атмосферы и океана как упрощение системы примитивных уравнений, описывающих потоки в атмосфере.

Представим уравнения мелкой воды в следующей форме:

$$\frac{\partial U}{\partial t} + F_{lin} + F_{nonlin} = \frac{\partial B}{\partial t} \quad (1.1.1)$$

$$U = \begin{pmatrix} \eta \\ u \\ v \end{pmatrix} \quad B = \begin{pmatrix} B(x, y, t) \\ 0 \\ 0 \end{pmatrix}$$

$$F_{lin} = \begin{pmatrix} \frac{\partial((H-B)u)}{\partial x} + \frac{\partial((H-B)v)}{\partial y} \\ g \frac{\partial \eta}{\partial x} \\ g \frac{\partial \eta}{\partial y} \end{pmatrix} \quad (1.1.2)$$

$$F_{nonlin} = \begin{pmatrix} 0 \\ \frac{1}{2} \frac{\partial(u^2)}{\partial x} + v \frac{\partial u}{\partial y} \\ \frac{1}{2} \frac{\partial(v^2)}{\partial y} + u \frac{\partial v}{\partial x} \end{pmatrix} \quad (1.1.3)$$

$$(x, y) \in \Omega, t > 0$$

$$u_0 = u(x, y, 0) \quad v_0 = v(x, y, 0) \quad \eta_0 = \eta(x, y, 0) \quad (1.1.4)$$

$$u(x, y, t), v(x, y, t), \eta(x, y, t) \rightarrow 0 \quad x^2 + y^2 \rightarrow \infty$$

Где  $H$  определяет неподвижную часть дна, а  $B$  - его динамическую составляющую ( $H - B$  – полная грубина слоя жидкости),  $\eta$  – функция свободной поверхности,  $u(x, y, t)$  – скорость по Ох,  $v(x, y, t)$  - скорость по Оу,  $g = 9,8$ .

Данные формулы являются гиперболической системой дифференциальных уравнений и описывают более общую постановку задачи в нелинейном виде без учёта сил Кориолиса, трения и вязкости в бесконечной области (более подробно об уравнениях мелкой воды см. [1], [2]). Линейная форма может быть получена как частный случай - для этого достаточно отбросить  $F_{nonlin}$ .

## § 1.2 Аппроксимация уравнений

Задача (1.1.1), (1.1.4) решалась методом сеток. Для решения в расчетной области  $\Omega$  введем обычным образом неравномерную в общем случае сетку  $\Omega_h$  с шагом  $h_{xi} = x_{i+1} - x_i$ ,  $h_{yi} = y_{i+1} - y_i$  и шагом по времени  $\tau = t^{n+1} - t^n$ .  $n$ -й слой по времени определим как множество узлов  $(i, j)$  при фиксированном  $t^n$ .

Обозначим значение сеточной функции  $\phi$  в узле  $(i, j)$ , как  $\phi_{ij}^n = \phi(x_i, y_i, t^n)$  и аппроксимируем уравнения (1.1.1), (1.1.4) в узлах сетки  $\Omega_h$  на  $n$ -м слое неявной конечно-разностной схемой второго порядка аппроксимации:

$$\frac{U_{ij}^{n+1} - U_{ij}^n}{\tau} + F_{lin}^{n+1} + F_{nonlin}^{n+1} = \frac{B_{ij}^{n+1} - B_{ij}^n}{\tau} \quad (1.2.1)$$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{i+1j} - B_{i+1j}^{n+1})u_{i+1j}^{n+1} - (H_{i-1j} - B_{i-1j}^{n+1})u_{i-1j}^{n+1}}{2h_x} + \\ \frac{(H_{ij+1} - B_{ij+1}^{n+1})v_{ij+1}^{n+1} - (H_{ij-1} - B_{ij-1}^{n+1})v_{ij-1}^{n+1}}{2h_y} \\ \frac{u_{ij}^{n+1} - u_{ij}^n}{\tau} + g \frac{\eta_{i+1j}^{n+1} - \eta_{i-1j}^{n+1}}{2h_x} \\ \frac{v_{ij}^{n+1} - v_{ij}^n}{\tau} + g \frac{\eta_{ij+1}^{n+1} - \eta_{ij-1}^{n+1}}{2h_y} \end{pmatrix} \quad (1.2.2)$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{ij} \frac{u_{ij+1}^{n+1} - u_{ij-1}^n}{2h_y} + \frac{(u_{i+1j}^{n+1})^2 - (u_{i-1j}^n)^2}{2h_x} \\ u_{ij} \frac{v_{ij+1}^{n+1} - v_{ij-1}^n}{2h_x} + \frac{(v_{ij+1}^{n+1})^2 - (v_{ij-1}^n)^2}{2h_y} \end{pmatrix} \quad (1.2.3)$$

$$u_{ij}^0 = u_0(x_i, y_i) \quad v_{ij}^0 = v_0(x_i, y_i) \quad \eta_{ij}^0 = \eta_0(x_i, y_i) \quad (1.2.4)$$

$$n = 0, 1 \dots N \quad i, j = 0, \pm 1, \pm 2 \dots$$

где  $H_{i,j} = H(x_i, y_j)$ ,  $B_{i,j,n} = B(x_i, y_i, t^n)$  - постоянная и динамическая составляющие функции дна.

Принято считать, что в неявной схеме можно без потери точности вычислений брать шаг по  $t$  гораздо большим, чем в явной, и этим существенно поднять скорость вычислений или повысить точность. Кроме того, подобные

схемы оптимально подходят для задач на установление, когда вместо решения уравнения  $AX = 0$  шагами по времени ищут состояние покоя системы  $\frac{dX}{dt} = AX$ , наступающее в отдаленном времени.

## § 1.3 Выбор граничных условий

Для численного решения задачи (1.1.1), (1.1.4) необходимо ограничить расчетную область. Основной проблемой при этом является поиск таких условий на искусственных границах, которые не приводят к отражениям или искажениям волны, проходящей через границу (так называемых, «прозрачных» или «неотражающих» краевых условий).

Как известно (см. [3], [4]) существуют способы задания искусственных краевых условий которые в некоторых случаях позволяют выходить волнам из  $Q$  без отражений и искажений.

### 1.3.1 Виды неотражающих краевых условий

В кратце опишем существующие виды прозрачных краевых условий:

- Неотражающие граничные условия

Основная идея постановки неотражающих граничных условий заключается в нахождении решения системы и представления его в виде суперпозиции разнонаправленных волн. Исходя из этого на границе ставятся такие условия, которые уничтожают ненужные волны (пример такой постановки условий [5]).

- Поглощающие граничные условия

Основная идея постановки поглощающих граничных условий - окружить расчетную область поглощающим слоем конечной ширины. В результате все волны должны попадать в этот слой и поглощаться независимо от частоты и угла падения(пример такой постановки условий [6]).

### 1.3.2 Замыкание системы уравнений

К сожалению, указанные выше методы оказались недостаточно гибкими и содержали множество ограничений. Например, для постановки неотражающих условий необходимо иметь точное решение системы уравнений, которое для нелинейных уравнений нельзя найти в общем случае. Постановка поглощающих граничных условий же черевата неустойчивостью полученного

решения на длительных промежутках времени. В результате был предложен следующий способ.

Т.к. искусственные границы принадлежат расчетной области и тем самым на них выполняются уравнения теории мелкой воды, то для замыкания схемы (1.2.1)-(1.2.3) на границах и в углах можно произвести аппроксимацию этих уравнений внутрь расчетной области. В итоге получим замкнутую систему уравнений, которую и будем решать.

Т.о. аппроксимация на границе  $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$  расчетной области выглядит следующим образом:

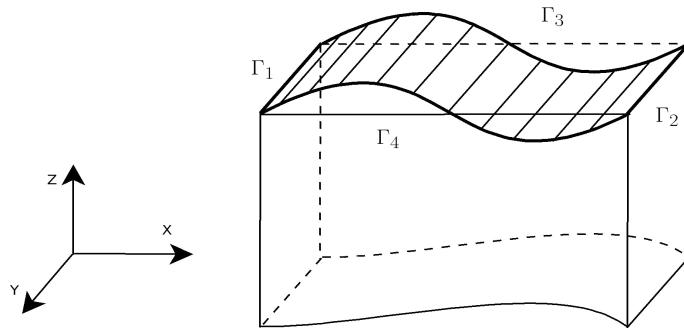


Рис. 1.2 Искусственные границы

- На границе  $\Gamma_1 = \{x = 0, y \in [0, 1]\}$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{1j} - B_{1j}^{n+1})u_{1j}^{n+1} - (H_{0j} - B_{0j}^{n+1})u_{0j}^{n+1}}{2h_x} + \\ \frac{(H_{0j+1} - B_{0j+1}^{n+1})v_{0j+1}^{n+1} - (H_{0j-1} - B_{0j-1}^{n+1})v_{0j-1}^{n+1}}{2h_y} \\ \frac{u_{0j}^{n+1} - u_{0j}^n}{\tau} + g \frac{\eta_{1j}^{n+1} - \eta_{0j}^{n+1}}{2h_x} \\ \frac{v_{0j}^{n+1} - v_{0j}^n}{\tau} + g \frac{\eta_{0j+1}^{n+1} - \eta_{0j-1}^{n+1}}{2h_y} \end{pmatrix} \quad (1.3.1)$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{0j} \frac{u_{0j+1}^{n+1} - u_{0j-1}^n}{2h_y} + \frac{(u_{1j}^{n+1})^2 - (u_{0j}^n)^2}{2h_x} \\ u_{0j} \frac{v_{1j}^{n+1} - v_{0j}^n}{2h_x} + \frac{(v_{0j+1}^{n+1})^2 - (v_{0j-1}^n)^2}{2h_y} \end{pmatrix} \quad (1.3.2)$$

- На границе  $\Gamma_2 = \{x = 1, y \in [0, 1]\}$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{nxj} - B_{nxj}^{n+1})u_{nxj}^{n+1} - (H_{nx-1j} - B_{nx-1j}^{n+1})u_{nx-1j}^{n+1}}{2h_x} + \\ \frac{(H_{nxj+1} - B_{nxj+1}^{n+1})v_{nxj+1}^{n+1} - (H_{nxj-1} - B_{nxj-1}^{n+1})v_{nxj-1}^{n+1}}{2h_y} \\ \frac{u_{nxj}^{n+1} - u_{nxj}^n}{\tau} + g \frac{\eta_{nxj}^{n+1} - \eta_{nx-1j}^{n+1}}{2h_x} \\ \frac{v_{nxj}^{n+1} - v_{nxj}^n}{\tau} + g \frac{\eta_{nxj+1}^{n+1} - \eta_{nxj-1}^{n+1}}{2h_y} \end{pmatrix} \quad (1.3.3)$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{nxj} \frac{u_{nxj+1}^{n+1} - u_{nxj-1}^n}{2h_y} + \frac{(u_{nxj}^{n+1})^2 - (u_{nx-1j}^n)^2}{2h_x} \\ u_{nxj} \frac{v_{nxj+1}^{n+1} - v_{nx-1j}^n}{2h_x} + \frac{(v_{nxj+1}^{n+1})^2 - (v_{nxj-1}^n)^2}{2h_y} \end{pmatrix} \quad (1.3.4)$$

- На границе  $\Gamma_3 = \{x = 0, y \in [0, 1]\}$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{i+10} - B_{i+10}^{n+1})u_{i+10}^{n+1} - (H_{i-10} - B_{i-10}^{n+1})u_{i-10}^{n+1}}{2h_x} + \\ \frac{(H_{i1} - B_{i1}^{n+1})v_{i1}^{n+1} - (H_{i0} - B_{i0}^{n+1})v_{i0}^{n+1}}{2h_y} \\ \frac{u_{i0}^{n+1} - u_{i0}^n}{\tau} + g \frac{\eta_{i+10}^{n+1} - \eta_{i-10}^{n+1}}{2h_x} \\ \frac{v_{i0}^{n+1} - v_{i0}^n}{\tau} + g \frac{\eta_{i1}^{n+1} - \eta_{i0}^{n+1}}{2h_y} \end{pmatrix} \quad (1.3.5)$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{i0} \frac{u_{i1}^{n+1} - u_{i0}^n}{2h_y} + \frac{(u_{i+10}^{n+1})^2 - (u_{i-10}^n)^2}{2h_x} \\ u_{i0} \frac{v_{i+10}^{n+1} - v_{i-10}^n}{2h_x} + \frac{(v_{i1}^{n+1})^2 - (v_{i0}^n)^2}{2h_y} \end{pmatrix} \quad (1.3.6)$$

- На границе  $\Gamma_4 = \{x = 0, y = 1\}$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{i+1ny} - B_{i+1ny}^{n+1})u_{i+1ny}^{n+1} - (H_{i-1ny} - B_{i-1ny}^{n+1})u_{i-1ny}^{n+1}}{2h_x} + \\ \frac{(H_{in_y} - B_{in_y}^{n+1})v_{in_y}^{n+1} - (H_{in_y-1} - B_{in_y-1}^{n+1})v_{in_y-1}^{n+1}}{2h_y} \\ \frac{u_{in_y}^{n+1} - u_{in_y}^n}{\tau} + g \frac{\eta_{i+1ny}^{n+1} - \eta_{i-1ny}^{n+1}}{2h_x} \\ \frac{v_{in_y}^{n+1} - v_{in_y}^n}{\tau} + g \frac{\eta_{in_y}^{n+1} - \eta_{in_y-1}^{n+1}}{2h_y} \end{pmatrix} \quad (1.3.7)$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{in_y} \frac{u_{i+1ny}^{n+1} - u_{in_y-1}^n}{2h_y} + \frac{(u_{i+1ny}^{n+1})^2 - (u_{i-1ny}^n)^2}{2h_x} \\ u_{in_y} \frac{v_{i+1ny}^{n+1} - v_{in_y-1}^n}{2h_x} + \frac{(v_{in_y}^{n+1})^2 - (v_{in_y-1}^n)^2}{2h_y} \end{pmatrix} \quad (1.3.8)$$

Аппроксимация в углах расчетной области выглядит следующим образом:

