Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский государственный университет

имени М.В.Ломоносова»

Механико-математический факультет

Кафедра вычислительной механики

Курсовая работа

**Численное решение уравнения переноса**

Выполнил студент 421 группы

Сенченок Григорий Антонович.

Научный руководитель:

д.ф.-м.н., профессор

Меньшов Игорь Станиславович.

Москва, 2020

Введение

В данной работе рассматривается проблема численного расчета движения твердого тела в сплошной среде.

Твердое тело задано как совокупность точек, лежащих на его границе. Сплошная среда задается векторным полем скоростей в любой точке, не принадлежащей твердому телу.

Данную проблему можно разбить на две подзадачи: нахождение векторного поля скоростей твердого тела и нахождение характеристической функции твердого тела и жидкости в рассматриваемой области на каждом временном шаге.

Для вычисления поля скоростей твердого тела необходимо рассчитать положение каждой точки твердого тела на каждом временном шаге. При этом мы считаем известными скорость центра масс и угловую скорость твердого тела в любой момент времени. Также нам известно начальное положение всех точек, характеризующих границы области, в которой находится твердое тело.

За расчет положения точек твердого тела в каждый момент времени отвечает формула Эйлера. Данное уравнение справедливо для любых двух точек твердого тела и выражает отношение скоростей этих точек в соответствии с угловой скоростью. В качестве одной из скоростей используется скорость центра масс, а другая представляется в виде производной положения этой точки. Численное решение данного дифференциального уравнения позволяет нам получить положение любой точки твердого тела на любом временном шаге.

Характеристическая функция твердого тела и жидкости в рассматриваемой области представляет собой индикатор и принимает значение 0, если в рассматриваемой точке находится твердое тело и 1, если жидкость. Изменение данной скалярной величины в пространстве и времени описывает уравнение переноса. Для численного данного дифференциального уравнения в частных производных широко используется метод объема жидкости (volume of fluid, VOF). Ключевым моментом реализации данного метода является численная аппроксимация поверхности, которая является границей двух сред. Для данного процесса были разработаны различные схемы геометрической реконструкции, позволяющие при использовании их в расчетах метода объема жидкости получить достаточную точность. Высокая точность расчетов была достигнута при использовании схемы THINC (tangent of hyperbola for INterface capturing, отслеживание поверхности с помощью гиперболического тангенса)[1].

Постановка задачи

1. Необходимо реализовать программный алгоритм расчета положения твердого тела при движении.

Твердое тело представляет собой совокупность точек, расстояния, между положениями которых не изменяются. Известно начальное положение всех необходимых для расчета точек. Одна из точек твердого тела является центром масс. Скорость данной точки известна в любой момент времени. Программа должна рассчитать положение всех точек твердого тела и построить траекторию движения на заданном временном отрезке.

Программный алгоритм должен обладать достаточно высокой точностью, для использования результатов его вычислений при расчете движения твердого тела в сплошной среде. Для подсчета точности вычисленного решения предполагается сравнение результатов численного метода с аналитическим решением.

Алгоритм должен работать для расчета положения точек тела, как в двумерном, так и в трехмерном пространстве.

1. Необходимо реализовать программный алгоритм расчета движения поверхности жидкости.

Сплошная среда – это среда, заполняющая некоторую область непрерывно, то есть в сколь угодно малом объеме этой области содержится масса. Движение жидкости описывается с помощью уравнения переноса. Данное дифференциальное уравнение в частных производных описывает изменение скалярной величины в пространстве и времени. При реализации программного алгоритма в задаче данной работы предполагается использовать схему THINC.

Необходимо рассмотреть реализацию данного алгоритма в одномерном случае, чтобы в дальнейшем использовать полученную схему для каждого из двух или трех измерений.

Решение данных задач необходимо для будущей реализации основной проблемы: расчет движения твердого тела в сплошной среде.

