

04, апрель 2016

УДК 004.92

Моделирование потока невязкого несжимаемого газа с использованием сетки Эйлера в компьютерной графике

Сикорский О.С., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

Научный руководитель: Русакова З.Н., к.т.н, доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

iu7@bmstu.ru

Введение

Моделирование движения газов и жидкостей является актуальной задачей в компьютерной графике. Оно используется как при создании специальных эффектов в фильмах, так и в интерактивных программах. Стандартные методы гидродинамики и газодинамики уделяют большое внимание явлениям, которые невидимы человеческому глазу (звуковые волны, ударные волны и другие) и поэтому требуют очень маленького шага по времени для того, чтобы правильно их смоделировать, иначе физическая точность экспоненциально падает и даже общая картина распространения газа или жидкости становится абсурдной. Для компьютерной графики такое неприемлемо: необходимо, чтобы алгоритмы были быстрые и стабильные, пусть и не очень физически точные. Для этих целей были разработаны алгоритмы, работающие намного быстрее классических алгоритмов гидродинамики и газодинамики. Некоторые из них рассмотрены в данной статье. Эти алгоритмы также могут позволить в первом приближении рассмотреть многие физические и инженерные задачи, не тратя огромное количество времени на моделирование потоков жидкостей или газов классическими методами вычислительной гидродинамики.

Целью данной работы является рассмотрение алгоритмов моделирования распространения невязкого несжимаемого дыма и построение программного обеспечения, реализующего их.

Физические основы движения газа

Движение жидкости или газа описываются уравнением Навье-Стокса:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = \nu \nabla^2 \vec{u} - \frac{1}{\rho} \nabla p + \vec{f}, \quad (1)$$

где $\vec{u} = \{u, v, w\}$ - поле скоростей

$(\vec{u} \cdot \nabla) = u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z}$ - оператор адвекции

ν – коэффициент кинематической вязкости

ρ - плотность

$\vec{f}$ - внешние силы

Эффектами, возникающими из-за сжатия газа (звуковые волны, ударные волны и другие), в компьютерной графике обычно можно пренебречь, поэтому будем считать, что газ несжимаемый. Другими словами, должно соблюдаться условие:

$$\nabla \cdot \vec{u} = 0 \quad (2)$$

В данной работе будет рассматриваться невязкий газ, то есть газ, у которого $\nu = 0$:

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{u} - \frac{1}{\rho} \nabla p + \vec{f} \quad (3)$$

Таким образом, из (2) и (3) получается система дифференциальных уравнений, описывающая поведение невязкого несжимаемого газа.

Для решения этой системы, можно использовать схему расщепления, предложенную Александре Чорином [1]:

$$1) \quad \frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{u}$$

$$2) \quad \frac{\partial \vec{u}}{\partial t} = \vec{f}$$

$$3) \quad \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \nabla p$$

Для каждой части есть свой метод решения: для первой части используется процедура *advect*; вторая часть, в предположении, что внешние силы особо не меняются, решается методом Эйлера; третья часть решается процедурой *project*. Процедуры *advect* и *project* будут более подробно рассмотрены ниже.

Таким образом, алгоритм решения уравнения (3) следующий:

$$1) \quad \vec{u}^A = \text{advect}(\vec{u}^n, \Delta t, \vec{u}^n)$$

$$2) \quad \vec{u}^B = \vec{u}^A + \vec{f} \Delta t$$

$$3) \quad \overrightarrow{u^{n+1}} = \text{project}(\overrightarrow{u^B}, \Delta t),$$

где $\overrightarrow{u^n}$ – известное поле скоростей на шаге n, а Δt – шаг времени симуляции.

Дискретизация пространства

Необходимо определить способ описания поля скоростей. Существует две основные точки зрения:

1) Система частиц. Каждая частица обладает определёнными характеристиками: положением в пространстве, скоростью – точка зрения Лагранжа

2) Фиксированная в пространстве система координат, в каждой точке которой можно измерить характеристику поля – точка зрения Эйлера

В данной работе была выбрана точка зрения Эйлера.

