

И.А. Широков

ИТЕРАЦИОННЫЙ МЕТОД РЕШЕНИЯ УРАВНЕНИЯ ПУАССОНА И ЕГО РЕАЛИЗАЦИЯ НА МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ*

Введение

Бурное развитие многопроцессорных систем за последние годы дало реальную возможность численного моделирования сложных физических явлений, в частности, газо- и гидродинамических течений в пространственной постановке. Для эффективного использования многих процессоров требуются соответствующие алгоритмы. Такие алгоритмы должны не только адекватно описывать физическую реальность, но и учитывать особенности вычислительных систем. В частности, представляют интерес алгоритмы, обладающие линейным (или близком к таковому) ростом производительности при увеличении числа используемых процессоров.

Построение математических моделей гидродинамических течений (а также ряда других задач: электронной оптики, изучения воздействия заряженных частиц на поверхность и др.) часто приводит к необходимости решения краевой задачи для уравнения Пуассона. Разработаны различные численные методы решения такого рода задач, в частности, учитывающие симметрию соответствующей системы линейных уравнений [1]. В настоящей работе используется метод решения уравнения Пуассона для задач в двумерной и трёхмерной пространственной постановке, являющийся вариантом метода прогонки [2]. К его достоинствам относятся простота многопроцессорной реализации, получаемая при этом высокая эффективность распараллеливания, лёгкость адаптации к различным типам граничных условий.

В данной работе в качестве примера применения предлагаемого метода решения уравнения Пуассона рассматривается моделирование течения вязкой несжимаемой жидкости в каверне. Такое течение – относительно сложный тест для верификации и оценки эффективности численных методов. Для невысоких чисел Рейнольдса $Re \leq 1000$ течение является ламинарным и стационарным. Течение в плоскости симметрии каверны хорошо описывается двумерными моделями, и может служить эталонным для трёхмерного расчёта [3]. В качестве математической модели используются квазигидродинамические (КГД) уравнения. По сравнению с традиционной системой Навье-Стокса, КГД уравнения имеют ряд преимуществ. В частности, они включают диссипативные слагаемые, обеспечи-

* Работа выполнена при финансовой поддержке грантов РФФИ 01-01-00061 и ИНТАС № 2000-0617

вающие регуляризацию при численном решении [4–6].

Алгоритм решения уравнения Пуассона

Пусть задана пространственная область в виде параллелепипеда со сторонами, параллельными осям X, Y, Z соответственно. Введём прямоугольную неравномерную сетку $x_0 \dots x_{N_x-1}, y_0 \dots y_{N_y-1}, z_0 \dots z_{N_z-1}$. Запишем разностный аналог уравнения Пуассона в общем виде (f – неизвестная функция):

$$Gf = Af_{i-1} + Bf_{i+1} + Cf_{j-1} + Df_{j+1} + Ef_{k-1} + Ff_{k+1} + H. \quad (1)$$

Величины $A \dots H$ могут зависеть от x, y, z (т. е. от i, j, k). Считаем, что если опущены какие-либо индексы, предполагается, что они равны текущему значению, т. е. $f \equiv f_{ijk}, f_{i+1} \equiv f_{i+1jk}, A \equiv A_{ijk}$ и т. п. Ищем решение (1) в виде

$$f = \alpha f_{i+1} + \beta f_{j+1} + \gamma f_{k+1} + \delta. \quad (2)$$

Прогоночные коэффициенты $\alpha, \beta, \gamma, \delta$ зависят от i, j, k . Подставим (2) в (1) так, чтобы исключить члены с $f_{i-1}, f_{j-1}, f_{k-1}$. Для этого запишем (2) для этих значений индексов.

$$\text{Для } i-1: \quad f_{i-1} = \alpha_{i-1} f + \beta_{i-1} f_{i-1,j+1} + \gamma_{i-1} f_{i-1,k+1} + \delta_{i-1}.$$

$$\text{Для } j-1: \quad f_{j-1} = \alpha_{j-1} f_{i+1,j-1} + \beta_{j-1} f + \gamma_{j-1} f_{j-1,k+1} + \delta_{j-1}.$$

$$\text{Для } k-1: \quad f_{k-1} = \alpha_{k-1} f_{i+1,k-1} + \beta_{k-1} f_{j+1,k-1} + \gamma_{k-1} f + \delta_{k-1}.$$

После подстановки имеем:

$$\begin{aligned} Gf = & A(\alpha_{i-1} f + \beta_{i-1} f_{i-1,j+1} + \gamma_{i-1} f_{i-1,k+1} + \delta_{i-1}) + Bf_{i+1} + \\ & + C(\alpha_{j-1} f_{i+1,j-1} + \beta_{j-1} f + \gamma_{j-1} f_{j-1,k+1} + \delta_{j-1}) + Df_{j+1} + \\ & + E(\alpha_{k-1} f_{i+1,k-1} + \beta_{k-1} f_{j+1,k-1} + \gamma_{k-1} f + \delta_{k-1}) + Ff_{k+1} + H. \end{aligned}$$

Перепишем это выражение в виде (2). Для этого перегруппируем слагаемые:

$$\begin{aligned} (G - A\alpha_{i-1} - C\beta_{j-1} - E\gamma_{k-1})f = & Bf_{i+1} + Df_{j+1} + Ff_{k+1} + \\ & + A(\beta_{i-1} f_{i-1,j+1} + \gamma_{i-1} f_{i-1,k+1} + \delta_{i-1}) \\ & + C(\alpha_{j-1} f_{i+1,j-1} + \gamma_{j-1} f_{j-1,k+1} + \delta_{j-1}) \\ & + E(\alpha_{k-1} f_{i+1,k-1} + \beta_{k-1} f_{j+1,k-1} + \delta_{k-1}) + H. \end{aligned}$$

Отсюда получаем выражения для прогоночных коэффициентов:

$$\alpha = B/k, \quad \beta = D/k, \quad \gamma = F/k,$$

$$\delta = \left[A(\beta_{i-1} f_{i-1,j+1} + \gamma_{i-1} f_{i-1,k+1} + \delta_{i-1}) + C(\alpha_{j-1} f_{i+1,j-1} + \gamma_{j-1} f_{j-1,k+1} + \delta_{j-1}) \right]$$

$$+E(\alpha_{k-1}f_{i+1k-1} + \beta_{k-1}f_{j+1k-1} + \delta_{k-1}) + H]/k, \quad (3)$$

где $k = G - A\alpha_{i-1} - C\beta_{j-1} - E\gamma_{k-1}$.

Заметим, что в выражении для δ присутствуют значения неизвестной функции f . Это — особенность случая двух и более измерений. Эта особенность является принципиальной для данного метода, предназначенного для решения задач в двумерной и трёхмерной постановке. В отличие от одномерной прогонки, формулы (3) нельзя использовать для прямого вычисления прогоночных коэффициентов. Однако их можно использовать для построения итерационного алгоритма. При этом значения f в выражении для δ берутся с нижнего итерационного слоя.

