

О МЕТОДАХ ОПТИМИЗАЦИИ И РАСПАРАЛЛЕЛИВАНИЯ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ ГИДРОДИНАМИКИ

ФИЛАТОВ Е.Ю., асп.

Решается задача моделирования обтекания воздухом одиночного здания в двумерном случае. Для решения данной задачи предлагаются методы оптимизации, позволяющие увеличить точность и сократить время расчета, и методы распараллеливания.

Ключевые слова: гидродинамика, метод сгущающихся сеток, метод асинхронного интегрирования по времени, распараллеливание.

OPTIMIZATION METHODS AND PROBLEM PARALLELING OF COMPUTATIONAL HYDRODYNAMICS

FILATOV E.Yu., postgraduate

The article concerns the problem of airflow around separate building simulation in two dimension case. The optimization methods are suggested to solve this problem which enables to increase the accuracy and to reduce the calculation time and paralleling methods.

Key words: hydrodynamics, thickened networks method, asynchronous time integration method, paralleling.

Постановка задачи. Дано одиночное здание в форме параллелепипеда и дана скорость ветра, направление которой параллельно одной из сторон здания. Требуется рассчитать поля скорости и давления вокруг данного здания.

Задача решается в двумерном случае, т.е. в плоскости, параллельной поверхности земли. Ниже будет предложено решение данной задачи, а также методы оптимизации и распараллеливания для предложенного решения. Распространение изложенных методов на трехмерный случай не представляет большого труда.

Решение задачи. Для решения данной задачи используется система «давление – скорость». Для нахождения поля скорости \vec{U} решаются уравнения Навье-Стокса:

$$\frac{\partial U_x}{\partial t} = -\vec{U}\nabla U_x + \nu\Delta U_x - \frac{1}{\rho} \frac{\partial P}{\partial x}; \quad (1)$$

$$\frac{\partial U_y}{\partial t} = -\vec{U}\nabla U_y + \nu\Delta U_y - \frac{1}{\rho} \frac{\partial P}{\partial y}. \quad (2)$$

Для нахождения поля давления P решается уравнение Пуассона:

$$\Delta P = -\rho \left(\frac{\partial D}{\partial t} + \vec{U}\nabla D - \nu\Delta D + Q \right); \quad (3)$$

$$Q = \sum_{i=1}^2 \sum_{j=1}^2 \frac{\partial U_i}{\partial x_j} \frac{\partial U_j}{\partial x_i} = \quad (4)$$

$$= \left(\frac{\partial U_x}{\partial x} \right)^2 + 2 \frac{\partial U_x}{\partial y} \frac{\partial U_y}{\partial x} + \left(\frac{\partial U_y}{\partial y} \right)^2;$$

$$D = \text{div } \vec{U}. \quad (5)$$

В качестве начальных условий для поля скорости на всей рассчитываемой области берется некоторое начальное значение \vec{U}_0 , а для поля давления – некоторое начальное значение P_0 .

В качестве граничных условий для расчета поля скорости на выходной границе (рис. 1, правая граница) используется граничное условие II рода, при котором производная по нормали к границе равна 0, на остальных свободных границах используются граничные условия I рода, при которых скорость равна

\vec{U}_0 , а на здании используется граничное условие I рода, при котором скорость равна 0. При решении уравнения Пуассона для нахождения поля давления везде (на свободных границах и на здании) используются граничные условия II рода, при которых производная по нормали к границе равна 0.

Поскольку рассматриваемое течение турбулентно (т.к. число Рейнольдса велико), то требуется также рассчитывать поле турбулентной вязкости. Для этого используется модель турбулентности А.Н. Секундова. Турбулентная вязкость ν_{turb} в данной модели рассчитывается из следующего уравнения:

$$\begin{aligned} \frac{\partial \nu_{\text{turb}}}{\partial t} + \sum_{j=1}^2 U_j \frac{\partial \nu_{\text{turb}}}{\partial x_j} = \\ = \sum_{j=1}^2 \frac{\partial}{\partial x_j} \left((\nu_{\text{lam}} + \alpha \nu_{\text{turb}}) \frac{\partial \nu_{\text{turb}}}{\partial x_j} \right) + \\ + \nu_{\text{turb}} f \left(\frac{\nu_{\text{turb}}}{8 \nu_{\text{lam}}} \right) D - \gamma L_{\text{min}}^{-2} (\nu_{\text{lam}} + \beta \nu_{\text{turb}}) \nu_{\text{turb}}, \end{aligned} \quad (6)$$