Ограничения и допущения

При реализации алгоритмов численных методов рассматриваемая область пространства и отрезок времени разбиваются на отрезки равной длины: вводится равномерная сетка с достаточно малой длиной отрезков. Значения искомых величин вычисляются в узлах данной сетки: в определенной клетке пространства и на определенном шаге по времени. Длина и количество отрезков разбиения выбираются таким образом, чтобы полученное численное решение аппроксимировало аналитическое с высокой точностью.

При реализации программного алгоритма расчета положения твердого тела при движении для данной задачи мы ограничиваемся двумерным случаем.

Граница твердого тела представляет собой многоугольник, состоящий из характерных точек поверхности. Точки считаются характерными точками поверхности, если линию поверхности между двумя соседними точками можно с высокой точностью аппроксимировать прямой линией.

При реализации программный алгоритм расчета движения поверхности жидкости для данной задачи мы ограничиваемся пока одномерным случаем.

При реализации метода VOF со схемой THINC характеристическая функция может принимать значения в диапазоне от 0 до 1. Значения характеристической функции в таком случае будут показывать объемную долю жидкого и твердого вещества в точке пространства x в момент времени t.

Описание процесса

Модель твердого тела представляет собой замкнутую область с границей . Движение твердого тела происходит в исследуемой области . В данной области вводится сетка с равномерным шагом по каждой осей координат.

Тело имеет центр масс - точку . Начальное положение данной точки известно и равняется . Остальные точки твердого тела имеют положения , а их начальные положения равны .

Скорость точки в любой момент времени задается вектор - функцией . Необходимо найти как производную от скорости при заданном начальном положении :

Угловая скорость твердого тела в любой момент времени задается вектор – функцией .

Для любых двух точек A и B твердого тела справедлива формула Эйлера:

Рассмотрим в качестве точки A скорость центра масс , а в качестве точки B k-ую точку твердого тела:

Где - радиус вектор от центра масс до k-ой точки твердого тела.

Поле скоростей внутри твердого тела определяется как скорость k-ой точки в точке :

Характеристическая функция представляет собой функцию – индикатор жидкости:

Уравнение переноса — дифференциальное уравнение в частных производных, описывающее изменение скалярной величины в пространстве и времени:

В случае соленоидального поля скоростей , данное уравнение принимает вид , что в одномерном случае дает:

Таким образом, мы имеем систему дифференциальных уравнений:

С начальными условиями:

Точное решение

Была решена вспомогательная задача о расчете точного положения точек твердого тела при заданных условиях на скорости и начальное положение.

При известном векторе скорости центра масс, векторе угловой скорости твердого тела и начальном положении всех точек твердого тела были численно рассчитаны положения всех точек твердого тела в моменты времени .

Расчет был произведен для двумерного случая, но его результаты можно применить и при расчете движения в трехмерном пространстве.

Постановка задачи

В двумерное пространство помещено твердое тело, представляющее собой совокупность точек, расстояния, между положениями которых не изменяются. Задана абсолютная система координат.

Начальное положение точек твердого тела: , в том числе центра масс:

Вектор скорости центра масс: . В случае двумерного пространства:

Вектор угловой скорости твердого тела: . В случае двумерного пространства: .

Возьмем отрезок времени и разобьем его на подотрезков . Длина каждого отрезка: . - узлы данного разбиения.

Необходимо численно рассчитать положение всех точек твердого тела в моменты времени .

Теоретическая часть

За скорость любой точки твердого тела отвечает формула Эйлера:

Где - радиус вектор от центра масс до k-ой точки твердого тела.

Для центра масс:

Интегрирование от 0 до t: .

Численное решение

Данный интеграл был рассчитан методом квадратур. В качестве квадратурной формулы была использована составная формула трапеции. Исходный код программы расчета на C++ находится в файле EulerEquation.cpp. Функция C++

VectorArray getXc(VectorArray v, double tMax, VectorXd xc0);

позволяет вычислить значения вектора

Таким образом, были получены значения в узлах .

