

УДК 519.6

ПРИМЕНЕНИЕ УНИВЕРСАЛЬНОЙ МНОГОСЕТОЧНОЙ ТЕХНОЛОГИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ДВУХФАЗНОЙ ФИЛЬТРАЦИИ НА МНОГОПРОЦЕССОРНОМ ВЫЧИСЛИТЕЛЬНОМ КОМПЛЕКСЕ¹

Е.А. Ярошенко

*Учреждение Российской академии наук**Институт проблем управления им. В.А. Трапезникова РАН*

Россия, 117806, Москва, Профсоюзная ул., 65

E-mail: ujinmuz@mail.ru

Рассматриваются особенности применения универсальной многосеточной технологии для решения задач моделирования двухфазной фильтрации воды и нефти с использованием многопроцессорных вычислительных комплексов.

Ключевые слова: многопроцессорные вычислительные системы, многосеточный метод, двухфазная фильтрация.

ROBUST MULTIGRID TECHNIQUE FOR TWO-PHASE FILTRATION PROBLEM SOLVING USING HPC / Ye.A. Yaroshenko (Institute of Control Sciences RAS, 65 Profsoyuznaya, Moscow, 117997, Russia). Application of robust multigrid technique for two-phase filtration of water and oil problem solving with help of HPC is considered in this paper.

Key words: high-performance computing, multigrid technique, two-phase filtration.

¹ Работа выполнена по гранту РФФИ № 08-00503а.

1. Введение

С развитием вычислительной техники все большее применение численные методы находят в нефтяной и газовой промышленности. При проектировании разработки нефтяных месторождений целесообразно предварительное построение модели разработки. Это позволяет опробовать несколько вариантов разработки и выбрать из них наиболее рациональный. Наиболее значимой, а иногда и единственной частью модели является гидродинамическая модель процессов фильтрации жидкостей в пластах нефтяных залежей. В процессе моделирования исходные дифференциальные уравнения в частных производных, описывающие физические процессы, заменяются соответствующими системами конечно-разностных уравнений большой размерности, решение которых требует большого объема вычислений.

Задача моделирования является подзадачей задачи оптимального управления процессом разработки нефтяного месторождения. При этом следует отметить, что в задачах оптимизации процесс моделирование стоит повторять многократно. Кроме того, учитывая отсутствие достоверных данных о свойствах пласта, перед решением задачи оптимизации, необходимо решать задачу идентификации параметров модели, что, в свою очередь, также требует многократно повторить процесс моделирования. Это ведет к необходимости использования таких алгоритмов и средств моделирования, которые позволят решать задачу моделирования в приемлемые сроки. Одним из таких средств является использование для решения поставленной задачи универсальной многосеточной технологии, которая обладает высокой степенью внутреннего параллелизма и в ряде случаев позволяет эффективно использовать в качестве аппаратного ресурса для моделирования многопроцессорные вычислительные комплексы. Универсальная многосеточная технология является развитием одного из экономичных методов решения систем дифференциальных уравнений и, кроме

того, позволяет решить ряд проблем влияющих на скорость расчета в процессе моделирования двухфазной фильтрации воды и нефти.

2. Постановка задачи

В качестве объекта для исследования эффективности применения универсальной многосеточной технологии будем рассматривать математическую модель двухфазной фильтрации воды и нефти, полученную на основе уравнений неразрывности и линейного обобщенного закона фильтрации Дарси. Модель предполагает следующие допущения: фильтрация изотермическая и подчиняется закону Дарси, гравитация, сжимаемость и взаиморастворимость фаз не учитываются. Система уравнений решается численно при заданных начальных и граничных условиях. Начальные условия заданы в соответствии с условиями: границы элемента считаются непроницаемыми, на скважинах задаются отбор, либо забойное давление.