Предположим, требуется решить задачу первого рода, когда заданы значения f на границе.

Зададим разумное начальное приближение f , соответствующее граничным условиям. Итерационный шаг состоит из прямого и обратного хода.

В прямом ходе по формулам (3) определяются прогоночные коэффициенты α , β , γ , δ . Значения f на плоскостях $x=0$, $y=0$, $z=0$ определяют коэффициент δ через (2): $\delta_{0jk} = f_{0jk}$, $\delta_{i0k} = f_{i0k}$, $\delta_{ij0} = f_{ij0}$. При этом значения α , β , γ на этих плоскостях равны нулю. Значения f в выражении для δ (3) на первом шаге берутся из начального приближения, а на последующих — с предыдущего итерационного слоя.

В процессе обратного хода с помощью (2) определяются значения f на текущем итерационном слое. При этом используются остальные граничные условия для f при $x = x_{\max}$, $y = y_{\max}$, $z = z_{\max}$. В качестве условия прекращения итераций используется

$$\max_{ijk}^{Up} (f - f)/(N_x \cdot N_y \cdot N_z) < \varepsilon_2, \quad (4)$$

где f^{Up} — значение на верхнем итерационном слое.

При других граничных условиях можно использовать этот же алгоритм, только на каждом итерационном шаге необходимо реализовывать эти условия. Например, пусть на одной из граней расчетной области ($x=0$) заданы условия второго рода $\partial f / \partial \bar{n} = 0$. Тогда после прямого и обратного хода выполняем присваивания

$$f_{0jk}^{Up} = f_{1jk}. \quad (5)$$

Если условия второго рода ставятся на всех гранях расчётной области (как это имеет место в данной работе), для каждой грани необходимо выполнить присваивания, аналогичные (5). В этом случае для устой-

чивости вычислений значение f в одной точке надо поддерживать постоянным. Например, можно поддерживать $f_{000} = 0$. Для этого необходимо выполнить присваивания $f_{ijk} = f_{ijk} - f_{000}$ для всех i, j, k .

Реализация алгоритма решения уравнения Пуассона на параллельной вычислительной системе

Рассмотрим случай трёхмерной постановки (для двумерной постановки рассуждения аналогичны). Разобьём область расчёта (рис. 1) на N подобластей плоскостями, перпендикулярными оси X так, чтобы соседние подобласти перекрывались. Размеры подобластей примерно одинаковы. Отношение перекрывающихся частей к длине области 10...20%. Задействуем N процессоров. Каждый процессор реализует вычисления в одной подобласти. Алгоритм вычислений в одной подобласти идентичен описанному выше алгоритму для одного процессора. В качестве условия на внутренней границе подобласти используются значения, вычисленные соседним процессором.

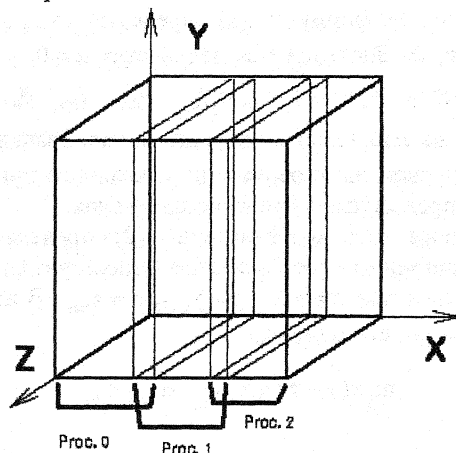


Рис. 1. Схема расчётной области

При этом результат вычислений (в т. ч. число итераций на каждом шаге по времени) зависит от количества подобластей (процессоров) и от размеров перекрывающихся частей подобластей, хотя и совсем слабо. При использовании данного алгоритма в конкретной задаче (расчёт № 1, см. ниже) справедлива оценка (роль f играет p)

$$\max_{ij} |(p_{ij}^{1proc} - p_{ij}^{5proc}) / p_{ij}^{1proc}| < 0.0005,$$

где p_{ij}^{1proc} и p_{ij}^{5proc} — конечное поле давления в центральном сечении

($z = 0.5$) при вычислениях на одном и пяти процессорах соответственно. Эта оценка показывает, что при использовании различного числа процессоров конечный результат меняется очень мало.

При использовании некоторых других многопроцессорных алгоритмов (например, для явной разностной схемы) результат вообще не зависит от числа процессоров [7]. Такой алгоритм можно построить и для уравнения Пуассона (для схемы (1)–(5)). Но при этом процессоры должны обмениваться очень короткими порциями данных. Попытка реализации такого алгоритма показала его неэффективность. Это связано с тем, что скорость межпроцессорных обменов зависит от длины пересылаемых массивов – чем больше массив, тем выше скорость. Если массивы слишком малы, пропускная способность сети падает и эффективность работы программы тоже.

Постановка задачи и математическая модель

Рассмотрим течение вязкой несжимаемой изотермической жидкости в кубической каверне с твёрдыми стенками (рис. 1). Длина ребра куба равна H . Верхняя крышка каверны ($y=H$) движется с постоянной скоростью U_0 в направлении оси X .

В качестве математической модели используем квазигидродинамические (КГД) уравнения для вязкой несжимаемой изотермической жидкости ([4, 5]):

$$\nabla_i u^i = \nabla_i w^i, \quad (6)$$

$$\frac{\partial}{\partial t} u^k + \nabla_i u^i u^k + \nabla^k p = \nabla_i \Pi_{NS}^{ik} + \nabla_i (u^i w^k) + \nabla_i (w^i u^k). \quad (7)$$

Полагаем, что $\rho = const = 1$. Тензор вязких напряжений имеет вид

$$\Pi_{NS}^{ik} = \nu (\nabla^k u^i + \nabla^i u^k). \quad (8)$$

Вектор потока массы вычисляется следующим образом:

$$j^k = (u^k - w^k), \quad (9)$$

где $w^k = \tau (u^i \nabla_i u^k + \nabla^k p)$.

Давление p отсчитывается от гидростатического. $\vec{u} = (u^1, u^2, u^3)$ – скорость, $\nu = \eta / \rho$ – коэффициент кинематической вязкости, τ – характерный масштаб временного сглаживания.

КГД уравнения отличаются от уравнений Навье-Стокса дополнительными дивергентными слагаемыми с параметром τ , имеющим размерность времени. При $\tau \rightarrow 0$ КГД уравнения переходят в систему Навье-Стокса. В стационарном случае дополнительные дивергентные члены имеют порядки малости $O(\tau^2)$.