Вычисление :

Интегрирование от 0 до t:

Метод численного решения данного уравнения аналогичен решению интегрального уравнения Вольтера второго рода. [2]

Функция в правой части в нашем случае является:

Значения данной функции были вычислены аналогично, методом квадратур с использованием составной формулы трапеции. Исходный код программы расчета на C++ находится в файле EulerEquation.cpp. Функция C++

VectorArray getRightFuncNoMargin(VectorArray v, Vector3Array omega, VectorArray xc, double maxT);

позволяет вычислить значения вектор – функции правой части

Таким образом, были получены значения вектор - функции в узлах .

Применим принцип численного решения интегрального уравнения Вольтера второго рода для вычисления значений вектор – функции в узлах равномерной сетки . Шаг сетки . Количество узлов:

Произведение функций , где - ядро интегрального уравнения в нашем случае является векторным произведением , и не зависит от t.

, при

При заменим интеграл в левой части уравнения на квадратурную формулу с коэффициентами :

Где - погрешность квадратурой формулы. Отбрасывая данную малую величину, решим систему уравнений относительно .

Введем новые обозначения: , , .

Заменим операцию векторного произведения на произведение кососимметрической матрицы на вектор.

Матрица была рассчитана для двумерного случая, однако можно использовать тот же способ для расчета в трехмерном пространстве.

Таким образом, мы выражаем вектор для . необходимо рассчитывать последовательно, так как в сумме используются значения .

Данный алгоритм был реализован в функции на C++:

VectorArray getXk(VectorArray rightFunc, Vector3Array omega, VectorArray xc, double maxT)

В качестве квадратурной формулы была использована комбинация составной формулы Симпсона и правила трех восьмых.

В результате была реализована программа для вычисления положений k-ой точки твердого тела в моменты времени . Результаты вычислений записываются в файл “output.txt”.

Пример расчета

Вектор – функции в данной программе задаются с помощью лямбда - выражений на языке программирования C++, что обеспечивает высокую гибкость в настройке программы под любую конфигурацию задачи. Например, вектор - функция скорости центра масс твердого тела может быть задана следующим образом:

function<VectorXd(double)> v = [=](double t)->VectorXd {

VectorXd v(n);

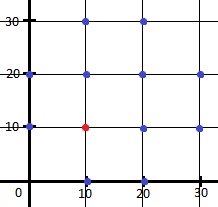
v(0) = 1;

v(1) = 50 - 9.81 \* t;

return v;

};

Для демонстрации работы программы было выбрано твердое тело, состоящее из 12 точек, которые в начальный момент времени имели следующие координаты:



Вектор угловой скорости:

function<Vector3d(double)> omega = [=](double t)->Vector3d {

return Vector3d(0, 0, 1);

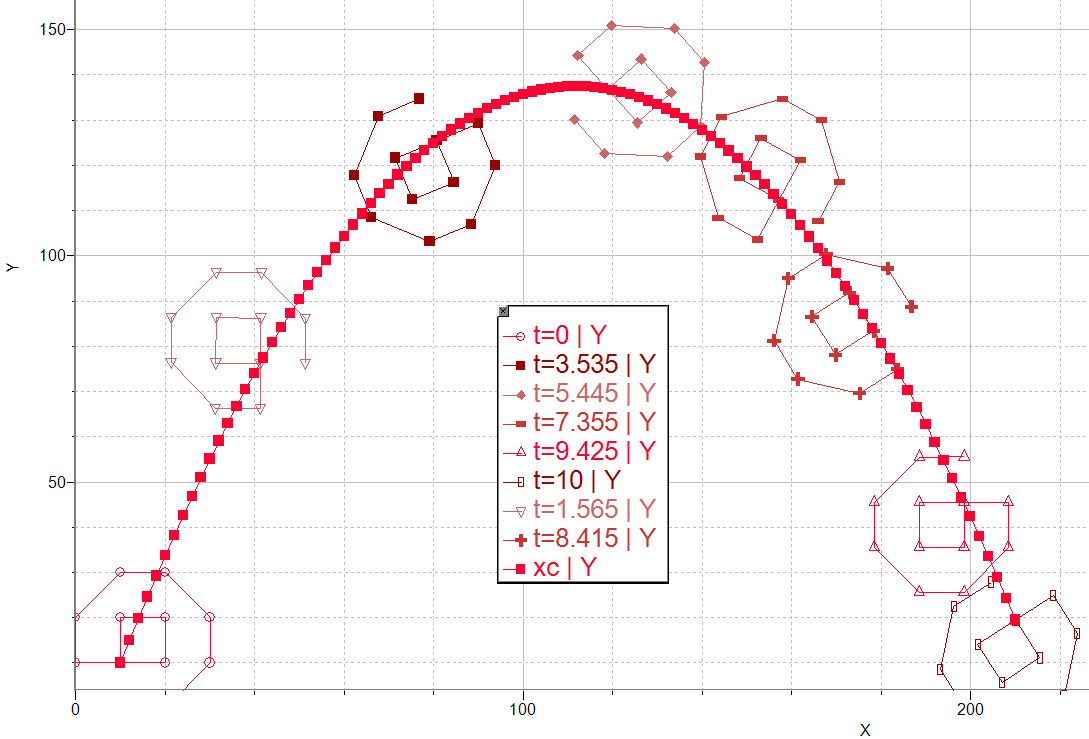
};