После несложных математических преобразований математическая модель принимает вид следующей системы дифференциальных уравнений в частных производных:

$$\operatorname{div}\left(\frac{K_{\text{фн}}(s)}{\mu_{\text{н}}} + \frac{K_{\text{фв}}(s)}{\mu_{\text{в}}}\right) \mathbf{K}_a(x) \operatorname{grad} P - \operatorname{div}\left(\frac{K_{\text{фв}}(s)}{\mu_{\text{в}}}\right) \mathbf{K}_a(x) \operatorname{grad} P_k(s) = 0, \quad (1)$$

$$m \frac{\partial s}{\partial t} = \operatorname{div}\left(\frac{K_{\text{фн}}(s)}{\mu_{\text{н}}}\right) \mathbf{K}_a(x) \operatorname{grad} P - \operatorname{div}\left(\frac{K_{\text{фв}}(s)}{\mu_{\text{в}}}\right) \mathbf{K}_a(x) \operatorname{grad} P_k(s), \quad (2)$$

$$P_k(s) = P - P_b \quad (3)$$

$$s_{\text{н}} + s = 1 \quad (4)$$

Здесь $\mathbf{K}_a(x)$ – распределение проницаемости, $x \in R^3$; $K_{\text{фн}}$ и $K_{\text{фв}}$ – фазовые проницаемости нефти и воды соответственно; $\mu_{\text{н}}$ и $\mu_{\text{в}}$ – динамическая вязкость нефти и воды соответственно; $P(x,t)$ и $P_b(x,t)$ – распределение давления в нефтяной и водной фазах, $s_{\text{н}}(x,t)$ и $s(x,t)$ – распределения насыщенности нефти и воды в пласте в момент времени t .

Решение системы уравнений (1)-(2) возможно хорошо известным IMPES-методом, описанным, например, в [1]. По методу IMPES последовательно проводится расчет поля давлений и распределения насыщенностей в каждый дискретный момент времени. При этом на каждом временном этапе уравнение (1) решается относительно неизвестных давлений по неявной схеме, а уравнение (2) относительно неизвестных насыщенностей по явной схеме. Последовательное решение уравнений (1) и (2) можно организовать в итерационный процесс, что является более строгим с математической точки зрения. Организация такого итерационного процесса позволяет учитывать в уравнении (1) значения насыщенностей с того же временного слоя, на котором находятся искомые значения давлений. Для рассмотрения данного вопроса подробнее рекомендуем обратиться к [1] и [3]. На рисунке 1 приведена блок-схема расчета для одного временного слоя.

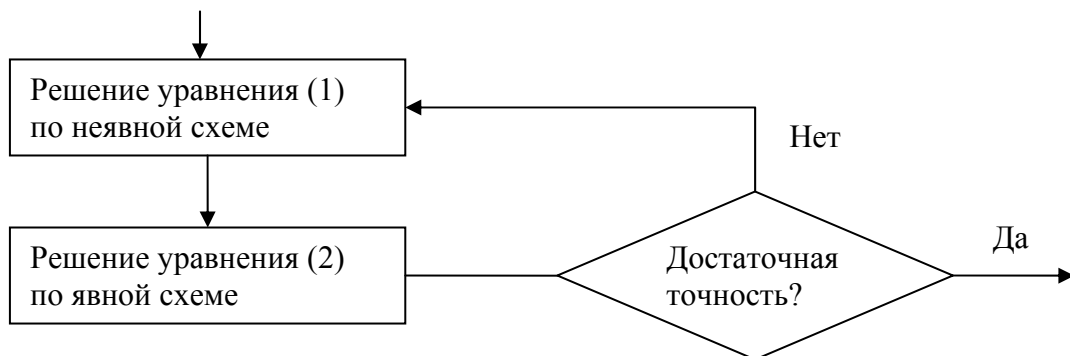


Рис. 1.

3. Применение УМТ

Численное решение уравнения (1) по неявной схеме является достаточно трудоемким процессом с точки зрения объема вычислений. Поэтому для его решения целесообразно использовать так называемые экономичные методы решения ДУ. Один из таких методов был предложен Р.П. Федоренко в 1961 г. Первая алгоритмическая реализация основополагающей идеи Р. П. Федоренко, получила название «классические многосеточные методы» (КММ). В наши дни получило распространение ее развитие, так называемая «универсальная многосеточная технология» (УМТ). Основополагающие идеи УМТ были изложены в работах С.И. Мартыненко, например в [2].