Далее необходимо определить, как дискретизировать моделируемое пространство. Можно использовать обычную регулярную сетку, в которой все величины находятся в центре каждой ячейки (которые также называют вокселями). Пример такой сетки изображён на рис.1.

Существуют и другие варианты разбиения. Одним из них является MAC-сетка, предложенная Харлоу и Уэлч в 1965 году [2]. В ней скалярные величины находятся в центре каждой ячейки, а векторные – в центре рёбер или граней в двумерном и трёхмерном случаях соответственно (рис 2). Основным преимуществом данной сетки является более точное, по сравнению с обычной сеткой, вычисление дивергенции векторных величин, что будет использовано позднее. Очевидно, что, если мы поделим моделируемое пространство на большее число ячеек, тем точнее будет полученный результат. В данной работе будет использоваться MAC-сетка.

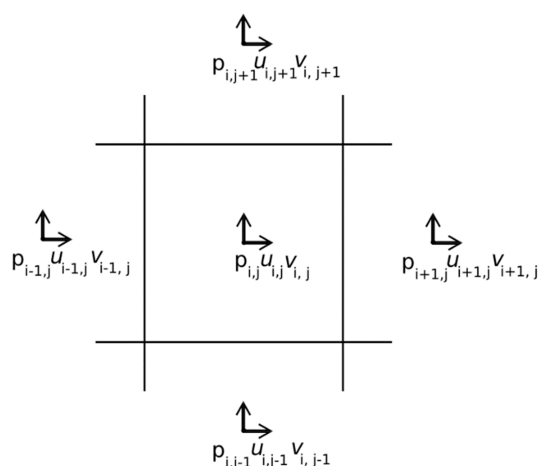


Рис. 1. Ячейка обычной сетки

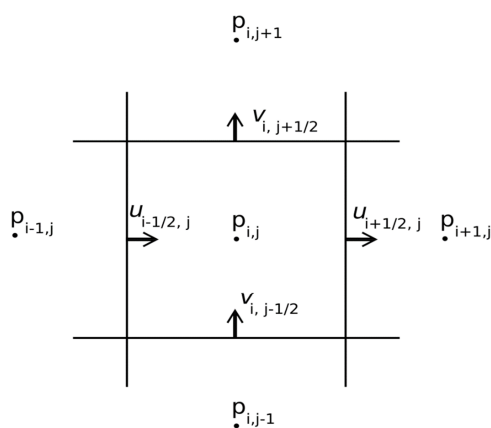


Рис. 2. Ячейка MAC-сетки

Адвекция

Адвекция – явление переноса частицей своих свойств. Первая часть уравнения (3) является уравнением адвекции для скорости:

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla) \vec{u} \quad (4)$$

Для решения уравнения (4) Джос Стэм предложил использовать полу-Лагранжевый метод, ранее используемый в метеорологии [3]. Общий алгоритм следующий:

Пускай необходимо узнать значение параметра q в точке x_0 в момент времени $t + dt$, если известно поле скоростей в момент времени t (рис 3). На эту задачу можно посмотреть с точки зрения Лагранжа – где находится во время t частица, которая в момент $t + dt$ находится в точке x_0 . Значение q этой частицы и будет искомым значением q . Положение этой частицы можно аппроксимировать, пойдя «обратно» по полю скоростей:

$$v(x_0, t_0) = v_0$$

$$q(x_0, t_0 + dt) \approx q(x_1, t_0) = q(x_0 - v_0 * dt, t_0) \quad (5)$$

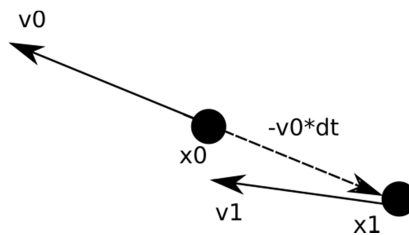


Рис. 3. Аппроксимация положения частицы методом Эйлера

Уравнение (5) соответствует методу Эйлера. Для увеличения точности можно использовать методы более высокого порядка, например, Рунге-Кутты второго порядка:

$$x_1 = x_0 - v_0 * \frac{dt}{2}$$

$$q(x_0, t_0 + dt) \approx q(x_2, t_0) = q(x_0 - v(x_1) * dt, t_0) = q(x_0 - v\left(x_0 - v_0 * \frac{dt}{2}\right) * dt, t_0)$$

Так как положения частиц в общем случае не лежат в узлах сетки, необходимо произвести интерполяцию по ближайшим известным значениям q . Джос Стэм использовал линейную интерполяцию.