Количество шагов и длина шага сетки:

int stepsN = 100;

double h = 0.1;

Координаты выбранных точек твердого тела записаны в файле output.txt и имеют следующий вид:



Для более наглядной визуализации результатов расчет была реализована программа на языке программирования Python, которая создает анимацию движения твердого тела по предрассчитанным данным о положениях точек.

Результат работы данной программы прилагается в файле TestMovement.gif.

Погрешность вычислиней

Для расчета ошибки численного моделирования в данном примере было вычислено аналитическое решение.

Ошибка была рассчитана по норме : как среднеквадратическое отклонение по следующей формуле:

Где – действительное положение k-ой точки твердого тела в момент времени , полученное из аналитического решения.

Ошибка расчета - центра твердого тела:

Среднеквадратическое отклонение при вычислении 4000 шагов на отрезке времени 10с (длина временного шага 0.0025) составляет 2.47756e-10.

Ошибка расчета

Среднеквадратическое отклонение при вычислении 4000 шагов на отрезке времени 10с (длина временного шага 0.0025) составляет 0.000469316.

Следует отметить сходимость рассчитанного решения к аналитическому: при измельчении шага , значение ошибки стремится к 0:

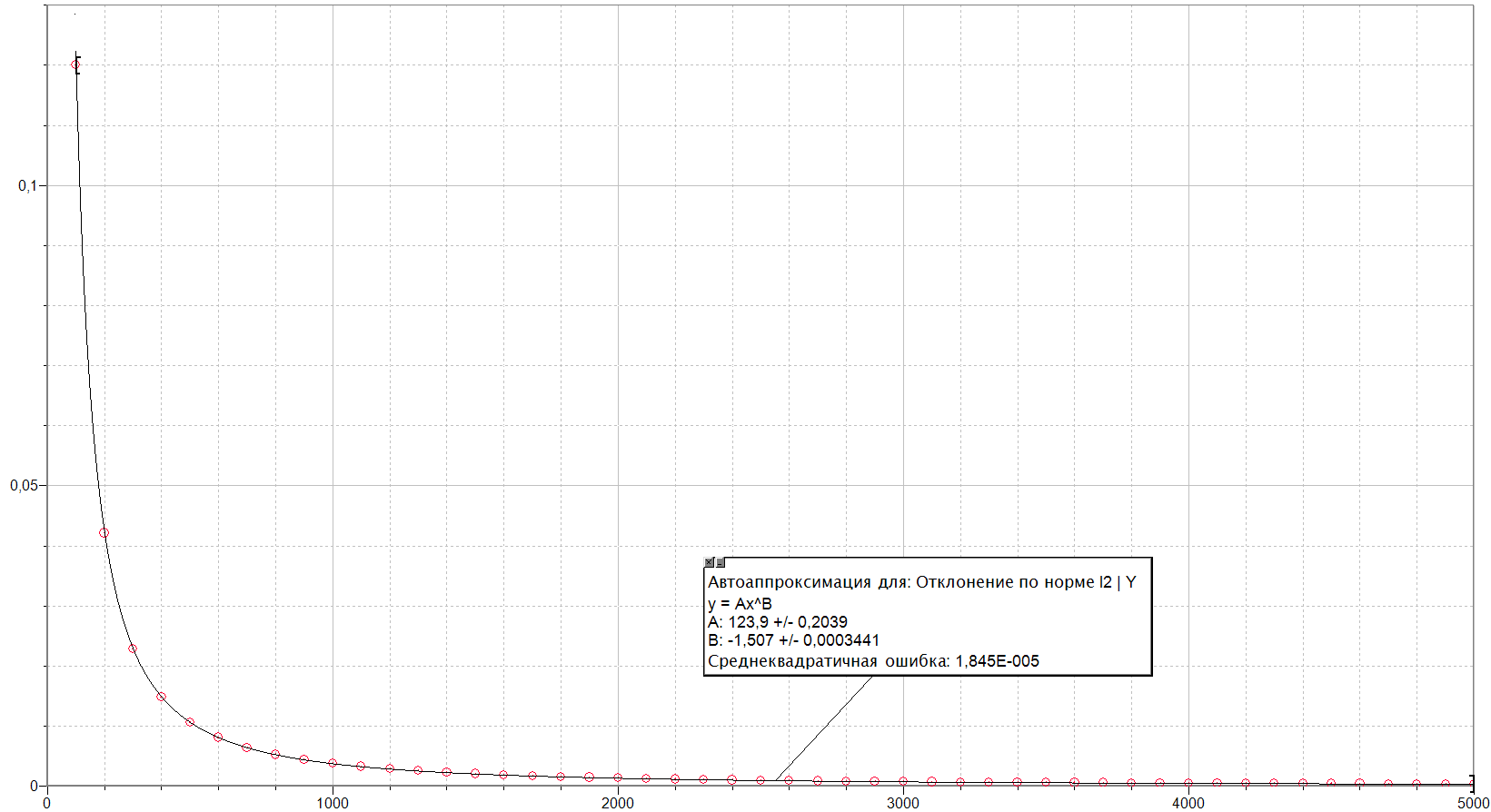


Схема THINC

Для численного решения уравнения переноса была рассмотрена схема THINC. Данный метод описан в статье [1].

