

ПАКЕТ ПРОГРАММ ЛОГОС. АЛГЕБРАИЧЕСКИЙ МНОГОСЕТОЧНЫЙ МЕТОД РЕШЕНИЯ СЛАУ ДЛЯ ЗАДАЧ ГИДРОДИНАМИКИ

Голубев А.А.¹, Дерюгин Ю.Н.¹, Зеленский Д.К.¹, Козелков А.С.¹, Лашкин С.В.¹,
Силаев Д.П.¹, Симонов П.Г.¹

¹ФГУП «Российский федеральный ядерный центр - Всероссийский научно-исследовательский институт экспериментальной физики» (607188, г. Саров, Нижегородская обл., пр. Мира, 37), e-mail: silaevdp@gmail.com

В данной статье рассматривается программная реализация алгебраического многосеточного метода решения систем линейных алгебраических уравнений (СЛАУ) с агрегативным способом огрубления. Реализация метода выполнена на основе структуры памяти программного пакета ЛОГОС для решения задач вычислительной гидродинамики методом SIMPLE. Представлены варианты решения СЛАУ с использованием V-, W- и F-циклов многосеточного метода. Рассмотрены гранево-ячеечный формат хранения матриц и адаптация метода Зейделя для эффективной работы с данным форматом. Особое внимание уделено распараллеливанию алгоритмов сглаживания и построения матриц грубого уровня. Показано решение проблемы с использованием грубых уровней в параллельном режиме, которая состоит в том, что из-за небольшого размера матриц на грубых уровнях время, затраченное на обмены, заметно превосходит время, затраченное на вычисления. В статье приводятся результаты тестовых расчетов и их сравнение с другими реализациями AMG.

Ключевые слова: система линейных уравнений, алгебраический многосеточный метод.

THE SOFTWARE PACKAGE LOGOS. ALGEBRAIC MULTIGRID METHOD FOR SOLUTION OF LINEAR EQUATIONS SYSTEMS FROM CFD-PROBLEMS

Golubev A.A.¹, Derjugin J.N.¹, Zelenskij D.K.¹, Kozelkov A.S.¹, Lashkin S.V.¹,
Silaev D.P.¹, Simonov P.G.¹

¹ Russian Federal Nuclear Center – VNIIEF (37, Mira prospect, Sarov, Niznij Novgorod region, 607188), e-mail: silaevdp@gmail.com

In the paper a software implementation of an algebraic multigrid method (AMG) with aggregative coarsening for solution of systems of linear equations was reviewed. Data and memory structure of the LOGOS, a software package for solving of CFD-problems by SIMPLE-method, were used during the implementation. V-, W-, F-cycles of multigrid method were considered for solution of systems of linear algebraic equations (SLAE). Face-by-cell storage format for a matrix and an adaptation of Gauss-Seidel method for efficient processing of the format were reviewed. Parallelization of algorithms of both smoothing and coarsening is examined in extra details. Also solution for a problem of parallel coarse level's usage is presented. The problem was the following: because of small matrix size at coarse level's time spent for data exchanges is longer comparing to the time of calculations. The paper contains results of test problems solutions, and comparison of results against other implementations of AMG.

Keywords: system of linear equations, algebraic multigrid.

При построении вычислительного алгоритма для расчета течений с использованием неявных разностных схем возникает важная задача, связанная с решением системы линейных алгебраических уравнений (СЛАУ) большой размерности. В задачах моделирования несжимаемых течений с использованием метода SIMPLE вычислительный алгоритм строится на основе метода расщепления по физическим процессам, сводящегося к последовательному решению систем дифференциальных уравнений [2]. Выбор метода решения разностных уравнений оказывает существенное влияние на устойчивость вычислительного алгоритма и скорость сходимости итерационного процесса.