Данный метод является безусловно стабильным. Действительно, наибольшее значение параметра q в поле не превосходит наибольшего значения этого параметра на предыдущем шаге.

Таким образом, алгоритм процедуры $\text{advect}(q, u, dt)$ с использованием метода Рунге-Кутты второго порядка выглядит следующим образом:

1. для каждого q_i с положением x_0 в поле q :
2. $x_1 = x_0 - 0.5 * dt * \text{interpolate}(x_0, u)$
3. $x_2 = x_0 - dt * \text{interpolate}(x_1, u)$
4. $q_i = \text{interpolate}(x_2, q)$,

где $\text{interpolate}(x, f)$ – интерполяция значения поля f в точке x .

Основной проблемой данного метода является явление численной диссипации [4] – полученные потоки слишком быстро затухают, теряются мелкие детали. Есть несколько способов борьбы с этим явлением:

1. применение более сложной схемы интерполяции, например, сплайнов Катмулла-Рома;
2. использование метода vorticity confinement.

Физическая основа метода vorticity confinement следующая: численная диссипация мешает потоку «вращаться» в соответствии с завихрённостью поля, поэтому предлагается найти места с максимальными завихрённостями и добавить там внешние силы, тем самым компенсировав численную диссипацию.

Предварительно вычисляется завихрённость поля

$$\vec{\omega} = \nabla \times \vec{u}.$$

Затем находится нормализованный вектор положений завихрённостей, ведущий от зон с малой завихрённостью к зонам с более высокой:

$$\vec{N} = \frac{\nabla |\vec{\omega}|}{|\nabla |\vec{\omega}||}.$$

Направление и величина силы определяются следующим образом:

$$\overrightarrow{f_{vort}} = \epsilon \Delta x (\vec{N} \times \vec{\omega}), \quad (6)$$

где ϵ – настраиваемый параметр

Таким образом, добавив (6) с правильно подобранным коэффициентом ко внешним силам, можно получить более качественный результат.

Внешние силы

За действие внешних сил отвечает вторая часть уравнения (4):

$$\frac{\partial \vec{u}}{\partial t} = \vec{f}$$

Внешние силы состоят из двух частей:

$$\vec{f} = \overrightarrow{f_{bouy}} + \overrightarrow{f_{vort}}$$

$\overrightarrow{f_{vort}}$ – определяется с помощью уравнения (6)

$\overrightarrow{f_{bouy}}$ – сила плавучести дыма

В [4] приводится простая модель влияния температуры и массы дыма на его распространение:

$$\overrightarrow{f_{bouy}} = -\alpha s \vec{z} + \beta (T - T_{amb}) \vec{z},$$

где s – количество дыма в ячейке;

T – температура дыма в ячейке;

T_{amb} – температура среды;

α, β – настраиваемые параметры.

Проекция

Третья часть уравнения (3) отвечает за давление:

$$\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \nabla p \quad (7)$$

Уравнение (7) можно представить как

$$\begin{aligned} \frac{u^* - u}{\Delta t} &= -\frac{1}{\rho} \nabla p \\ u^* &= u - \frac{\Delta t}{\rho} \nabla p, \quad (8) \end{aligned}$$

где u^* – решение уравнения (7), удовлетворяющее условию несжимаемости (2), а u – известное поле скоростей. Применив к (8) оператор набла, получаем

$$\nabla u^* = \nabla u - \frac{\Delta t}{\rho} \nabla^2 p$$

Учитывая, что дивергенция конечного поля скоростей равна нулю, получаем

$$\frac{\Delta t}{\rho} \nabla^2 p = \nabla u \quad (9) \text{ – уравнение Пуассона}$$