где

$$D = \left(\sum_{i,j} \frac{\partial U_i}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right)^{1/2}; \quad (7)$$

$$f(z) = 0,2 \frac{z^2 + 1,47z + 0,2}{z^2 - 1,47z + 1}; \quad (8)$$

$$\alpha = 2; \quad \beta = 0,06; \quad \gamma = 1; \quad (9)$$

L_{min} – кратчайшее расстояние до ближайшей неподвижной стенки; ν_{lam} – ламинарная вязкость.

Таким образом, вязкость ν , входящая в уравнение Навье-Стокса и Пуассона, состоит из ламинарной и турбулентной составляющих:

$$\nu = \nu_{\text{lam}} + \nu_{\text{turb}}. \quad (10)$$

Для того чтобы поле турбулентной вязкости сформировалось, требуется задать его начальное значение отличным от 0, например,

$$\nu_{\text{turb}}^0 = 1000 \nu_{\text{lam}}. \quad (11)$$

В качестве граничных условий на входной границе (рис. 1, левая граница) и на здании используются граничные условия I рода, при которых турбу-

лентная вязкость равна 0, а на остальных свободных границах используются граничные условия II рода, при которых производная по нормали к границе равна 0.

Методы оптимизации. Изменение искомых полей в рассматриваемой области происходит неравномерно. Наиболее динамично изменение происходит вблизи здания. В удалении же от здания поля меняются довольно медленно. Кроме того, вблизи здания желательнее иметь большую точность расчета, так как поля вблизи здания в значительной степени влияют на расчет полей во всей области. Исходя из этого, расчет полей был бы более эффективным, если бы шаг сетки и шаг по времени вблизи здания были меньше, чем в удалении от здания. Этого позволяют добиться методы сгущающихся сеток и асинхронного интегрирования по времени.

Метод сгущающихся сеток состоит в том, что в разных частях рассматриваемой области выбирается разный шаг по расстоянию. Так, разделим всю область на 3 зоны (рис. 1). Малая зона будет расположена вокруг здания, средняя зона – вокруг малой зоны и большая – вокруг средней. В большой зоне возьмем шаг сетки h , в средней – $h/2$, а в малой – $h/4$.

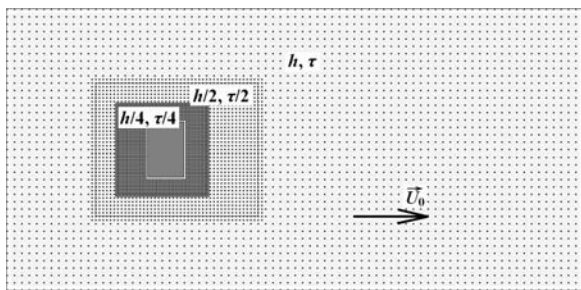


Рис. 1. Метод сгущающихся сеток и метод асинхронного интегрирования по времени

Кроме того, при уменьшении шага по расстоянию для сохранения устойчивости требуется также уменьшать шаг по времени, причем примерно в той же степени, что и шаг по расстоянию. Так, если в большой зоне шаг по времени равен τ , то в средней зоне возьмем шаг по времени $\tau/2$, а в малой зоне – $\tau/4$. Метод, при котором в разных частях расчетной области используются разные шаги по времени, будем называть *методом асинхронного интегрирования по времени*. Шаг по времени τ будем называть *большим* шагом по времени.

Упрощенно алгоритм расчета полей в условиях сгущающихся сеток выглядит следующим образом:

1. Расчет большого шага по времени в каждой зоне.
2. Обмен значениями полей на границах между зонами.