При решении задач вычислительной гидродинамики методом SIMPLE до 90% вычислительного времени приходится на решение СЛАУ, составленной относительно поправки давления. Данная матрица СЛАУ симметрична и не имеет строгого диагонального преобладания. Из мировой практики известно, что наиболее подходящим методом для решения таких СЛАУ является алгебраический многосеточный метод (AMG) [5], который существенно позволяет увеличить скорость сходимости и минимизировать затраты машинного времени и памяти. Кроме вышеуказанных свойств этого метода, он обеспечивает монотонную сходимость, что немаловажно, так как в методе SIMPLE нет смысла решать СЛАУ с высокой точностью – многократное повышение точности лишь незначительно сокращает общее число шагов решателя SIMPLE. Это дает преимущество на данных задачах перед, например, предобусловленным методом сопряженных градиентов (PCG), что показано на рисунке 1. На данном графике сравнивается сходимость алгебраического многосеточного метода, которому посвящена данная статья, и метода сопряженных градиентов с предобуславливателем Эйзенштата [4].

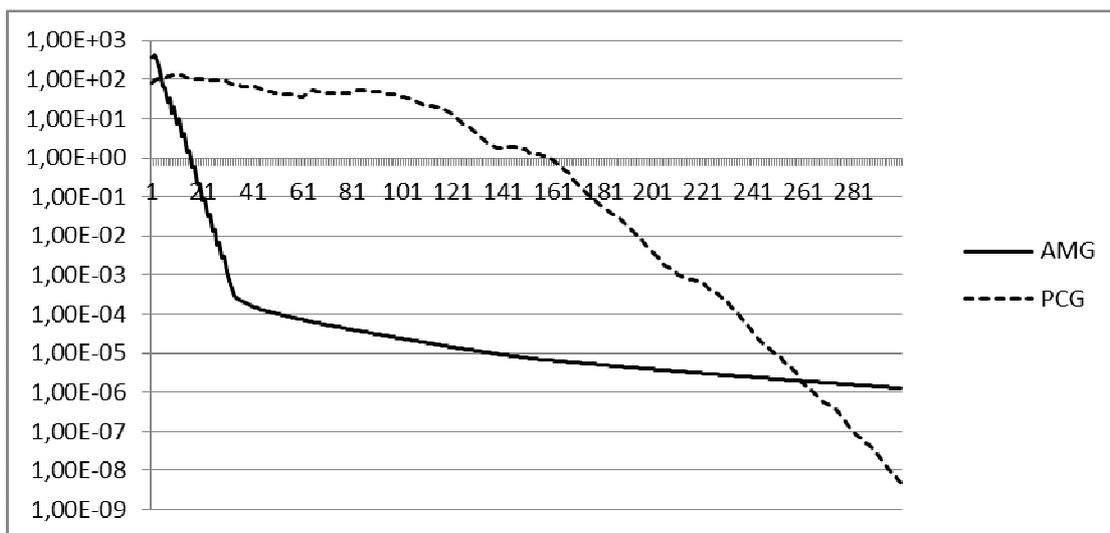


Рисунок 1. Сходимость методов решения СЛАУ.

В многосеточном методе обычный итерационный процесс комбинируется с коррекцией решения на последовательности грубых сеток. Рассмотрим систему уравнений:

$$A_h x^h = b^h. \quad (1)$$

Оператор интерполяции P с грубой сетки H на подробную сетку h позволяет представить оператор A_h на грубой сетке в виде

$$A_H = R A_h P, \quad (2)$$

где $R = P^T$. Шаг коррекции решения имеет вид:

$$x_{new}^h = x_{old}^h + P e^H. \quad (3)$$

Коррекция решения e^H является точным решением уравнения

$$A_H e^H = r^H, \quad (4)$$

где $r^H = Rr^h$, $r^h = b^h - A_h x_{old}^h$.

Таким образом, многосеточный метод, использующий схему коррекции решения, представляет собой следующую последовательность шагов.