Подставив (8) и (9) в (6)–(7), запишем КГД систему в следующем виде:

$$\nabla_j u^j = \nabla_i \tau (u^j \nabla_j u^i + \nabla^i p), \quad (10)$$

$$\begin{aligned} \frac{\partial u^i}{\partial t} + \nabla_j u^j u^i + \nabla^i p = \nabla_j \nu (\nabla^j u^i + \nabla^i u^j) + \\ + \nabla_j \tau \left[(u^k \nabla_k u^j + \nabla^j p) u^i + u^j (u^k \nabla_k u^i + \nabla^i p) \right]. \end{aligned} \quad (11)$$

Введём безразмерные величины (помечены знаком \sim) с помощью соотношений:

$$\begin{aligned} x = \tilde{x}H, \quad y = \tilde{y}H, \quad z = \tilde{z}H, \quad u_x = \tilde{u}_x U_0, \quad u_y = \tilde{u}_y U_0, \quad u_z = \tilde{u}_z U_0, \\ p = \tilde{p} U_0^2, \quad t = \tilde{t}H/U_0, \quad \text{Re} = (U_0 H)/\nu. \end{aligned}$$

В дальнейшем знак \sim опускается. Запишем систему (10)–(11) в безразмерном виде:

$$\begin{aligned} \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = \frac{1}{\tau} \left[\frac{\partial}{\partial x} u_x + \frac{\partial}{\partial y} u_y + \frac{\partial}{\partial z} u_z \right] - \\ - \left[\frac{\partial}{\partial x} \left(u_x \frac{\partial}{\partial x} u_x + u_y \frac{\partial}{\partial y} u_x + u_z \frac{\partial}{\partial z} u_x \right) + \right. \\ \left. + \frac{\partial}{\partial y} \left(u_x \frac{\partial}{\partial x} u_y + u_y \frac{\partial}{\partial y} u_y + u_z \frac{\partial}{\partial z} u_y \right) + \right. \\ \left. + \frac{\partial}{\partial z} \left(u_x \frac{\partial}{\partial x} u_z + u_y \frac{\partial}{\partial y} u_z + u_z \frac{\partial}{\partial z} u_z \right) \right]. \quad (12) \\ \frac{\partial}{\partial t} u_x + \frac{\partial}{\partial x} u_x^2 + \frac{\partial}{\partial y} u_x u_y + \frac{\partial}{\partial z} u_x u_z + \frac{\partial}{\partial x} p = \\ = \frac{\partial}{\partial x} \left[\frac{2}{\text{Re}} \frac{\partial}{\partial x} u_x + 2\tau (u_x^2 \frac{\partial}{\partial x} u_x + u_x u_y \frac{\partial}{\partial y} u_x + u_x u_z \frac{\partial}{\partial z} u_x + u_x \frac{\partial}{\partial x} p) \right] + \\ + \frac{\partial}{\partial y} \left[\frac{1}{\text{Re}} \left(\frac{\partial}{\partial y} u_x + \frac{\partial}{\partial x} u_y \right) + \tau (u_x^2 \frac{\partial}{\partial x} u_y + u_x u_y \frac{\partial}{\partial y} u_y + u_x u_z \frac{\partial}{\partial z} u_y + \right. \\ \left. + u_x \frac{\partial}{\partial y} p + u_y u_x \frac{\partial}{\partial x} u_x + u_y^2 \frac{\partial}{\partial y} u_x + u_y u_z \frac{\partial}{\partial z} u_x + u_y \frac{\partial}{\partial x} p) \right] + \\ + \frac{\partial}{\partial z} \left[\frac{1}{\text{Re}} \left(\frac{\partial}{\partial z} u_x + \frac{\partial}{\partial x} u_z \right) + \tau (u_x^2 \frac{\partial}{\partial x} u_z + u_x u_y \frac{\partial}{\partial y} u_z + u_x u_z \frac{\partial}{\partial z} u_z + \right. \end{aligned}$$

$$+u_x \frac{\partial}{\partial z} p + u_z u_x \frac{\partial}{\partial x} u_x + u_z u_y \frac{\partial}{\partial y} u_x + u_z^2 \frac{\partial}{\partial z} u_x + u_z \frac{\partial}{\partial x} p \Big]. \quad (13)$$

$$\begin{aligned} & \frac{\partial}{\partial t} u_y + \frac{\partial}{\partial x} u_x u_y + \frac{\partial}{\partial y} u_y^2 + \frac{\partial}{\partial z} u_y u_z + \frac{\partial}{\partial y} p = \\ & = \frac{\partial}{\partial x} \left[\frac{1}{\text{Re}} \left(\frac{\partial}{\partial x} u_y + \frac{\partial}{\partial y} u_x \right) + \tau(u_x u_y \frac{\partial}{\partial x} u_x + u_y^2 \frac{\partial}{\partial y} u_x + u_y u_z \frac{\partial}{\partial z} u_x + \right. \\ & \quad \left. + u_y \frac{\partial}{\partial x} p + u_x^2 \frac{\partial}{\partial x} u_y + u_x u_y \frac{\partial}{\partial y} u_y + u_x u_z \frac{\partial}{\partial z} u_y + u_x \frac{\partial}{\partial y} p \right) \Big] + \\ & + \frac{\partial}{\partial y} \left[\frac{2}{\text{Re}} \frac{\partial}{\partial y} u_y + 2\tau(u_x u_y \frac{\partial}{\partial x} u_y + u_y^2 \frac{\partial}{\partial y} u_y + u_y u_z \frac{\partial}{\partial z} u_y + u_y \frac{\partial}{\partial y} p) \right] + \\ & + \frac{\partial}{\partial z} \left[\frac{1}{\text{Re}} \left(\frac{\partial}{\partial z} u_y + \frac{\partial}{\partial y} u_z \right) + \tau(u_x u_y \frac{\partial}{\partial x} u_z + u_y^2 \frac{\partial}{\partial y} u_z + u_y u_z \frac{\partial}{\partial z} u_z + \right. \\ & \quad \left. + u_y \frac{\partial}{\partial z} p + u_x u_z \frac{\partial}{\partial x} u_y + u_y u_z \frac{\partial}{\partial y} u_y + u_z^2 \frac{\partial}{\partial z} u_y + u_z \frac{\partial}{\partial y} p \right) \Big]. \quad (14) \end{aligned}$$