Под расчетом большого шага по времени понимается расчет 4-х шагов по времени $\tau/4$ в малой зоне и расчет 2-х шагов по времени $\tau/2$ в средней зоне. При расчете полей на границах зон используется граничное условие I рода и значения, полученные от соседней зоны.

Существуют два подхода в реализации метода сгущающихся сеток. Они различаются распределением узлов на границе между зонами. При 1-м подходе (рис. 2) граничные (0-е) узлы меньшей ($q+1$)-й зоны не совпадают с узлами большей q -й зоны. При 2-м подходе (рис. 3) граничные (0-е) узлы меньшей ($q+1$)-й зоны совпадают с ($A-1$)-ми узлами большей

q -й зоны. Здесь граничными узлами большей q -й зоны являются A -е узлы.

Таким образом, обмен значениями полей на границах схематично можно представить следующим образом. Для 1-го подхода:

$$q+1 \rightarrow q: G_{[q]A} = G_{[q+1]1}, \quad (12)$$

$$q \rightarrow q+1: G_{[q+1]0} = \frac{G_{[q]A-1} + G_{[q]A}}{2}. \quad (13)$$

Для 2-го подхода:

$$q+1 \rightarrow q: G_{[q]A} = G_{[q+1]2}, \quad (14)$$

$$q \rightarrow q+1: G_{[q+1]0} = G_{[q]A-1}. \quad (15)$$

Здесь G – данное поле; в квадратных скобках указывается номер зоны; последний индекс – номер узла по одной из координат.

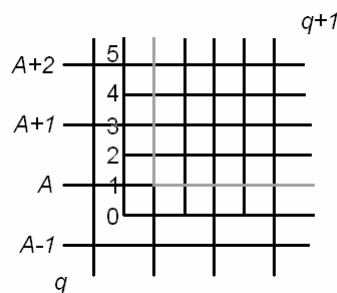


Рис. 2. Метод сгущающихся сеток: 1-й подход

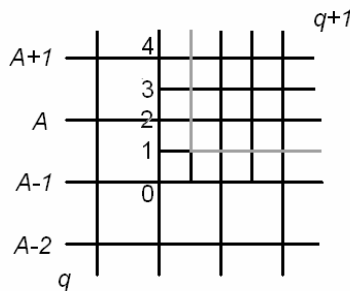


Рис. 3. Метод сгущающихся сеток: 2-й подход

Если сравнивать эти два подхода, то реализация 2-го подхода будет заметно проще. Это связано с тем, что при 1-м подходе при передаче значений узлов на границе с q -й зоны на ($q+1$)-ю (13) используется дополнительная линейная интерполяция. Кроме того, при 2-м подходе в трехмерном случае расчет полей в приземной области около здания будет точнее, чем при 1-м. Однако отметим, что поскольку при 1-м подходе сетки перекрываются по одному узлу меньшей зоны, а при 2-м подходе – по двум узлам, то при 2-м подходе число узлов сеток будет больше, чем при 1-м, и следовательно, время расчета также будет больше.

Данные методы оптимизации – метод сгущающихся сеток и метод асинхронного интегрирования по времени – позволяют увеличить точность вычислений в тех местах рассматриваемой области, где она актуальна, и уменьшить время расчета вследствие уменьшения шагов по расстоянию и по времени в остальной части рассматриваемой области. Результаты расчета полей показаны на рис. 4.

Метод распараллеливания. Опишем метод распараллеливания в условиях изложенных методов оптимизации. Реализуем распараллеливание по пространству. Заметим, что структура сгущающихся сеток заметно усложняет алгоритм распараллеливания.