- $x = 0 \ y = 0$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{10}-B_{10}^{n+1})u_{10}^{n+1}-(H_{00}-B_{00}^{n+1})u_{00}^{n+1}}{2h_x} + \\ \frac{(H_{01}-B_{01}^{n+1})v_{01}^{n+1}-(H_{00}-B_{00}^{n+1})v_{00}^{n+1}}{2h_y} \\ \frac{u_{00}^{n+1}-u_{00}^n}{\tau} + g \frac{\eta_{10}^{n+1}-\eta_{00}^{n+1}}{2h_x} \\ \frac{v_{00}^{n+1}-v_{00}^n}{\tau} + g \frac{\eta_{01}^{n+1}-\eta_{00}^{n+1}}{2h_y} \end{pmatrix}$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{00} \frac{u_{01}^{n+1}-u_{00}^n}{2h_y} + \frac{(u_{10}^{n+1})^2-(u_{00}^n)^2}{2h_x} \\ u_{00} \frac{v_{10}^{n+1}-v_{00}^n}{2h_x} + \frac{(v_{01}^{n+1})^2-(v_{00}^n)^2}{2h_y} \end{pmatrix}$$

- $x = 1 \ y = 0$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{nx0}-B_{nx0}^{n+1})u_{nx0}^{n+1}-(H_{nx-10}-B_{nx-10}^{n+1})u_{nx-10}^{n+1}}{2h_x} + \\ \frac{(H_{nx1}-B_{nx1}^{n+1})v_{nx1}^{n+1}-(H_{nx0}-B_{nx0}^{n+1})v_{nx0}^{n+1}}{2h_y} \\ \frac{u_{nx0}^{n+1}-u_{nx0}^n}{\tau} + g \frac{\eta_{nx0}^{n+1}-\eta_{nx-10}^{n+1}}{2h_x} \\ \frac{v_{nx0}^{n+1}-v_{nx0}^n}{\tau} + g \frac{\eta_{nx1}^{n+1}-\eta_{nx0}^{n+1}}{2h_y} \end{pmatrix}$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{nx0} \frac{u_{nx1}^{n+1}-u_{nx0}^n}{2h_y} + \frac{(u_{nx0}^{n+1})^2-(u_{nx-10}^n)^2}{2h_x} \\ u_{nx0} \frac{v_{nx1}^{n+1}-v_{nx0}^n}{2h_x} + \frac{(v_{nx1}^{n+1})^2-(v_{nx0}^n)^2}{2h_y} \end{pmatrix}$$

- $x = 0 \ y = 1$

$$F_{lin}^{n+1} = \begin{pmatrix} \frac{(H_{1ny}-B_{1ny}^{n+1})u_{1ny}^{n+1}-(H_{0ny}-B_{0ny}^{n+1})u_{0ny}^{n+1}}{2h_x} + \\ \frac{(H_{0ny}-B_{0ny}^{n+1})v_{0ny}^{n+1}-(H_{0ny-1}-B_{0ny-1}^{n+1})v_{0ny-1}^{n+1}}{2h_y} \\ \frac{u_{0ny}^{n+1}-u_{0ny}^n}{\tau} + g \frac{\eta_{1ny}^{n+1}-\eta_{0ny}^{n+1}}{2h_x} \\ \frac{v_{0ny}^{n+1}-v_{0ny}^n}{\tau} + g \frac{\eta_{0ny}^{n+1}-\eta_{0ny-1}^{n+1}}{2h_y} \end{pmatrix}$$

$$F_{nonlin}^{n+1} = \begin{pmatrix} 0 \\ v_{0ny} \frac{u_{0ny}^{n+1}-u_{0ny-1}^n}{2h_y} + \frac{(u_{1ny}^{n+1})^2-(u_{0ny}^n)^2}{2h_x} \\ u_{0ny} \frac{v_{1ny}^{n+1}-v_{0ny}^n}{2h_x} + \frac{(v_{0ny}^{n+1})^2-(v_{0ny-1}^n)^2}{2h_y} \end{pmatrix}$$

$$\bullet \quad x = 1 \quad y = 1$$

$$F_{lin}^{n+1} = \left( \begin{array}{c} \frac{(H_{nxny} - B_{nxny}^{n+1})u_{nxny}^{n+1} - (H_{nx-1ny} - B_{nx-1ny}^{n+1})u_{nx-1ny}^{n+1}}{2h_x} + \\ \frac{(H_{nxny} - B_{nxny}^{n+1})v_{nxny}^{n+1} - (H_{nxny-1} - B_{nxny-1}^{n+1})v_{nxny-1}^{n+1}}{2h_y} \\ \frac{u_{nxny}^{n+1} - u_{nxny}^n}{\tau} + g \frac{\eta_{nxny}^{n+1} - \eta_{nx-1ny}^{n+1}}{2h_x} \\ \frac{v_{nxny}^{n+1} - v_{nxny}^n}{\tau} + g \frac{\eta_{nxny}^{n+1} - \eta_{nxny-1}^{n+1}}{2h_y} \end{array} \right)$$

$$F_{nonlin}^{n+1} = \left( \begin{array}{c} 0 \\ v_{ij} \frac{u_{ij+1}^{n+1} - u_{ij-1}^n}{2h_y} + \frac{(u_{i+1j}^{n+1})^2 - (u_{i-1j}^n)^2}{2h_x} \\ u_{ij} \frac{v_{i+1j}^{n+1} - v_{i-1j}^n}{2h_x} + \frac{(v_{ij+1}^{n+1})^2 - (v_{ij-1}^n)^2}{2h_y} \end{array} \right)$$

## § 1.4 Алгоритм решения

В итоге, схема (1.2.1)-(1.2.3) с замыканием на  $\Gamma$  представляет собой систему нелинейных алгебраических уравнений (СНАУ)

$$AU = f \quad A = A_{lin} + A_{nonlin} \quad (1.4.1)$$

относительно переменных  $U^n = (\eta_{00}^{n+1}, \dots, \eta_{n_x n_y}^{n+1}, u_{00}^{n+1}, \dots, u_{n_x n_y}^{n+1}, v_{00}^{n+1}, \dots, v_{n_x n_y}^{n+1} 0)$ , где вектор  $f$  известен и зависит от  $\eta_{ij}^{n+1}, u_{ij}^{n+1}, v_{ij}^{n+1}, H_{ij}, B_{ij}^{n+1}, B_{ij}^n$ , а  $A_{lin}$  - оператор аппроксимации линейной части (1.1.1)-(1.1.4),  $A_{nonlin}$  - оператор аппроксимации нелинейной части (1.1.1)-(1.1.4).

Для решения задачи (1.1.1)-(1.1.4) использовался следующий алгоритм:

- По известному вектору  $\{u_{ij}^n, v_{ij}^n, \eta_{ij}^n\}$  на каждом временном слое каким-либо итерационным методом находим приближенное решение системы (1.4.1).
- Оно принимается за решение схемы (1.2.1)-(1.2.3) на  $n+1$  слое.

Матрица полученной системы уравнений не обладает свойствами, необходимыми для решения системы известными методами, поэтому для ее решения мы воспользовались методом неполной аппроксимации минимальных невязок, шаг которого выглядит следующим образом:

$$u^{k+\frac{1}{2}} = u^k - \tau_{k+1} r^k \quad (1.4.2)$$

$$u^{k+1} = u^{k+\frac{1}{2}} - \alpha_{k+1} z^k \quad (1.4.3)$$

где  $\tau$  - итерационный параметр, выбираемый из условия минимума  $\|r^{k+\frac{1}{2}}\|, z^k$  - заданный вектор,  $\alpha$  - матрица итерационных параметров, элементы которой выбираются из минимума последовательности норм невязок.

## § 1.5 Описание двухслойной схемы неполной аппроксимации

Пусть в конечномерном вещественном гильбертовом пространстве  $H_m$  задана система уравнений

$$Au = f \quad (1.5.1)$$

с невырожденной незнакоопределенной матрицей  $A = (a_{ij})$  порядка  $m$ . Здесь  $u, f$  - неизвестный и известный  $m$ -мерные векторы из  $H_m$ .

В дальнейшем будем считать, что  $\det A$  или одно из собственных чисел могут быть близкими к машинному нулю. Если матрица незнакоопределена, то, умножая систему (1.5.1) на  $A^*$  (1-ая трансформация Гаусса), получим новую систему с положительно определенной матрицей  $\bar{A} = A^*A$  и правой частью  $A^*f$ . Для решения этой системы можно использовать достаточно широкий арсенал итерационных схем, но в нашем случае такой путь решения системы (1.5.1) не приемлем, т.к.  $\det A$  или одно из собственных чисел могут быть близкими к машинному нулю.

Для решения системы (1.5.1) рассмотрим итерационную схему неполной аппроксимации (более подробное описание метода см. [7])

$$u^{n+\frac{1}{2}} = u^n - \tau_{n+1}B(Au^n - f) \quad (1.5.2)$$

$$u^{n+1} = u^{n+\frac{1}{2}} - \alpha_{n+1}z^n, \quad n = 0, 1, 2, \dots \quad (1.5.3)$$

где  $B$  - неособенная квадратная матрица,  $z^n \in H_m$  - произвольный вектор,  $\tau_{n+1}, \alpha_{n+1}$  - итерационные параметры,  $u^0$  - произвольный начальный вектор. Заметим, что  $\alpha_{n+1}$  может быть как константой, так и матрицей.

### 1.5.1 Случай, когда $\alpha_{n+1} = const$

Пусть в схеме (1.5.2)-(1.5.3)  $\alpha_{n+1} = const$ , а  $D$  - самосопряженный положительно определенный оператор, действующий в  $H_m$ . Тогда относительно  $v^n = D^{\frac{1}{2}}(u^n - u)$  система имеет вид

$$v^{n+\frac{1}{2}} = S_{n+1}v^n \quad (1.5.4)$$

$$v^{n+1} = v^{n+\frac{1}{2}} - \alpha_{n+1}D^{\frac{1}{2}}z^n, \quad n = 0, 1, 2, \dots \quad (1.5.5)$$

где  $S_{n+1} = E - \tau_{n+1}C$ ,  $C = D^{\frac{1}{2}}z^n$ ,  $v^{n+\frac{1}{2}} = D^{\frac{1}{2}}(u^{n+\frac{1}{2}} - u)$ .

Тогда

$$\|v^{n+1}\|^2 = \|v^{n+\frac{1}{2}}\|^2 - 2\alpha_{n+1}(v^{n+\frac{1}{2}}, D^{\frac{1}{2}}z^n) + \alpha_{n+1}^2 \|D^{\frac{1}{2}}z^n\|^2 \quad (1.5.6)$$

Если вектор  $z^n$  такой, что  $D^{\frac{1}{2}}z^n \neq 0$ , то, выбирая  $\alpha_{n+1}$  из условия минимума  $\|v^{n+1}\|^2$ , получаем

$$\|v^{n+1}\|^2 = \|v^{n+\frac{1}{2}}\|^2 - \frac{(v^{n+\frac{1}{2}}, D^{\frac{1}{2}}z^n)^2}{\|D^{\frac{1}{2}}z^n\|^2} \quad (1.5.7)$$

при

$$\alpha_{n+1} = \frac{(v^{n+\frac{1}{2}}, D^{\frac{1}{2}}z^n)^2}{\|D^{\frac{1}{2}}z^n\|^2} \quad (1.5.8)$$

Параметр  $\tau_{n+1}$ , например, можно выбирать из условия минимума  $\|v^{n+\frac{1}{2}}\|^2$

$$\|v^{n+\frac{1}{2}}\|^2 = \|v^n\|^2 - 2\tau_{n+1}(v^n, Cv^n) + \tau_{n+1}^2 \|Cv^n\|^2 \quad (1.5.9)$$

тогда

$$\tau_{n+1} = \frac{(Cv^n, v^n)}{\|Cv^n\|^2} \quad (1.5.10)$$

и формулу (1.5.7) можно пререписать в виде

$$\|v^{n+1}\|^2 = \Theta_{n+1}^{(1)} \|v^n\|^2 \quad (1.5.11)$$

где

$$\Theta_{n+1}^{(1)} = \rho_{n+1}^2 - R_{n+1} \leq 1 \quad (1.5.12)$$

$$R_{n+1} = \frac{(v^{n+\frac{1}{2}}, D^{\frac{1}{2}}z^n)^2}{\|D^{\frac{1}{2}}z^n\|^2 \|v^n\|^2} \quad (1.5.13)$$

$$\rho_{n+1}^2 = 1 - \frac{(Cv^n, v^n)^2}{\|Cv^n\|^2 \|v^n\|^2} \quad (1.5.14)$$

Справедлива

**Теорема.** Если для всех  $n$  вектор  $z^n$  такой, что  $(D^{\frac{1}{2}}v^{n+\frac{1}{2}}, z^n) \neq 0$ , то итерационная схема (1.5.6),

(1.5.8), (1.5.10) сходится, т.е.  $\|v^{n+1}\| \rightarrow 0$  при  $n \rightarrow \infty$  для любого  $v_0$

В [7] показано, что строить векторы  $z^n$  такие, чтобы для величины  $\Theta_n^{(1)}$  выполнялась оценка  $\Theta_n^{(1)} \leq \Theta_* < 1$ , достаточно просто.

**Теорема.** Итерационная схема (1.5.6), (1.5.8), (1.5.10) является сходящейся, если задать вектор  $z^n$  в виде

$$z^n = x^n + \frac{\sigma}{(D^{\frac{1}{2}}v^{\frac{1}{2}}, v^{\frac{1}{2}})} v^{n+\frac{1}{2}}$$

со следующей оценкой:

$$\|v^{n+1}\|^2 \leq (1 - \xi)\rho_{n+1}^2 \|v^n\|^2 \quad (1.5.15)$$

где  $\xi = \frac{\gamma_1}{\gamma_2}$ , а  $\rho_{n+1}^2$  определяется равенством (1.5.14)

### 1.5.2 Случай, когда $\alpha_{n+1}$ является матрицей

Для схемы (1.5.2)-(1.5.3) введем параметр  $\alpha_{n+1}$  в виде диагональной матрицы размерности  $m$ .