1. Делается μ_1 приближений решения на сетке h при помощи метода Зейделя (предварительное сглаживание).
2. Невязка $r^h = b^h - A_h x_{old}^h \in V_h$ проецируется на пространство V_H , т.е. $r^H = Rr^h$.
3. Находится приближенное решение $A_H e^H = r^H$ на грубой сетке. Для этого рекурсивно делается γ циклов многосеточного метода.
4. Коррекция e^H интерполируется на подробную сетку и производится коррекция решения на подробной сетке $x_{new}^h = x_{old}^h + P e^H$.
5. Делается μ_2 приближений решения на подробной сетке для подавления ошибки интерполяции (заключительное сглаживание).

В зависимости от числа рекурсивных вызовов метода γ на каждом сеточном уровне выделяют различные типы циклов.

При $\gamma = 1$ имеет место быть V-цикл, а при $\gamma = 2$ – W-цикл (рис. 2).

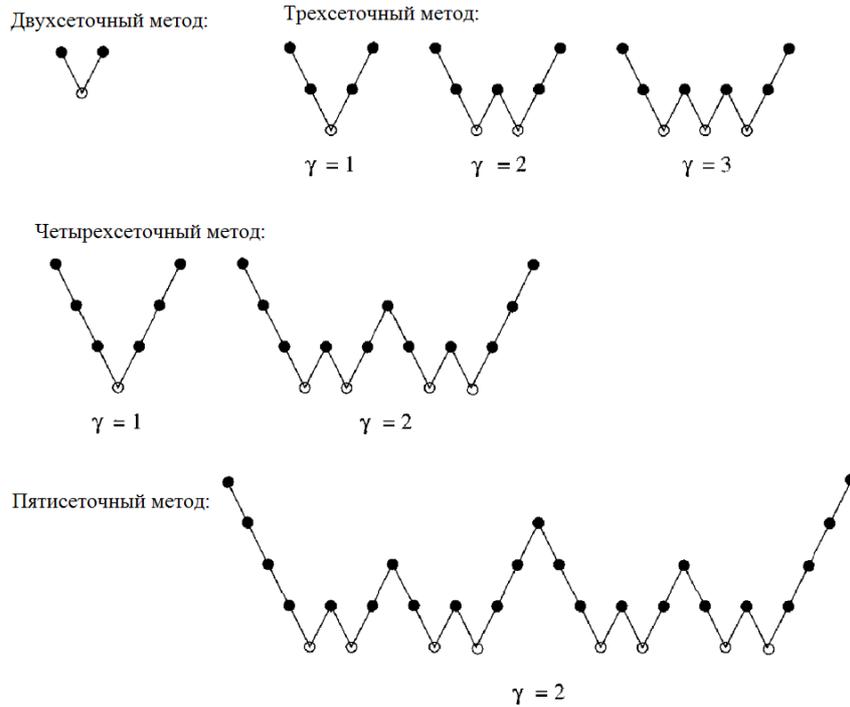


Рисунок 2. V- и W-циклы.

Если на каждом уровне рекурсивно сначала вызывать один W-цикл, а затем V-цикл, получим F-цикл (рис. 3).

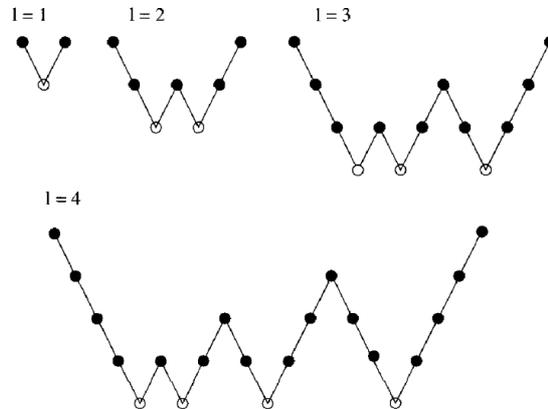


Рисунок 3. F-цикл.

В данной работе реализован агрегативный метод огрубления с постоянной интерполяцией [6]. Все переменные разделяются на агрегаты I_k , где I_k содержит все индексы i , соответствующие ячейкам, которые включены в агрегат k .