$$\begin{aligned} & \frac{\partial}{\partial t} u_z + \frac{\partial}{\partial x} u_x u_z + \frac{\partial}{\partial y} u_y u_z + \frac{\partial}{\partial z} u_z^2 + \frac{\partial}{\partial z} p = \\ & = \frac{\partial}{\partial x} \left[\frac{1}{\text{Re}} \left(\frac{\partial}{\partial x} u_z + \frac{\partial}{\partial z} u_x \right) + \tau(u_x u_z \frac{\partial}{\partial x} u_x + u_y u_z \frac{\partial}{\partial y} u_x + u_z^2 \frac{\partial}{\partial z} u_x + \right. \\ & \quad \left. + u_z \frac{\partial}{\partial x} p + u_x^2 \frac{\partial}{\partial x} u_z + u_x u_y \frac{\partial}{\partial y} u_z + u_x u_z \frac{\partial}{\partial z} u_z + u_x \frac{\partial}{\partial z} p \right) \Big] + \\ & + \frac{\partial}{\partial y} \left[\frac{1}{\text{Re}} \left(\frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z} \right) + \tau(u_x u_z \frac{\partial}{\partial x} u_y + u_y u_z \frac{\partial}{\partial y} u_y + u_z^2 \frac{\partial}{\partial z} u_y + \right. \\ & \quad \left. + u_z \frac{\partial}{\partial y} p + u_x u_y \frac{\partial}{\partial x} u_z + u_y^2 \frac{\partial}{\partial y} u_z + u_y u_z \frac{\partial}{\partial z} u_z + u_y \frac{\partial}{\partial z} p \right) \Big] + \\ & + \frac{\partial}{\partial z} \left[\frac{2}{\text{Re}} \frac{\partial}{\partial z} u_z + 2\tau(u_x u_z \frac{\partial}{\partial x} u_z + u_y u_z \frac{\partial}{\partial y} u_z + u_z^2 \frac{\partial}{\partial z} u_z + u_z \frac{\partial}{\partial z} p) \right] \quad (15) \end{aligned}$$

Заметим, что уравнение неразрывности записано в форме уравнения Пуассона для давления (именно это уравнение будем решать с помощью описанного выше итерационного метода). Неизвестными величинами являются компоненты вектора скорости и давление. Дополним систему граничными условиями.

На неподвижной твёрдой стенке используются условия прилипания:

$$\vec{u} = 0.$$

На скользящей стенке задаются условия:

$$u_x = 1, u_y = 0, u_z = 0.$$

Граничные условия для давления следуют из условия непротекания массы и имеют вид:

$$\frac{\partial p}{\partial \vec{n}} = 0, \quad (16)$$

где \vec{n} – нормаль к поверхности (см. [6]).

В качестве начальных условий задаются состояние покоя для скорости (кроме движущейся крышки) и постоянное поле давления. Так как поле давления является решением задачи второго рода для уравнения Пуассона, оно определяется с точностью до произвольной аддитивной постоянной. Поэтому для устранения неоднозначности на протяжении расчёта давление поддерживается постоянным в вершине $x = 0, y = 0, z = 0$.

Алгоритм решения уравнений движения

Для использования метода конечных разностей введём внутри кубической каверны пространственную сетку $x_0 \dots x_{N_x-1}, y_0 \dots y_{N_y-1}, z_0 \dots z_{N_z-1}$. Будем вычислять компоненты скорости и давление в её узлах везде, кроме границы области. Границы расчётной области расположены в полуцелых точках. Для аппроксимации граничных условий используются фиктивные ячейки. Аппроксимируем пространственные производные центральными разностями. Значения величин в полуцелых узлах определим как полусуммы их значений в целых узлах, а значения величин в центрах двумерных ячеек вычислим как среднее арифметическое значений в прилегающих узлах. Для производных первого порядка:

$$\frac{\partial}{\partial x} f \approx \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}.$$

Для производных второго порядка вдоль одной оси:

$$\frac{\partial}{\partial x} g \frac{\partial}{\partial x} f \approx \frac{1}{x_{i+1} - x_{i-1}} \cdot \left[\frac{g_{i+1} + g_i}{x_{i+1} - x_i} (f_{i+1} - f_i) - \frac{g_i + g_{i-1}}{x_i - x_{i-1}} (f_i - f_{i-1}) \right].$$

Для смешанных производных второго порядка:

$$\begin{aligned} \frac{\partial}{\partial x} g \frac{\partial}{\partial y} f \approx & \frac{1}{2 \cdot (y_{j+1} - y_{j-1}) \cdot (x_{i+1} - x_{i-1})} \times \\ & \times \left[(g_{i+1} + g_i) \cdot (f_{i+1j+1} + f_{ij+1} + f_{i+1j-1} + f_{ij-1}) - \right. \end{aligned}$$

$$-(g_i + g_{i-1}) \cdot (f_{ij+1} + f_{i-1,j+1} + f_{ij-1} + f_{i-1,j-1})].$$

Производные по времени аппроксимируются так:

$$\frac{\partial}{\partial t} f \approx \frac{f^{Up} - f}{ht},$$

где f^{Up} — значение на верхнем по времени слое.

На основе приведённых выражений для производных построим явную разностную схему для уравнений (13)–(15). В дальнейшем сетка считается равномерной с шагом h . Будем искать установившееся решение методом последовательных приближений. При этом на каждом временном слое потребуется решать разностный аналог уравнения Пуассона (1) с граничными условиями (16). Течение считается установившимся, если

$$\max_{\lambda} \left(\frac{\sum_{ijk}^{Up} (u_{\lambda} - u_{\lambda})}{N_x \cdot N_y \cdot N_z} \right) < \varepsilon_1, \quad (17)$$

где u_{λ}^{Up} — значения скорости на верхнем временном слое, $\lambda = x, y, z$.

Объединение алгоритмов

Метод конечных разностей приводит к двум вложенным итерационным алгоритмам. Внешний (разностная схема для уравнений движения) будем называть «шаги по времени», внутренний (решение уравнения Пуассона) назовём «итерации». В качестве начальных условий для итераций используем значения давления с предыдущего слоя по времени. На самом первом временном слое используем условие $p_{ijk} = 0$.

На начальном этапе расчёта при заданной точности решения уравнения Пуассона ε_2 (4) требуется большое (порядка 100) число итераций для каждого шага по времени, и на итерации уходит большая часть машинного времени (90%). В дальнейшем поле давления постепенно устанавливается, число итераций на каждом шаге по времени снижается, и с какого-то шага (1000) достаточно лишь одной итерации на каждый шаг по времени. На Рис. 2 для однопроцессорного расчёта (в Табл. 1 это расчёт № 1) пунктирной линией показано число итераций для одного шага по времени, а сплошной линией процентное отношение общего времени, расходуемого на итерации к общему времени расчёта до текущего шага по времени (его номер отложен по оси абсцисс). Как видно из Рис. 2, доля общего машинного времени расчёта задачи, необходимая для итераций, в данном случае около 30%, если все вычисления ведутся на одном процессоре.