Уравнение (9) можно дискретизировать с помощью центральных разностей следующим образом:

$$\frac{\Delta t}{\rho} \left(\frac{p_{i+1,j,k} + p_{i,j+1,k} + p_{i,j,k+1} + p_{i-1,j,k} + p_{i,j-1,k} + p_{i,j,k-1} - 6p_{i,j,k}}{\Delta x^2} \right) \\ = \frac{u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}}{\Delta x} + \frac{v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}}{\Delta x} + \frac{w_{i,j,j+\frac{1}{2},k} - w_{i,j,j-\frac{1}{2},k}}{\Delta x} \quad (10)$$

(10) является системой линейных уравнений вида $Ax = b$, причём A является разреженной симметричной положительно определённой матрицей [5], поэтому можно воспользоваться методами, хорошо работающими с такими видами уравнений, например, методом Гаусса-Зейделя, методом сопряжённых градиентов или других.

Алгоритм процедуры $\text{project}(u, dt, \rho)$ следующий:

1. составить матрицу A
2. составить вектор дивергенции скоростей b
3. решить уравнение $Ap = b$
4. $u = u - \text{grad}(p) * (dt / \rho)$,

где $\text{grad}(p)$ – вычисление градиента поля p .

Данная процедура была названа проекцией из-за того, что она проецирует поле скоростей в поле скоростей с нулевой дивергенцией. Применение этой процедуры дважды даст тот же результат, что и её применение один раз, что соответствует понятию «проекция» в других областях математики.

Структура данных разреженной матрицы

Как было указано выше, матрица A является разреженной, симметричной и положительно определённой. Её можно держать в стандартных структурах для хранения разреженных матриц. Но из уравнения (10) видна следующая особенность: в ряду, соответствующем уравнению ячейки (i, j, k) , ненулевых коэффициентов не более семи: $A_{(i,j,k),(i,j,k)}$ и шесть соседних ячеек: $A_{(i,j,k),(i+1,j,k)}$, $A_{(i,j,k),(i-1,j,k)}$, $A_{(i,j,k),(i,j+1,k)}$ и так далее. В [5] был предложен эффективный способ хранения этой матрицы с использованием четырёх трёхмерных массивов: Adiag , AplusI , AplusJ , AplusK . В $\text{Adiag}[i,j,k]$ хранится значение диагонального элемента: $A_{(i,j,k),(i,j,k)}$; в $\text{AplusI}[i,j,k]$ – значение элемента $A_{(i,j,k),(i+1,j,k)}$. Аналогично хранятся элементы и в массивах AplusJ и AplusK . Для доступа к элементу с отрицательным смещением относительно диагонали,

например, $A_{(i,j,k),(i-1,j,k)}$, можно воспользоваться свойством симметричности:

$$A_{(i,j,k),(i-1,j,k)} = A_{(i-1,j,k),(i,j,k)} = Aplusi[i-1, j, k]$$

Поэтому при размерности матрицы k нужно хранить всего $7k$ элементов, причём обращение к каждому элементу происходит без дополнительных затрат в отличие от стандартных структур хранения разреженных матриц.

Алгоритм моделирования

Таким образом, алгоритм одного шага моделирования выглядит следующим образом:

1. `advect(s, u, dt)`
2. `advect(T, u, dt)`
3. `advect(u, u, dt)`
4. `applyForce(dt)`
5. `project(u, dt, rho)`

На основе полученной концентрации дыма s затем производится рендеринг.

Рендеринг

Рендеринг дыма производится с помощью метода Volumetric Ray Casting, позволяющий из скалярного трёхмерного поля получить двумерное изображение. Алгоритм Volumetric Ray Casting следующий: из точки наблюдения в пиксель пускается луч, пронизывающий сцену, состоящую из вокселей. Через каждый интервал d делается замер значения поля. На основе всех замеров получается конечный цвет пикселя. Принцип работы и блок схема алгоритма представлен на рис. 4.