Построение оператора на грубой сетке производится при помощи соотношения:

$$A_H = RA_{\bar{h}}P = (a_{kl}^H), \quad a_{kl}^H = \sum_{i \in I_k} \sum_{j \in I_l} a_{ij}^h \quad (k, l \in \mathcal{C}). \quad (5)$$

Рассмотрим сетку, огрубление которой показано на рисунке 4.

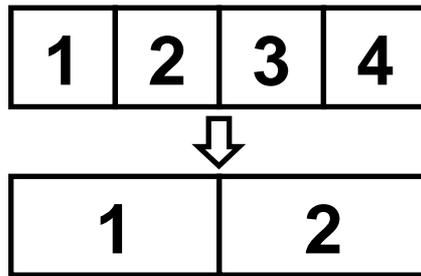


Рисунок 4. Пример огрубления сетки.

Вычисление матрицы грубого уровня, в данном случае, будет происходить следующим образом:

$$A_H = RA_kP = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11} + a_{12} + a_{21} + a_{22} & a_{23} \\ a_{32} & a_{33} + a_{34} + a_{43} + a_{44} \end{pmatrix}$$

Для хранения матрицы использован ячеечно-граневый формат. Отдельно хранятся диагональ D , верхнетреугольная U и нижнетреугольная L части матрицы. Соответственно, $A = L + D + U$. Эти три массива являются одномерными. Массив, хранящий диагональ, индексируется номером соответствующей ячейки. Два других массива индексируются номерами граней, определяющих значения соответствующих коэффициентов. Упорядочение массивов представлено на рисунке 5.

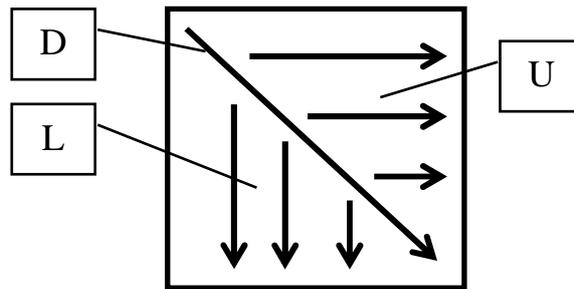


Рисунок 5. Расположение матрицы в памяти.

В данной работе в качестве сглаживателя используется метод Зейделя [1]:

$$x_i^{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^N a_{ij} x_j^k}{a_{ii}}, i \in 1, \dots, N, \quad (6)$$

где N – число ячеек.

Но в данном представлении метода (6) возникают сложности при работе с матрицей в ячеечно-граневом формате, кроме того, доступ к данным осуществляется не по порядку,

плохо используется кэш (рис. 5). Преобразовав данный алгоритм к виду (7), можно добиться строго последовательного доступа к элементам массивов хранения коэффициентов матрицы.

$$\left. \begin{aligned} x_i^{k+1} &= \frac{\tilde{b}_i - \sum_{j=i+1}^N a_{ij} x_j^k}{a_{ii}} \\ \tilde{b}_j &= \tilde{b}_j - a_{ji} x_i, j \in i+1, \dots, N \end{aligned} \right\} i \in 1, \dots, N, \quad (7)$$

где \tilde{b} – копия вектора b , подвергающаяся модификации в процессе работы алгоритма.

Распараллеливание организовано с помощью фиктивных ячеек. На рисунке 6 показана передача информации, возникающая при применении данного подхода.

