

УДК 519.6

**ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ
ЛАМИНАРНОГО ПЛАМЕНИ**А. А. Зоткевич¹

Предложен новый алгоритм моделирования распространения ламинарного пламени, основная идея которого состоит в расчете сеточной энтальпии на грубой пространственно-временной сетке, в отличие от концентрации, которая считается на точной сетке. Показана эффективность алгоритма. Кроме того, проведено исследование эффективности табулирования экспоненты и распараллеливания программного кода на системах с общей памятью при помощи OpenMP. Рассмотрены некоторые аспекты алгоритмической и программной реализации моделирования процесса ламинарного горения, в том числе описана структура применяемой пространственной сетки. Работа выполнена при финансовой поддержке РФФИ (код проекта 07-01-00164).

Ключевые слова: ламинарное горение, адаптивные сетки, явные схемы.

1. Введение. Известно, что задачи теории горения и, в частности, задача о распространении ламинарного газового пламени отличаются большой разномасштабностью протекающих процессов. На эту тему имеется обширная литература (см., например, [1] и цитируемые там работы). При газовом горении обычно выделяются три масштаба: внешний (газодинамический), средний (диффузионный) и внутренний (кинетический). Наибольшие трудности для численного моделирования представляет взаимодействие диффузионного и кинетического масштабов. Это взаимодействие порождает такую важную макрохарактеристику, как нормальная скорость распространения пламени u_n . Попытки численно определить величину u_n не только качественно, но и количественно встречаются наибольшие трудности.

Численные расчеты для одномерных моделей проводятся уже давно. По-видимому, первой в этом направлении работой является [2], а наибольший успех был достигнут в работе [3], в которой было осуществлено численное моделирование всего нестационарного процесса, начиная с инициализации горения и заканчивая выходом на стационарный режим распространения пламени. При этом основу численной методики составляли рассмотренные в статье [4] балансные монотонные неявные разностные схемы с использованием сгущающихся в зоне горения сеток. Работы, связанные с адаптацией одномерных разностных сеток под структуру волны горения, достаточно популярны и в настоящее время (см., например, [5] и цитируемые там статьи). К исследованиям по численным методикам решения многомерных задач о распространении пламени и их применению следует отнести [6–11] и многие другие. Все эти исследования используют адаптацию разностной сетки под структуру фронта горения (см., например, [12] и цитируемую там литературу).

Помимо адаптации пространственной сетки, существует возможность сгущать шаги по времени в некоторой области. В работе [13] численные расчеты распространения фронта ламинарного пламени показали хорошую работоспособность так называемых двухуровневых явных схем [14–16]. Полный алгоритм численного решения задачи моделирования ламинарного пламени в двумерном случае приводится в работе [17]. Алгоритм основан на двухуровневой явной схеме, описанной в работе [16], а также на использовании адаптивной пространственной сетки. При этом уравнения ламинарного пламени выписываются в энтальпийной постановке. Однако необходимо отметить, что для сложных примеров с относительно большим количеством степеней свободы время расчета достаточно велико.

В настоящей статье описана одна модификация алгоритма решения задачи, основанная на априорных знаниях о характере протекающих процессов и позволяющая существенно сократить общее время расчета. Не менее важной частью работы является описание некоторых ключевых особенностей алгоритма и структуры данных сетки, не опубликованных в [13, 17]. Необходимо отметить, что многие идеи, использованные при построении сетки, обсуждались с автором [18]. Наконец, несколько частей статьи посвящены краткому сравнительному исследованию различных подходов к программной реализации некоторых частей алгоритма.

¹ Институт вычислительной математики и математической геофизики СО РАН, просп. акад. Лаврентьева, 6, 630090, Новосибирск; науч. сотр., e-mail: aaz@labchem.sccc.ru

Предлагаемая статья организована следующим образом. В разделе 2 рассматривается постановка задачи и аппроксимация уравнений. В третьем разделе приводятся данные о структуре пространственной сетки и ее программной реализации. Обсуждается наглядный пример. Приводятся также важные особенности алгоритма, касающиеся построения матриц МКЭ. Четвертый раздел посвящен описанию физических и алгоритмических параметров. В пятом разделе описан алгоритм решения, а также его модификация, основанная на переносе сеточной энтальпии на грубую пространственно-временную сетку. Дан пример, проведен подбор необходимых параметров и приведены результаты численного эксперимента. В шестом разделе описаны некоторые подробности программной реализации алгоритма, касающиеся вычисления экспоненты и сборки матриц метода конечных элементов (МКЭ). В седьмом разделе говорится о распараллеливании части программы на основе OpenMP. Наконец, восьмой раздел посвящен выводам и некоторым перспективным направлениям для дальнейших исследований.

2. Постановка задачи. Аппроксимация уравнений. Рассматривается простейшая модель движения фронта пламени между двумя бесконечными в направлении одной из осей пластинами. Область решения представляет собой произвольное прямоугольное сечение $\Omega = (0, L) \times (0, l)$, $l \ll L$. Пусть T — температура газа, η — относительная концентрация реагирующего компонента горючей смеси, c_p — удельная теплоемкость газа при постоянном давлении, a — коэффициент температуропроводности, Q — тепловой эффект реакции, проходящей в соответствии с законом Аррениуса

$$W(T, \eta) = k_0 \eta e^{-E/RT}, \quad (2.1)$$

где k_0 — предэкспонент, E — энергия активации, R — универсальная газовая постоянная и Θ — энтальпия: $T = \Theta - \frac{Q}{c_p} \eta$.

