

УДК 533.6.011.3; 519.688

## **Опыт создания программного кода на основе метода Галеркина с разрывными базисными функциями высокого порядка точности**

**Подаруев В.Ю.**

*Центральный аэрогидродинамический институт имени профессора Н. Е. Жуковского,*

*ЦАГИ, улица Жуковского, 1, Жуковский, Московская обл., 140180, Россия*

*e-mail: [vladimir.podaruev@tsagi.ru](mailto:vladimir.podaruev@tsagi.ru)*

### **Аннотация**

Приведено краткое описание программного кода, основанного на методе Галеркина с разрывными базисными функциями (РМГ) высокого порядка точности. Обсуждаются основные черты метода, такие, как реконструкция консервативных переменных, аппроксимация конвективных, диффузионных и источниковых членов, Гауссовы квадратуры, учет кривизны поверхности, преобразования координат при помощи “серендиповых” элементов. Приводятся примеры, демонстрирующие работоспособность кода и качество получаемых результатов.

**Ключевые слова:** разрывный Метод Галеркина, базисные функции, высокий порядок, суперкомпьютер, верификация, валидация, крыло с механизацией.

### **Введение**

В настоящее время существует большое количество методов высокого порядка точности. Они предназначены для решения задач различных типов. Для удобства

классификации выделяют три большие группы: конечно-разностные методы (КРМ), методы конечного объема (МКО) и методы конечных элементов (МКЭ). Метод РМГ относится к группе МКЭ. Так как тема существует достаточно давно, то можно найти обзоры (например [1, 2]) в которых многие из перечисленных методов досконально описаны. Отмечается, что построение КРМ для решения систем уравнений в частных производных является наиболее простой задачей. Для этого строится разностная сетка, непрерывное распределение функций заменяется дискретным аналогом, а значения в промежутках между узлами сетки реконструируются полиномами разной степени. Для реализации метода второго порядка достаточно линейной реконструкции, третьего — квадратичной, и так далее. В целом, КРМ редко используются для практических расчетов на сильно неравномерных и неортогональных сетках. Основа МКО заключается в том, что расчетная область с помощью сетки разбивается на совокупность конечных объемов (ячеек). Узлы, в которых ищется решение, находятся в центрах ячеек. Для каждой ячейки должны выполняться законы сохранения массы, количества движения и энергии. Фактически, все многообразие методов конечного объема определяется двумя факторами: способом реконструкции функции внутри ячейки и способом вычисления потоков газодинамических параметров через стороны ячейки. Если не вдаваться в подробности, реконструкция функции в трехмерном случае также может осуществляться двумя способами. В первом случае реконструкция осуществляется при помощи построения квазиодномерного шаблона и аппроксимации на этом шаблоне. В случае структурированной сетки шаблон строится естественным

образом вдоль сеточных линий. В случае неструктурированной сетки обсуждение численной схемы обычно сводится к вопросу, как построить шаблон и вычислить значения функций в узлах шаблона. После этого используются, например, различные варианты схемы типа WENO [3-7]. Отдельный вопрос заключается в том, чтобы определить сколько квадратурных точек участвует в расчете потоков газодинамических параметров на стороне ячейки. Напомним, что переход от первого порядка точности (метод Годунова) ко второму осуществлен путем увеличения порядка аппроксимации реконструируемых функций в ячейке (от константы до линейного распределения). На первый взгляд, переход от второго порядка точности к более высоким может быть осуществлен аналогично. Однако, в работе [5] показано, что в случае использования одной квадратурной точки, невозможно реализовать схему порядка выше второго. Тем не менее, такой подход имеет право на существование. Он прост в реализации, при этом реконструкция высокого порядка помогает существенно улучшить диссипативные свойства схемы практически без увеличения времени расчета. В литературе такие схемы обычно называют слабодиссипативными или схемами высокого разрешения. Метод РМГ [8-11] относится к классу конечно-элементных методов. Основное отличие РМГ от традиционных конечно-элементных методов заключается в том, что функциональное пространство, соответствующее методу РМГ, состоит из кусочных многочленов. На границах элементов используется методика расчета численных потоков из метода конечного объема. К достоинствам метода относится строгая математическая постановка, компактный шаблон для

формально любого порядка точности, простота распараллеливания и гибкость в выборе расчетной сетки. Однако, РМГ существенно более требователен, чем МКО к вычислительным ресурсам.

## 1. Система уравнений

Система уравнений, описывающая течение вязкого газа в трехмерной области  $D$  может быть представлена в виде

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}, \mathbf{G}) = 0, \quad (1)$$

где  $\mathbf{U} = (\rho, \rho u, \rho v, \rho w, \rho E)$  — столбец консервативных переменных,  $\mathbf{F}$  — вектор потоков импульса и энергии,  $\mathbf{G} = \nabla \mathbf{U}$  — вектор градиентов. Полная энергия газа  $E$  определяется формулой

$$E = \frac{u^2 + v^2 + w^2}{2} + \frac{RT}{\gamma - 1},$$

где  $R = 287.065$  Дж/(кг К) — газовая постоянная воздуха,  $\gamma = 1.4$  — постоянная адиабаты. Используется уравнение состояния совершенного газа  $p = \rho RT$ . В невязком газе система упрощается. Например, потоки вдоль координатной оси  $k$  задаются формулой

$$F_k = F_k^{\text{conv}} = (\rho u_k, \rho u u_k + p \delta_{xk}, \rho v u_k + p \delta_{yk}, \rho w u_k + p \delta_{zk}, \rho E u_k + p u_k).$$

В вязком газе появляются дополнительные потоки, вызванные молекулярным переносом параметров:  $F_k = F_k^{\text{conv}} + F_k^{\text{diff}}$ , где

$$F_k^{\text{diff}} = (0, \tau_{xk}, \tau_{yk}, \tau_{zk}, \tau_{xk}u + \tau_{yk}v + \tau_{zk}w + q_k).$$

Молекулярные потоки импульса и тепловой энергии определяются

$$\text{как } \tau_{ij} = -\mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \operatorname{div} \mathbf{V} \delta_{ij} \right), \quad q_k = -\frac{\mu C_p}{\operatorname{Pr}} \frac{\partial T}{\partial x_k}.$$

где  $\mu$  - коэффициент вязкости,  $\operatorname{Pr}$  - число Прандтля,  $C_p$  - коэффициент теплоемкости.