Пусть  $z^n$  - вектор, все элементы которого равны единице, тогда произведение  $\alpha_{n+1}z^n$  можно представить в виде

$$\alpha_{n+1}z^n = \sum_{i=1}^m \alpha_{n+1}^i z_i^n \quad (1.5.16)$$

где  $\alpha_{n+1}^i$  - элемент главной диагонали матрицы  $a_{n+1}$ , стоящий в  $i$ -й строке,  $z_i^n$  - вектор с ненулевой  $i$ -й компонентой. Следовательно,

$$v^{n+1} = v^{n+\frac{1}{2}} - \sum_{i=1}^m \alpha_{n+1}^i D^{\frac{1}{2}} z_i^n \quad (1.5.17)$$

Будем находить элементы матрицы  $\{\alpha_{n+1}^i\}$  из условия минимума  $\|v^{n+1}\|$

$$\|v^{n+1}\|^2 = \|v^{n+\frac{1}{2}}\|^2 - \sum_{i=1}^m \alpha_{n+1}^i (v^{n+\frac{1}{2}}, D^{\frac{1}{2}} z_i^n) + \sum_{i=1}^m \sum_{j=1}^m \alpha_{n+1}^i \alpha_{n+1}^j (D^{\frac{1}{2}} z_i^n, D^{\frac{1}{2}} z_j^n) \quad (1.5.18)$$

Для функций многих переменных справедлива

**Теорема.** Если  $f(x) \in C^2(R^m)$ , где  $x \in R^m$ , и выполнено условие

$$\frac{\partial f}{\partial x_i}(x_0) = 0 \quad (1.5.19)$$

а также  $d_{xx}^2 f(x_0)$  - матрица частных производных второго порядка строго положительно определена, то точка  $x_0$  является точкой глобального минимума

В [7] показано, что все условия теоремы выполняются, т.к. по условию матрица  $D$  положительно определенная и неособенная, а векторы  $\{z_m^n\}$  по построению линейно независимы.

### 1.5.3 Метод минимальных невязок

Пусть  $D = A^*A$ , тогда (1.5.5) перепишется в виде

$$v^{n+1} = v^{n+\frac{1}{2}} - \alpha_{n+1}(A^*A)^{\frac{1}{2}}z^n, \quad n = 0, 1, 2, \dots \quad (1.5.20)$$

и минимизируя  $\|r^{n+1}\|$ , получим

$$\alpha_{n+1} = \frac{(r^{n+\frac{1}{2}}, Az^n)}{\|Az^n\|^2} \quad (1.5.21)$$

Если же  $\alpha_{n+1}$  - диагональная матрица, то ее элементы будем находить как решение системы

$$\begin{pmatrix} (Az_1^n, Az_1^n) & (Az_1^n, Az_2^n) & \cdots & (Az_1^n, Az_m^n) \\ (Az_2^n, Az_1^n) & (Az_2^n, Az_2^n) & \cdots & (Az_2^n, Az_m^n) \\ \vdots & \vdots & & \vdots \\ (Az_m^n, Az_1^n) & (Az_m^n, Az_2^n) & \cdots & (Az_m^n, Az_m^n) \end{pmatrix} \times \begin{pmatrix} \alpha_{n+1}^1 \\ \alpha_{n+1}^2 \\ \vdots \\ \alpha_{n+1}^m \end{pmatrix} = \begin{pmatrix} (Az_1^n, r^n) \\ (Az_2^n, r^n) \\ \vdots \\ (Az_m^n, r^n) \end{pmatrix}$$

## Глава 2. Результаты расчетов

В главе приведены результаты серии расчетов движения волн при различных параметрах - например, шаг расчетной сетки, начальные данные, форма дна и т.д.

Для того, чтобы продемонстрировать пригодность метода решения, описанного в разделе §1, проводились расчеты с разной начальной поверхностью и над разным дном. В них проверялось, насколько «хорошо» волна будет выходить из расчетной области, появятся ли искажения или отражения.

После этого проводились расчеты, направление на сравнение линейных ( $F_{nonlin} = 0$ ) и нелинейных ( $F_{nonlin} \neq 0$ ) уравнений (1.1.1). Данные, полученные для линейного и нелинейного случая, анализировались и сравнивались, для выявления качественных различий. Для удобства восприятия, результаты обрабатывались отдельным модулем программы и визуализировались.

Помимо этого отдельно проводились расчеты для исследования явления «возникновения волны».

Все расчеты представлены в двух проекциях - стандартной и вид сбоку. Сеточная поверхность, расположенная под свободной поверхностью - дно, над которым движется жидкость.

## § 2.1 Расчет со сложной начальной формой волны

Рассмотрим задачу о движении начальной волны следующего вида:

$$u_0 = 0 \quad v_0 = 0 \quad \eta_0 = 7 \cdot 10^{-3} e^{(-100((x-0.5)^2 + (y-0.5)^2))} + 4 \cdot 10^{-3} e^{(-300((x-0.7)^2 + (y-0.7)^2))}$$

в единичном квадрате глубиной  $H(x, y) = 10^{-2}$ .

Таблица 2.1 Параметры для расчета со сложной формой поверхности

|                  |                                       |
|------------------|---------------------------------------|
| Размер области   | $1 \times 1$                          |
| Количество шагов | 300                                   |
| Шаг по времени   | 0.01                                  |
| Шаг сетки        | 0.01                                  |
| Точность метода  | $10^{-7}$                             |
| Форма дна        | $H(x, y) = 0.01 \quad B(x, y, t) = 0$ |

На рис.2.1 представлена динамика выхода волны из области решения.

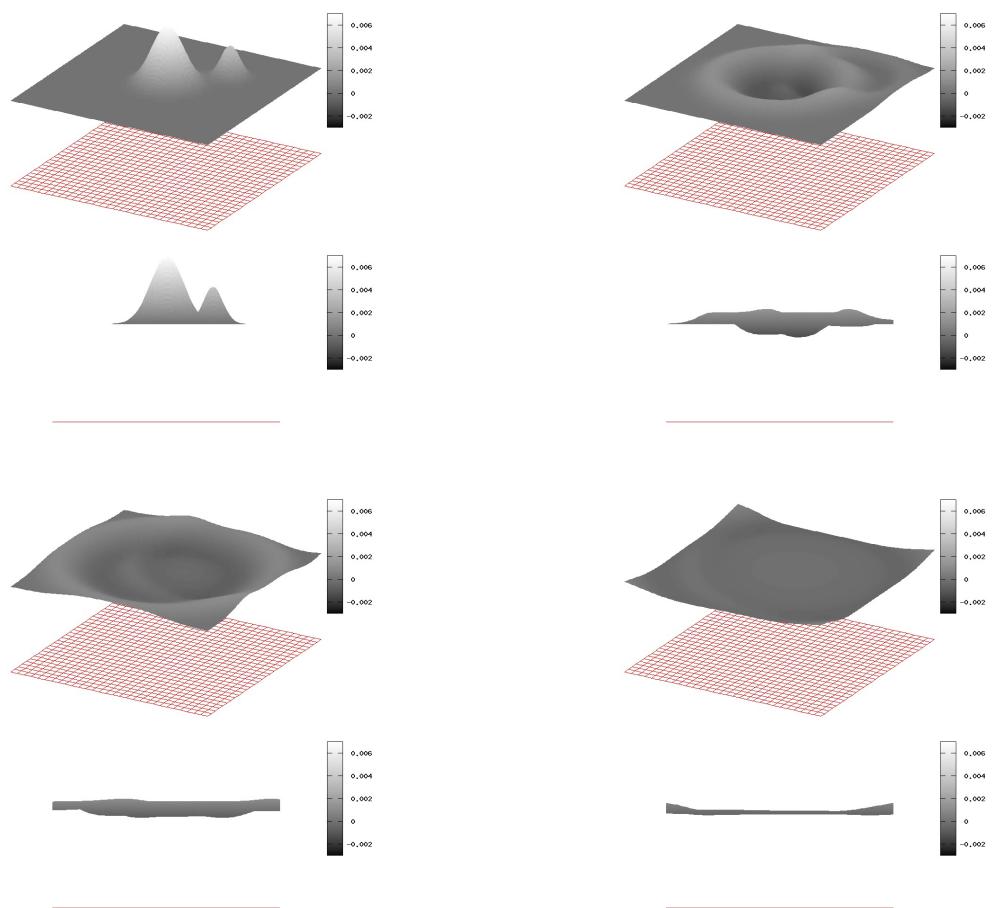


Рис. 2.1 Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) со сложной начальной формой на нулевом, 80, 180 и 280 шаге

Рассмотрим задачу о движении начальной волны следующего вида:

$$u_0 = 0 \quad v_0 = 0 \quad \eta_0 = 4 \cdot 10^{-3} e^{-300((x-0.3)^2)} + 4 \cdot 10^{-3} e^{(-300((x-0.7)^2+(y-0.7)^2))}$$

в единичном квадрате глубиной  $H(x, y) = 10^{-2}$ .

Таблица 2.2 Параметры для расчета со сложной формой поверхности

|                  |                                       |
|------------------|---------------------------------------|
| Размер области   | $1 \times 1$                          |
| Количество шагов | 300                                   |
| Шаг по времени   | 0.01                                  |
| Шаг сетки        | 0.01                                  |
| Точность метода  | $10^{-7}$                             |
| Форма дна        | $H(x, y) = 0.01 \quad B(x, y, t) = 0$ |

На рис.2.2 представлена динамика выхода волны из области решения.

Как видно из представленных расчетов, предлагаемый подход к решению задачи (1.1.1)-(1.1.4) позволяет находить в ней решение без отражений или искажений. Волна, движущаяся в расчетной области, проходит через искусственные границы корректно.

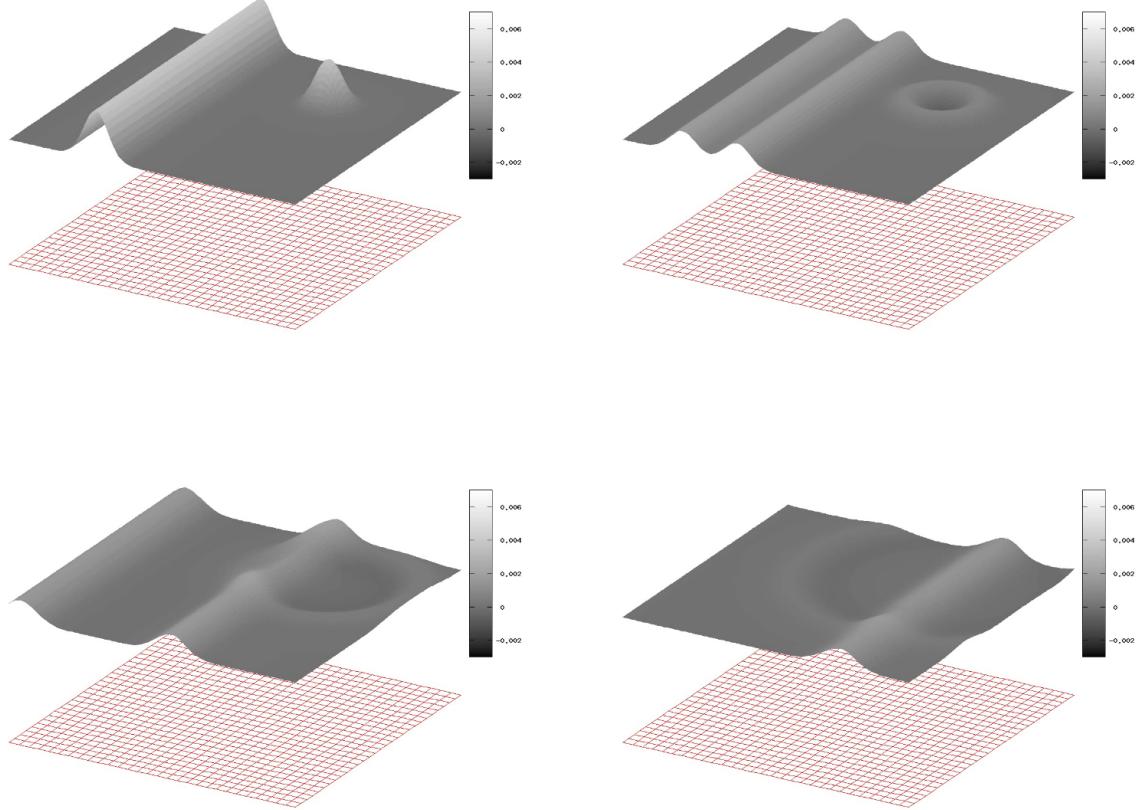


Рис. 2.2 Результат расчета для нелинейной системы ( $F_{nonlin} \neq 0$ ) со сложной начальной формой (длинная волна и «капля») на нулевом, 80, 180 и 280 шаге

## § 2.2 Расчет движения над сложным дном

Рассмотрим задачу о движении начальной волны следующего вида:

$$u_0 = 0 \quad v_0 = 0 \quad \eta_0 = 4 \cdot 10^{-3} e^{(-100(x-0.7)^2)}$$

в единичном квадрате глубиной  $H(x, y) = 10^{-2}$ .

Таблица 2.3 Параметры для расчета с коническим выступом

|                  |   |
|------------------|---|
| Размер области   | $1 \times 1$  |
| Количество шагов | 1200  |
| Шаг по времени   | 0.001   |
| Шаг сетки        | 0.01  |
| Точность метода  | $10^{-6}$   |
| Форма дна        | $H(x, y) = 0.01, \sqrt{(x-5)^2 + (y-0.5)^2} > 0.05$<br>$H(x, y) = 0.005, \sqrt{(x-5)^2 + (y-0.5)^2} < 0.05$<br>$B(x, y, t) = 0$ |

На рис.2.3 представлена динамика выхода волны из области решения в двух проекциях для каждого шага.

На рис.2.4 представлен тот же расчет в проекции свеху.

Видно, что волна, наталкиваясь на препятствие, замедляется в центре, откуда затем по поверхности начинают распространяться колебания.