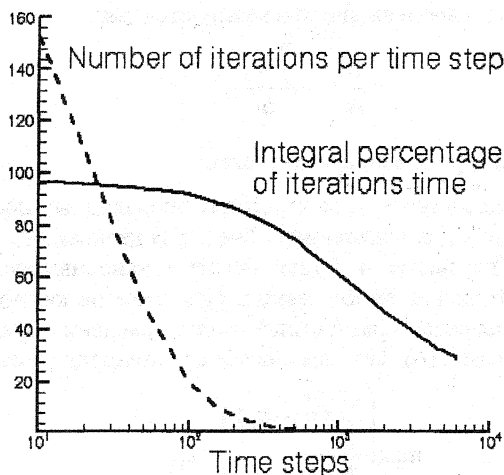


Рис. 2. Число итераций и время их выполнения в зависимости от шага по времени при расчёте № 1 на одном процессоре

Оба вложенных алгоритма могут быть распараллелены. Как видно из Рис. 2, для построения эффективного алгоритма необходимо реализовать многопроцессорный алгоритм как для решения уравнений движения, так и для решения уравнения Пуассона. В настоящей работе моделирование течения в каверне рассматривается только как пример применения итерационного параллельного метода решения уравнения Пуассона, и распараллелен только внутренний алгоритм – итерации, а вычисления, соответствующие уравнениям движения, производятся на одном процессоре. Оценки эффективности распараллеливания приведены только для решения уравнения Пуассона.

Результаты расчётов

Моделирование течения в каверне проводилось для чисел Рейнольдса Re , равных 100, 400, 800, 1000 для различных пространственных сеток. В Табл. 1 приведены параметры расчётов. Значение параметра τ выбиралось равным $\tau = 1/Re$, исходя из условий точности расчёта, с одной стороны, и устойчивости, с другой. Для всех расчётов критерий сходимости $\varepsilon_1 = 10^{-6}$ (17). Все пространственные сетки равномерные с одинаковыми размерами ячеек по X, Y, Z направлениям. Вычисления проводились на МВС-1000М [8]. Использовался стандарт MPI (Message Passing Interface). Задействовались от 1 до 6 процессоров.

На Рис. 4–13 показаны векторы скорости, линии тока и поля давления в сечении $z=0.5$. Это сечение является плоскостью симметрии в данной задаче, на этой плоскости $u_z \equiv 0$, так что приведённые результаты хорошо соответствуют результатам, полученным на основе двумерных моделей ([3, 6]). Видно, что увеличение числа Рейнольдса требует более подробной сетки, т. к. при расчёте на грубой сетке линии уровня давления теряют гладкость (ср. Рис. 11 и 13).

На Рис. 14 и 15 для расчёта № 5 приведены векторы скорости и «линии тока» в сечениях $x=0.5$ и $y=0.5$ соответственно (т. к. эти сечения не являются плоскостями симметрии, выражение «линии тока» в данном случае является условным).

На Рис. 16–20 показаны одномерные распределения компонентов скоростей в трёх центральных сечениях для различных чисел Рейнольдса (для $Re=1000$ используются результаты расчёта № 5). На Рис. 16–17 приведены профили скоростей $u_x(y)$ в плоскости $x=0.5, z=0.5$ и $u_y(x)$ в плоскости $y=0.5, z=0.5$. Эти кривые хорошо соответствуют графикам из [6]. На Рис. 18–20 показаны профили u_x, u_y и u_z в зависимости от z в плоскости $x=0.5, y=0.5$. Поведение этих кривых объясняется различным расположением вихря относительно центральных сечений при разных числах Рейнольдса. Одномерные профили можно использовать для количественного сравнения результатов моделирования течения в трёхмерной каверне, полученных на основе различных методик.

№ расчёта	Число Рейнольдса Re	Сетка $N_x \times N_y \times N_z$	Шаг по времени ht	Точность решения уравнения Пуассона на $\xi_1^{(4)}$	Число шагов по времени до сходимости Nt	Результаты представлены на рисунках
1	100	41×41×41	0.001	10^{-4}	6088	4, 5, 16–20
2	400	41×41×41	0.001	10^{-9}	14825	6, 7, 16–20
3	800	41×41×41	0.001	10^{-9}	19222	8, 9, 16–20
4	1000	41×41×41	0.0005	10^{-9}	37992	10, 11
5	1000	81×81×81	0.0005	10^{-8}	38869	12, 13, 14–20

Таблица 1. Параметры расчётов

Оценка эффективности распараллеливания

Расчёт № 1 (см Табл. 1) был выполнен на 1...6 процессорах. В Табл. 2 приведены результаты измерения полного времени расчёта и времени итераций (т. е. решения уравнения Пуассона). Число шагов по вре-

мени до сходимости при каждом числе процессоров одинаково. Последний столбец таблицы (производительность P) графически изображён на Рис. 3. Производительность вычисляется как $P = T_1 / T_{Np}$, где T_1 и T_{Np} – времена решения уравнения Пуассона при расчёте на одном и на Np процессорах. Видно, что получен быстрый (суперлинейный) рост производительности. Это можно объяснить уменьшением размеров перекрывающихся частей областей (см. Рис. 1) с увеличением числа процессоров, а также более эффективным задействованием кэш-памяти процессоров с уменьшением объёма данных, приходящихся на один процессор. Такая высокая эффективность распараллеливания является обнадеживающим результатом и оправдывает применение предлагаемого метода решения уравнения Пуассона для более сложных задач гидродинамики.

Число процессоров Np	Полное время расчёта, мин	Время на итерации (решение уравнения Пуассона) T_{Np} , мин	Производительность $P = T_1 / T_{Np}$
1	258.934	76.2734	1
2	217.852	35.3702	2.156
3	203.3	22.5548	3.382
4	198.333	17.1892	4.437
5	192.022	11.026	6.918
6	189.524	7.998	9.536

Таблица 2. Измерения времени расчёта № 1

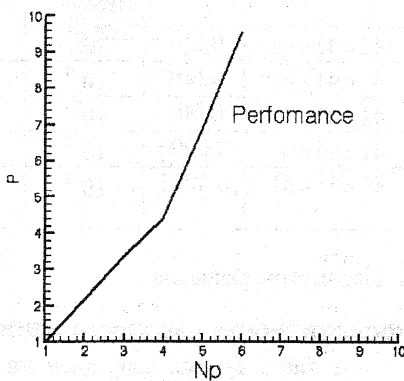


Рис. 3. Производительность в зависимости от числа процессоров

Заклучение

Реализован итерационный многопроцессорный метод решения уравнения Пуассона для задач в двумерной и трёхмерной пространственной постановках. Метод применён к моделированию пространственного гидродинамического течения. Моделирование проведено на основе квази-гидродинамических (КГД) уравнений. Результаты расчётов сопоставлены с известными из литературы данными. Получена высокая эффективность распараллеливания.

Автор выражает признательность Т.Г. Елизаровой за внимание к работе.