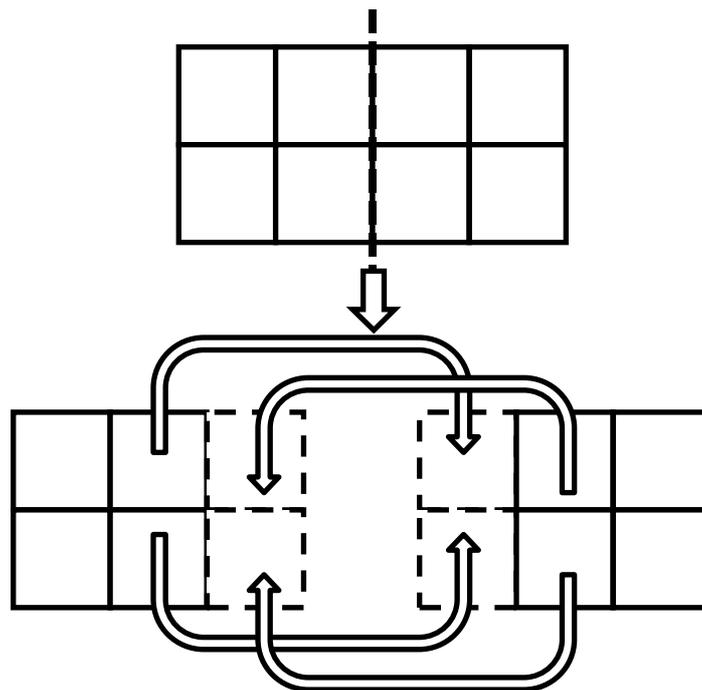


Рисунок 6. Передача информации в фиктивные ячейки.

Огрубление происходит независимо на каждом процессе. Фиктивные ячейки огрубляются в соответствии с огрублением их действительных прообразов на соответствующих процессах. Таким образом, в процессе огрубления число связей между процессами уменьшается.

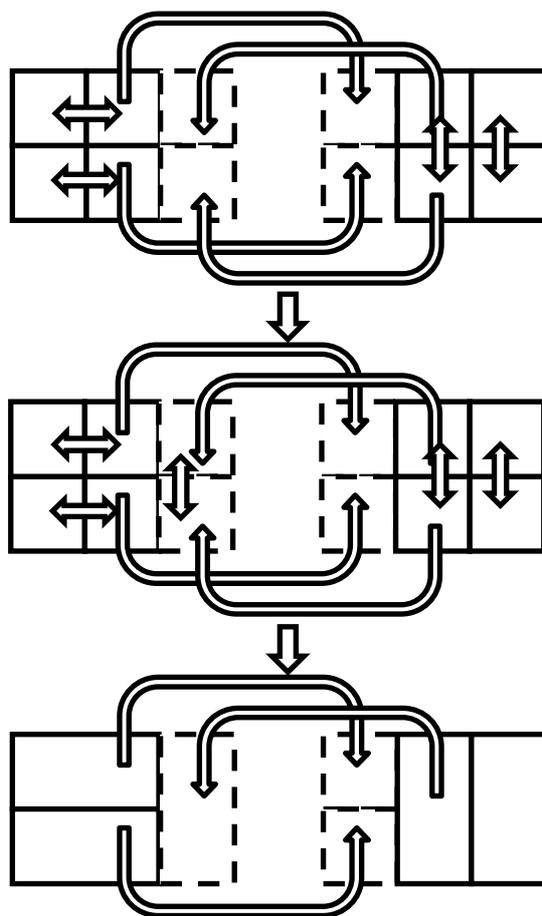


Рисунок 7. Огрубление ячеек в параллельном режиме.

Следует отметить, что данный подход к распараллеливанию огрубления порождает две проблемы. Во-первых, огрубление останавливается в случае, если на каждом процессе осталось по одной действительной ячейке. Во-вторых, на грубых уровнях, где размерность матриц невелика, время, затрачиваемое на межпроцессорные обмены, из-за латентности коммуникационной среды начинает многократно превосходить время, затрачиваемое на вычисления. Для решения данных проблем в рамках данной работы предложено выполнять сбор всех матриц небольшого размера на одном процессе, формировать из них глобальный уровень и продолжать огрубление и решение в последовательном режиме (рис. 8).