## 2. Метод Галеркина с разрывными функциями

Дискретный аналог системы уравнений (1) решается в дискретной области  $D_h$ , которая аппроксимирует физическую область  $D$  с заданным порядком точности. При этом,  $D_h$  разбита на  $N_e$  непересекающихся элементов  $\Omega_i$ :

$$D_h = \bigcup_{i=1}^{N_e} \Omega_i.$$

Будем раскладывать консервативные переменные с использованием полиномов:

$$\mathbf{U}(\mathbf{x}, t) = \sum_{j=1}^{K_f} u_j(t) \varphi_j(\mathbf{x}). \quad (2)$$

Коэффициенты разложения  $u_j$  после дискретизации по времени становятся основными неизвестными величинами в РМГ. В качестве функций  $\varphi_j(\mathbf{x})$ ,  $j=1, \dots, K_f$  возьмем базис пространства полиномов степени  $K$ . При этом количество базисных функций  $K_f$  связано с  $K$  соотношением  $K_f = (K+1)(K+2)(K+3)/6$ . Следуя [12], построим базис из ортонормированных функций. На первом этапе определим координаты центра масс элемента  $\Omega_i$ :

$$x_0 = \frac{\int_{\Omega} x d\Omega}{\int_{\Omega} d\Omega}, \quad y_0 = \frac{\int_{\Omega} y d\Omega}{\int_{\Omega} d\Omega}, \quad z_0 = \frac{\int_{\Omega} z d\Omega}{\int_{\Omega} d\Omega}.$$

Затем вычислим компоненты тензора инерции  $\Omega_i$ :

$$\mathbf{I} = \begin{bmatrix} \int_{\Omega} (\tilde{y}^2 + \tilde{z}^2) d\Omega & -\int_{\Omega} \tilde{x}\tilde{y} d\Omega & -\int_{\Omega} \tilde{x}\tilde{z} d\Omega \\ -\int_{\Omega} \tilde{x}\tilde{y} d\Omega & \int_{\Omega} (\tilde{x}^2 + \tilde{z}^2) d\Omega & -\int_{\Omega} \tilde{y}\tilde{z} d\Omega \\ -\int_{\Omega} \tilde{x}\tilde{z} d\Omega & -\int_{\Omega} \tilde{y}\tilde{z} d\Omega & \int_{\Omega} (\tilde{x}^2 + \tilde{y}^2) d\Omega \end{bmatrix}.$$

Здесь  $\tilde{x} = x - x_0$ ,  $\tilde{y} = y - y_0$ ,  $\tilde{z} = z - z_0$  — координаты в системе, перемещенной в центр масс элемента. Из симметричности и положительной определенности тензора  $\mathbf{I}$  следует, что его единичные собственные векторы  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ ,  $\mathbf{e}_3$  взаимно ортогональны.

Пусть набор  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ ,  $\mathbf{e}_3$  является правой тройкой векторов. Тогда преобразование

$$\begin{bmatrix} x_{\Omega} \\ y_{\Omega} \\ z_{\Omega} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}$$

задает поворот системы  $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z})^T$  в систему  $\mathbf{x}_{\Omega} = (x_{\Omega}, y_{\Omega}, z_{\Omega})^T$  главных осей элемента  $\Omega_i$ . Определим в системе координат  $\mathbf{x}_{\Omega}$  исходные (предварительные)

базисные функции  $\psi_j(\mathbf{x}_{\Omega})$  как одночлены

$$\psi_j(\mathbf{x}_{\Omega}) = \frac{x_{\Omega}^{\alpha_j} y_{\Omega}^{\beta_j} z_{\Omega}^{\gamma_j}}{s_j}, \quad \alpha_j, \beta_j, \gamma_j \in \mathbf{Z}_+, \quad 0 \leq \alpha_j + \beta_j + \gamma_j \leq K.$$

Нормировочный множитель  $s_j = \sqrt{\int_{\Omega} (x_{\Omega}^{\alpha_j} y_{\Omega}^{\beta_j} z_{\Omega}^{\gamma_j})^2 d\Omega}$  выбран таким образом,

чтобы обеспечить выполнение условия  $\int_{\Omega} \psi_j^2(\mathbf{x}_{\Omega}) d\Omega = 1$ . Проведем ортогонализацию по

методу Грама–Шмидта и нормируем набор функций  $\{\psi_j(\mathbf{x}_\Omega)\}$ , что в результате даст базис  $\{\psi_j^*(\mathbf{x}_\Omega)\}$ , который обладает свойством  $\int_\Omega \psi_i^*(\mathbf{x}_\Omega)\psi_j^*(\mathbf{x}_\Omega)d\Omega = \delta_{ij}$ . В исходной системе координат базисные функции будут определяться как

$$\varphi_j(\mathbf{x}) = \psi_j^*(\mathbf{x}_\Omega(\mathbf{x})), \quad \mathbf{x}_\Omega(\mathbf{x}) = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} (\mathbf{x} - \mathbf{x}_0).$$