Двумерная модель распространения ламинарного пламени с числом Льюиса $Le = 1$ в энтальпийной постановке выглядит следующим образом:

$$\frac{\partial \Theta}{\partial t} = a \left(\frac{\partial^2 \Theta}{\partial x^2} + \frac{\partial^2 \Theta}{\partial y^2} \right), \quad (2.2)$$

$$\frac{\partial \eta}{\partial t} = a \left(\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} \right) - W(T, \eta). \quad (2.3)$$

Формально первое уравнение этой системы не зависит от второго. Фактическая зависимость Θ от η реализуется через краевые условия:

$$\text{— условия при } x = 0: \quad \Theta = T_0 + \frac{Q}{c_p}, \quad \eta = 1; \quad (2.4)$$

$$\text{— условия при } x = L: \quad \frac{\partial \Theta}{\partial x} = \frac{\partial \eta}{\partial x} = 0; \quad (2.5)$$

$$\text{— условия при } y = 0: \quad \frac{\partial \Theta}{\partial y} - \alpha_0 \left(\Theta - \frac{Q}{c_p} \eta - T_0 \right) = \frac{\partial \eta}{\partial y} = 0; \quad (2.6)$$

$$\text{— условия при } y = l: \quad \frac{\partial \Theta}{\partial y} + \alpha_l \left(\Theta - \frac{Q}{c_p} \eta - T_0 \right) = \frac{\partial \eta}{\partial y} = 0. \quad (2.7)$$

Здесь α_0 и α_l — положительные коэффициенты теплообмена с внешней средой (с температурой T_0) через “боковые” стенки ($y = 0$, $y = l$). Кроме того, заданы начальные распределения температуры и концентрации в момент времени $t = 0$:

$$T = T^0(x, y), \quad \eta = \eta^0(x, y), \quad \Theta = T^0(x, y) + \frac{Q}{c_p} \eta^0(x, y). \quad (2.8)$$

Аппроксимированная по пространственным переменным задача (2.2)–(2.8) может быть записана как задача Коши для системы обыкновенных дифференциальных уравнений вида

$$M \frac{dv}{dt} + Av = f(v), \quad t > 0, \quad (2.9)$$

$$v = v^0, \quad t = 0, \quad (2.10)$$

где A — сеточный оператор диффузии и M — диагональная матрица масс. Компоненты вектора v включают в себя значения сеточных функций Θ и η . Разобьем вектор v на два подвектора, содержащих две группы переменных: к первой группе отнесем значения, соответствующие сеточным узлам из

диффузионной зоны, а ко второй группе — из зоны горения (сгущения сетки). Отметим, что поскольку в правую часть не входят аппроксимации дифференциальных операторов, то $f_k(v) = f_k(u_k)$, $k = 1, 2$.

Соответствующим образом разобьется на блоки матрица A : $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix}$, $\|A_{11}\|_{(1)} \ll \|A_{22}\|_{(2)}$.

Для введенных обозначений двухуровневая разностная схема решения задачи (2.9), (2.10) имеет вид

$$M \frac{u_1^{n+1} - u_1^n}{\Delta t} + A_{11}u_1^n + A_{12}u_2^n = f_1(u_1^n), \tag{2.11}$$

$$u_1^{n+k/p} = u_1^n + \frac{k}{p}(u_1^{n+1} - u_1^n), \quad k = 1, \dots, p-1, \tag{2.12}$$

$$M \frac{u_2^{n+(k+1)/p} - u_2^{n+k/p}}{\tau} + A_{12}^T u_1^{n+k/p} + A_{22}u_2^{n+k/p} = f_2(u_2^{n+k/p}), \quad k = 0, \dots, p-1. \tag{2.13}$$

Отметим, что в отличие от [17] нами была использована схема из работы [14]. Как показал численный эксперимент, при реализации задачи с заданными параметрами и используемым пространственным делением по зонам, первый порядок точности по времени в диффузионной зоне является достаточным. В то же время количество требуемых на одном шаге по времени операций очевидным образом меньше у схемы из [14]. Основным же преимуществом этой схемы является более простая программная реализация и возможная дальнейшая модификация. Результаты численного эксперимента не приводятся по причине малого различия между временами выполнения программы с использованием разных алгоритмов. Это обусловлено тем, что различия присутствуют только в зоне диффузии, где количество узлов сетки, а также пространственных шагов на порядок меньше, чем в зоне реакции.

3. Сетка. Пространственная дискретизация задачи. Описанная в настоящей статье сетка предназначена для триангуляции прямоугольных областей с возможностью сгущений в заданных под-областях. Как будет видно далее, существуют возможности описания более сложных областей, однако в рамках проделанной работы это не требовалось. Описать структуру сетки будет проще всего при помощи алгоритма ее построения.

Пусть, для удобства, границы области параллельны координатным осям OX и OY . Применяемая здесь сетка состоит из конечного числа так называемых “уровней”, отчет начинается с нулевого. Пусть исходная прямоугольная область разбита прямыми, параллельными границам исходной области таким образом, что расстояние между параллельными прямыми фиксировано, а через каждый из четырех граничных отрезков области проходит строго одна прямая. Полученное разбиение области на равные прямоугольники $\sigma_k^{(0)}$, $k = 0, \dots, K_0$, с длинами сторон $h_X^{(0)}$ и $h_Y^{(0)}$ будет образовывать нулевой уровень сетки $\Omega_h^{(0)}$: $\Omega_h^{(0)} = \bigcup \sigma_k^{(0)}$, $k = 0, \dots, K_0$. При этом по каждой из координат область будет разбита на слои равной толщины. Здесь и далее образующий уровень прямоугольник $\sigma^{(0)}$ назовем “ячейкой”. Каждая ячейка имеет две координаты, каждая из которых есть номер слоя, которому она принадлежит. Ключевым моментом структуры сетки является то, что четыре ячейки уровня $k+1$ могут получиться только путем разбиения ячейки уровня k на четыре равных прямоугольника с одной общей для всех прямоугольников точкой в центре исходной ячейки. Уровень 1 строится следующим образом: в соответствии с некоторым критерием \tilde{K}_1 ячеек уровня 0 разбиваются на $K_1 = 4\tilde{K}_1$ ячеек уровня 1. Количество слоев по каждой из координат для уровня 1 в два раза больше, чем для уровня 0. Отношение длин сторон ячеек уровней 0 и 1 есть 2:1. Теперь сетка состоит из двух уровней: $\Omega_h^{(0)} = \bigcup \sigma_k^{(0)}$, $k = 0, \dots, K_0 - \tilde{K}_1$, и $\Omega_h^{(1)} = \bigcup \sigma_k^{(1)}$, $k = 0, \dots, K_1$. Аналогичным путем получают остальные уровни сетки. В итоге получим

$$\begin{aligned} \Omega_h^{(0)} &= \bigcup \sigma_k^{(0)}, & k &= 0, \dots, K_0 - \tilde{K}_1; & \Omega_h^{(1)} &= \bigcup \sigma_k^{(1)}, & k &= 0, \dots, K_1 - \tilde{K}_2; \\ \dots & & \dots & & \dots & & \dots & \\ \Omega_h^{(N-1)} &= \bigcup \sigma_k^{(N-1)}, & k &= 0, \dots, K_{N-1} - \tilde{K}_N; & \Omega_h^{(N)} &= \bigcup \sigma_k^{(N)}, & k &= 0, \dots, K_N. \end{aligned}$$