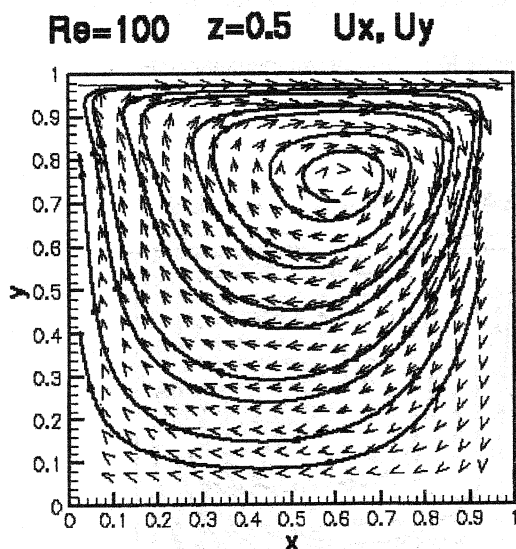


Рис. 4. Расчёт № 1. Векторы U_x, U_y

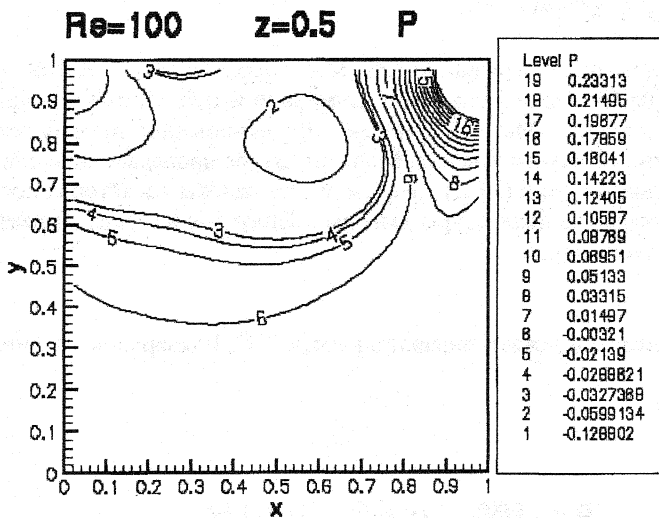


Рис. 5. Расчёт № 1. Линии уровня давления

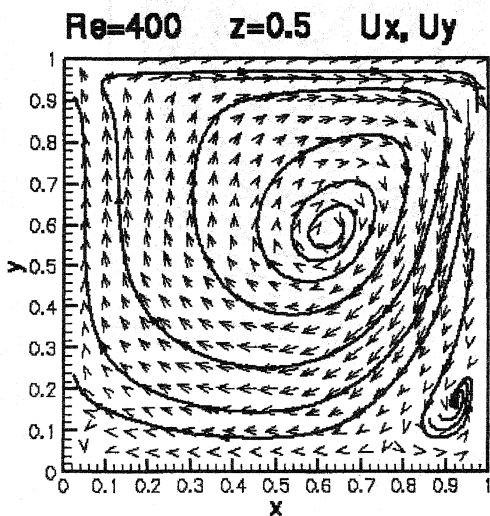


Рис. 6. Расчёт № 2. Векторы U_x, U_y

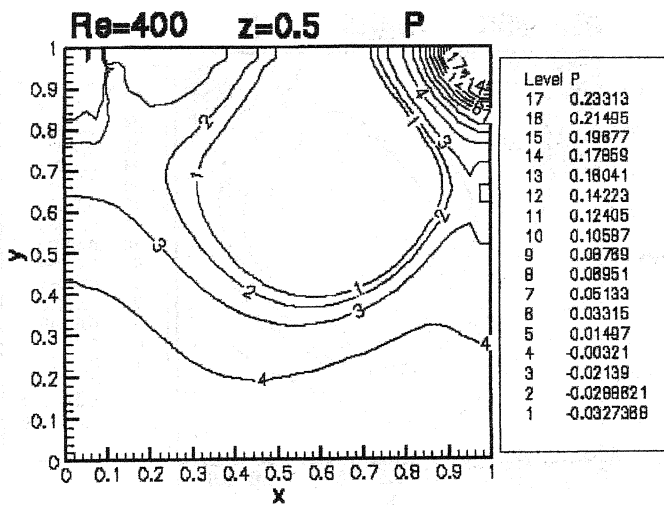


Рис. 7. Расчёт № 2. Линии уровня давления

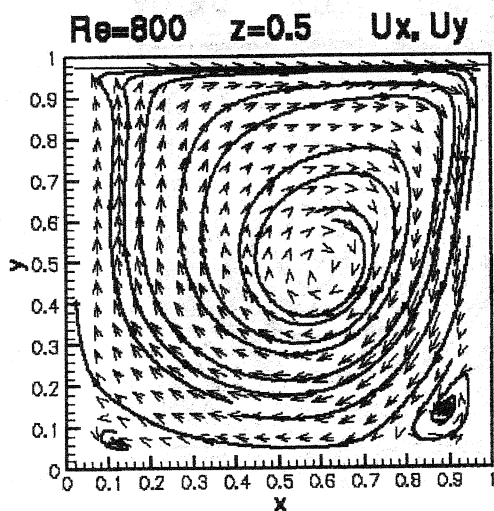


Рис. 8. Расчёт № 3. Векторы U_x, U_y

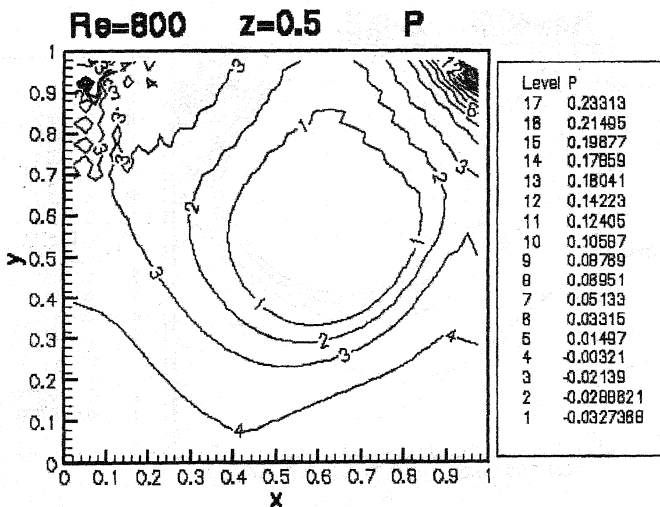


Рис. 9. Расчёт № 3. Линии уровня давления

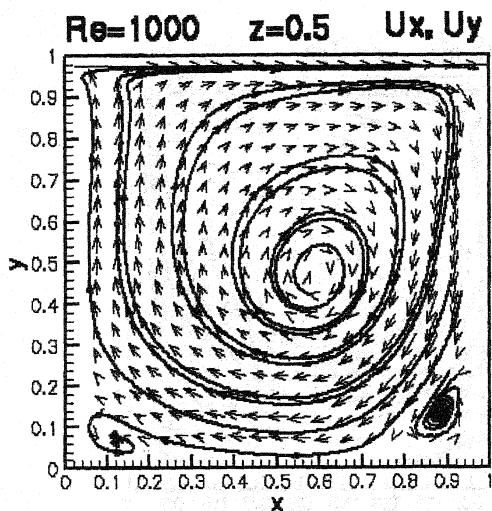


Рис. 10. Расчёт № 4. Сетка 41x41x41. Векторы Ux, Uy