В данной работе проводится исследование данного метода, а также программная реализация THINC для одномерного случая на языке C++. Полученная программа необходима для дальнейшей разработки численного решения уравнения переноса.

Предполагается использовать схему THINC (tangent of hyperbola for interface capturing: гиперболический тангенс для отслеживания поверхности) для численного решения уравнения переноса, представленного в следующем виде:

Где – векторное поле скоростей, - переносимая скалярная величина. Определим как функцию Хевисайда, принимающую значения 0 и 1:

– оператор дивергенции.

В одномерном случае данная задача рассматривается в виде:

Также мы предполагаем, что поле скоростей соленоидально, то есть . В этом случае задача рассматривается в виде:

Отрезок , на котором рассматривается данное уравнение, разбивается на последовательных подотрезков, длиной каждый. . Положения , являются узлами данной сетки. . Для реализации программы была выбрана равномерная сетка с подотрезками равной длины . Зададим длину временного шага и построим схему THINC для вычисления значений функции

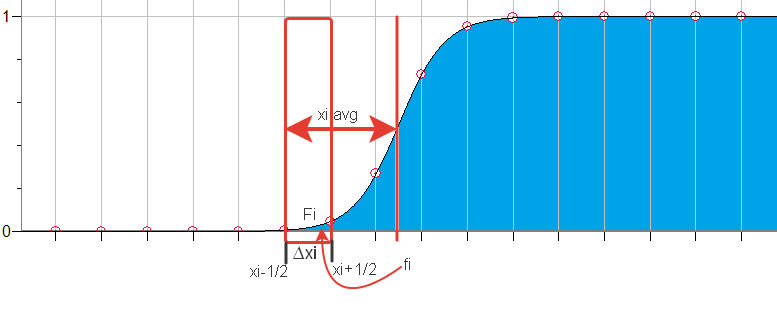
Рассмотрим - среднее значение функции на -ом отрезке на -ом временном шаге:

Для аппроксимации функции на каждом отрезке рассчитывается функция:

Где параметры определяются следующим образом:  
.