Сетка есть объединение всех уровней: $\Omega_h = \bigcup \Omega_h^{(n)}$, $n = 0, \dots, N$. Далее выполняется процедура упорядочивания сетки таким образом, чтобы общие точки могли быть только у “соседних” уровней n и $n+1$. Важно, что в любой момент времени расчет ведется только на упорядоченной таким образом сетке. Множество ячеек, образующих переходную зону от множества $\Omega_h^{(n+1)}$ к множеству $\Omega_h^{(n)}$, обозначим через $\Omega_h^{(n,n+1)}$, $\Omega_h^{(n,n+1)} \subseteq \Omega_h^{(n)}$.

Триангуляция ячеек из множества $\Omega_h^{(n,n+1)}$ порождает ячейки восьми типов, приведенных на рис. 1. Ячейки, не принадлежащие множеству $\bigcup_{n=0}^{N-1} \Omega_h^{(n,n+1)}$, триангулируются диагональю.

Итак, нулевой уровень сетки состоит из ячеек максимальной величины, последний уровень — из ячеек минимальной. Переход от большого пространственного шага к меньшему осуществляется постепенным сгущением, при этом стыки уровней организованы при помощи ячеек восьми типов. Теперь рассмотрим механизм, в соответствии с которым разбивается ячейка. В основе такого механизма лежит функция плотности сетки $\Phi(x, y)$. Рассмотрим некоторую ячейку $\sigma^{(n)}$ с локально пронумерованными вершинами $z_i = (x_i, y_i)$, $i = 1, \dots, 4$, и пусть $\bar{z}_\sigma = \frac{1}{4} \sum_{i=1}^4 z_i$, $\bar{\Phi}_\sigma = \frac{1}{4} \sum_{i=1}^4 \Phi(z_i)$.

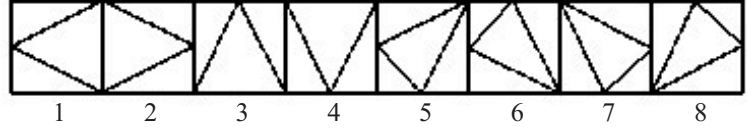


Рис. 1. Типы ячеек

Критерием разбиения ячейки $\sigma^{(n)}$ на четыре ячейки $(n+1)$ -го уровня является условие $|\bar{\Phi}_\sigma - \Phi(\bar{z}_\sigma)| > \varepsilon_1$. Для сетки, которая пересчитывается во времени, необходимо объединить четыре ячейки $\sigma_j^{(n)}$ с общей вершиной в одну ячейку $(n-1)$ -го уровня. Критерием такого объединения является условие $|\bar{\Phi}_{\sigma_j} - \Phi(\bar{z}_{\sigma_j})| < \varepsilon_2$, $j = 1, \dots, 4$. Экспериментальным путем было найдено оптимальное соотношение параметров ε_1 и ε_2 , позволяющее наилучшим образом отслеживать при изменении сетки вид функции плотности: $\varepsilon_1/\varepsilon_2 = 4$.

Перейдем теперь к программной реализации. Напомним, что границы области взяты, для удобства, параллельными осям координат. Алгоритмически каждый уровень состоит из двух частей, назовем их Ног и ЕС. Первая часть отвечает за регулярные ячейки, которые триангулируются диагональю. Вторая отвечает за ячейки, изображенные на рис. 1. Пусть количество слоев для описываемого уровня равно SNX и SNY. И Ног, и ЕС описываются следующей структурой:

```
struct MESH
{
    массив Level[SNY][NP];
    массив Index[SNY];
};
```

Для Ног структура будет заполнена следующим образом: элемент в массиве Index укажет на номер слоя i , следовательно, на положение ячейки в пространстве по координате Y . Пусть в Index[m]-м слое по Y присутствуют несколько соседних ячеек описываемого уровня. Тогда все эти ячейки будут описаны в массиве Level[m] всего парой чисел, а именно номером слоя по X первой ячейки и номером слоя по X последней ячейки.

Структура для ЕС отличается от структуры для Ног лишь тем, что пары чисел в Level[m] составляют не ссылки на первую и последнюю ячейки непрерывного массива, а номер слоя, в котором находится нестандартная ячейка, а затем ее тип. Здесь необходимо заметить, что замечание о возможности расширения возможностей сетки касалось именно этой части статьи: очень просто увеличить множество “нестандартных” ячеек.

Простой пример приведен на рис. 2. Ячейки с максимальной длиной стороны образуют нулевой уровень. Таким образом, количество уровней равно 3, номер максимального уровня равен 2. Хранимая в памяти вычислительной машины информация будет выглядеть следующим образом.

```
===== Уровень 0 =====
Ног:{
Index: [0,1,2,3]
Level: [[0,0] [0,0] [0,0] [0,0]]
// четыре пары чисел, каждая описывает одну ячейку
}
ЕС:{
```