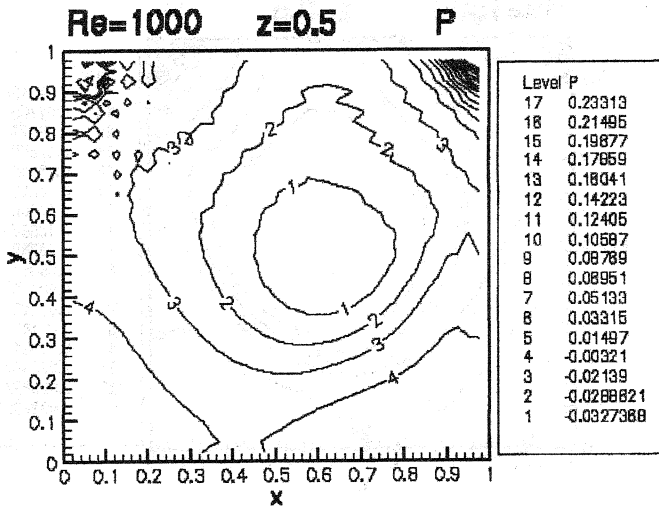


Рис. 11. Расчёт № 4. Сетка 41x41x41. Линии уровня давления

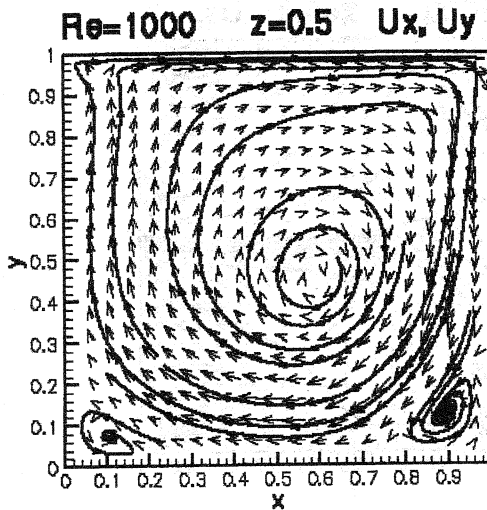


Рис. 12. Расчёт № 5. Сетка 81x81x81. Векторы Ux, Uy

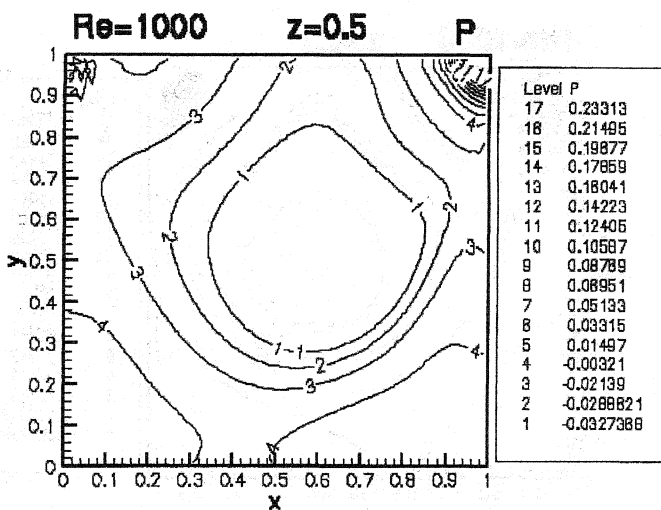


Рис. 13. Расчёт № 5. Сетка 81x81x81. Линии уровня давления

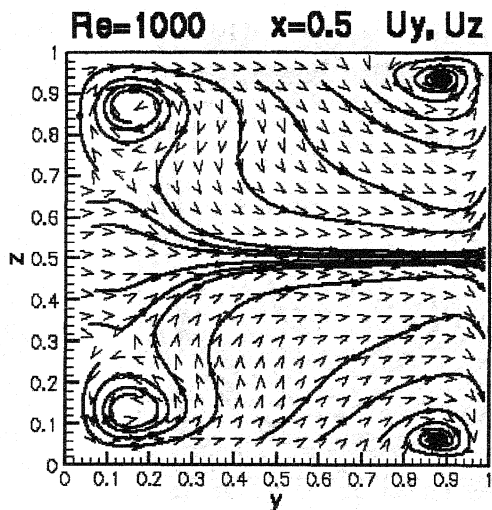


Рис. 14. Расчёт № 5. Сетка 81x81x81. Векторы Uy, Uz

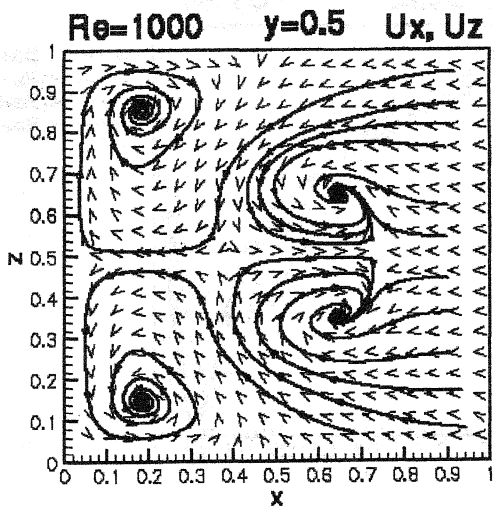


Рис. 15. Расчёт № 5. Сетка 81x81x81. Векторы U_x, U_z

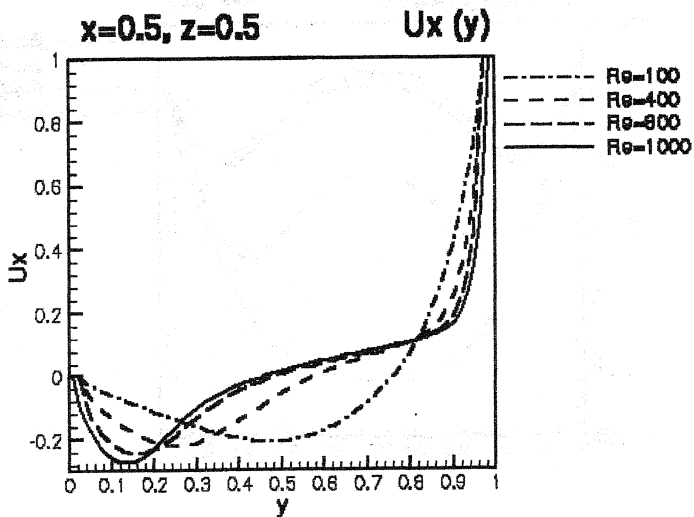


Рис. 16. Профили U_x(y)