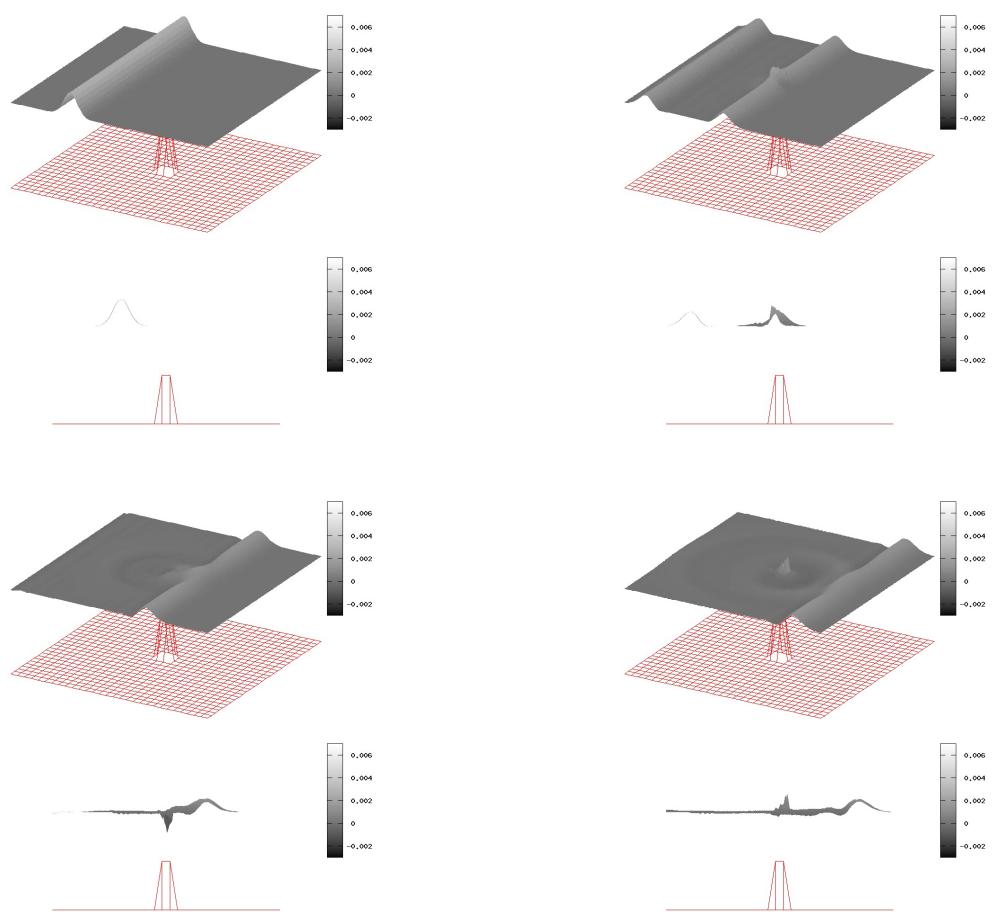


Рис. 2.3 Результат расчета для нелинейной системы ( $F_{nonlin} \neq 0$ ) с коническим выступом дна на нулевом, 400, 800 и 1200 шаге

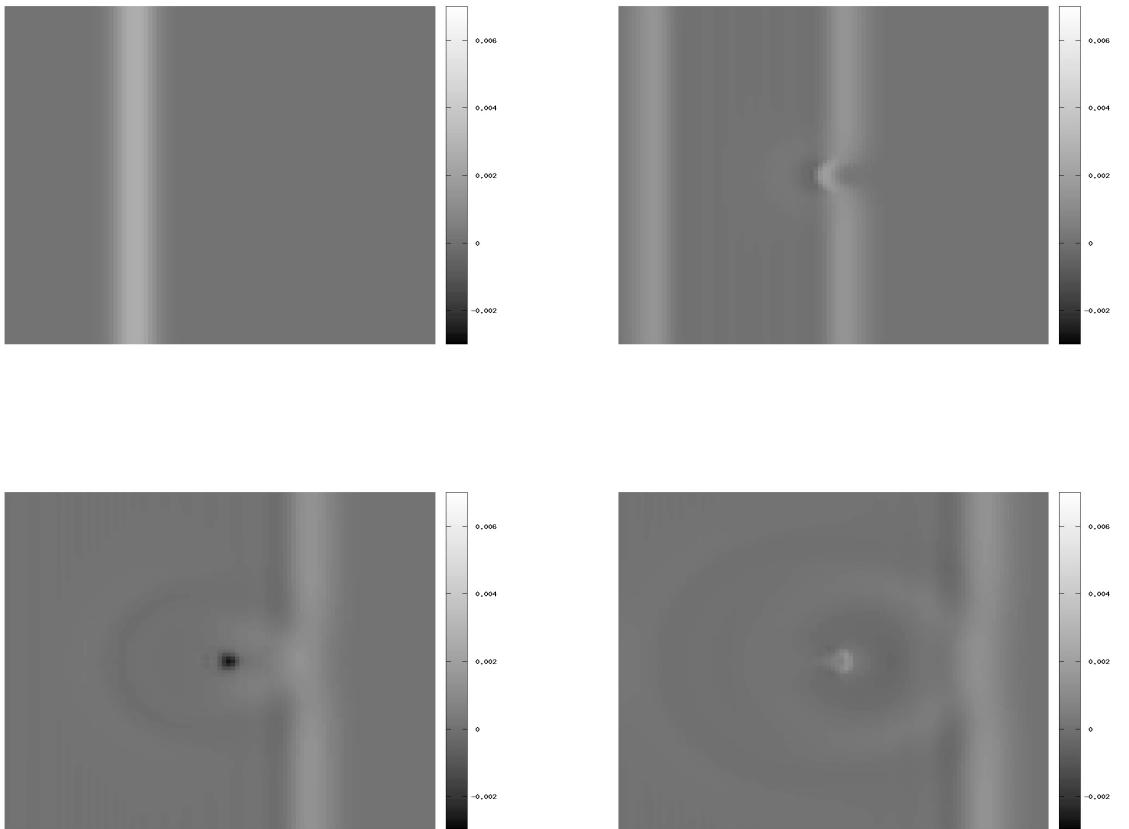


Рис. 2.4 Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) с коническим выступом дна на нулевом, 400, 800 и 1200 шаге (проекция сверху)

Рассмотрим задачу о движении начальной волны следующего вида:

$$u_0 = 0 \quad v_0 = 0 \quad \eta_0 = 7 \cdot 10^{-3} e^{(-100(x-0.7)^2)}$$

в единичном квадрате глубиной  $H(x, y) = 10^{-2}$ .

Таблица 2.4 Параметры для расчета с экспоненциальным выступом

|                  |  |
|------------------|--|
| Размер области   | $1 \times 1$   |
| Количество шагов | 300  |
| Шаг по времени   | 0.001  |
| Шаг сетки        | 0.01   |
| Точность метода  | $10^{-7}$  |
| Форма дна        | $H(x, y) = 0.01 - 3 \cdot 10^{-3} e^{(-100(x-0.7)^2)}$<br>$B(x, y, t) = 0$ |

На рис.2.5 представлена динамика выхода волны из области решения в двух проекциях для каждого шага.

Из расчетов, представленных в этом параграфе, видно, что предлагаемый подход к решению задачи (1.1.1)-(1.1.4) над неровным дном также позволяет находить решение без искажений или отражений. Волна, движущаяся в расчетной области, по-прежнему корректно проходит через искусственные границы, соответствующим образом изменяя свою форму над неровностями дна (например, замедляясь на подъемах).

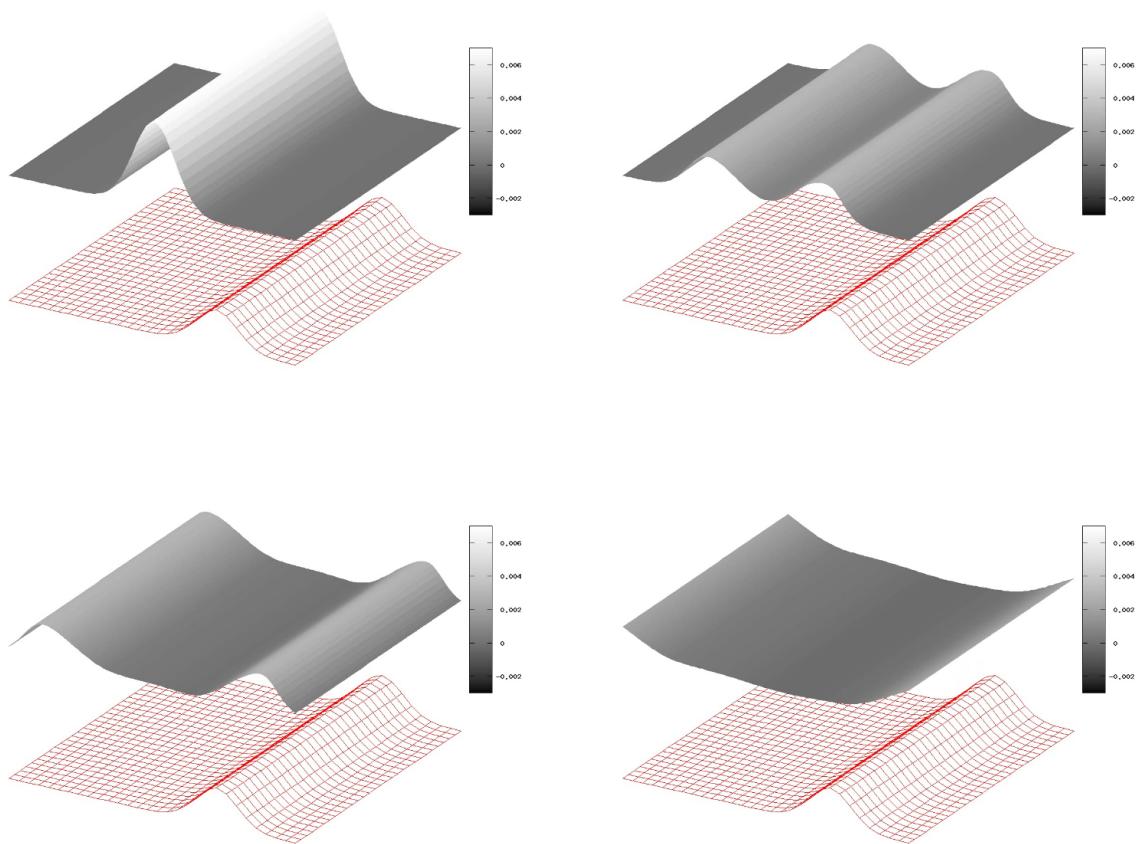


Рис. 2.5 Результат расчета для нелинейной системы ( $F_{nonlin} \neq 0$ ) с экспоненциальным выступом дна на нулевом, 80, 180 и 280 шаге

## § 2.3 Сравнение линейных и нелинейных уравнений

Рассмотрим задачу о движении начальной волны следующего вида:

$$u_0 = 0 \quad v_0 = 0 \quad \eta_0 = 7 \cdot 10^{-3} e^{(-100(x-0.5)^2)}$$

в единичном квадрате глубиной  $H(x, y) = 10^{-2}$ .

Таблица 2.5 Параметры для расчетов со сравнением линейных и нелинейных уравнений

|                  |                                       |
|------------------|---------------------------------------|
| Размер области   | $1 \times 1$                          |
| Количество шагов | 300                                   |
| Шаг по времени   | 0.01                                  |
| Шаг сетки        | 0.01                                  |
| Точность метода  | $10^{-7}$                             |
| Форма дна        | $H(x, y) = 0.01 \quad B(x, y, t) = 0$ |

На рис.2.6 представлена динамика выхода волны из области решения в случае линейных уравнений.

На рис.2.7 представлена динамика выхода волны из области решения в случае нелинейных уравнений.

На рис.2.8 представлена динамика разности этих расчетов.

Заметно, что расчет с нелинейными уравнениями отличается от аналогичного с линейными уравнениями. В линейном случае волна более симметрична относительно своей оси, в нелинейном - форма волны наклоняется в сторону движения.

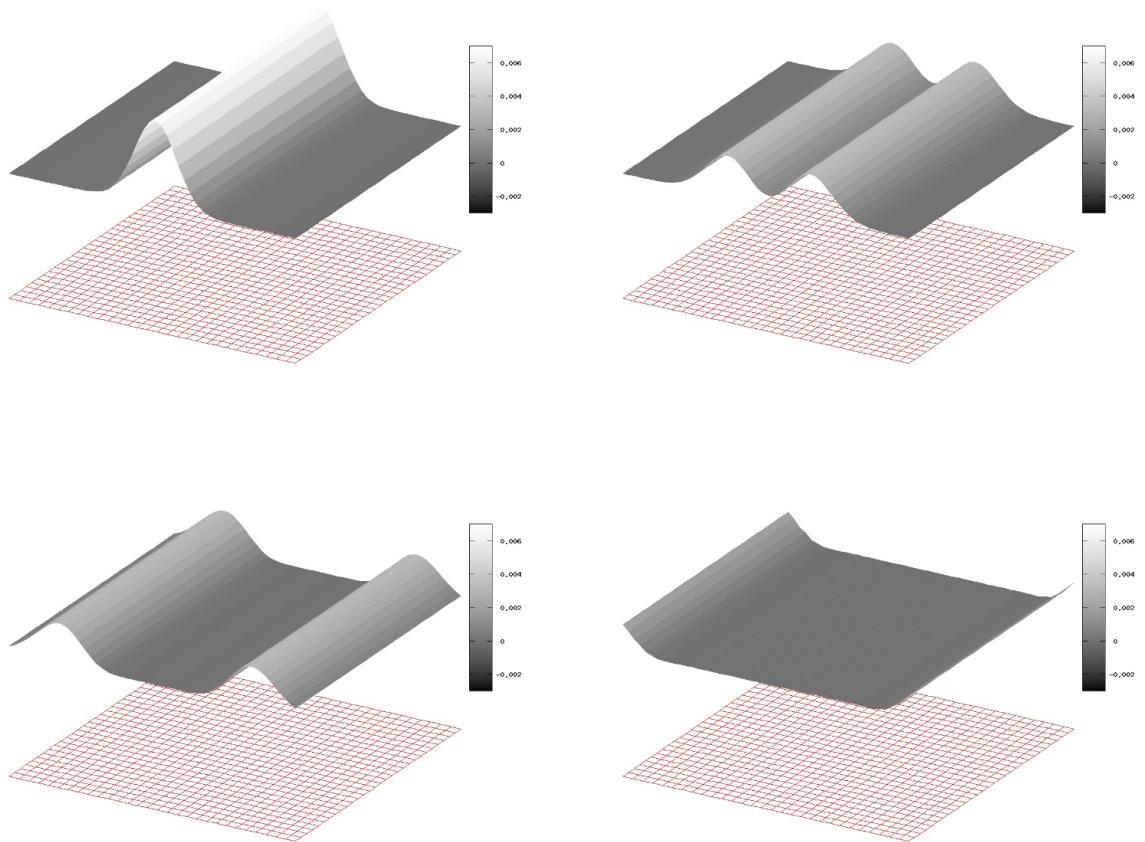


Рис. 2.6 Результат расчета для линейной системы ( $F_{nonlin} = 0$ ) с ровным дном на нулевом, 80, 180 и 280 шаге