Одно из основных отличий КММ от УМТ заключается в процедуре построения грубых сеток. Для примера рассмотрим построение грубых сеток, используемых при решении одномерной задачи. Исходная (самая мелкая) сетка, на которой аппроксимируется решение системы уравнений (1)-(2), представляется как объединение множества узлов и границ конечных объемов и обозначается $\Omega(0,1)$.

$$\begin{aligned}\Omega(0,1) &= \Omega^v(0,1) \cup \Omega^f(0,1); \\ \Omega^v(0,1) &= \{x_k^v : x_k^v \in \Omega, h_k = x_{k+1}^v - x_k^v, k = \overline{0, n}\}; \\ \Omega^f(0,1) &= \{x_k^f : x_k^f = 0,5(x_k^v + x_{k+1}^v), k = \overline{1, n}\}.\end{aligned}$$

Исходную сетку $\Omega(0,1)$ представим как объединение трех более грубых и не пересекающихся сеток первого уровня (как показано на рис. 2), т.е. $\Omega(0,1) = \bigcup_{\alpha=1}^3 \Omega(1,\alpha)$, $\Omega(1,\alpha) \cap \Omega(1,\beta) = \emptyset, \alpha \neq \beta$. Далее рекуррентным способом каждая из сеток $\Omega(1,\alpha), \alpha = \overline{1,3}$, рассматривается как исходная для сеток $\Omega(2,\alpha), \alpha = 1, \dots, 3^2$, а полученные девять еще более грубых сеток образуют второй уровень т.д.

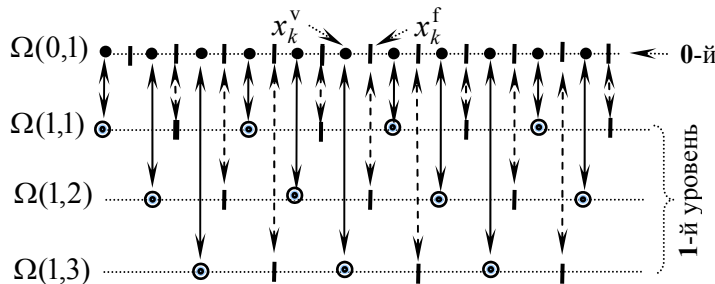


Рис. 2. Прямой (обратный) переход от мелких (грубых) сеток к грубым (мелким).

Построение грубых сеток, которое используется в УМТ, состоит в удалении двух точек из каждой системы Ω^v и Ω^f , как показано на рис. 2. Шаг сетки L -го уровня есть $\Delta 3^L$, где Δ – шаг самой мелкой сетки $\Omega(0,1)$. Совокупность всех сеток, в том числе и самой мелкой, будем называть многосеточной структурой.

Посещение уровней многосеточной структуры представляет собой пилообразный цикл. Решение ДУ начинается на самых грубых сетках, на этом этапе необходимо получить решение на множестве грубых сеток, которые смещены друг относительно друга на длину шага самой мелкой сетки. Здесь следует отметить, что сам по себе процесс решения на каждой из грубых сеток является независимым от аналогичных процессов на других сетках. Подробнее о том, как согласуются между собой решения на грубых сетках, будет изложено позже.

После того, как решение получено на каждой из самых грубых сеток (уровня L), необходимо перейти к сеткам более мелкого уровня ($L-1$) и провести на них несколько сглаживающих итераций. Следует заметить, что данный переход не вносит каких-либо погрешностей в численное решение, поскольку все грубые сетки являются подсетками самой мелкой сетки. Затем необходимо перейти к еще более мелким сеткам, и выполнить сглаживающие итерации на них. Такой процесс продолжается до тех пор, пока сглаживающие итерации не пройдут на самой мелкой сетке.

После завершения сглаживающих итераций на самой мелкой сетке, необходимо оценить, удовлетворяет ли полученное нами решение требуемой точности. Если решение не достаточно точное, необходимо опять перейти на уровень с самыми грубыми сетками, и еще раз повторить расчет на каждом из уровней многосеточной структуры.

Рассмотрим теперь вопрос о том, как согласуются между собой решения ДУ на каждой из независимых грубых сеток. Для того чтобы решения на грубых сетках были согласованы между собой и с решением на самой мелкой сетке необходимо, чтобы решения, полученные на грубых сетках, учитывали граничные условия (первого или второго рода), которые заданы на самой мелкой сетке. Однако, поскольку множества узлов грубых сеток не пересекаются, большинство грубых сеток не содержит в себе тех узлов, в которых должны выполняться граничные условия исходной задачи.