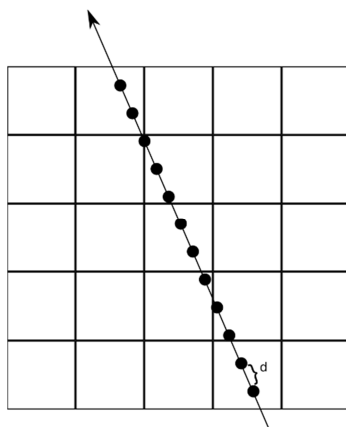


Рис. 4. Принцип работы Volumetric Ray Casting

Вычислительная сложность

Пусть моделируемое пространство разделено на $n \times n \times n$ ячеек. Очевидно, что сложность процедур `advect` и `applyForce` равна $O(n^3)$.

Самая интенсивная с точки зрения вычислительного аппарата часть – процедура проекции, а именно – решение дискретного уравнения Пуассона методом сопряжённых градиентов. Использование модифицированного неполного разложения Холецкого позволяет решению сойтись за $O(\sqrt{k})$ итераций [6], где $k = n^3$ – размерность матрицы. Умножение разреженной матрицы на вектор занимает $O(k)$ операций. Таким образом, процедура проекции и, следовательно, вся программа, требует $O(k\sqrt{k}) = O(n^{4.5})$ операций.

Программная реализация

На основе методов, изложенных в статье, построено программное обеспечение, реализующее моделирование дыма. В качестве сетки использована MAC-сетка. Для адвекции применён полу-Лагранжевый метод со сплайнами Катмулла-Рома в качестве схемы интерполяции. Для решения системы линейных уравнений в процедуре проекции был имплементирован метод сопряжённых градиентов, использующий модифицированное неполное разложение Холецкого в качестве предобуславливателя.

В качестве языка программирования был выбран язык C++, обеспечивающий необходимую скорость вычислений и, в то же время, поддерживающий современные возможности высокоуровневых языков (классы, шаблоны, исключения, лямбда-функции, умные указатели и другие). Для разработки интерфейса была использована библиотека Qt.

Интерфейс программы представлен на рис. 5.

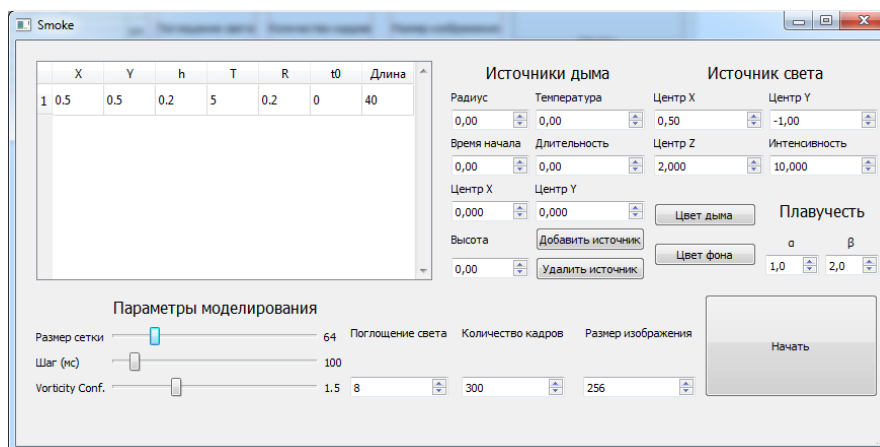


Рис. 5. Интерфейс программы

В программе можно задавать количество источников дыма, их положение, размер сетки, шаг моделирования и другие параметры. После окончания моделирования полученные изображения конвертируются в видеофайл посредством программы ffmpeg. Диаграмма классов разработанной программы представлена на рис. 6.