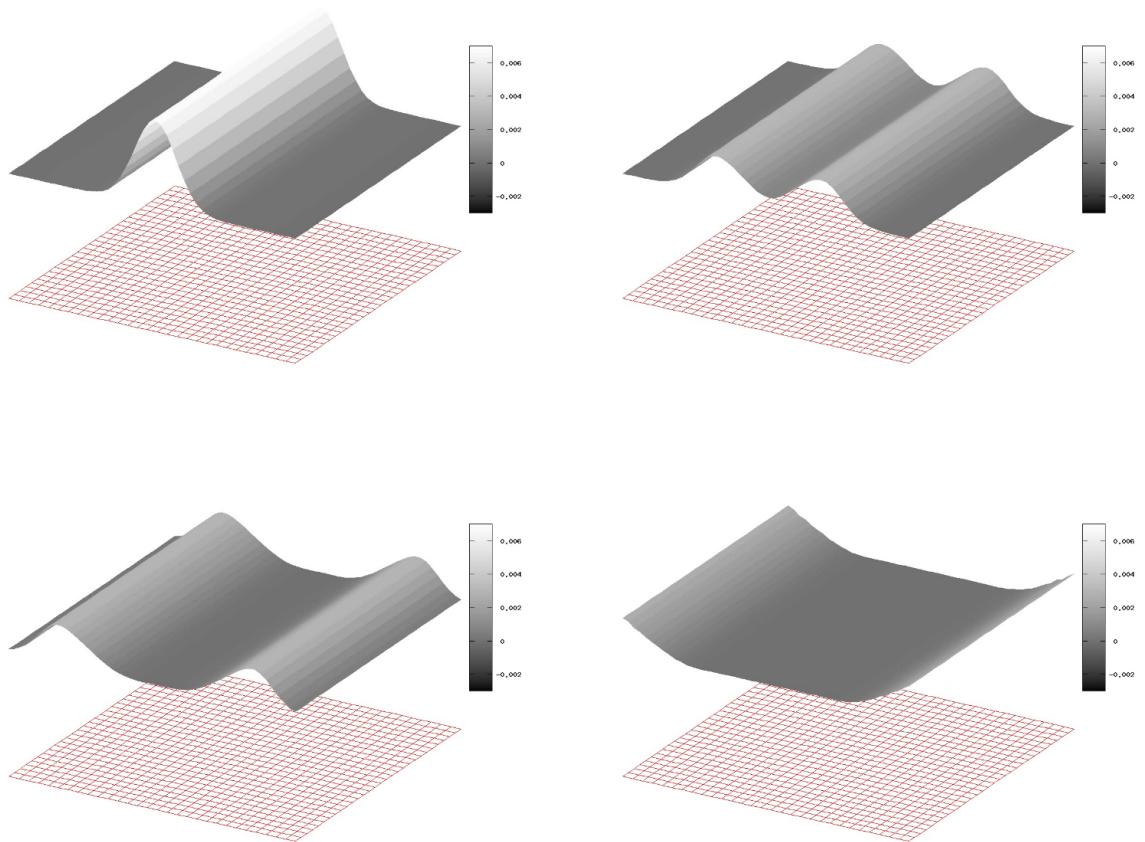


Рис. 2.7 Результат расчета для нелинейной системы ( $F_{nonlin} \neq 0$ ) с ровным дном на нулевом, 80, 180 и 280 шаге

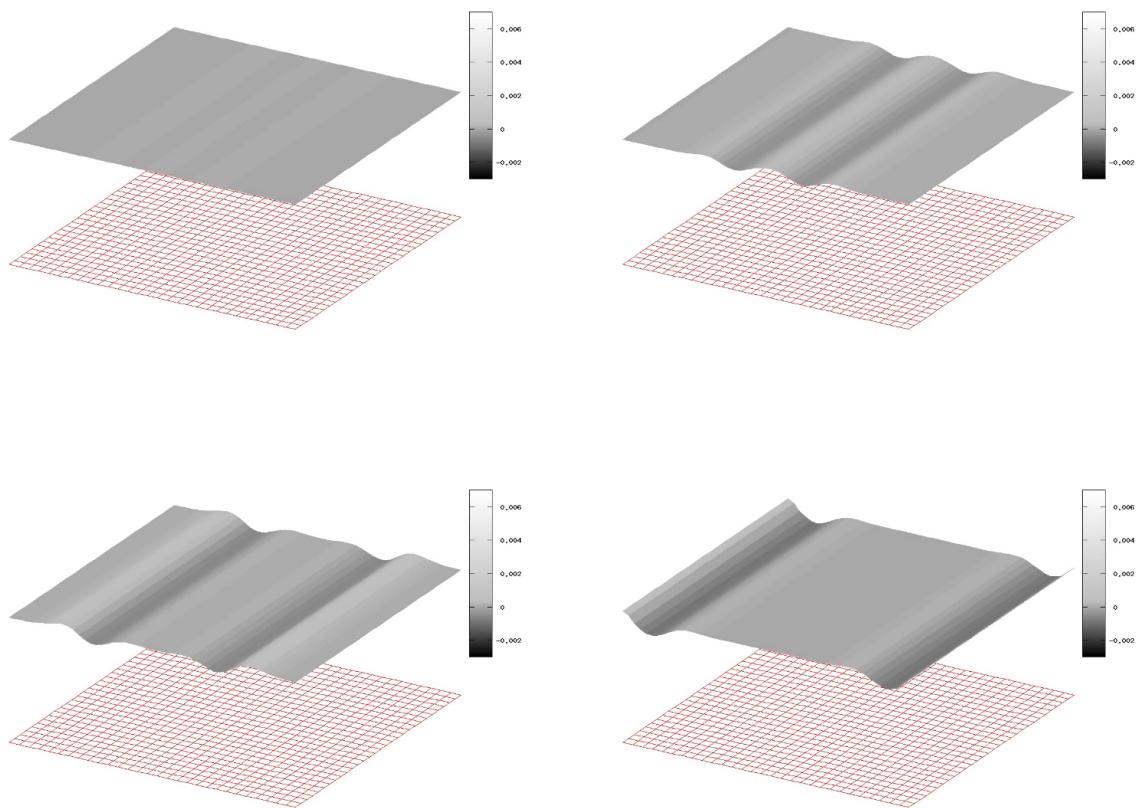


Рис. 2.8 Динамика разности расчетов в линейном и нелинейном случае с ровным дном на нулевом, 80, 180 и 280 шаге

Рассмотрим задачу о движении начальной волны следующего вида:

$$u_0 = 0 \quad v_0 = 0 \quad \eta_0 = 7 \cdot 10^{-3} e^{(-100(x-0.5)^2)}$$

в единичном квадрате глубиной  $H(x, y) = 10^{-2}$ .

Таблица 2.6 Параметры для расчетов со сравнением линейных и нелинейных уравнений в случае дна с препятствием на границе

|                  |   |
|------------------|---|
| Размер области   | $1 \times 1$  |
| Количество шагов | 300   |
| Шаг по времени   | 0.01  |
| Шаг сетки        | 0.01  |
| Точность метода  | $10^{-7}$   |
| Форма дна        | $H(x, y) = 10^{-2} - 3 \cdot 10^{-3} e^{(-100(x-0.7)^2)}$<br>$B(x, y, t) = 0$ |

На рис.2.9 представлена динамика выхода волны из области решения в случае линейных уравнений.

На рис.2.10 представлена динамика выхода волны из области решения в случае нелинейных уравнений.

На рис.2.11 представлена динамика разности этих расчетов в проекции сверху.

Исходя из представленных расчетов для линейных и нелинейных уравнений, можно отметить, что в нелинейном случае:

- Волна со временем меняет свою форму. Она становится немного «наклоненной» вперед.
- После прохождения основной волны за ней тянется «хвост» из небольших колебаний.
- С течением времени данные эффекты проявляются более явно.

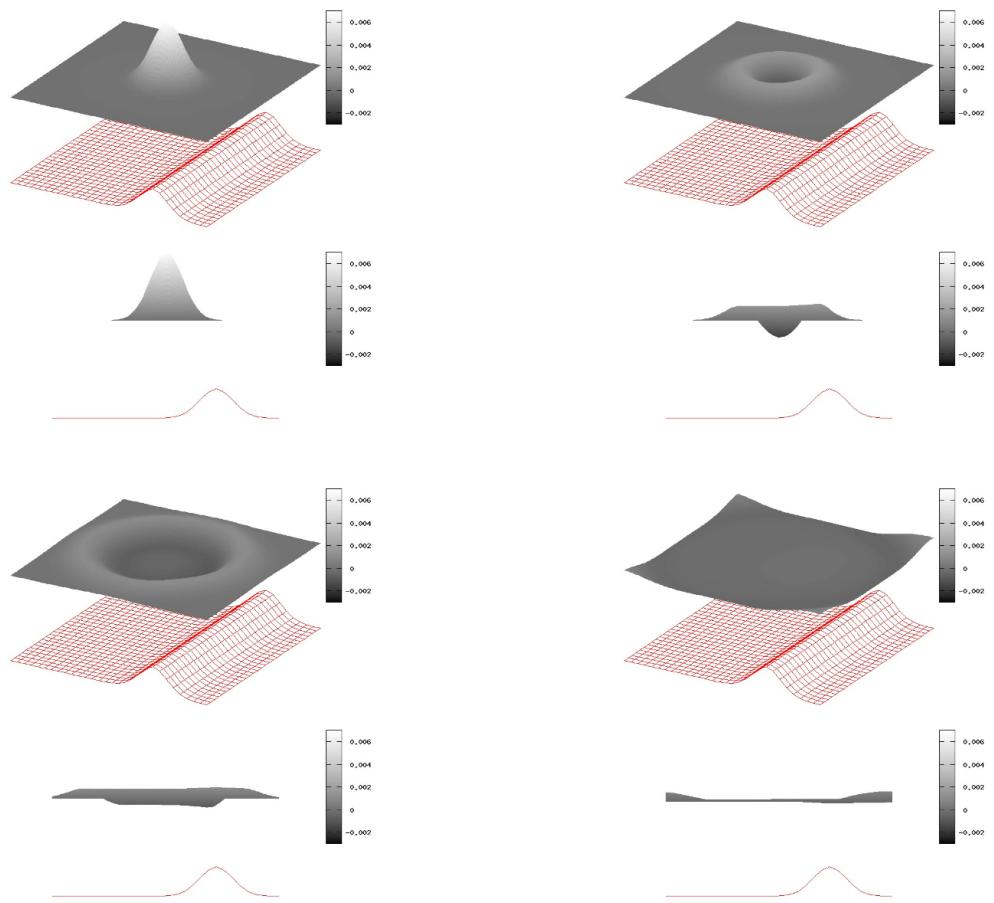


Рис. 2.9 Результат расчета для линейной системы( $F_{nonlin} = 0$ ) с препятствием у границы на нулевом, 80, 200 и 380 шаге

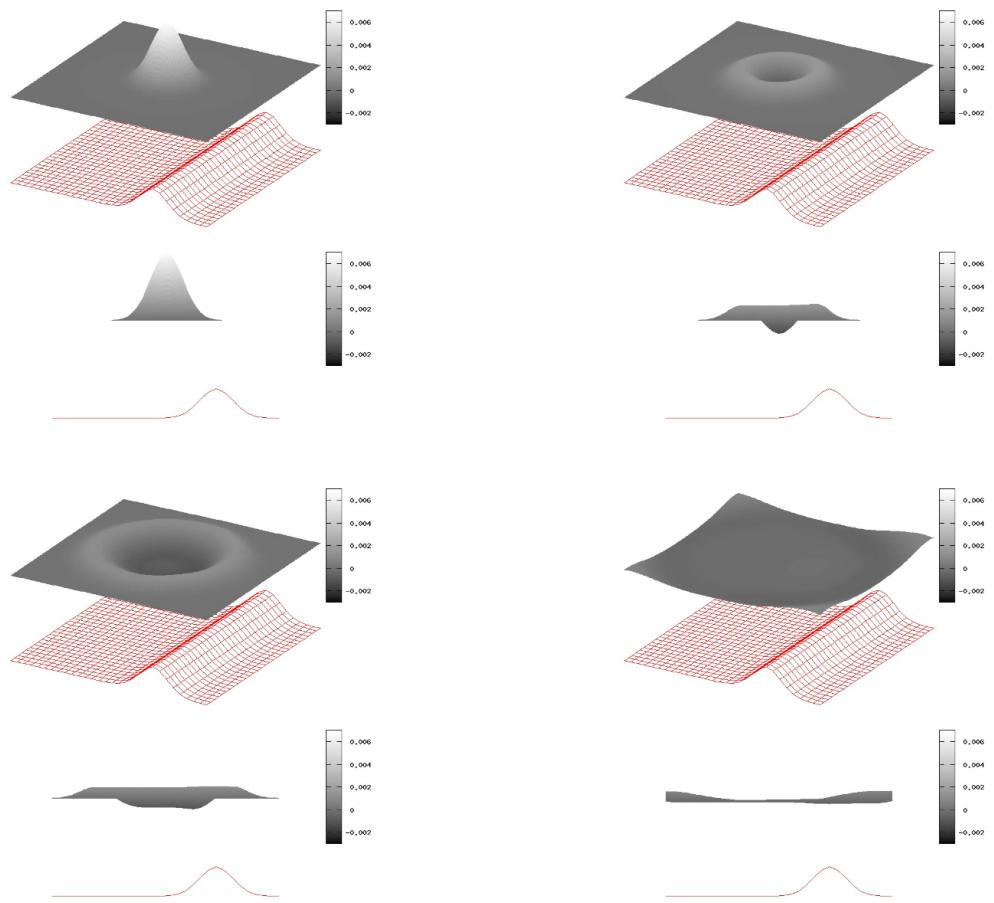


Рис. 2.10 Результат расчета для нелинейной системы ( $F_{nonlin} \neq 0$ ) с препятствием у границы на нулевом, 80, 200 и 380 шаге

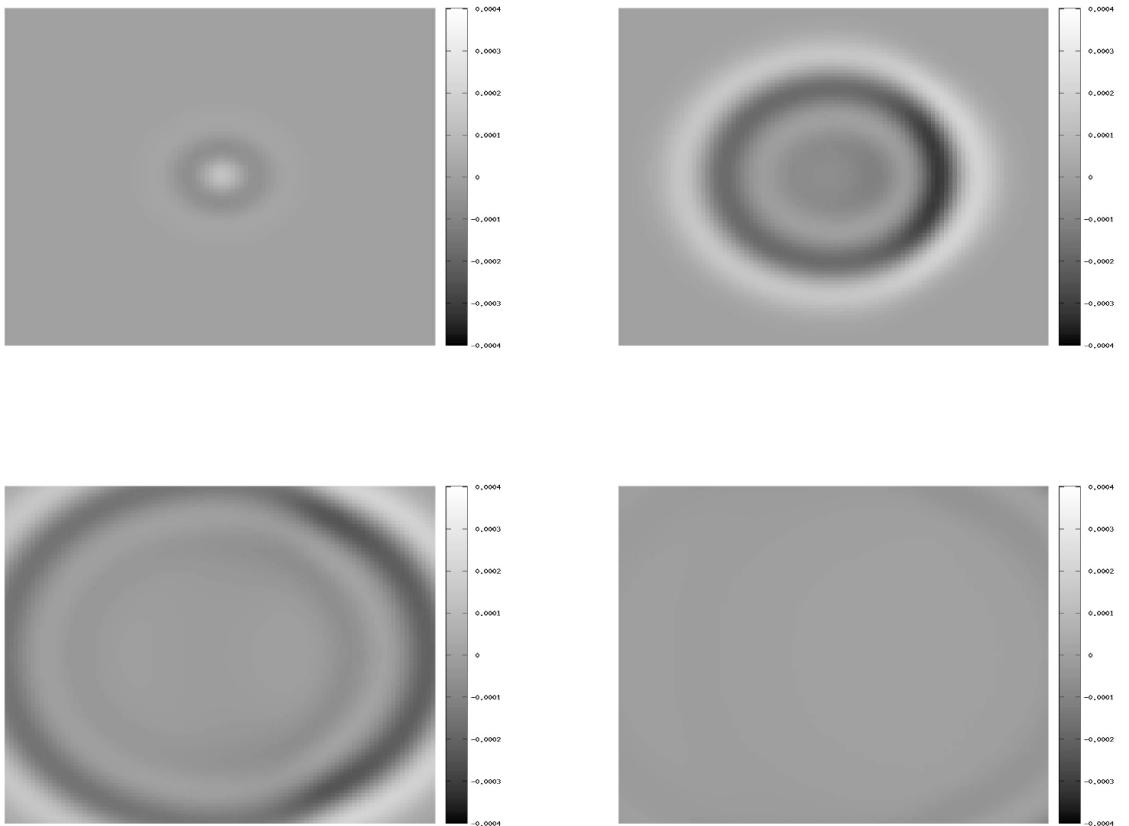


Рис. 2.11 Динамика разности расчетов в линейном и нелинейном случае с препятствием у границы на 20, 80, 200 и 380 шаге

## § 2.4 Эффект возникновения волны

В процессе проведения расчетов был обнаружен любопытный эффект, который ранее нигде встречался. Данный эффект достаточно устойчив, для того, чтобы не списать его на ошибки или неточности, поэтому привлек внимание.