Для решения этой проблемы возможны два подхода. Первый подход изложен в работах С.И. Мартыненко, суть его заключается в добавлении к грубым сеткам так называемых фиктивных точек, значения искомой функции в которой связывается со значениями в граничных точках при помощи интерполяционных формул. Эти фиктивные точки выступают в качестве точек, в которых должны выполняться граничные условия для грубых сеток. Второй подход заключается в том, чтобы включать в каждую из грубых сеток граничные точки из самой мелкой сетки. В результате такого подхода грубые сетки имеют нерегулярный шаг по пространству.

Каждый из подходов имеет свои достоинства и недостатки. С одной стороны, использование нерегулярной сетки позволяет более строго, с математической точки зрения, учитывать граничные условия на грубых сетках. С другой стороны, наличие мелких шагов в грубой сетке ухудшает выполнимость условия Куранта, что ведет к необходимости увеличивать количество шагов моделирования по времени, а это ведет к увеличению общего объема вычислений. При программной реализации методов, изложенных в данной статье, использовались оба подхода, однако, для получения выводов о предпочтительности одного из них требуются дополнительные исследования.

Следует отметить, что в конечном итоге при программной реализации алгоритмов каждый массив данных, которые соответствует той или иной грубой сетке, содержит в себе данные о граничных точках самой мелкой сетки, а не о фиктивных граничных точках. Эта ситуация не зависит от выбранного способа учета граничных узлов в грубых сетках. Этот выбор влияет только на формулы, которые используются для связи граничных точек грубой сетки с внутренними узлами той же грубой сетки.

Рассмотрим теперь еще некоторые особенности решения дифференциального уравнения (1) при помощи УМТ. Для дискретизации уравнения предлагается использовать метод контрольного объема. Вычислительная область разделяется на ряд непересекающихся контрольных объемов таким образом, что только один объем окружает узел сетки. Дифференциальные уравнения интегрируются по каждому контрольному объему. Наиболее привлекательная особенность данной дискретизации состоит в том, что полученные численные решения будут удовлетворять интегральным законам сохранения над любой группой контрольных объемов.

Важным здесь является то, что для вычисления контрольного объема на той или иной грубой сетке можно использовать тот факт, что любой из контрольных объемов грубой сетки является объединением нескольких контрольных объемов самой мелкой сетки. Т.е., для вычисления интегралов на грубой сетке можно использовать самую мелкую сетку, что естественно повышает точность вычислений.

Следует учитывать еще одну особенность математической модели месторождения. Она заключается в том, что область моделирования может включать в себя несколько таких объектов как скважины (с тем или иным режимом эксплуатации) и границы пластов (проницаемые или непроницаемые). Таким образом, в одномерном случае исходная мелкая сетка $\Omega(0,1)$ может содержать достаточно большое количество точек, в которых должны выполняться различные граничные условия. При программной реализации УМТ для

решения уравнения (1) в рамках модели нефтяного месторождения необходимо реализовать процедуру, которая будет однозначно составлять набор точек, которые необходимо включить в ту или иную грубую стеку в зависимости от сеточного уровня и от номера сетки в данном уровне. В этом случае, помимо множества внутренних точек, которые не пересекаются с точками других грубых сеток данного уровня, к данной грубой сетке должны быть добавлены все точки, в которых расположены скважины или границы нефтеносных пластов.

Основные этапы алгоритма решения уравнения (1) по неявной схеме в рамках одного временного этапа при помощи УМТ представлены на рисунке 3.