Все описанные преобразования сохраняют полиномиальность базиса. Ортонормированный базис, построенный таким образом, обладает малой чувствительностью к ошибкам округления, что актуально для РМГ высокого порядка точности [12]. Вернемся к определению коэффициентов разложения решения. Умножим систему (1) на базисную функцию  $\varphi_i(\mathbf{x})$  и проинтегрируем ее по объему элемент  $\Omega$ :

$$\int_\Omega \left( \frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} \right) \varphi_i d\Omega = 0, \quad i = 1, \dots, K_f.$$

Подставим сюда разложение (2), и запишем  $(\nabla \cdot \mathbf{F})\varphi_i$  как  $\nabla \cdot (\mathbf{F}\varphi_i) - \mathbf{F} \cdot \nabla \varphi_i$ , а после применим формулу Остроградского–Гаусса. Учтем, что в силу ортонормированности базиса выполняется соотношение  $\int_\Omega \varphi_i \varphi_j d\Omega = \delta_{ij}$ , тогда получим, что

$$\frac{du_i}{dt} + \oint_\Sigma \hat{\mathbf{F}} \cdot \mathbf{n} \varphi_i d\Sigma = \int_\Omega \mathbf{F} \cdot \nabla \varphi_i d\Omega, \quad (3)$$

где  $\Sigma$  — поверхность рассматриваемого элемента,  $\mathbf{n}$  — внешняя единичная нормаль к ней. Вектор  $\hat{\mathbf{F}}$  введен из-за того, что на грани элемента решение в общем случае

терпит разрыв, и потоки надо вычислять по модифицированным формулам, зависящим от решения с двух сторон от грани ( $U_L$  и  $U_R$ ) и вектора  $\mathbf{n}$ . Для конвективных потоков используется формула

$$\hat{\mathbf{F}}^{\text{conv}} \cdot \mathbf{n} = \frac{1}{2}(\mathbf{F}^{\text{conv}}(U_L) + \mathbf{F}^{\text{conv}}(U_R)) \cdot \mathbf{n} - \alpha |\mathbf{A}_n| \frac{U_R - U_L}{2}. \quad (4)$$

В этой формуле  $\mathbf{A}_n = \partial(\mathbf{F}^{\text{conv}} \cdot \mathbf{n})/\partial U$  — матрица Якоби конвективных потоков по нормали к грани, вычисленная по осредненным величинам

$$\tilde{u} = \frac{\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad \tilde{v} = \frac{\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad \tilde{w} = \frac{\sqrt{\rho_L} w_L + \sqrt{\rho_R} w_R}{\sqrt{\rho_L} + \sqrt{\rho_R}},$$

Пусть  $\{\lambda_i\}$  — собственные числа матрицы  $\mathbf{A}_n$ , а матрица  $\mathbf{T}_n$  сформирована из собственных векторов  $\mathbf{A}_n$ . Введем матрицу  $|\mathbf{A}_n| = \text{diag}(|\lambda_i|)$ . В этих обозначениях матрица  $|\mathbf{A}_n|$  в (4) равна  $\mathbf{T}_n |\mathbf{A}_n| \mathbf{T}_n^{-1}$ , а коэффициент  $\alpha$  может принимать значения от 0 (центрально-разностная схема без искусственной диссипации) до 1 (стандартная схема Рунге).

Диффузионные потоки на грани вычисляются по симметричной формуле

$$\hat{\mathbf{F}}^{\text{diff}} \cdot \mathbf{n} = \frac{1}{2}(\mathbf{F}^{\text{diff}}(U_L) + \mathbf{F}^{\text{diff}}(U_R)) \cdot \mathbf{n}.$$

Градиенты  $\mathbf{G}$  вычисляются по методу Bassi–Rebay 2 [13]. Интегралы по поверхности и объему элемента берутся методом квадратур Гаусса. В стандартном кубе  $[-1, 1]^3$  и квадрате  $[-1, 1]^2$  распределяются декартовы произведения множеств



одномерных точек Гаусса–Лежандра, которые переносятся в физическое пространство с помощью “серендипова” преобразования.

Рассмотрим временную аппроксимацию уравнений РМГ (3). Для этого интеграл по поверхности элемента перенесем в правую часть равенства, которую можно после этого назвать невязкой. Обозначим столбец правых частей символом  $R$ , а столбец искомых коэффициентов  $c_i$  символом  $U$ . Система уравнений приобретет вид

$$\frac{dU}{dt} = R.$$

Ориентируясь на нестационарные задачи, воспользуемся явной аппроксимацией этой системы по времени с помощью метода Рунге–Кутты:

$$U^{(0)} = U^n,$$

$$U^{(1)} = c_{00}U^{(0)} + c_0\Delta tR^{(0)},$$

$$U^{(2)} = c_{10}U^{(0)} + c_{11}U^{(1)} + c_1\Delta tR^{(1)},$$

...

$$U^{(k+1)} = c_{k0}U^{(0)} + c_{k1}U^{(1)} + \dots + c_{kk}U^{(k)} + c_k\Delta tR^{(k)},$$

...

$$U^{n+1} = U^{(M)}.$$

Здесь индексы без скобок обозначают номер шага по времени, со скобками - номер внутреннего шага метода Рунге–Кутты. За один шаг по времени  $\Delta t$  делается  $M$  внутренних шагов, после которых значения переменных  $U^{(M)}$  присваиваются новому

временному слою  $U^{n+1}$ . Невязка  $R^{(k)}$  пересчитывается на каждом внутреннем шаге. Коэффициенты метода постоянны. В зависимости от выбора  $M$  и коэффициентов  $\{c_{kl}, c_m\}$  порядок аппроксимации по времени может варьироваться. В настоящей реализации РМГ используется метод Рунге–Кутты семейства SSP [14] с 5 внутренними шагами, который обеспечивает 4-й порядок аппроксимации по времени

### **3. Особенности реализации программного кода**

В данном разделе описаны некоторые приемы, позволяющие конструировать высокотехнологичные коды, с максимальным использованием возможностей языка C++. Как известно, язык программирования помогает выразить идеи в виде кода. Для этого выполняются две взаимосвязанные задачи, а именно, предоставляются средства для арифметических действий и набор концепций, для поддержания архитектуры кода. Инструменты, добавленные к C (при создании C++), такие как классы и шаблоны, в первую очередь, предназначались для этих целей. В результате, получается прямое отображение встроенных операций и типов на аппаратное обеспечение, и, как следствие, эффективное использование памяти при выполнении низкоуровневых операции.