После того как волна полностью выходит из расчетной области, в ней остаются некоторые регулярные колебания, волны, амплитуда которых меньше, чем исходная. Если продолжить расчет в этой ситуации, либо взять указанные колебания в качестве начальных данных для нового расчета, то произойдет «возвращение» волны - т.е. свободная поверхность начинает изменяться и принимает форму, обратную той, что была задана в качестве начального положения.

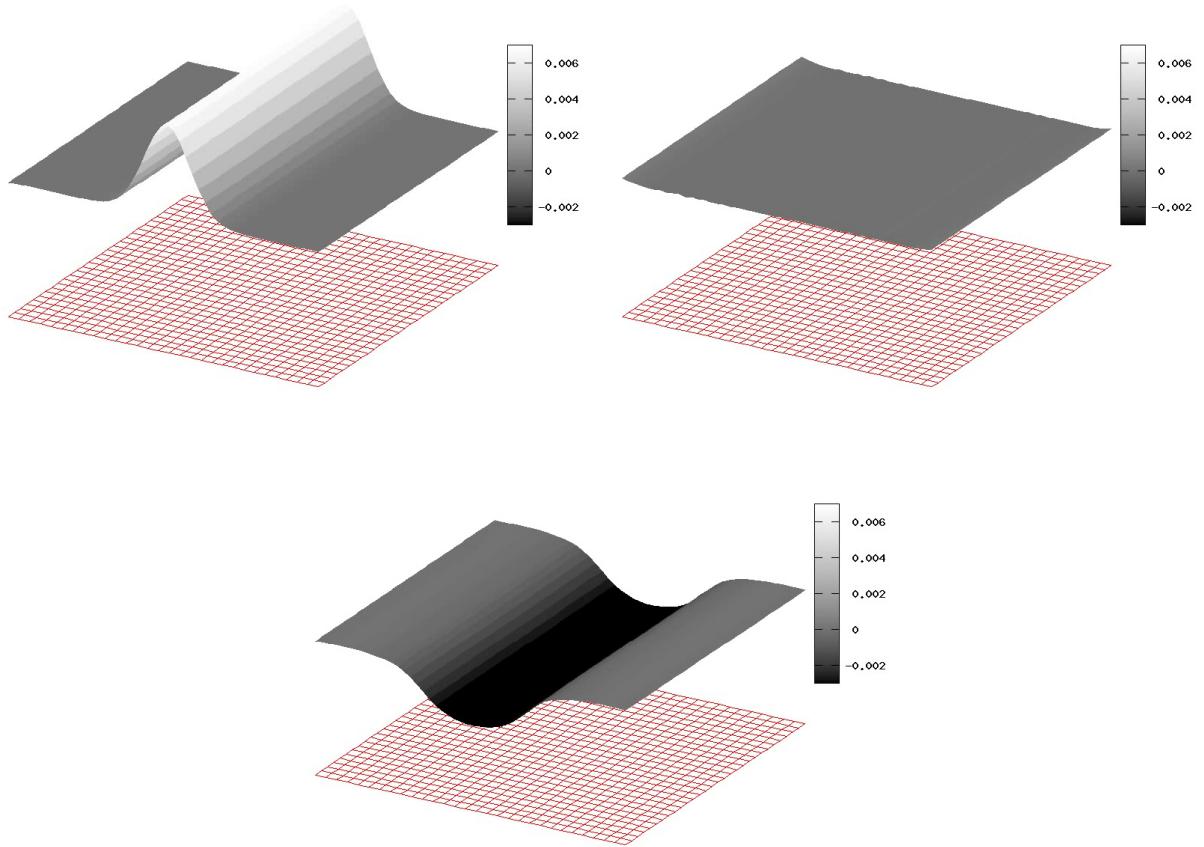


Рис. 2.12 Графическое представление о возвращенной волне

На рис.2.13 представлена динамика движения волны следующего вида  $u_0 = 0$   $v_0 = 0$   $\eta_0 = 7 \cdot 10^{-3} e^{(-100(x-0.5)^2)}$  и ее возвращения. В данном случае произошло возвращение волны с противоположной по знаку амплитуды.

Таблица 2.7 Параметры для расчетов с возвращением волны над ровным дном

|                  |                                      |
|------------------|--------------------------------------|
| Размер области   | $1 \times 1$                         |
| Количество шагов | 600                                  |
| Шаг по времени   | 0.01                                 |
| Шаг сетки        | 0.01                                 |
| Точность метода  | $10^{-7}$                            |
| Форма дна        | $H(x, y) = 0.01$<br>$B(x, y, t) = 0$ |

На рис.2.14 представлено продолжение расчета с начальных данных, взятых после выхода волны следующего вида  $u_0 = 0$   $v_0 = 0$   $\eta_0 = 7 \cdot 10^{-3} e^{(-100(x-0.3)^2)}$  из расчетной области.

Таблица 2.8 Параметры для расчетов с продолженным движением волны над ровным дном

|                  |                                      |
|------------------|--------------------------------------|
| Размер области   | $1 \times 1$                         |
| Количество шагов | 300                                  |
| Шаг по времени   | 0.001                                |
| Шаг сетки        | 0.01                                 |
| Точность метода  | $10^{-7}$                            |
| Форма дна        | $H(x, y) = 0.01$<br>$B(x, y, t) = 0$ |

По мере исследования выяснилось, что возвращенная волна может быть в точности такой, как и изначальная, а может иметь меньшую/большую длину. Более того, задание в качестве начальных данных «колебательного фона» разных видов (например, произвольные колебания маленькой амплитуды, либо колебания, задаваемые по определенной формуле) тоже приводит к появлению некоторых волн. Общим остается тенденция к образованию из небольшого «колебательного фона» волн, амплитуда которых на несколько порядков больше (в наших расчетах чаще всего на два порядка).

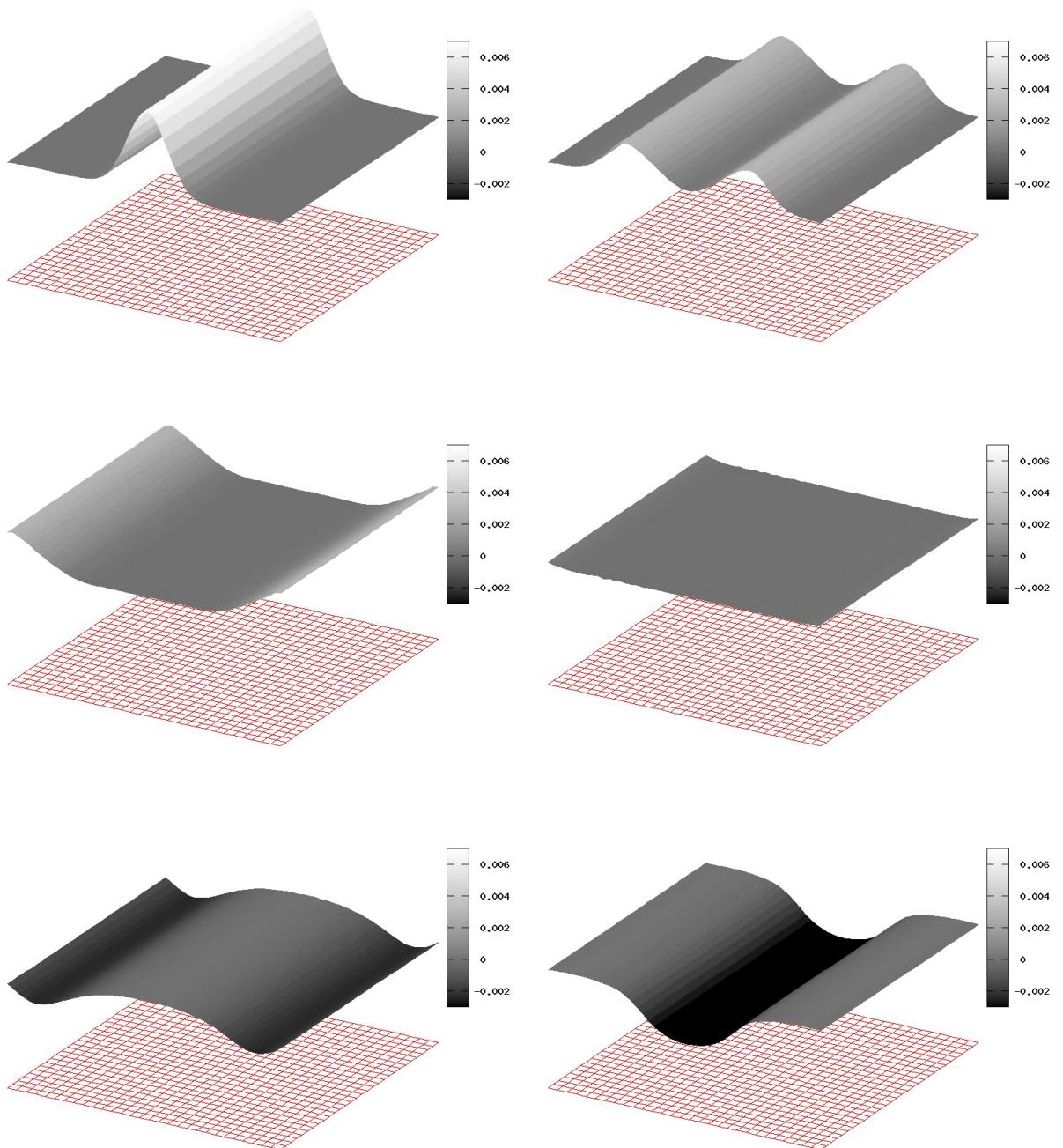


Рис. 2.13 Динамика движения волны в нелинейном случае на нулевом, 80, 160, 250, 400 и 460 шагах

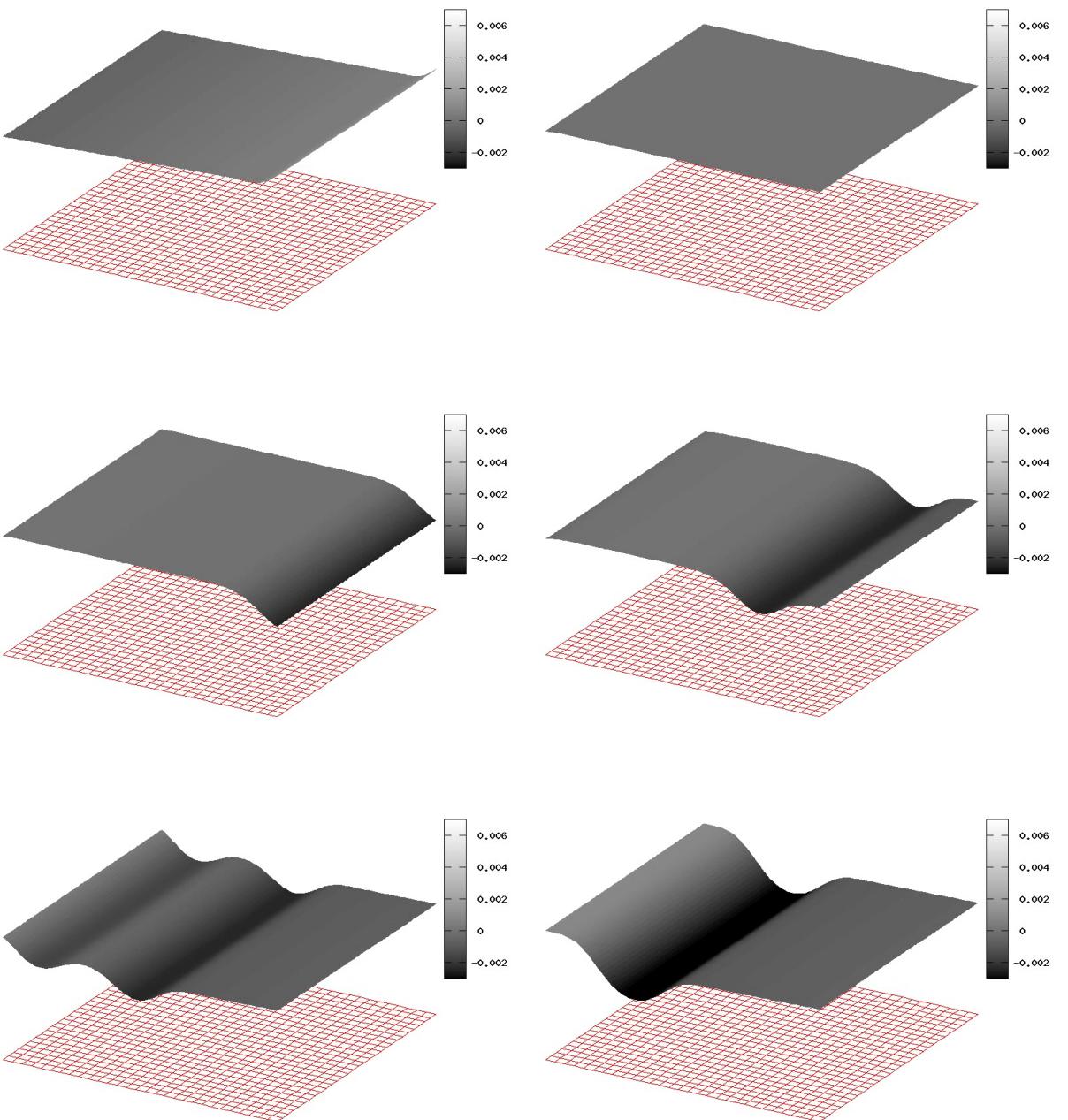


Рис. 2.14 Динамика продолженного движения волны в нелинейном случае на нулевом, 500, 1000, 1500, 2000 и 2500 шагах

#### 2.4.1 Зависимость от параметров

Была предпринята попытка выяснить причину возникновения такого эффекта с помощью проведения серии расчетов, в которых изменялись бы основные параметры, такие как шаг по пространству, шаг по времени, точность вычислений, начальные данные, параметры, входящие в систему уравнений - т.е. глубина дна  $H(x, y)$ .