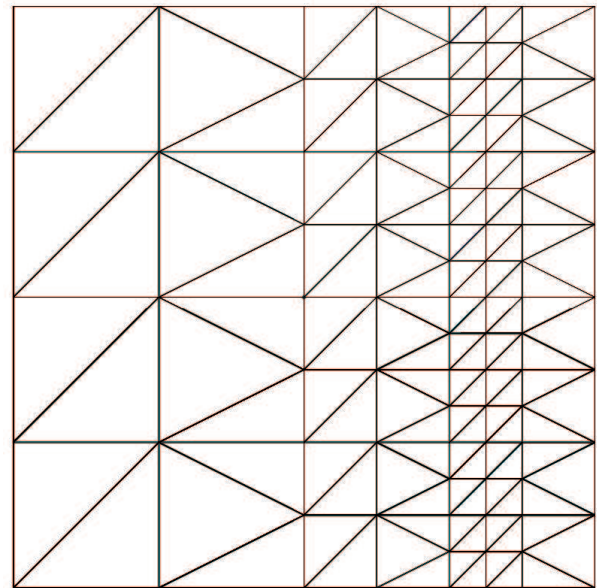


Рис. 2. Пример сетки

```

Index: [0,1,2,3]
Level: [[1, 2], [1, 2], [1, 2], [1, 2]]
// четыре "нестандартные" ячейки типа 2 в первом слое
}
===== Уровень 1 =====
Ног:{
Index: [0,1,2,3,4,5,6,7]
Level: [[4,4] [4,4] [4,4] [4,4], [4,4] [4,4] [4,4] [4,4]]
// восемь пар чисел, каждая описывает одну ячейку
}
ЕС:{
Index: [0,1,2,3,4,5,6,7], алгоритм решения
Level: [[5,2,7,1], [5,2,7,1], [5,2,7,1], [5,2,7,1], [5,2,7,1],
[5,2,7,1], [5,2,7,1], [5,2,7,1]]
// в каждой строке две "нестандартные" ячейки:
// типа 2(5й слой по X) и типа 1(7й слой по X)
}
===== Уровень 2 =====
Ног:{
Index: [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
Level: [[12,13], [12,13], [12,13], [12,13], [12,13], [12,13], [12,13],
[12,13], [12,13], [12,13], [12,13], [12,13], [12,13], [12,13],
[12,13]]
// шестнадцать пар, каждая описывает две ячейки
}
ЕС://{// нестандартные ячейки отсутствуют}

```

Перейдем теперь собственно к задаче моделирования ламинарного горения. Для пространственной дискретизации задачи (2.2)–(2.8) используется стандартный конформный метод конечных элементов с кусочно-линейной аппроксимацией на триангулированной сетке [19]. При этом осуществляется диагонализация матрицы масс (элементами диагонализированной матрицы являются суммы элементов строки исходной матрицы). Как известно [20], такая процедура не снижает точности вычислений. Отметим, что без диагонализации матрицы масс исчезают все алгоритмические преимущества явных схем. Аналогичным образом аппроксимируется нелинейное слагаемое: $\int_{\Omega} W(T_h, \eta_h) \varphi_i dx dy \approx \text{mes } \omega_i W(T_i, \eta_i)$, где φ_i —

кусочно-линейная базисная функция-“крышка” с носителем $s_i \subset \Omega$, ω_i — барицентрическое множество [20] сеточного узла (x_i, y_i) , $T_i = T_h(x_i, y_i)$, $\eta_i = \eta_h(x_i, y_i)$. Здесь и далее $T_h(x, y)$ и $\eta_h(x, y)$ — кусочно-линейные функции. В приведенных терминах диагонализированная матрица масс есть $M = \text{diag} \{ \text{mes } \omega_i \}$.

В качестве функции плотности используется функция скорости реакции (2.1) на n -м временном слое: $\Phi(x, y) = W(T_h^n(x, y), \eta_h^n(x, y))$. Необходимо подчеркнуть, что критерием сгущения сетки являются не большой градиент концентрации $\eta_h(x, y)$, а градиент потока массы, который и определяется, главным образом, функцией скорости реакции.

Описанная выше структура сетки позволяет заведомо построить локальные матрицы МКЭ для ячеек сетки, остается лишь решить вопрос с подстановкой их в глобальные матрицы. Для этого очень удобно ввести вектор, который обозначим как RefAg. Введем сначала равномерную сетку, покрывающую всю область и имеющую длину стороны ячейки, равную длине стороны ячейки максимального уровня N сетки Ω_h . Обозначим такую сетку через Ω_h^N . Очевидно, что множество узлов Ω_h вложено в множество узлов Ω_h^N . Для построенной сетки введем некоторую фиксированную нумерацию узлов. Назовем эту нумерацию “глобальной”, а нумерацию узлов сетки Ω_h “локальной”. Длина вектора RefAg равна количеству узлов Ω_h^N , а номер элемента соответствует номеру узла в глобальной нумерации. При этом элемент вектора равен -1 в случае, если соответствующий узел не принадлежит Ω_h , и равен номеру узла в локальной нумерации, если принадлежит. При поэлементной сборке легко получить по сеточной информации номер узла в глобальной нумерации, а затем, при помощи вектора RefAg, номер в “локальной”.

Серьезнейший недостаток такого вектора — его большой размер. Например, в приведенном в данной статье численном эксперименте при числе степеней свободы около 18000 в нем более 3 миллионов элементов. Поэтому в случае недостатка оперативной памяти вектор следует заменить, например, на ассоциативный контейнер map из стандартной библиотеки STL, как это сделано в нашей работе. Этот

контейнер хранит пары элементов <ключ, значение> и оптимизирован относительно процедуры поиска. Элементы тар хранятся упорядоченными по ключу. В качестве ключа тар используется номер в глобальной нумерации, а в качестве значения — в локальной. Недостаток контейнера тар по сравнению с вектором — дополнительное время на процедуру поиска. В статье будет проведено сравнительное тестирование вариантов программы с тар и с вектором.

На рис. 3 приведен портрет матрицы, строящейся при помощи цикла по сетке от верхнего по номеру уровня к нижнему. Вертикальными и горизонтальными прямыми разделены уровни.