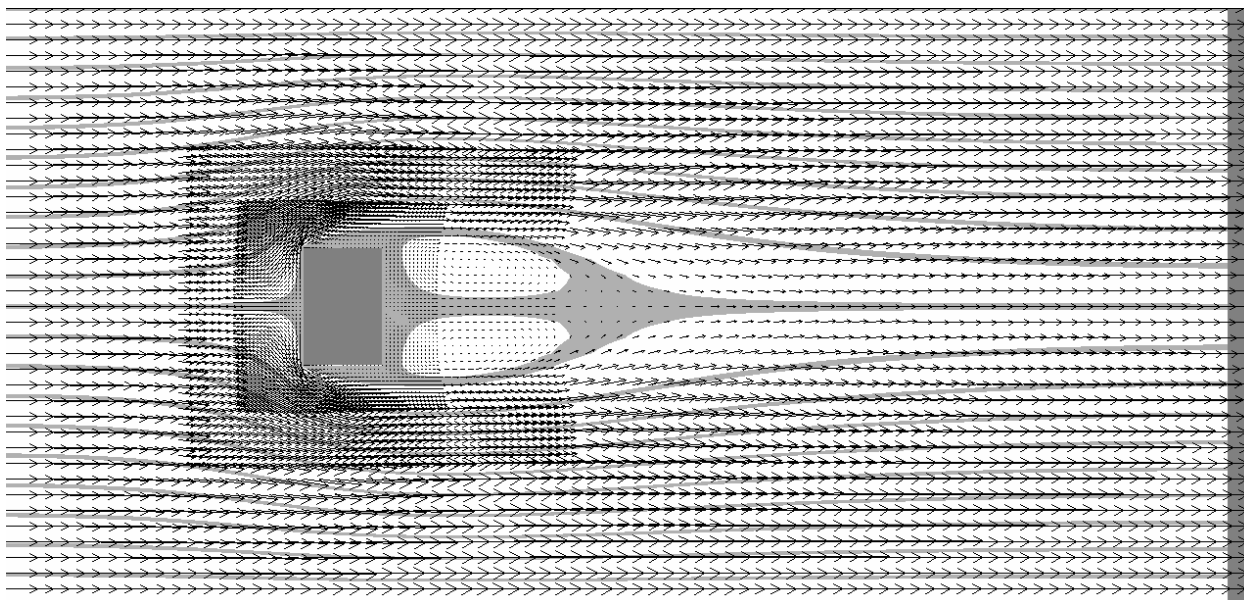


Рис. 4. Результаты расчета поля скорости и линий тока: стрелками изображено поле скорости, а серыми жирными линиями – поле функции тока, вычисленное из поля скоростей

Прежде всего отметим, что при распараллеливании используется клиент-серверная архитектура. Так, среди всех процессоров выбирается один процессор – сервер, который выполняет организационную работу, в частности, разделяет область расчета на участки между другими процессорами. Остальные процессоры – клиенты – производят расчет искомых полей, каждый на своем участке.

Предлагаемый метод распараллеливания включает в себя три алгоритма:

- 1) алгоритм деления области расчета на участки между процессорами;
- 2) алгоритм регистрации границ;
- 3) алгоритм обмена значениями граничных узлов.

Остановимся подробнее на первом алгоритме.

Сначала все процессоры распределяются между зонами в зависимости от количества узлов в зонах. Число узлов в зонах подсчитывается с учетом асинхронного интегрирования по времени, т.е. в малой зоне число узлов умножается на 4, а в средней – на 2. Это связано с предположением, что время расчета пропорционально числу узлов в области расчета и числу выполняемых шагов по времени.

Процессоры между зонами следует распределять, по возможности, равномерно. Для этого каждой зоне выделяется сначала по одному процессору. Затем для каждой q-й зоны вычисляется величина N_q / r_q , где N_q – число узлов в q-й зоне, а r_q – число процессоров, ей принадлежащих. Той зоне, где данное отношение наибольшее, выделяется еще один процессор. Эти действия выполняются до тех пор, пока все процессоры не будут распределены.

Далее каждая зона разделяется на участки между принадлежащими ей процессорами. Для этого она разделяется сначала на вертикальные полосы, а затем каждая полоса разделяется горизонтальными линиями на участки (рис. 5).

Объясним, как это происходит. Очевидно, распараллеливание будет наиболее эффективным, если все участки будут содержать примерно равное число узлов и будут иметь форму квадратов. Квадратная форма участка сводит к минимуму периметр

участка, и следовательно, уменьшает время на обмена граничными узлами.