В процессе реализации данной программы на языке C++ была разработана библиотека классов, позволившая обеспечить функциональные возможности на максимально возможном уровне. Библиотеке выдвигались следующие требования и ограничения:

- Решения трехмерной системы уравнений Рейнольдса, замкнутых различными моделями турбулентности ( $q-\omega$ , SST, SA, DRSM, ...).
- Возможность расчета нестационарных течений.
- Возможность задания нестационарных граничных условий.
- Возможность прямого численного моделирования крупномасштабной турбулентности (LES, DES, DDES, ...).
- Возможность распараллеливания вычислений вплоть до 50 000 ядер.

Описанная выше задача достаточно сложна и требует детальной проработки. Для ее решения применялись приемы как объектно-ориентированного, так и обобщённого программирования. Попытки прямого использования готовых библиотек не привели к положительному результату, однако изучение текстов этих библиотек оказалось очень полезным. Создание программного интерфейса для конкретных кодов вызвано необходимостью тестирования. Для этого использован язык программирования Python.

Основными *элементами* при описании метода Галеркина с разрывными базисными функциями являются узел расчетной сетки, расчетная ячейка, сторона ячейки. Прежде, чем перейти к описанию топологии расчетной области, рассмотрим основные отличия представления *элементов* для структурированной и неструктурированной расчетной сетки. Существует два основных отличия между способом представления структурированной и неструктурированной расчетной сетки в компьютере. В структурированной сетке узел идентифицируется при помощи

вычислительных координат. Узел имеет столько вычислительных координат, сколько размерностей имеет сетка. Например, для трехмерной сетки, положение любого узла в сетке однозначно определяется тремя вычислительными координатами. Векторы физических координат узлов используют вычислительные координаты в виде индексов трехмерных массивов. Однако, отображение физических координат на вычислительные возможно только для структурированной сетки. Для неструктурированной сетки, узлы идентифицируются просто номером от 1 до  $N$ , где  $N$  – количество узлов сетки. Другим основным отличием структурированной и неструктурированной сеток является представление других *элементов* (ячеек и сторон ячеек). В структурированной сетке, данные, ассоциированные с ячейкой (или сами ячейки) используют трехмерные индексы. Максимальное значение индексов ячеек на единицу меньше, чем индексов координат. Аналогично можно рассмотреть данные, ассоциированные со сторонами ячеек (или сами стороны ячеек). Т.е. местоположение *элемента* и координаты его узлов могут быть легко восстановлены из размерности сетки и не требуют хранения дополнительных данных. Для неструктурированной сетки, *элементы* должны хранить дополнительную информацию про свой тип (или форму), список узлов, принадлежащий элементу и информацию про своих соседей.

Ключевое место в программе занимают элементы, предназначенные для работы с геометрическими объектами. В качестве примера приведем описание реализации “серендиповых” элементов [11]. Алгоритмы для поверхностных и объемных элементов, хотя и схожи, но имеют ряд принципиальных отличий. Во-первых,

количество координат в локальной криволинейной системе для двумерных элементов равно двум, а для трёхмерных – трём. Во-вторых, различаются входные параметры для расчета поверхностных и объемных интегралов. С другой стороны, эти алгоритмы не зависят от порядка аппроксимации элементов и абсолютно идентичны для линейных, квадратичных, кубичных и прочих элементов более высокого порядка. Меняется только набор функций формы. Учитывая эти особенности, для описания элементов в данной программе созданы две иерархии классов: одна для поверхностных элементов, и вторая для объёмных элементов. Базовые классы хранят набор функций формы и их производных и предоставляют метод, выполняющий преобразование в локальную систему координат. Конкретные наборы функций формы задаются при помощи производных классов.

Функции формы реализованы в виде так называемых «лямбда»-функций, которые недавно появились в стандарте C++. Это придаёт программе гибкость и лёгкость в расширении функциональности. При этом код обладает компактностью, а функции формы сохраняют локальность по отношению к конструктору соответствующего класса, что исключает проблемы именования функций и возможные конфликты с другими частями программы. Для уменьшения ошибок, связанных с конечностью представления вещественных чисел в ЭВМ, все вычисления, связанные с преобразованием элементов сетки проводятся с повышенной точностью. Для этого в качестве вещественно типа QReal используется встроенный тип языка C++ `long double`, который обеспечивает 3 дополнительных десятичных знака мантиссы числа.

Примеры «линейных» элементов представлены на рисунке 1. Слева изображена кубическая тестовая ячейка. Якобиан преобразования равен единице, поскольку мы преобразуем куб в куб. Для любой ячейки в виде прямоугольного параллелепипеда поле якобиана также будет постоянным.

На этом же рисунке справа приведена ячейка, полученная из прямоугольного параллелепипеда путём небольшого смещения узлов. Результирующий, немного скошенный и скрученный шестигранник, представляет собой типичную ячейку расчётной области. Видно, что форма граней элемента искривляется, и поле якобиана также меняется нелинейно от точки к точке. Нелинейность якобиана преобразования повышает вычислительную сложность расчёта интегралов.