В таблице 2.9 представлены результаты расчетов и указаны следующие параметры: размер сетки, шаг по времени, соотношение  $h$  к  $\tau$  и результат, который может быть одним из двух - волна возвращается либо нет.

Если в таблице указаны два расчета с одинаковыми параметрами, но разным результатом, это значит, что другие параметры, не отраженные в таблице, были различными (точность вычислений, начальные данные).

Графическое представление этих данных дает некоторое представление о зависимостях между параметрами и результатом. На рис.2.15 изображены характеристики уравнения, которое описывает движение волны  $\frac{\partial x}{\partial t} = g \frac{\partial x}{\partial t} = H$ , где  $H$  - постоянная часть дна (в данных расчетах  $H = 0.1$ ). На графике точками указаны проведенные расчеты, в соответствии с отношением  $h$  к  $\tau$ . Кругами отмечены расчеты, в которых волна возвращается, квадратами - в которых волна не возвращается.

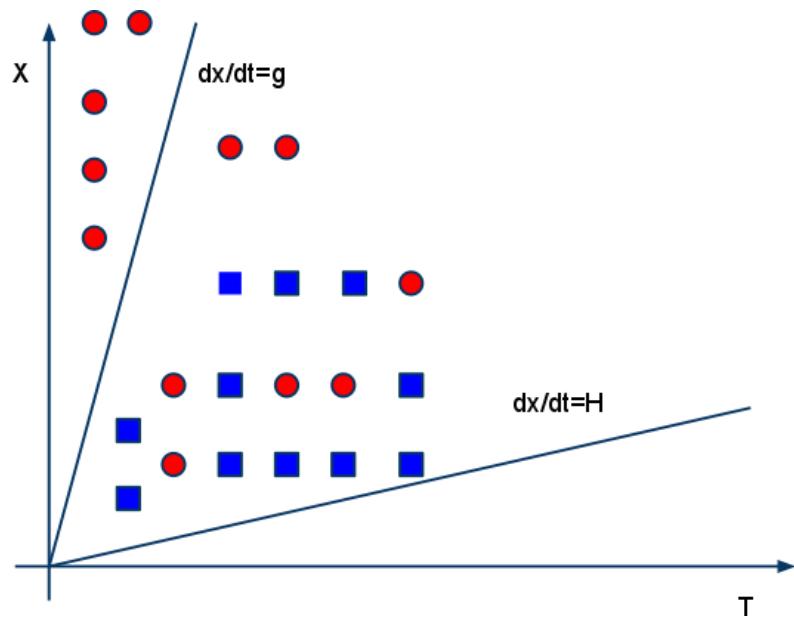


Рис. 2.15 Графическое представление данных о возвращении

Таблица 2.9 Результаты расчетов с различными параметрами

| Размер сетки | Соотношение $h\tau$ | Шаг по времени | Результат       |
|--------------|---------------------|----------------|-----------------|
| 100x100      | 1.0101...           | 0.01           | не возвращается |
| 100x100      | 1.0101...           | 0.01           | не возвращается |
| 100x100      | 1.0101...           | 0.01           | возвращается    |
| 50x50        | 20,4081632          | 0.001          | возвращается    |
| 50x50        | 2,04081632          | 0.01           | не возвращается |
| 150x150      | 0,67114093          | 0.01           | не возвращается |
| 150x150      | 0,67114093          | 0.01           | возвращается    |
| 100x100      | 1.0101...           | 0.01           | возвращается    |
| 100x100      | 1.0101...           | 0.01           | возвращается    |
| 200x200      | 0,50251256          | 0.01           | не возвращается |
| 200x200      | 5,0251256           | 0.001          | возвращается    |
| 50x50        | 2,04081632          | 0.01           | не возвращается |
| 50x50        | 20,4081632          | 0.001          | возвращается    |
| 50x50        | 40,963264           | 0.0005         | возвращается    |
| 50x50        | 20,4081632          | 0.001          | возвращается    |
| 50x50        | 2,04081632          | 0.01           | не возвращается |
| 50x50        | 0,204081632         | 0.1            | не возвращается |
| 50x50        | 40,8163264          | 0.0005         | возвращается    |
| 50x50        | 4,08163264          | 0.005          | возвращается    |
| 50x50        | 0,408163264         | 0.05           | не возвращается |
| 50x50        | 0,408163264         | 0.05           | не возвращается |
| 100x100      | 0,20202...          | 0.05           | не возвращается |
| 100x100      | 1,0101...           | 0.01           | не возвращается |
| 100x100      | 1,0101...           | 0.01           | возвращается    |

В результате такого представления были сделать предположение о причинах возникновения эффекта возвращения:

- если соотношение  $h$  и  $\tau$  больше, чем верхняя характеристика ( $g$ ), то волна всегда возвращается.
- Если меньше - то возвращение зависит от других параметров.
- В области ниже характеристики  $H$  расчетов нет, но близкие к ним (соотношение  $h$  и  $\tau$  примерно 0.2) характеризуются тем, что волна в них почти никогда не возвращается. Помимо этого можно отметить тот факт, что в тех расчетах, где волна возвращалась, при увеличении  $H$  волна переставала возвращаться. В связи с этим можно предположить, что все расчеты, соотношение  $h$  и  $\tau$  в которых меньше нижней характеристики ( $H$ ), характеризуются тем, что волна не возвращается

#### 2.4.2 Расчет с начальных данных

В качестве интересного результата были получены начальные данные небольшой «нулевой» амплитуды, при подстановке которых в расчет наблюдается появление движущейся в области волны. Т.о. более правильно говорить не о «возвращении», а о «возникновении» волны на поверхности жидкости, что приводит к мысли о возможности моделирования такого явления, как «волна-убийца».

Таблица 2.10 Параметры для расчетов с возникновением волны

|                  |                                  |
|------------------|----------------------------------|
| Размер области   | $1 \times 1$                     |
| Количество шагов | 2000                             |
| Шаг по времени   | 0.001                            |
| Шаг сетки        | 0.01                             |
| Точность метода  | $10^{-4}$                        |
| Форма дна        | $H(x, y) = 0.1$ $B(x, y, t) = 0$ |

На рис.2.16 представлена динамика движения такой волны.

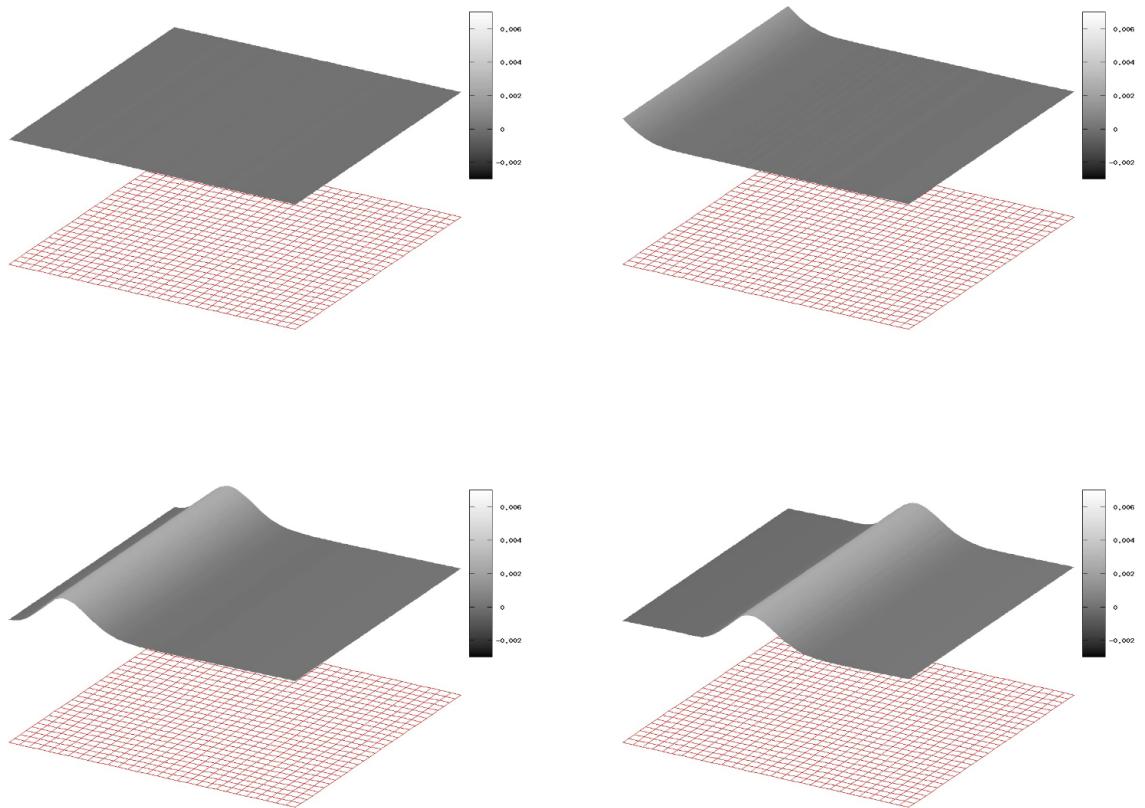


Рис. 2.16 Движение волны с «нулевого фона» на 200, 700, 1200 и 1600 шагах

## § 2.5 Выводы

Проанализировав представленные расчеты, можно сделать несколько выводов:

- Предлагаемый подход к решению задачи (1.1.1)-(1.1.4) с использованием неявных схем и аппроксимации исходной системы уравнений на границе конечной области позволяет находить в ней решение без отражений или искажений. Для всех случаев (сложная начальная форма свободной поверхности, неровное дно, линейная и нелинейная форма уравнения) было получено решение, соответствующее реальным физическим представлениям о задаче, возмущения свободной поверхности безпрепятственно выходили из области, при этом не искажаясь и не порождая отражений.
- Расчет в линейном и нелинейном случае показал, что нелинейная форма уравнений (1.1.1)-(1.1.4) приводит к тому, что волны свободной поверхности со временем меняют свою форму, становятся менее симметричными относительно своего центра.

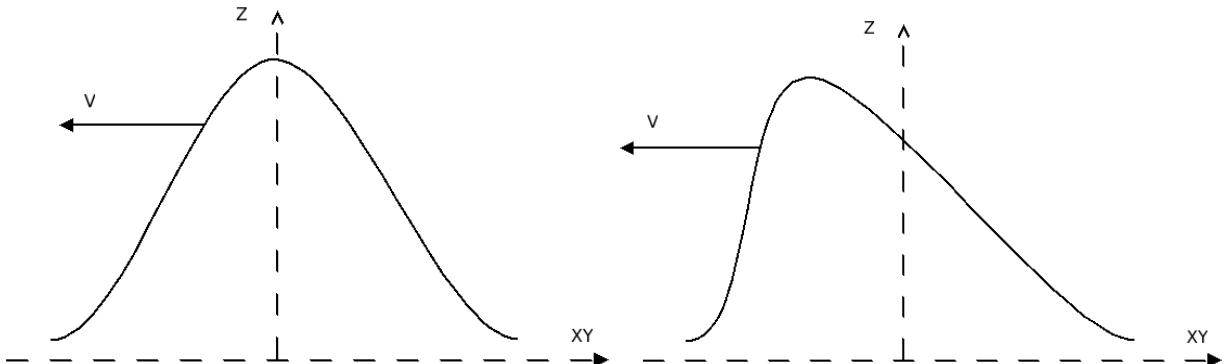


Рис. 2.17 Форма волны в линейном(слева) и нелинейном(справа) случае

Для нелинейных уравнений за основной волной формируются вторичные колебания, амплитуда которых много меньше исходной. При этом указанные эффекты со временем проявляются все более сильно.

# Глава 3. Приложения

## § 3.1 Программные средства

Для проведения расчетов была написан программный комплекс, позволяющий эффективно решать поставленную задачу и ряд сопутствующих проблем. Его разработка велась на языке программирования C++ с использованием компилятора из GNU Compiler Collection и системы автоматизации сборки программного обеспечения из исходного кода CMake. Для решения некоторых задач использовались сторонние инструменты. Так, для пофилирования использовались Valgrind и AMD CodeAnalyst, для визуализации полученных данных - Gnuplot, для генерации документации - Doxygen. Разработка велась под управлением системы контроля версий Git, исходный код был расположен в репозитории code.google.com под академической лицензией MIT.

Написанный программный комплекс реализует алгоритмы, описанные в предыдущих разделах, и позволяет моделировать движение свободной поверхности жидкости при различных условиях. Моделирование может происходить на основе линейных или нелинейных уравнений, с разными начальными данными, параметрами расчетов и формой дна.

Изменяемыми являются:

- начальное значение свободной поверхности
- начальное значение распределения скоростей
- уравнение, описывающее статическую часть дна
- уравнение, описывающее динамическую часть дна
- параметры расчетной области
- параметры численного метода
- значение шага по времени и продолжительность моделирования

В процессе работы программного комплекса формируются файлы, содержащие расчетные данные. Они полностью описывают итоговое распределение

ние скоростей и положение свободной поверхности на каждом временном шаге. Помимо этого, можно выполнить сравнение полученных результатов и также получить файл с данными, описывающими отличия.