4. Значения параметров. Платформа тестирования. Все дальнейшие эксперименты проводились при следующих значениях физических параметров: $L = 0.1$ м, l варьируется для получения различных значений размерности вектора правой части, $a = 8 \times 10^{-5}$ м²/сек, $T_0 = 300$ К, $T_b = 1400$ К, $k_0 = 10^{12}$ сек⁻¹, $E/R = 2 \times 10^4$ К, $\alpha_0 = \alpha_l = 0.1$, т.е. рассматривается задача с теплоотводом через “боковые” стенки. В качестве начальных данных в области Ω задавались функции $\Theta^0(x, y) \equiv T_b$, $\eta^0(x, y) = 1$ при $x \leq 0.09$ и $\eta^0(x, y) = 0$ при $x > 0.09$.

Подбор алгоритмических параметров, обеспечивающих устойчивость и воспроизводимость численных результатов, был показан в [17]. Для настоящей статьи набор был лишь немного изменен. Использованы следующие оптимальные величины: $\Delta t = 2 \times 10^{-5}$ сек., $p = 20$. Что касается пространственной дискретизации, отношение между максимальной и минимальной длиной стороны базовой квадратной ячейки составляло величину 2^6 . Это означает, что сетка содержит семь “уровней”, включая базовый нулевой: $N = 6$. Пересчет сетки производился каждый шестой шаг по времени: $RS = 6$. Поскольку целью данной работы являлось исследование эффективности оптимизации задачи, а не собственно моделирование процесса ламинарного горения, как в [17], то с целью экономии можно взять небольшой промежуток времени. Во всей серии экспериментов $\Delta T = 2 \times 10^{-3}$ с.

Все расчеты проводились на компьютере с четырехъядерным процессором Intel®Core 2 Quad Q6600, операционная система Ubuntu 8.04 (hardy), компилятор Intel®Compiler (icpc) версии 10.1.015. Для матрично-векторного умножения использовались функции библиотеки Intel®Math Kernel Library (MKL) версии 10.0.3.020 [21].

5. Оптимизация с переносом Θ на грубую сетку. Немодифицированный алгоритм решения задачи выглядит следующим образом.

Обработка входных данных

Инициализация глобальных переменных

Построение сетки в начальный момент времени

Инициализация сеточных данных

Построение матриц МКЭ

```
for (int i=0; i<TT; i++)
```

```
{
```

```
    Реализация уравнения (2.11) для концентрации
```

```
    Реализация уравнения (2.11) для энтальпии
```

```
    Интерполяция: реализация уравнения (2.12) для концентрации и энтальпии
```

```
    for (int k=0; k<p; k++)
```

```
    {
```

```
        Реализация уравнения (2.13) для концентрации
```

```
        Реализация уравнения (2.13) для энтальпии
```

```
    }
```

```
    if (i+1 кратно RS)
```

```
    {
```

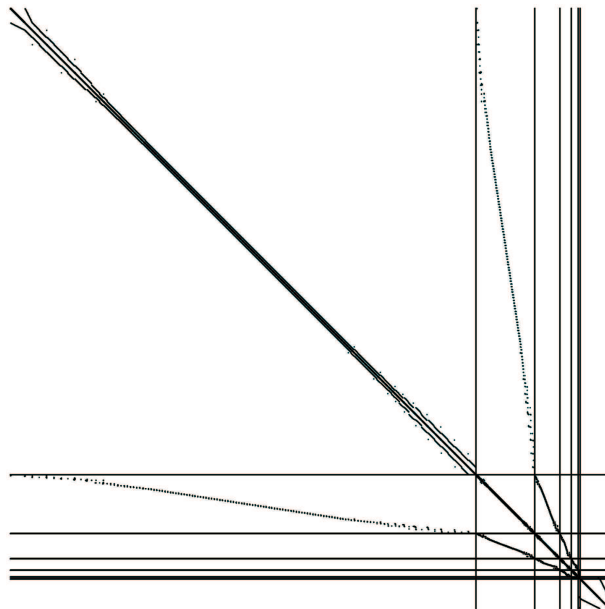


Рис. 3. Портрет матрицы

```

    Перестройка сетки
    Перенос сеточного решения на новую сетку
    Построение новых матриц МКЭ
}
}

```

Здесь $TT = \Delta T / \Delta t$ — количество шагов по времени в области больших градиентов. Сетка перестраивается каждые RS шагов по времени. Отметим здесь, что, поскольку элементы матриц МКЭ не зависят от времени, пересчитывать матрицы необходимо только при изменении расчетной сетки.

Методика оптимизации основана на том факте, что энтальпия Θ является значительно более гладкой функцией, чем концентрация η , причем как по пространственным, так и по временным переменным. Это позволяет использовать большие шаги по времени и более грубую пространственную сетку. Пусть N_{con} есть максимальный номер уровня для концентрации, а N_{enth} — для энтальпии. По аналогии, пусть p_{con} и p_{enth} — количество подшагов в уравнениях (2.12) и (2.13) для соответствующих величин. Модифицированный алгоритм выглядит следующим образом:

```

Обработка входных данных
Инициализация глобальных переменных
Построение точной сетки в начальный момент времени
Построение грубой сетки в начальный момент времени
Инициализация сеточных данных
Построение матриц МКЭ для энтальпии
Построение матриц МКЭ для концентрации
for (int i=0; i<TT; i++)
{
    Реализация уравнения (2.11) для концентрации
    Реализация уравнения (2.11) для энтальпии
    Интерполяция: реализация уравнений (2.12) для концентрации и энтальпии

    for (int k=0; k<pEnth; k++)
    {
        Пересчет сеточной энтальпии на точную сетку
        for (int l=0; l<(pCon/pEnth); l++)
        {
            Реализация уравнений (2.13) для концентрации
        }
        Реализация уравнений (2.13) для энтальпии
    }
    if (i+1 кратно RS)
    {
        Перестройка точной сетки
        Перестройка грубой сетки
        Перенос сеточного решения на новые сетки
        Построение новых матриц МКЭ для энтальпии
        Построение новых матриц МКЭ для концентрации
    }
}
}

```

Как мы видим, такой модифицированный алгоритм, с одной стороны, предполагает некоторое увеличение числа операций. Рассмотрим, за счет чего это происходит. Во-первых, это построение грубой сетки. Но данная операция не требует вычислительных затрат, сравнимых с построением точной сетки, поскольку благодаря иерархической структуре уровни с 0 по $N_{\text{enth}} - 1$ просто копируются из точной сетки. Во-вторых, требует времени построение матриц МКЭ для энтальпии. Однако эта операция также сравнительно малозатратна. В-третьих, требует времени пересчет сеточной энтальпии на точную сетку. Это затратная операция, учитывая то, что такой пересчет необходимо делать p_{enth} раз на каждом шаге по времени. Наконец, как видно из уравнений, учет граничных условий для энтальпии также предполагает некоторые дополнительные вычисления и операции с памятью, поскольку такой учет требует проекции сеточной концентрации на грубую сетку.