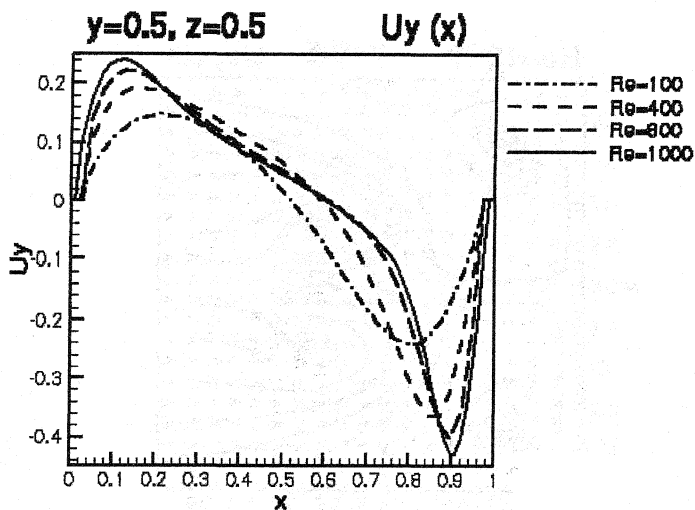


Рис. 17. Профили $U_y(x)$

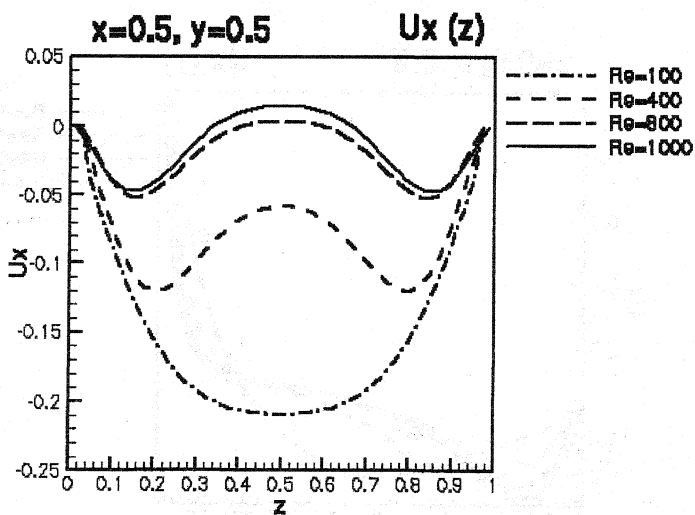


Рис. 18. Профили $U_x(z)$

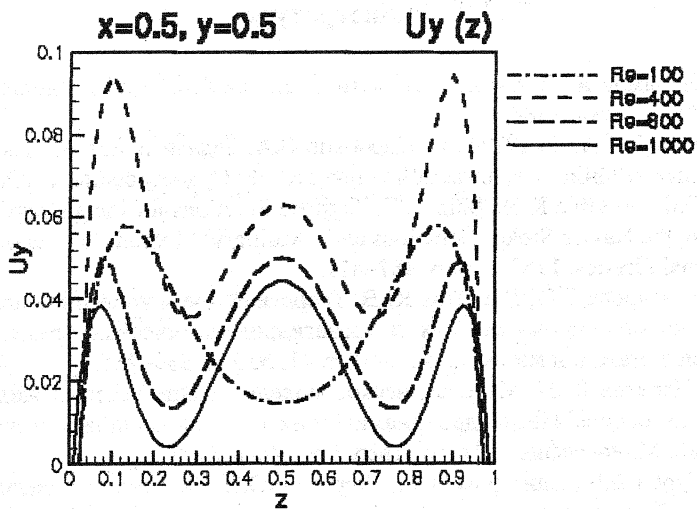


Рис. 19. Профили $U_y(z)$

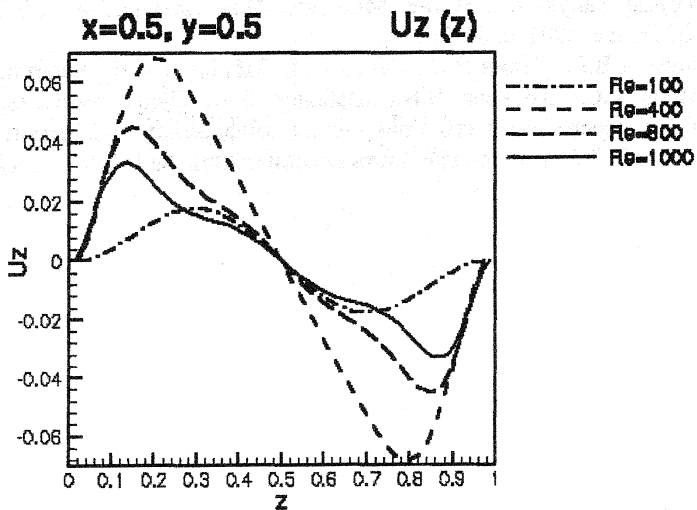


Рис. 20. Профили $U_z(z)$

Литература

- [1] Икрамов Х. Д. Численные методы для симметричных линейных систем. М.: Наука, 1988.
- [2] Белов И.А., Исаев С.А., Коробков В.А. Задачи и методы расчёта отрывных течений несжимаемой жидкости. Л.: Судостроение, 1989.
- [3] Ghia U., Ghia K.N., Shin C. T. High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. *Journal of Computational Physics*, 1982, v. 4, p. 387–411.
- [4] Елизарова Т.Г., Шеретов Ю.В. Теоретическое и численное исследование квазигазодинамических и квазигидродинамических уравнений. *Ж. вычисл. матем. и матем. физ.*, 2001, т. 41, № 2. с. 239–255.
- [5] Шеретов Ю.В. Математическое моделирование течений жидкости и газа на основе квазигидродинамических и квазигазодинамических уравнений. Монография. – Тверь: Твер. Гос. Ун-т, 2000.
- [6] Гуров Д.Б., Елизарова Т.Г., Шеретов Ю.В. Численное моделирование течений жидкости в каверне на основе квазигидродинамической системы уравнений. *Математическое моделирование*, 1996, т. 8, № 7, с. 33–44.
- [7] Широков И.А., Елизарова Т.Г. Применение многопроцессорных систем для расчёта струйных течений. *Прикладная математика и информатика: Труды факультета ВМиК МГУ им. М.В. Ломоносова: № 7.* – М.: МАКС Пресс, 2001, с. 63–72.
- [8] Фортон В.Е., Левин В.К., Савин Г.И., Забродин А.В., Каратанов В.В., Елизаров Г.С., Корнеев В.В., Шабанов Б.М. Суперкомпьютер МВС-1000М и перспективы его применения. *Информационно-аналитический журнал «Наука и промышленность России»*, 2001, № 11, (55), с. 49–52.