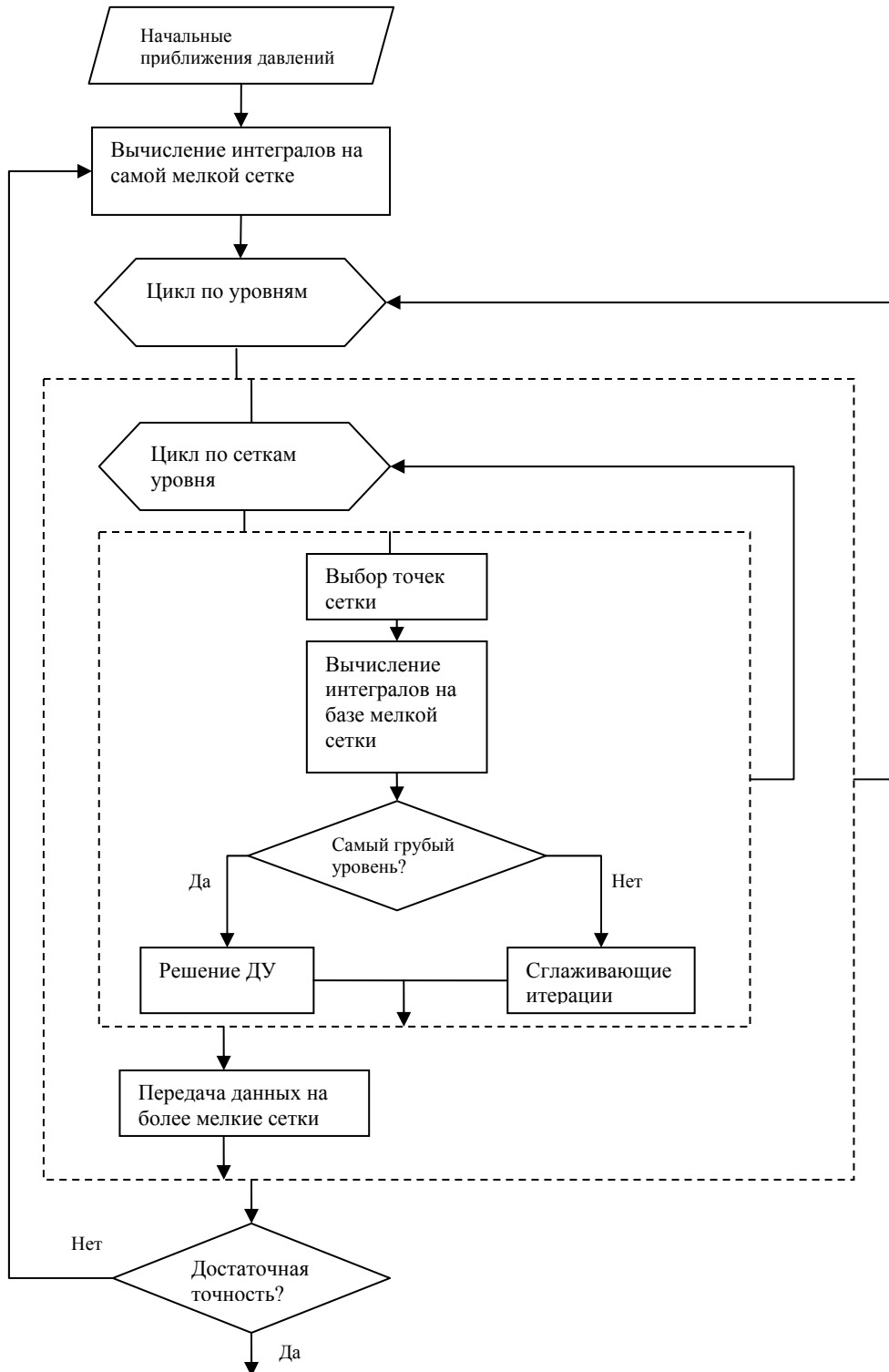


Рис. 3.

Для иллюстрации эффективности применения УМТ для решения уравнения (1) в модельной задаче приведем результаты численного эксперимента. В таблице 1 представлены результаты замеров времени счета уравнения (1) на двух разных временных этапах. Расчет проводился на одном ядре процессора без распараллеливания. В первой колонке указано количество многосеточных уровней, которые использовались для расчета, во второй – время счета уравнения (1) на первом временном шаге, в третьей – время счета уравнения (1) на одном из последующих временных шагов.

Количество уровней	Время счета 1	Время счета 2
1	59.3096	3.1212
2	13.2462	2.2254
3	3.20679	2.0783
4	1.627573	1.575
5	1.672344	1.5532

Табл. 1.