определяет сжатие по оси X – скорость прыжка

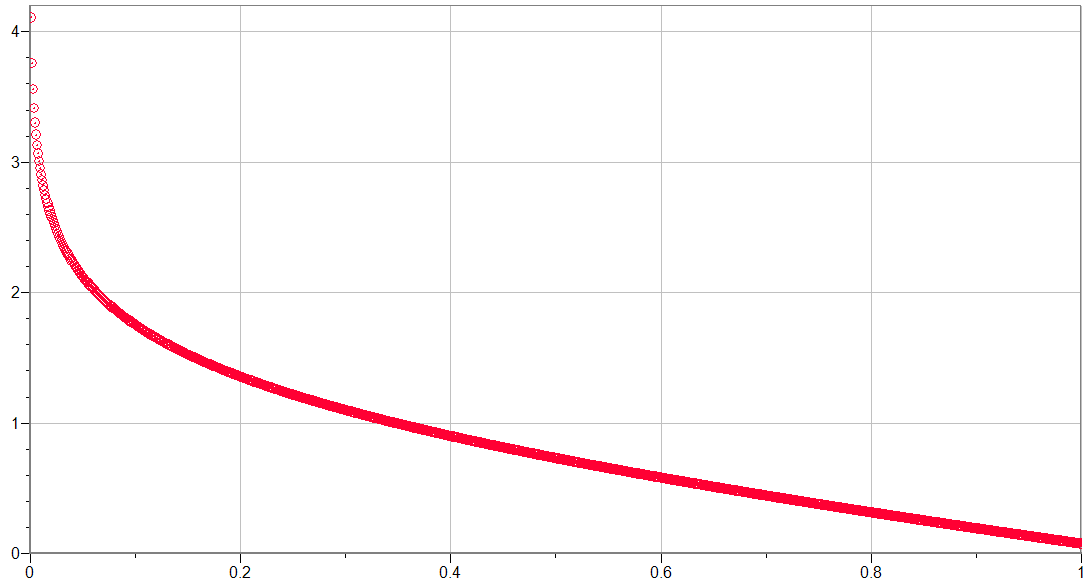
Параметр – относительное расстояние до середины прыжка от левой границы отрезка .



определяется из интегрального уравнения:

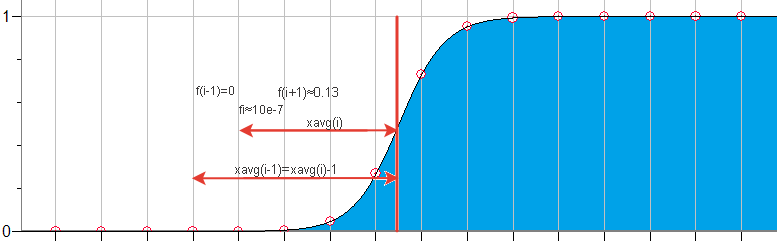
Аналитическое решение данного уравнения:

Зависимость от при приближении к отрезку, на котором происходит прыжок, выглядит следующим образом:



Полученная функция при удалении от середины скачка стремится к 0 или 1. В силу особенностей расчета данной функции на компьютере (ограничения на длину числа, хранимого в памяти) она будет принимать данные значения уже на небольшом удалении от середины прыжка. В таком случае, до некоторого номера отрезка , , а при расчете мы будем получать неопределенность типа .