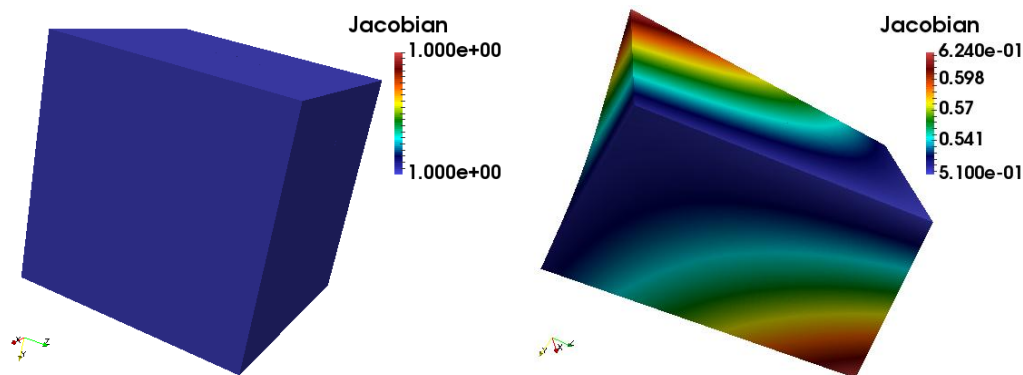


Рисунок 1. «Линейные» элементы

На рисунке 2 приведены примеры «квадратичных» элементов. Помимо вершин здесь заданы центры всех 12 рёбер. Слева представлена сильно искажённая кубическая ячейка. Минимальное и максимальное значения якобиана в пределах ячейки отличаются на 3 порядка. Появление подобных элементов в расчётной сетке

желательно исключить. Они сильно повышают степень якобиана преобразования и требуют самых больших вычислительных затрат при интегрировании.

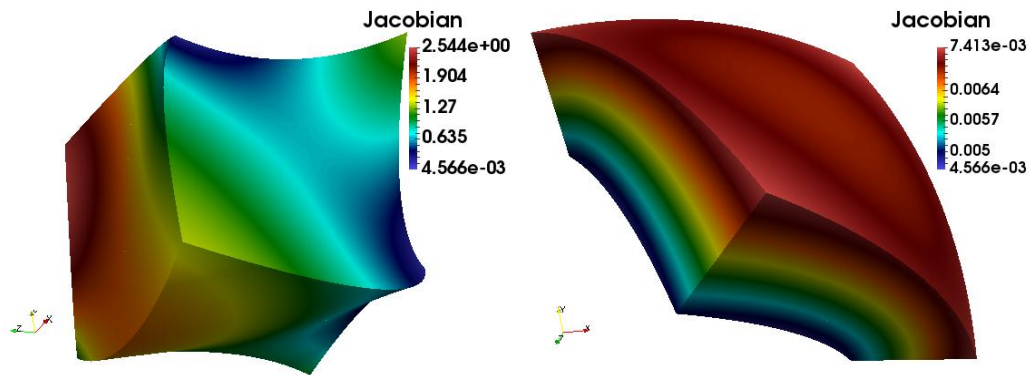


Рисунок 2. «Квадратичные» элементы

Справа приведена ячейка, узлы верхней и нижней грани которой расположены на цилиндрической поверхности. Здесь поле якобиана тоже меняется нелинейным образом, тем не менее, эти изменения не слишком велики: минимальное и максимальное значения являются величинами одного порядка.

В соответствии с идеологией методов конечного элемента, численное решение для каждой из консервативных переменных представляется в виде линейной комбинации базисных функций. При этом набор коэффициентов при базисных функциях определяется для каждой ячейки отдельно. Базисные функции, реализованные в коде программы, представляют собой полиномы степени не выше 3. Это определяет максимальный порядок аппроксимации схемы по пространству, а именно, четвёртый. Исходный набор функций определён не только в исходной системе координат, но и в любой иной. Например, перед ортонормированием базисных функций выполняется переход в систему координат, связанную с центром масс

элемента расчётной сетки и направленную вдоль главных осей его тензора инерции. Расширяя набор базисных функций, можно увеличивать точность и порядок пространственной аппроксимации численной схемы. В программе создан базовый класс `BasicFunctionSet3D`, который предоставляет хранилище для базисных функций и их производных, а также минимальный интерфейс для пользователей. Доступ осуществляется по каждому направлению, а функция `size()` позволяет вычислить количество доступных базисных функций. Другой класс `HierarchicalSet3D`, созданный с наследованием от базового класса, предоставляет вложенный набор дополнительных функций. При этом дополнительные функции для больших значений показателя степени полинома добавляются в конец массива, и их индексы больше, чем индексы для меньших значений. Конечный класс в этой цепочке `FullPolynomial3D` наследует от абстрактного класса `HierarchicalSet3D` и представляет собой набор базисных функций, основанных на полных полиномах. Приведенный пример не дает полное описание, но позволяет судить о структуре программы. В заключение следует отметить, что базисные функции, как и функции формы для “серендиповых” элементов, описанных в ранее, реализованы в виде так называемых «лямбда»-функций.

#### **4. Тестирование программы на примере обтекания цилиндра**

При тестировании программы проведен расчёт плоского обтекания полуцилиндра дозвуковым потоком идеального газа. Расчеты проведены на серии из четырех вложенных сеток размерностью от  $16 \times 4$  до  $128 \times 32$ . На рисунке 3 представлена расчетная сетка в одном из сечений, например, в плоскости симметрии.