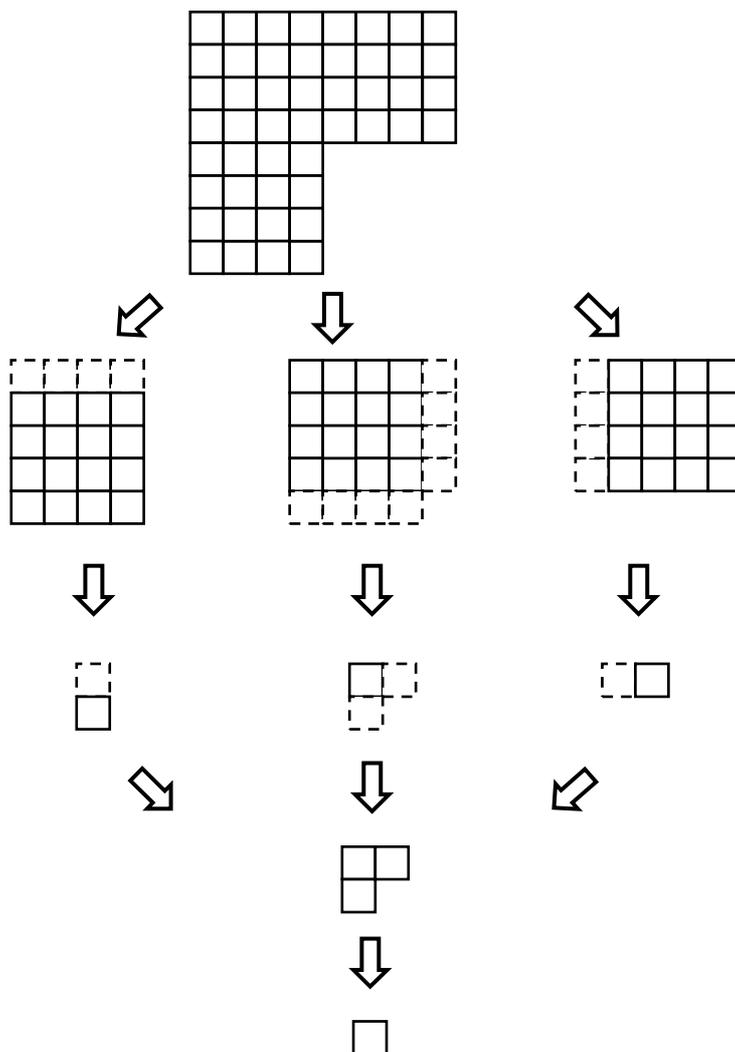


Рисунок 8. Глобальный уровень.

Алгебраический многосеточный метод был протестирован на нескольких задачах. СЛАУ давления в этих задачах решались многосеточным решателем, которому посвящена данная работа, и внешним алгебраическим многосеточным решателем BoomerAMG [3] с оптимизированными параметрами (Strong threshold = 0.25, Interpolation type: direct, Coarsen type: HMIS, Relax type: HSGS). Относительная точность в обоих случаях была равна 10^{-1} , максимальное число циклов многосеточного метода – 30. Выполнялись 1000 шагов метода SIMPLE. Результаты вычислительного эксперимента задачи расчёта турбулентного обтекания потоком вязкого несжимаемого газа (воздуха) грузового автомобиля КАМАЗ (рис. 9) представлены в таблице 1, а задачи течения охлаждающей жидкости в монолитной головке рубашки охлаждения блока цилиндров двигателя автомобиля (рис. 10) – в таблице 2.

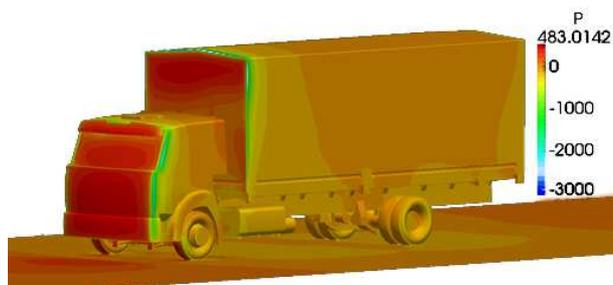


Рисунок 9. Задача расчёта турбулентного обтекания потоком вязкого несжимаемого газа грузового автомобиля КАМАЗ.