Для решения данной задачи на практике значения и не вычисляются для ячеек на достаточном удалении от прыжка. А при приближении к прыжку, отслеживается первое ненулевое значение , для него рассчитывается значение , а для предыдущей ячейки оно вычисляется как . Аналогично рассчитываются значения для отрезков после прыжка. Это позволяет корректно вычислять значения и в ячейках сетки. Данная особенность расчета была упомянута в статье [3]



Расчет был реализован в функции на C++:

double getXiavg(double beta, double gamma, double fi)

Для начальной аппроксимации функции (в момент времени t=0) и определения на сетке , положения были вычислены исходя из прыжка функции . Таким образом была реализована функция:

VectorXd aproximateHeaviside(VectorXd Heaviside, int cellCount, double deltaX, double beta)

После вычисления значений на n-ом временном шаге, совершается переход к шагу: рассчитываются значения справа от левой границы отрезка и слева от правой границы отрезка по следующим формулам [4]:

Где

Тестовый расчет для движения волны в одномерном случае был произведен в функции

void waveMovementTest()

Начальные условия:

Функция :

int cellCount = 400;

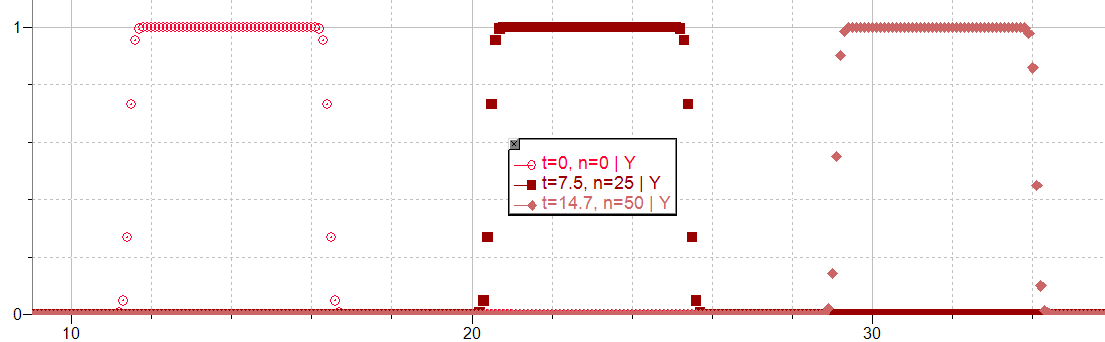
int stepN = 50;

double h = 0.1;

double timeStep = 0.3;

double u = 1.2; // векторное поле статическое

В результате расчета были получены значения для моментов времени . Некоторые положения при движении:



Таким образом, была написана программа, реализующая метод THINC для одномерного случая. Данный алгоритм можно применить и в случае нескольких измерений, используя одномерный расчет по направлению каждой из осей.

Заключение

Таким образом, были исследованы методы численного решения уравнения Эйлера и уравнения переноса. Был реализован программный алгоритм расчета положения твердого тела при движении в двух и трехмерном случае, а также программный алгоритм расчета движения поверхности жидкости в одномерном случае.

Численное решение уравнения Эйлера позволяет с высокой точностью рассчитать положение любой точки твердого тела в заданный момент времени.

Метод VOF с использование схемы THINC позволяет рассчитать изменение характеристической функции, описывающей твердое тело в жидкости в пространстве и времени. Полученная реализация одномерного алгоритма необходима для дальнейшей реализации многомерного расчета движения жидкости и твердого тела в пространстве с использованием результатов вычислений поля скоростей твердого тела.

Список литературы

[1] A simple algebraic interface capturing scheme using hyperbolic tangent function. F. Xiao, Y. Honma and T. Kono (2005)

[2] Применение метода квадратур для численного решения интегральных уравнений второго рода. И. О. Арушанян (2018)

[3] Revisit to the THINC scheme: A simple algebraic VOF algorithm. Feng Xiao, Satoshi Ii, Chungang Chen (2011)

[4] An Eulerian interface sharpening algorithm for compressible two-phase flow: The algebraic THINC approach. Keh-Ming Shyue, Feng Xiao (2014)

[5] Теоретическая Механика. С.В.Болотин, А.В.Карапетян, Е.И.Кугушев, Д.В.Трещев (2010)