Однако, с другой стороны, преимущества модификации очевидны: это сокращение времени матрично-векторного умножения, сокращение времени учета граничных условий для энтальпии и сокращение алгебраических вычислений для энтальпии. Экспериментально были подобраны следующие параметры для модифицированного алгоритма: $N_{\text{con}} = 6$, $N_{\text{enth}} = 4$, $p_{\text{Con}} = 20$, $p_{\text{Enth}} = 1$. Здесь следует отметить, что изначально предполагалось использовать статичную по времени грубую сетку для энтальпии. Это позволило бы избежать пересчета сетки для соответствующего вектора решения. Однако, как мы видим из определенных выше параметров, для этого пришлось бы слишком сильно сгущать сетку во всей области решения, что потребовало бы больших временных затрат. Кроме того, к требуемой в части алгоритма проекции концентрации на грубую сетку в этом случае добавилась бы еще интерполяция.

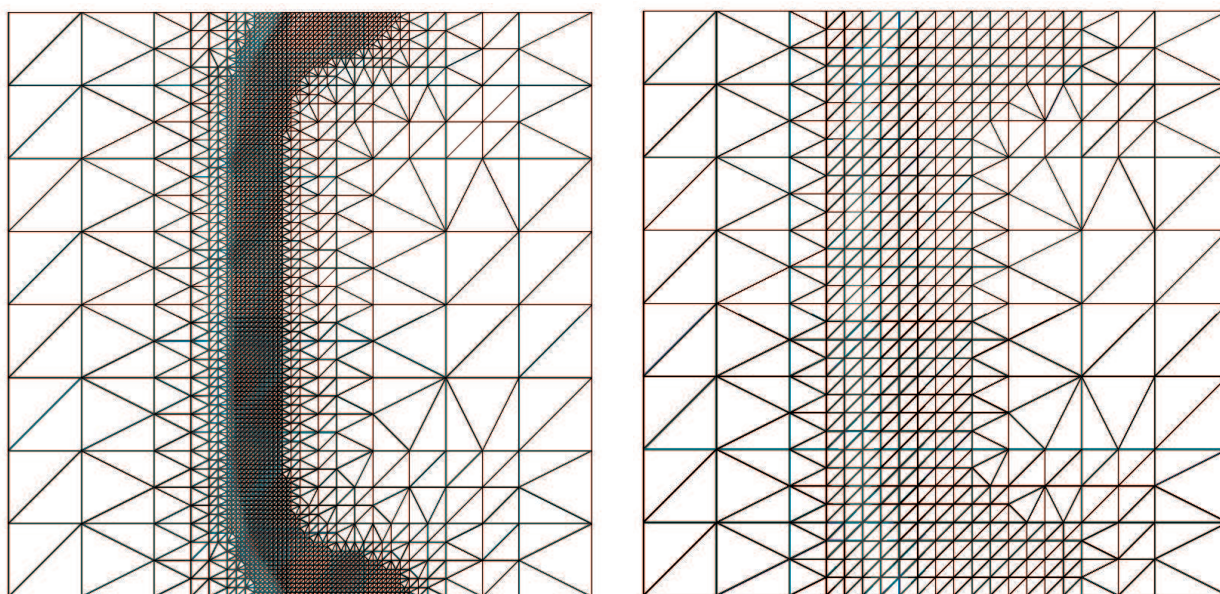


Рис. 4. Сетки для концентрации и энтальпии

На рис. 4 приведен пример сеток для концентрации и энтальпии в один и тот же момент времени при моделировании процесса с указанными выше физическими и алгоритмическими параметрами.

Таблица 1 содержит количество степеней свободы для точной сетки (ТС), грубой сетки (ГС), в диффузионной зоне (ДЗ), а также число ненулевых элементов матриц МКЭ и размерности вектора RefAr. Все данные получены в момент времени $T = 2 \times 10^{-3}$. Номера проставлены для удобства: во всех последующих таблицах будет приводиться номер эксперимента, соответствующий столбцу в табл. 1.

Таблица 1

Размерности массивов

Номер эксперимента	1	2	3	4	5	6	7
Степени свободы ТС	5026	9382	18094	35518	70366	140062	279494
Степени свободы ГС	1054	1954	3754	7354	14544	28954	57754
Степени свободы ДЗ	256	468	892	1740	3436	6828	13612
Матрица на ТС	26534	49818	98386	189522	375794	748338	1493426
Матрица на ГС	5500	10352	20056	39464	78280	155912	311176
RefAr	822657	1642113	3281025	6558849	13114497	26225793	52448385

Как уже сказано выше, выбранные параметры алгоритма являются оптимальными. Это означает, что большие сеточные шаги приведут к неудовлетворительным результатам моделирования. Обратим здесь внимание на размерности вектора RefAr. Именно столько степеней свободы было бы при использовании во всей области равномерной сетки. Для наглядности приведем в табл. 2 объем памяти в мегабайтах, занимаемой векторами правой части и RefAr.