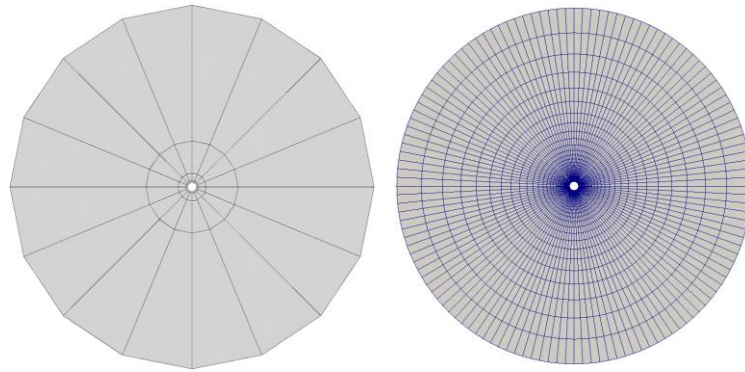


Рисунок 3. Расчетная сетка на цилиндре:  $16 \times 4$  ячеек и  $128 \times 32$  ячеек

Геометрические размеры цилиндра и расчетной области равны соответственно  $R_{cylinder} = 0.5$ ,  $R_{outer} = 20$ ,  $\Delta z = 0.1$ . На поверхности цилиндра и в ставилось условие непротекания (уравнения Эйлера). На внешней границе задавались параметры набегающего потока. Расчеты проводились при следующих значениях следующих параметров:  $p_{\infty} = 100,000 \text{ Па}$ ,  $T_{\infty} = 293^{\circ} \text{ К}$ ,  $u_{\infty} = 50 \text{ м/сек}$  ( $M_{\infty} \approx 0.15$ ). На рисунке 4 представлены поля полного давления на самой подробной сетке, состоящей из  $128 \times 32$  ячеек. Так как число Маха существенно меньше единицы, потери полного давления должны равняться нулю. В левой части рисунка 4 представлены поля, полученные с использованием линейной реконструкции параметров в ячейке (для РМГ — степень полинома  $K = 1$ ). В этом случае за цилиндром видны зоны пониженного полного давления примерно одинакового размера. При использовании схемы 4-го порядка точности с криволинейной сеткой потери полного давления исчезают, см. рисунок 4 справа.

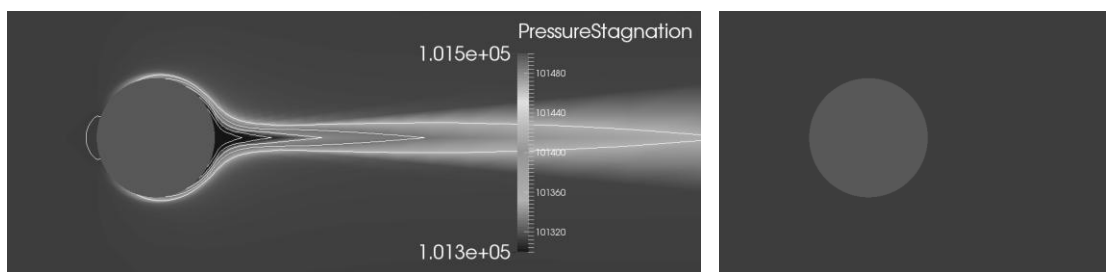


Рисунок 4. Поле полного давления,  $K = 1$  (слева),  $K = 3$  (справа)

Для определения порядка сходимости решения по расчетной сетке проведем анализ значений ошибки при определении энтропийной функции по норме L2. В данном решении энтропия, так же как и полное давление, должна оставаться постоянной. Воспользуемся соотношениями:

$$e_{entropy} = \left( \frac{p}{p_\infty} \right) / \left( \frac{\rho}{\rho_\infty} \right)^\gamma - 1, \quad Order = \frac{\log(e_{i-1} / e_i)}{\log(NDOF_i / NDOF_{i-1})},$$

где  $e_{entropy}$  — энтропийная ошибка,  $Order$  — порядок сходимости,  $NDOF$  — произведение размера сетки на число базисных функций,  $i$  - номер сетки в серии расчетных сеток. На рисунке 5 представлены результаты расчетов. Видно, что ошибочный рост энтропии практически прекращается в том случае, когда порядок точности выше, чем 3.

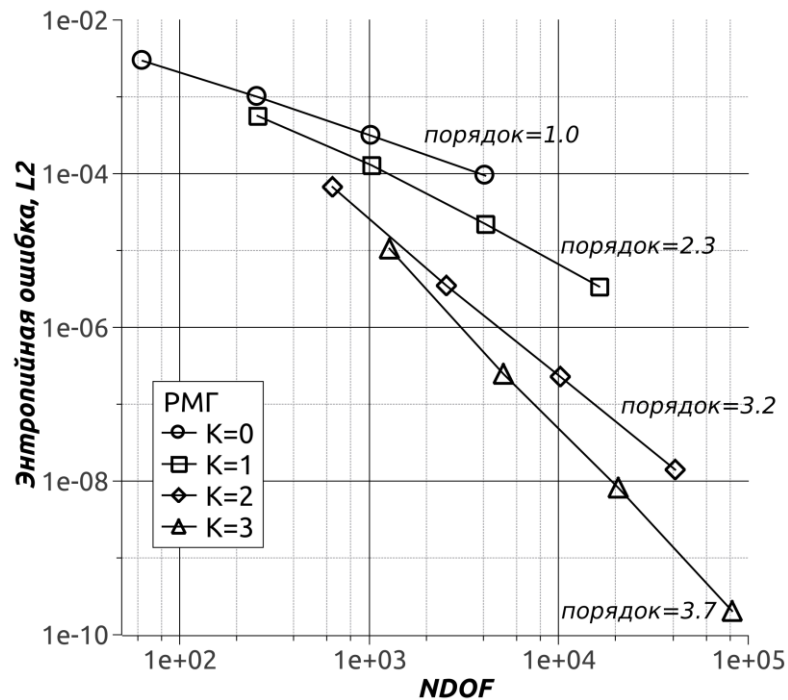


Рисунок 5. Ошибка энтропии по норме L2

Минимальная ошибка величиной порядка  $10^{-10}$  достигается при  $NDOF \approx 10^5$ .