Таблица 1 – Результаты вычислительного эксперимента задачи расчёта турбулентного обтекания потоком вязкого несжимаемого газа грузового автомобиля КАМАЗ

Тип решателя	Число процессов	Время решения, секунд
Внутренний многосеточный	288	1198
Внешний многосеточный	288	3418

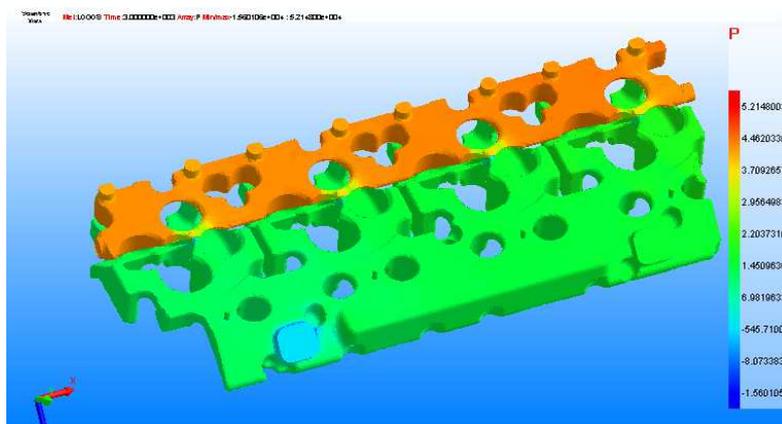


Рисунок 10. Задача течения охлаждающей жидкости в монолитной головке рубашки охлаждения блока цилиндров двигателя автомобиля.

Таблица 2 – Результаты вычислительного эксперимента задачи течения охлаждающей жидкости в монолитной головке рубашки охлаждения блока цилиндров двигателя автомобиля

Тип решателя	Число процессов	Время решения, секунд
Внутренний многосеточный	96	218
Внешний многосеточный	96	608

Таким образом, на модели памяти ЛОГОС был реализован алгебраический многосеточный метод. Тестирование метода на различных задачах показало его устойчивую работу и высокую скорость решения СЛАУ. Реализованный метод в составе программного комплекса ЛОГОС был передан на предприятия в сентябре 2012 г.

Список литературы

1. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М. : Лаборатория Базовых Знаний, 2002.
2. Ferziger J.H. Computational methods for fluid dynamics. - 3 rev. ed. / J.H. Ferziger, M. Peric - Berlin; New York; Barcelona; Hong Kong; London; Milan; Paris; Tokyo : Springer, 2002.
3. Henson V. BoomerAMG: A parallel algebraic multigrid solver and preconditioner / V. Henson, U. Yang // Appl. Numer. Math. – 2002. – Vol. 41. – P. 155-177.
4. Meurant G. Computer solution of large linear systems / G. Meurant – Amsterdam, Lausanne, New York, Oxford, Shannon, Singapore, Tokyo : Elsevier, 1999.
5. Stuben K. Algebraic multigrid (AMG): Experiences and comparisons // Appl. Math. Comp. – 1983. – Vol. 13. – P. 419-452.
6. Weiss J.M. Implicit solution of preconditioned Navier–Stokes equations using algebraic multigrid / J.M. Weiss, J.P. Maruszewski, W.A. Smith // AIAA Journal. – 1997. – Vol. 37. No. 1. – P. 29–36.

Рецензенты:

Щенников В.Н., д.ф.-м.н., профессор, заведующий кафедрой дифференциальных уравнений факультета математики и информационных технологий ФГБОУ ВПО «Мордовский государственный университет им. Н.П. Огарёва», г. Саранск.

Малыханов Ю.Б., д.ф.-м.н., профессор, профессор кафедры физики и методики обучения физике физико-математического факультета ФГБОУ ВПО «Мордовский государственный педагогический институт имени М.Е. Евсевьева», г. Саранск.