Таблица 2

Размеры массивов

Номер эксперимента	1	2	3	4	5	6	7
Вектор правой части	0,0383	0,0716	0,138	0,271	0,537	1,07	2,13
RefAr	6,28	12,5	25,0	50,0	100,1	200,1	400,1

Таблица 3

Без модификации

Номер эксперимента	1	2	3	4	5	6	7
Перестройка сетки	0,1783	0,364	0,7754	1,7230	4,085	10,52	30,34
Умножение матрицы на вектор	0,6139	1,151	2,28	4,764	11,61	23,91	49,39
Вычисление экспоненты	0,3552	0,6643	1,309	2,543	5,134	10,17	20,17
Остальное	0,5462	0,9641	1,837	4,531	11,14	25,58	61,47
Всего	1,6936	3,1434	6,2014	13,561	31,969	70,18	161,37

Таблица 4

С модификацией

Номер эксперимента	1	2	3	4	5	6	7
Перестройка сетки	0,1913	0,4165	0,8161	1,808	4,2400	10,9	33,97
Умножение матрицы на вектор	0,3619	0,6733	1,302	2,652	5,948	12,45	26,18
Вычисление экспоненты	0,3492	0,6607	1,282	2,535	5,05	10,13	20,19
Остальное	0,3145	0,6	1,168	2,457	5,624	13,17	31,48
Всего	1,2169	2,3505	4,5681	9,452	20,862	46,65	111,82

Перейдем теперь к результатам численного эксперимента. В табл. 3 и 4 приведено время расчета в секундах для двух вариантов программы: с модификацией и без нее. В таблицах для наглядности специально выделены времена выполнения отдельных частей программы. При этом пункт “Остальное” включает в себя:

- интерполяцию сеточной энтальпии на точную сетку; это та самая операция, которая уменьшает эффективность модификации с грубой сеткой;
- все операции с векторами: умножение, сложение, деление и др.;
- учет граничных условий.

При этом учет граничных условий составляет единицы процентов пункта, а в табл. 5 на примере однопоточного приложения показано соотношение времен, затрачиваемых на пункты 1 и 2 из списка выше.

Таблица 5

Отношение времени на арифметические операции с векторами ко времени пересчета вектора решения

1	2	3	4	5	6	7
1,18	1,07	1,03	1,04	1,12	1,2	1,32

Таблица 6

Уменьшение времени выполнения программы при переносе энтальпии на грубую сетку

1	2	3	4	5	6	7
1,39	1,34	1,36	1,43	1,53	1,5	1,44

Небольшое увеличение времени на перестройку сетки и пересчет, как и ожидалось, не сильно влияют на общую эффективность модификации. В табл. 6 приведено отношение времен полного выполнения программы для варианта с модификацией и без. Модификация с переносом сеточной энтальпии на грубую сетку позволяет сократить время выполнения программы в 1,3–1,6 раза, что, несомненно, является очень

хорошим результатом.

6. Оптимизация вычисления экспоненты. В рассмотренных экспериментах существенное время занимает вычисление экспоненты. В то же время существует известная простая методика сокращения необходимого для этого времени, основанная на табулировании экспоненты. Однако вычислительные мощности современных процессоров очень велики, и узким местом часто является пропускная способность памяти. Первая часть данного раздела посвящена исследованию эффективности табулирования экспоненты. Отметим, что, помимо очевидных плюсов табулирования, присутствуют и минусы: необходимость вычисления номера массива, что подразумевает несколько арифметических операций. Кроме того, необходимо время для доступа к элементу массива. Все эти факторы, как будет видно далее, существенно влияют на результат.

Ряд экспериментов показал, что достаточным для исследуемого случая является массив из 12000 элементов. Его инициализация не требует сравнимых с общим временем выполнения программы затрат.

Таблица 7

Времена расчета экспоненты

Номер эксперимента	1	2	3	4	5	6	7
Прямое	0,3492	0,6607	1,282	2,535	5,05	10,13	20,19
Из массива	0,2757	0,4871	0,9346	1,8689	6,7860	13,4900	27,3300
Отношение	1,27	1,36	1,37	1,36	0,74	0,75	0,74

Как видно из табл. 7, лишь только задача выпадает из кэша процессора, эффективность табулирования отрицательная. Поэтому необходимым условием для обеспечения эффективности табулирования экспоненты является постоянное присутствие в кэше процессора всего массива со значениями.

Обратимся теперь к RefAg. Очевидно, что при применении ассоциативного контейнера время вычисления должно увеличиться за счет процедур поиска. Это отлично видно в табл. 8.

Таблица 8

Время выполнения программы для вариантов с различными RefAg

Номер эксперимента	1	2	3	4	5	6	7
Ассоциативный контейнер	1,4828	2,7754	5,538	11,583	25,56	56,77	131,46
Вектор	1,2169	2,3505	4,5681	9,452	20,862	46,65	111,82
Отношение	1,22	1,18	1,21	1,23	1,23	1,22	1,18

Таким образом, до тех пор, пока RefAg полностью умещается в оперативную память процессора, имеет смысл использовать вектор. Как только задача становится достаточно большой, необходимо использовать ассоциативные контейнеры.

7. Распараллеливание. В данной работе не ставится цель полностью распараллелить программный комплекс, состоящий более чем из 6000 строк кода, без учета блока визуализации. Для этого, фактически, необходимо написать новый программный код, особенно это касается алгоритма построения сетки, который предполагает работу с динамическими структурами в памяти. Однако некоторые части кода можно легко распараллеливать на системах с общей памятью на основе OpenMP. Добавление одной строки обеспечивает параллельную реализацию соответствующего цикла: `#pragma omp parallel for private(i)`.

Следующие части программы были распараллелены таким образом: вычисление экспоненты; интерполяция сеточной энтальпии на точную сетку; все операции с векторами: умножение, сложение, деление и др.

Помимо использования OpenMP, легко применить заложенное в библиотеку MKL распараллеливание. Для этого лишь необходимо в качестве параметра MKLThr указать требуемое количество потоков: `mkl_set_num_threads(MKLThr)`.

Таким образом, распараллеливается матрично-векторное умножение. В табл. 9 приведены величины, показывающие уменьшение в разгах времени выполнения частей программы для двух и четырех потоков по сравнению с однопоточным выполнением. Экспонента вычисляется напрямую, вектор RefAg хранится как ассоциативный контейнер, энтальпия считается на грубой сетке.