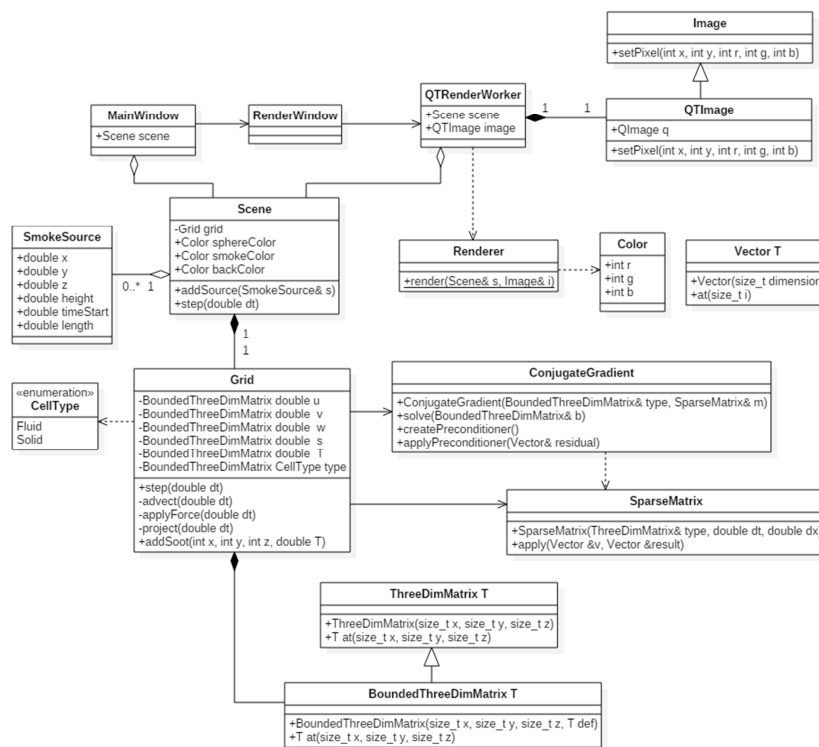


Рис.6. Диаграмма классов программы

Результаты работы программы представлены на рис. 7-10. Как видно, при увеличении коэффициента vorticity confinement с 0 (рис. 7) до 1.5 (рис. 8) повышает визуальное качество полученного изображения, но при слишком большом значении (рис 9) появляется слишком много вихрей, из-за чего реалистичность изображения резко падает. Из этого следует, что при практическом применении данного метода необходимо аккуратно подбирать значение параметра vorticity confinement.