Аналогичные результаты были получены и на других тестах.

### 5. Пример расчета обтекания модели крыла с фюзеляжем

Одной из задач, в которых является важным высокий порядок аппроксимации расчетной схемы является задача об обтекании крыла с выпущенной механизацией на режиме взлета-посадки [15]. В качестве примера приведены результаты сравнения расчетных данных с экспериментом High Lift Workshop [16]. Расчетная сетка представлена на рисунке 6 и содержала 2 097 344 ячеек.

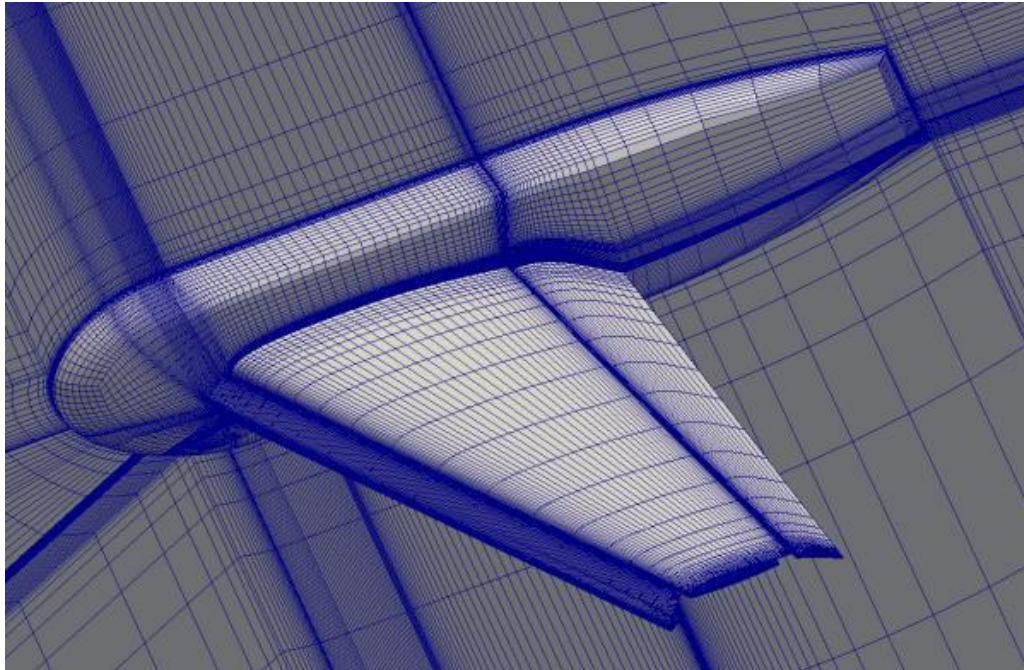


Рисунок 6. Геометрия модели и расчетная сетка

Расчет выполнен по схеме третьего порядка точности (квадратичные полиномы) на компьютере, содержащем 10 000 ядер. Максимальная масштабируемость кода не превышала значения 1.85, что для данной геометрии было признано нормальным. Поле статического давления приведено на рисунке 7. Хорошо видны все особенности течения, характерные для крыла с вырощенными предкрылками и закрылками. Следует отметить зону интерференции крыла и фюзеляжа, где распределение давления «переходит» с крыла на фюзеляж.

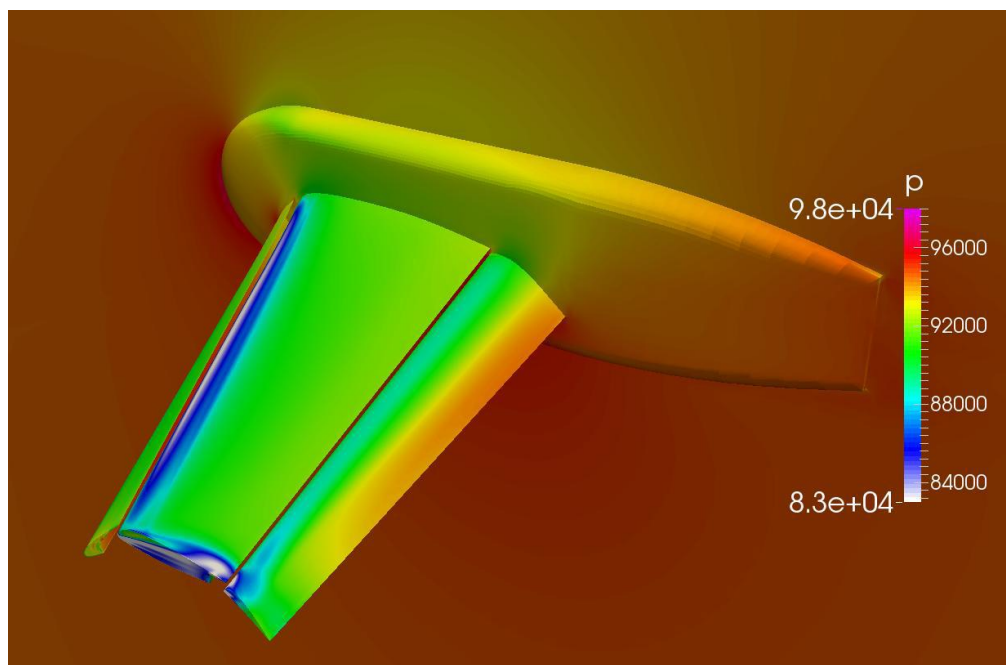


Рисунок 7. Поле статического давления в окрестности модели

Сопоставление расчета, выполненного на различных сетках с экспериментальными данными показывает, что на подробной сетке (Fine и Medium) имеет место хорошее соответствие расчетных и экспериментальных данных, см. Рисунок 8. При этом на передней кромке основного крыла отмечается более высокое разрежение, что может быть связано с работой полиномов в методе РМГ. На закрылке ситуация усложняется и только наилучшее сетка Fine позволяет достичь совпадения расчетных и экспериментальных данных.