Таблица 9

Уменьшение времени выполнения блоков программы

Номер эксперимента	1	2	3	4	5	6	7
Умножение матрицы на вектор 1/2	1,55	1,75	1,82	1,75	1,4	1,37	1,38
Умножение матрицы на вектор 1/4	1,84	2,31	2,61	2,15	1,53	1,32	1,33
Вычисление экспоненты 1/2	1,83	1,82	1,82	2,16	2,11	1,81	1,8
Вычисление экспоненты 1/4	3,55	3,59	3,61	4,37	4,22	3,62	3,61
Остальное 1/2	1,74	1,78	1,83	1,77	1,68	1,58	1,47
Остальное 1/4	2,49	2,76	2,9	2,92	2,65	2,38	2,18

В табл. 10 приведен итог: во столько раз уменьшилось время выполнения программы при переходе с одного потока к четырем.

8. Заключение. Главным итогом проделанной работы следует считать то, что показана эффективность аппроксимации энтальпии на грубой пространственно-временной сетке. Такая аппроксимация позволяет серьезнейшим образом увеличить эффективность вычислений. Исследования различных подходов к хранению вектора ссылок, предназначенного для построения глобальных матриц МКЭ, и двух подходов к вычислению экспоненты, также могут представлять интерес.

Получены хорошие результаты в разделе, касающемся распараллеливания процесса. Наконец, разработчикам алгоритмов пространственной дискретизации может быть полезным описание структуры пространственной сетки, используемой в программном комплексе.

Кратко остановимся на перспективных направлениях для дальнейших исследований. Во-первых, интересно было бы построить параллельный алгоритм построения пространственной сетки: это единственная часть программы, которой не коснулось распараллеливание. Во-вторых, дальнейшее увеличение эффективности возможно при переходе на гибридные системы, совмещающие MPI- и OpenMP-подходы: известно, что архитектура настольного процессора, на котором проводились эксперименты, не лучшим образом подходит для параллельных вычислений. Наконец, существует возможность переупорядочивания строк матрицы для создания более компактного портрета, что должно уменьшить время матрично-векторного умножения, а также увеличить эффективность распараллеливания.

Таблица 10

Уменьшение времени выполнения всей программы

1	2	3	4	5	6	7
1,85	1,97	2	1,99	1,75	1,56	1,47

СПИСОК ЛИТЕРАТУРЫ

1. Зельдович Я.Б., Баренблатт Г.И., Либрович В.Б., Махмеддзе Г.М. Математическая теория горения и взрыва. М.: Наука, 1980.
2. Spalding D.B. Theoretical aspect of flame stabilization // Aircraft. Eng. 1953. **25**, N 295. 264–276.
3. Мержанов А.Г., Хайкин Б.И., Шкадинский К.Г. Установление стационарного распространения пламени при зажигании газа накаливаемой поверхностью // Прикл. мех. техн. физ. 1969. № 5. 42–48.
4. Шкадинский К.Г. О разностном счете задач зажигания и горения с учетом диффузии и гидродинамики // Физ. горения и взрыва. 1969. **5**, № 2. 264–272.
5. Демин М.М., Мажукин В.И., Шапранов А.В. Метод динамической адаптации в проблеме ламинарного горения // Журн. вычисл. матем. и матем. физ. 2001. **41**, № 4. 648–661.
6. Aly S.L., Simpson R.B., Hermance C.E. Numerical solution of the two-dimensional premixed laminar flame equations // AIAA J. 1979. **17**. 56–63.
7. Winters K.H., Rae J., Jackson C.P., Cliffe K.A. The finite element method for laminar flow with chemical reactions // Int. J. Numer. Methods Eng. 1981. **17**. 239–253.
8. Braack M., Becker R., Rannacher R., Warnatz J. An adaptive finite element method for combustion problems // Proc. 3-d Summer Conf. Numer. Modelling in Cont. Mechanics. Prague, 1998. 91–100.
9. Рычков А.Д., Шокина Н.Ю. Математические модели фильтрационного горения и их приложения // Вычисл. технологии. 2003. **8**, спец. вып., часть 2. 124–144.
10. Graziadei M. Using local defect correction for laminar flame simulation by Marialuce Valentina Graziadei. Eindhoven: Technische Universiteit Eindhoven, 2004.
11. Ten Thije Boonkkamp J.H.M., Rook R., Mattheij R.M.M. A multi-level local defect correction technique for laminar

- flame simulation. CASA Report N 07-27. Eindhoven: Technische Universiteit Eindhoven, 2007.
12. *Braack M.* An adaptive finite element method for reactive-flow problems. Ph.D. Thesis. Ruprecht-Karls University of Heidelberg, 1998.
 13. *Зоткевич А.А.* Численное моделирование распространения фронта пламени в двумерном случае // Тр. конференции молодых ученых. Новосибирск: ИВМиМГ СО РАН, 2004. 61–67.
 14. *Лаевский Ю.М., Банушкина П.В.* Составные явные схемы // Сиб. журн. вычисл. матем. 2000. **3**, № 2. 165–180.
 15. *Banushkina P.V., Laevsky Yu.M.* Multi-level explicit schemes and their stability // Rus. J. Numer. Anal. Math. Model. 2001. **16**, N 3. 215–233.
 16. *Зоткевич А.А., Лаевский Ю.М.* Об одном классе двухуровневых явных схем // Сибир. журн. вычисл. матем. 2002. **5**, № 2. 163–173.
 17. *Зоткевич А.А., Лаевский Ю.М.* Численное моделирование распространения ламинарного пламени на основе двухуровневых явных схем // Вычисл. технологии. 2006. **11**, № 6. 31–43.
 18. *Sander I.A.* The program of Delaunay triangulation construction for the domain with the piecewise smooth boundary // Bull. of the Novosibirsk Computing Center. 1998. Ser.: Numerical Analysis. Issue 8. 71–79.
 19. *Сьярле Ф.* Метод конечных элементов для эллиптических задач. М.: Мир, 1980.
 20. *Лаевский Ю.М.* Метод конечных элементов решения многомерных параболических уравнений. Новосибирск: Изд-во НГУ, 1993.
 21. <http://www.intel.com/cd/software/products/asmo-na/eng/266853.htm>

Поступила в редакцию
26.03.2009