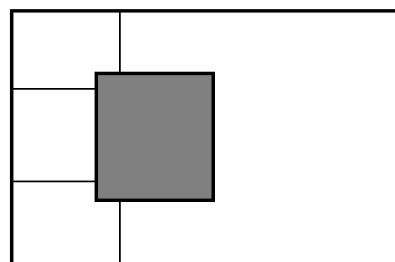


Рис. 5. Разделение зоны на участки между процессорами

Найдем, чему равна сторона такого квадратного участка. Пусть N – число узлов в зоне, а P – число процессоров в зоне. Тогда m – среднее число узлов в участке:

$$m = \left\lfloor \frac{N}{P} \right\rfloor; \quad (16)$$

a – средняя сторона участка:

$$a = \left\lceil \sqrt{m} \right\rceil. \quad (17)$$

Отрежем вертикальную полосу шириной a . Данная ширина – предварительная. Впоследствии она будет уточняться. Подсчитаем n – число узлов в данной предварительной полосе. Заметим, что полоса может пересекать другую зону или собственно здание. Затем найдем p – число процессоров в данной полосе:

$$p = \left\lfloor \frac{n}{m} \right\rfloor. \quad (18)$$

Здесь округление следует производить до ближайшего целого. Причем заметим, что полоса при пересечении с другой зоной или зданием может разделиться на две несвязные половины. В этом случае предпочтительнее, если число процессоров на ней p будет четным.

Затем следует уточнить ширину полосы для обеспечения равномерности деления зоны на участки, поскольку при нахождении величины p про-

исходит достаточно сильное округление (если число процессоров, принадлежащих зоне, мало). Для этого находится уточненное число узлов на данной полосе n' , равное произведению числа процессоров в полосе p и среднему числу узлов в участке m :

$$n' = pm. \quad (19)$$

Затем находится уточненная ширина полосы a' , при которой полоса содержит примерно n' узлов.

Заметим также, что эффективность распараллеливания повышается, если граница полосы совпадает с вертикальной границей следующей зоны или здания, поскольку таким образом уменьшается общая протяженность границ.

После того как была найдена очередная полоса, ее следует разделить горизонтальными линиями на участки. Алгоритм такого разделения достаточно прост. Здесь лишь необходимо следить за тем, чтобы участки содержали примерно равное число узлов. Эффективность распараллеливания при этом также повышается, если горизонтальные линии совпадают с горизонтальными границами следующей зоны или здания.

Перед тем как начинать расчет полей, необходимо зарегистрировать все границы между соседними процессорами и сохранить их в реестре границ.

Алгоритмы регистрации границ и обмена границами идейно просты, однако технически довольно сложны. В частности, это связано с тем, что для каждого процессора существует 3 вида границ: граница с процессором из той же зоны, граница с про-

цессором из меньшей зоны и граница с процессором из большей зоны.

Алгоритм расчета полей при распараллеливании напоминает алгоритм расчета полей в однопроцессорном режиме в условиях сгущающихся сеток:

1. Расчет большого шага по времени каждым процессором.

2. Обмен значениями полей на границах между процессорами.

Данная задача была запущена на вычислительной системе МВС-100 на 6 процессорах (сервер и 5 клиентов). Эффективность распараллеливания получилась равной 76 %:

$$E_p = \frac{T_1}{pT_p}. \quad (20)$$

Здесь T_1 – время расчета в однопроцессорном режиме; T_p – время расчета в многопроцессорном режиме на p клиентских процессорах; p – число клиентских процессоров.

Это показывает высокую эффективность предложенного алгоритма распараллеливания.

Список литературы

1. **Роуч П.** Вычислительная гидродинамика. – М.: Мир, 1980.
2. **Численные** методы и параллельные вычисления для задач механики жидкости, газа и плазмы: Учеб. пособие / Э.Ф. Балаев, Н.В. Нуждин, В.В. Пекунов и др.; Иван. гос. энерг. ун-т. – Иваново, 2003.

Филатов Евгений Юрьевич,
ГОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,
аспирант кафедры программного обеспечения компьютерных систем,
телефон (4932) 26-98-29.