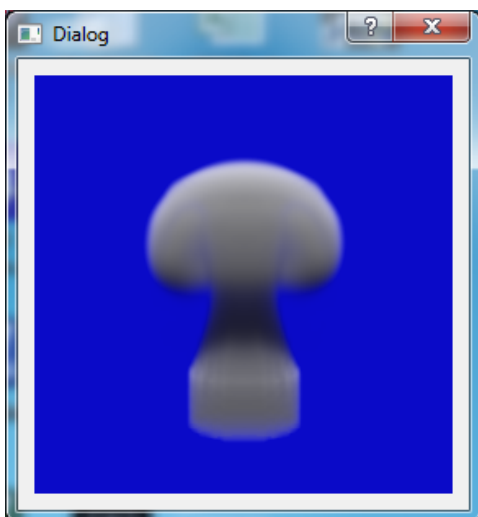


Рис. 8. Коэффициент vorticity confinement 0

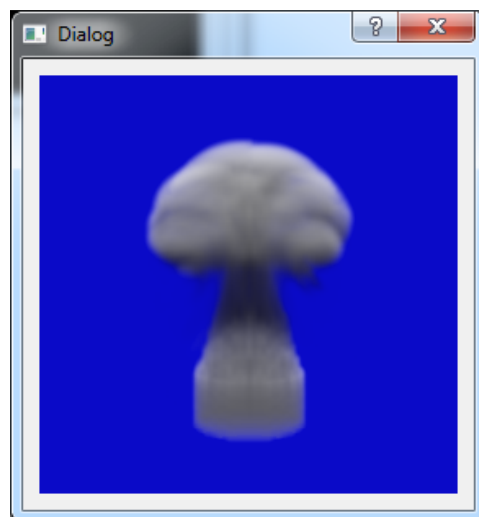


Рис. 9. Коэффициент vorticity confinement 1,5

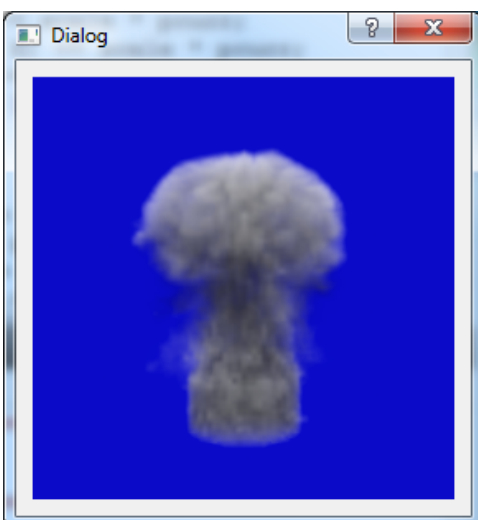


Рис. 10. Коэффициент vorticity confinement 5

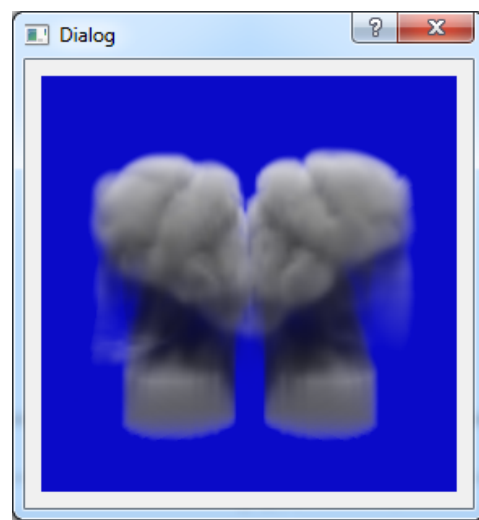


Рис. 11. Два источника дыма, коэффициент vorticity confinement 1,5

Оценка производительности

Были проведены замеры времени моделирования дыма от количества ячеек в моделируемом пространстве. Полученные результаты представлены на рис. 12. Как видно из рисунка 12, зависимость времени моделирования от разбиения сетки не соответствует полученным выше результатам – время моделирования растёт намного медленнее, чем предсказывает анализ. Возможно, это связано с явлением сверхлинейной сходимости, присущей методу сопряжённых градиентов [7].



Рис. 12. График зависимости среднего времени моделирования от разбиения сетки

Заключение

В данной работе был представлен метод моделирования дыма, позволяющий получить необходимые визуальные характеристики дымового облака, и метод рендеринга данного облака. Было разработано программное обеспечение, в основу которого были положены данные методы.

Список литературы

- [1] Chorin A. J. Numerical solution of the Navier-Stokes equations // Mathematics of computation. 1968. Vol. 22. № 104. pp. 745-762. DOI: 10.1090/S0025-5718-1968-0242392-2.
- [2] Harlow F., Welch. J. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface // Physics of fluids. 1965. Vol. 8. №12. pp. 2182-2189. DOI: 10.1063/1.1761178.
- [3] Stam J. Stable fluids // Proceedings of the 26th annual conference on Computer graphics and interactive techniques. NY: ACM Press/Addison-Wesley Publishing Co. 1999. P. 121-128.
- [4] Fedkiw R., Stam J., Jensen H. W. Visual simulation of smoke // Proceedings of the 28th annual conference on Computer graphics and interactive techniques. NY: ACM Press, 2001. P. 15-22.
- [5] Bridson R. Fluid simulation for computer graphics. Boca Raton, FL: CRC Press, 2008. 246 p.

- [6] Barrett R. et al. Templates for the solution of linear systems: building blocks for iterative methods. Philadelphia, PA: SIAM, 1994. P.118. DOI: 10.1137/1.9781611971538.
- [7] Herzog R., Sachs E. Superlinear Convergence of Krylov Subspace Methods for Self-Adjoint Problems in Hilbert Space // SIAM Journal on Numerical Analysis. 2015. Vol. 53. № 3. P. 1304-1324. DOI: 10.1137/140973050.