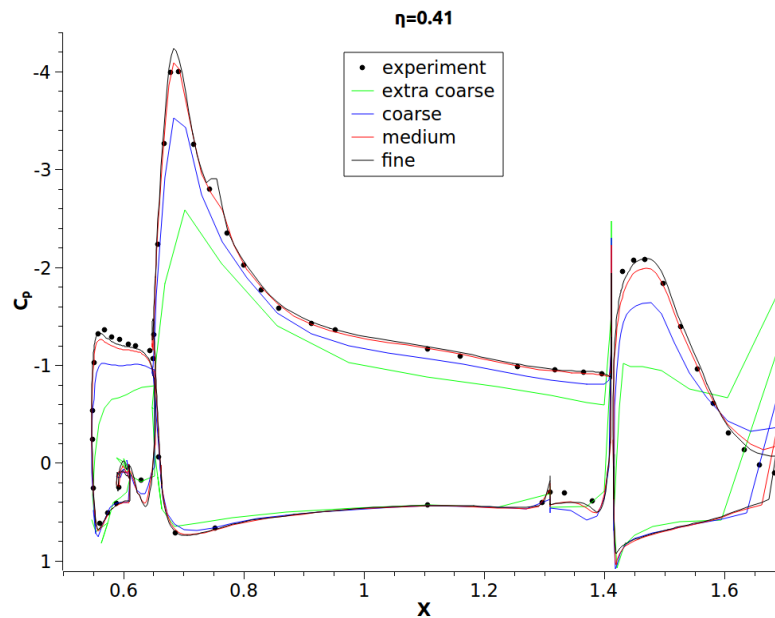


Рисунок 8. Сопоставление расчета с экспериментом

*Работа была поддержана Министерством образования и науки в рамках федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014 - 2020 годы», проект № 14.628.21.0005, уникальный идентификатор проекта:*

*RFMEFI62815X0005*

### **Библиографический список**

1. Shu C.-W. High Order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD // International Journal of Computational Fluid Dynamics, 2003, vol. 17, no. 2, pp. 107-118.
2. Nogueira X., Cueto-Felgueroso L., Colominas I., Gómez H., Navarrina F., Casteleiro M. On the accuracy of finite volume and discontinuous Galerkin discretizations for compressible

flow on unstructured grids // *International Journal for Numerical Methods in Engineering*, 2009, vol. 78, no. 13, pp. 1553–1584.

3. Shu C.-W., Osher S. Efficient implementation of essentially non-oscillatory shock capturing schemes, I // *Journal of Computational Physics*, 1988, vol. 77, no. 2, pp. 439–471.

4. Shu C.-W., Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II // *Journal of Computational Physics*, 1989, vol. 83, no. 1, pp. 32–78.

5. Zhang R., Zhang M., Shu C.W. On the order of accuracy and numerical performance of two classes of finite volume WENO schemes // *Communications in Computational Physics*, 2011, vol. 9, no. 3, pp. 807–827.

6. Balsara D.S., Shu C.-W. Monotonicity Preserving Weighted Essentially Non-oscillatory Schemes with Increasingly High Order of Accuracy // *Journal of Computational Physics*, 2000, vol. 160, no. 2, pp. 405–452.

7. Titarev V., Toro E. Finite-volume WENO schemes for three-dimensional conservation laws // *Journal of Computational Physics*, 2004, vol. 201, no. 1, pp. 238–260.

8. Bassi F., Rebay S. A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations // *Journal of Computational Physics*, 1997, vol. 131, no. 2, pp. 267–279.

9. Cockburn B., Shu C.-W. The Runge–Kutta Discontinuous Galerkin Method for Conservation Laws V // *Journal of Computational Physics*, 1998, vol. 141, no. 2, pp. 199–224.

10. Волков А. Методы решения сеточных уравнений конечно-элементной аппроксимации пространственных течений // Ученые записки ЦАГИ. 2010. Т. XLI. № 3. С. 52-68.
11. Arnold D. N., Awanou G. The Serendipity Family of Finite Elements // Foundations of Computational Mathematics, 2011, vol. 11, no. 3, pp. 337–344.
12. Bassi F., Botti L., Colombo A., Di Pietro D.A., Tesini P. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations // Journal of Computational Physics, 2012, vol. 231, no. 1, pp. 45–65.
13. Bassi F., Rebay S., Mariotti G., Pedinotti S., Savini M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows // In Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics. Antwerpen, Belgium, 1997, pp. 99–109.
14. Gottlieb S., Shu C.-W., Tadmor E. Strong Stability-Preserving High-Order Time Discretization Methods // SIAM Review, 2001, vol. 43, no. 1, pp. 89–112.
15. Губский В.В. Применение адаптивной механизации крыла на легком транспортном самолете // Труды МАИ. 2013. № 68. URL: <http://trudymai.ru/published.php?ID=41737>
16. Christopher Rumsey, NASA Langley Research Center, The 1st AIAA CFD High Lift Prediction Workshop (HiLiftPW-1), 2017, URL: <http://hiliftpw.larc.nasa.gov/index-workshop1.html>