Все расчетные данные могут быть визуализированы в двух основных представлениях - с учетом дна и в проекции сверху.

### **3.1.1 Решенные задачи**

В рамках разработки расчетного программного комплекса были решены некоторые проблемы:

#### **Гибкость и документирование**

Для гибкого конфигурирования и кросплатформенности для сборки проекта были выбраны компилятор `g++` из состава `GCC`, а также система автоматической сборки `Cmake`. Данные инструменты могут быть использованы на любой платформе, а также поддерживают достаточно удобное управление компиляцией с помощью разнообразных ключей, которые подробно описаны в документации к соответствующим инструментам.

Чтобы улучшить понимаемость исходного кода был использован генератор документации `Doxxygen`. Соответствующим образом был доработан проект, в него были добавлены описания всех используемых классов и методов, а также формат генерируемой документации. В итоге получено достаточно полное описание кода в виде связанных `html` страниц, которое поможет разработчику гораздо легче разобраться в проекте.

#### **Визуализация расчетных данных**

В качестве инструмента для визуализации расчетных данных был выбран `Gnuplot`, т.к. он позволяет задавать параметры и способ визуализации с помощью скриптов. Это позволяет достаточно легко сформировать сложную 3d композицию на основе расчетных данных, в которой каждый элемент легко может быть настроен необходимым образом. Результатом работы `Gnuplot` является `gif` файл, который в дальнейшем преобразуется в видео файл с помощью выбранного кодека (`mpeg2` для совместимости или `mpeg4` для лучшего отображения).

Помимо этого была написана небольшая утилита для конвертирования результатов, отформатированных для Tecplot, в формат Gnuplot.

На листинге (3.1) представлено тело скрипта, использованного для визуализации.

```
#!/bin/bash
MAX=$4
echo "clear" >> $1
echo "reset" >> $1
echo "set terminal gif animate delay 10 giant size 1280,1024" >> $1
echo "set palette defined (0 \"black\", 0.001 \"blue\", 0.004 \"green\",0.006
      \"yellow\",0.007 \"red\")" >> $1
#echo "set pal gray" >> $1
echo "set output \"\"$2\"\" >> $1
echo "unset xtics">>>$1
echo "unset ytics">>>$1
echo "unset ztics">>>$1
echo "unset border">>>$1

echo "set xrange [0:1]" >> $1
echo "set yrange [0:1]" >> $1
echo "set zrange [-0.011:0.007]" >> $1
echo "set cbrange[-0.003:0.007]" >> $1
for i in `seq 0 ${MAX}`
do
echo "set multiplot" >> $1
echo "set size 0.7,0.7" >> $1
echo "set origin 0.15,-0.15" >> $1
echo "set view 90,0" >> $1
echo "splot \"\"h.dat\"\" with lines t \"\",\"\"$3\"\" index $i using 1:2:3 with
pm3d t \"\" >> $1

echo "set size 0.7,0.7" >> $1
echo "set origin 0.15,0.35" >> $1
echo "set view 60,30" >> $1
echo "splot \"\"h.dat\"\" with lines t \"\",\"\"$3\"\" index $i using 1:2:3 with
pm3d lc rgb \"black\" t \"\" >> $1

echo "unset multiplot" >> $1
done
```

Listing 3.1 Пример скрипта для визуализации

## **Оптимизация времени работы**

В процессе разработки часть усилий была прорачена на оптимизацию программы. Увеличение производительности происходило 2-мя путями:

- Выявление ненужных действий и «узких» мест в программной реализации алгоритма
- Использование возможностей компилятора. Были проанализированы различные настройки компилятора gcc и их влияние на производительность генерируемого кода. Здесь стоит отметить, что современные компиляторы ориентированы на создание более универсальных программ, поэтому добавляют в генерируемый машинный код некоторое количество полезного в общем случае функционала. Но расчетные программы являются узкоспециализированными и не нуждаются в нем, соответственно, отключая данные возможности можно оптимизировать по скорости получающийся код.

Оба вида оптимизации не затрагивали алгоритм решения задачи, т.е. неоптимизированный код эквивалентен оптимизированному.

Таблица 3.1 Используемые флаги компиляции

| Ключ   | Описание   |
|--|--|
| o3   | Наивысший уровень оптимизации (компилятор выбирает почти все опции, позволяющие генерировать быстрый код)  |
| march=native                                       | Указывает компилятору определить тип процессора и оптимизировать код для него  |
| ffast-math (либо только funsafe-math-optimization) | Устанавливает режим «быстрой математики» для программ, в которых не требуется точной реализации IEEE/ISO стандартов для математических функций   |
| fomit-frame-pointer                                | Разрешает не хранить в регистрах указатель на фрейм для функции, которая больше не используется (эта опция уже включена в o3, но иногда явное ее указание заметно улучшает производительность) |
| fno-stack-protector                                | Убирает проверку целостности стека после выхода из функции   |
| fprefetch-loop-array                               | Генерирует инструкции для предварительной выборки в циклах с большими массивами (если позволяет архитектура целевой машины)  |
| funsafe-loop-optimizations                         | Указывает, что циклы в программе не переполняются и не являются бесконечными, что позволяет сгенерировать более быстрый код  |
| funroll-loops                                      | Позволяет раскрыть циклы, размер которых может быть вычислен на этапе компиляции   |
| mpreferred-stack-boundary=num                      | Указывает выравнивание для стека и в некоторых случаях влияет на скорость (по умолчанию num=2, в случае данной программы наибольшая скорость выполнения была получена при num=6)               |
| ftracer  | Форсирует хвостовую дубликацию, делая управление потоком выполнения более простым  |

К возможностям компилятора также относится и компиляция с обратной связью (Profile-guided optimization). В отличие от традиционных способов оптимизации анализирующих исключительно исходные коды, PGO использует результаты измерений тестовых запусков оптимизируемой программы для генерации более оптимального кода. Тестовые запуски выявляют какие части программы исполняются чаще, а какие реже. Преимущество такого подхода в том что компилятор не строит предположений при выборе способа оптимизации, а использует реальную статистику, собранную во время выполнения программы. В силу специфики расчетных программ данный вид оптимизации оказывается достаточно действенным. Техники оптимизации PGO реализованы, во многих компиляторах, в частности: Intel C++ Compiler, Inter Fortran Compiler, GCC, Sun Studio, Microsoft Visual C++, поэтому они могут быть использованы практически в любом расчетном проекте вне зависимости от платформы.

Пример использования техники PGO для gcc выглядит следующим образом:

```
g++ -O3 -fprofile-generate -c -fmessage-length=0 -MMD -MP -MF program.d -MT  
program.d -o program.o program.cpp
```

```
g++ -fprofile-generate -o program ./program.o
```

Listing 3.2 Пример компиляции для сбора статистики

Для сбора статистики используется ключ **fprofile-generate**, а результатом компиляции являются объектные файлы. Затем тот же ключ необходимо отдельно указать линковщику. Полученная в итоге программа будет работать медленнее, т.к. собирает статистику о действиях программы, которая затем будет помещена в файл с расширением **\*.gcda**.

После этого необходимо вновь скомпилировать программу, но уже с учетом набранной статистики - для этого аналогичным образом используется флаг **fprofile-use**.

```
g++ -O3 -fprofile-use -c -fmessage-length=0 -MMD -MP -MF program.d -MT  
program.d -o program.o program.cpp
```

```
g++ -fprofile-use -o program ./program.o
```

Listing 3.3 Пример компиляции с учетом статистики

### 3.1.2 Сервер вычислений

Моделирование движения жидкости с помощью разработанной программы требует выполнения последовательности действий и вовлечения в процесс обработки некоторых инструментов. Это, например, сборка программы с изменениями, внесенными в алгоритм, запуск программы с определенными параметрами, визуализация полученных данных. В обычном случае многие из этих действий выполняются исследователем вручную. Для облегчения работы было создано небольшое приложение («сервер вычислений»), которое позволяет объединить необходимые для моделирования действия в рамках одной среды, а также делает работу более удобной.

Требования, которые могут предъявляться к серверу:

- Сборка и выполнение расчетной программы
- Управление запущенными программами
- Пост- и преобработка результатов
- Управление файлами и хранение информации, описывающей расчет
- Загрузка полученных результатов

Основные принципы, используемые в процессе его разработки:

- Объединение различных процессов, составляющих проведение расчета
- Создание «обертки» вокруг уже существующего кода
- Гибкость доступа и работы с процессами расчета (запуск, получение информации состояния запущенных задач, отмена задачи и т.д.)
- Легкая адаптируемость под разные задачи

«Сервер вычислений» представляет собой TCP сервер, написанный на скриптовом языке python. В нем заложен общий алгоритм проведения расчета, каждый конкретный этап которого реализуется отдельным модулем. Т.о. реализуется шаблон «стратегия», и достигается высокий уровень гибкости, т.к. python позволяет создавать новые модули в замен старых без необходимости создания структуры интерфейсов и изменения исходного кода.

Сервер принимает сообщения в формате xml, описывающие команды к выполнению. Например, запустить указанную программу с выбранными параметрами, что позволяет достаточно легко адаптировать сервер к новым задачам.

При этом на исходный код расчетной программы не накладывается практически никаких ограничений. Сервер просто запускает ее и анализирует результат ее работы, получаемый из стандартного потока вывода. Т.о. например, реализована возможность отслеживания состояния запущенной программы, тогда как, например, во многих планировщиках задач, используемых на вычислительных кластерах (Altair PBS), эта информация недоступна.

На данный момент реализован основной функционал сервера вычислений, позволяющий использовать его в работе. Написаны модули, позволяющие производить предварительную обработку кода программы - компиляцию с помощью GCC и утилиты Make, и визуализировать результаты с использованием Gnuplot.

# Список иллюстраций

|      |  |    |
|------|--|----|
| 1.1  | Постановка задачи . . . . .  | 5  |
| 1.2  | Искусственные границы . . . . .  | 10 |
| 2.1  | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) со сложной начальной формой на нулевом, 80, 180 и 280 шаге . . . . .                           | 21 |
| 2.2  | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) со сложной начальной формой (длинная волна и «капля») на нулевом, 80, 180 и 280 шаге . . . . . | 23 |
| 2.3  | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) с коническим выступом дна на нулевом, 400, 800 и 1200 шаге . . . . .                           | 25 |
| 2.4  | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) с коническим выступом дна на нулевом, 400, 800 и 1200 шаге (проекция сверху) . . . . .         | 26 |
| 2.5  | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) с экспоненциальным выступом дна на нулевом, 80, 180 и 280 шаге . . . . .                       | 28 |
| 2.6  | Результат расчета для линейной системы( $F_{nonlin} = 0$ ) с ровным дном на нулевом, 80, 180 и 280 шаге . . . . .  | 30 |
| 2.7  | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) с ровным дном на нулевом, 80, 180 и 280 шаге . . . . .   | 31 |
| 2.8  | Динамика разности расчетов в линейном и нелинейном случае с ровным дном на нулевом, 80, 180 и 280 шаге . . . . .   | 32 |
| 2.9  | Результат расчета для линейной системы( $F_{nonlin} = 0$ ) с препятствием у границы на нулевом, 80, 200 и 380 шаге . . . . .                                   | 34 |
| 2.10 | Результат расчета для нелинейной системы( $F_{nonlin} \neq 0$ ) с препятствием у границы на нулевом, 80, 200 и 380 шаге . . . . .                              | 35 |
| 2.11 | Динамика разности расчетов в линейном и нелинейном случае с препятствием у границы на 20, 80, 200 и 380 шаге . . . . .   | 36 |
| 2.12 | Графическое представление о возвращенной волне . . . . .   | 37 |

|  |    |
|--|----|
| 2.13 Динамика движения волны в нелинейном случае на нулевом, 80,<br>160, 250, 400 и 460 шагах . . . . .                    | 39 |
| 2.14 Динамика продолженного движения волны в нелинейном случае<br>на нулевом, 500, 1000, 1500, 2000 и 2500 шагах . . . . . | 40 |
| 2.15 Графическое представление данных о возвращении . . . . .  | 41 |
| 2.16 Движение волны с «нулевого фона» на 200, 700, 1200 и 1600 шагах   | 44 |
| 2.17 Форма волны в линейном(слева) и нелинейном(справа) случае . . .   | 45 |

## Список таблиц

|      |  |    |
|------|--|----|
| 2.1  | Параметры для расчета со сложной формой поверхности . . . . .  | 20 |
| 2.2  | Параметры для расчета со сложной формой поверхности . . . . .  | 22 |
| 2.3  | Параметры для расчета с коническим выступом . . . . .  | 24 |
| 2.4  | Параметры для расчета с экспоненциальным выступом . . . . .  | 27 |
| 2.5  | Параметры для расчетов со сравнением линейных и нелинейных<br>уравнений . . . . .  | 29 |
| 2.6  | Параметры для расчетов со сравнением линейных и нелинейных<br>уравнений в случае дна с препятствием на границе . . . . . | 33 |
| 2.7  | Параметры для расчетов с возвращением волны над ровным дном .  | 38 |
| 2.8  | Параметры для расчетов с продолженным движением волны над<br>ровным дном . . . . .                                       | 38 |
| 2.9  | Результаты расчетов с различными параметрами . . . . .   | 42 |
| 2.10 | Параметры для расчетов с возникновением волны . . . . .  | 43 |
| 3.1  | Используемые флаги компиляции . . . . .  | 50 |

# Литература

1. Монахов В.Н. Овсянников Л.В. *Нелинейные проблемы теории поверхностных и внутренних волн.* Н.: Наука, 1985. 318с.
2. Шокин Ю.И. Марчук А.Г., Чубаров Л.Б. *Численное моделирование волн цунами.* Н.: Наука, 1983. 168с.
3. Гильманов А.Н. Ильгамов М.А. *Неотражающие условия на границах расчетной области.* М.: ФИЗМАТЛИТ, 2003. 208с.
4. Рябенький В.С. *Метод разностных потенциалов и его приложения.* М.:ФИЗМАТЛИТ, 2002. 496с.
5. McDonald A. A step toward transparent boundary conditions for meteorological models. //*Monthly weather review*, №130:140–151, 2001.
6. Hussaini M.Y. Navon I.M., Neta B. *A perfectly matched layer formulation for the nonlinear shallow water equations models: The split equation approach.* 2001. 69с.
7. Захаров Ю.Н. *Градиентный итерационные методы решения задач гидродинамики.* Н.: Наука, 2004. 238с.
8. Яненко Н.Н. Рождественский Б.Л. *Системы квазилинейных уравнений и их приложения к газовой динамике.* М.: Наука, 1978. 688с.