Из представленных данных видно, что для модели, используемой в эксперименте, на первом временном шаге за счет применения УМТ можно сократить время счета больше, чем в 36 раз. Однако, для той же самой модели, на других временных шагах эффективность применения УМТ значительно ниже. Здесь следует отметить, что на первом временном шаге в качестве начального приближения значений давления в узлах сетки используются значения, которые могут значительно отличаться от точного решения, а на последующих временных шагах в качестве начальных значений используются значения давлений, полученные на предыдущем временном шаге, которые достаточно близки к решению.

Рассмотрим несколько случаев, которые возможны с точки зрения эффективности применения УМТ:

1. Начальное приближение к решению уравнения (1) является достаточно “плохим”. В этом случае для решения уравнения только на самой мелкой сетке может потребоваться провести сотни или тысячи итераций. Здесь применение УМТ позволяет получить существенный эффект, поскольку предварительное решение на грубых сетках позволяет достаточно быстро приблизить решение к точному, а затем при помощи сглаживающих итераций на мелких сетках получить удовлетворяющее нас решение.
2. Начальное приближение к решению является таким, что для решения уравнения только на мелкой сетке потребовалось бы провести несколько десятков или сотен итераций. Здесь УМТ позволяет получить небольшое увеличение производительности.
3. Начальное приближение к решению является настолько “хорошим”, что для решения уравнения на самой мелкой сетки потребовалось бы всего несколько итераций. В данном случае применение УМТ может дать отрицательный эффект. Это связано с тем, что в УМТ необходимо провести количество итераций, которое как минимум совпадает с количеством сеток в многосеточной структуре, а на практике приходится провести несколько проходов по всем уровням многосеточной структуры.

Как было отмечено, мы имеем “плохое” начальное приближение на первом временном шаге моделирования. Кроме того, такое возможно на тех временных шагах, на которых изменяются режимы эксплуатации скважин. В остальных случаях перед началом расчета поля давлений на временном шаге имеется достаточно хорошее начальное приближение к решению. Таким образом, применение УМТ на различных этапах расчета модели может быть оправданно или не оправданно.

4. Распараллеливание алгоритмов решения задачи двухфазной фильтрации

Как было показано в предыдущем разделе, эффективность использования УМТ зависит как от модели конкретного нефтяного месторождения (т.е. от данных), так и от временного этапа, который подвергается моделированию. Если в последовательных алгоритмах при оценке временных затрат на расчеты речь идет в основном непосредственно о затратах времени на вычисления, то в случае параллельных алгоритмов возникают также временные затраты на организацию параллельных вычислений. К затратам на организацию параллельных вычислений можно отнести следующие: время на пересылку данных, время на выделение из общего массива данных те данные, которые используются для расчета в той или иной ветви алгоритма, синхронизация расчета между параллельными ветвями алгоритма. На этом фоне необходимо оценить эффективность применения к модельной задаче распараллеленного варианта УМТ в сравнении с другими методами распараллеливания. В качестве метода для сравнения будем использовать широко известный метод разбиения области решения на части, который описан, например, в [4].

Для сравнения параллельных алгоритмов необходимо выяснить, как часто происходит обмен данными между различными ветвями алгоритма, сколько времени занимает этот обмен, как много времени тратится на подготовку данных для обмена информацией и сравнить, как эти временные затраты соотносятся с затратами времени непосредственно на вычисления. Для этой цели приведем блок-схемы алгоритмов для численного решения уравнения (1) в рамках одного временного шага, на которых будут обозначены моменты, связанные с обменом информацией между ветвями параллельных алгоритмов.

На рисунке 4 представлена блок-схема для алгоритма, использующего разбиение области.



Рис. 4.

Обмен данными между вычислителями является достаточно затратным процессом. Как видно на рисунке 4, недостатком алгоритма использующего разбиение области является необходимость частого (после каждой итерации) обмена информацией.

Рассмотрим теперь особенности распараллеливания УМТ. Удобство распараллеливания УМТ обеспечивается следующими факторами:

1. Грубые сетки каждого уровня не имеют общих точек. Следовательно, вычислительные процессы на них могут проводиться параллельно.
2. Отсутствие общих сеточных точек уменьшает необходимость обмена данными между ветвями параллельного алгоритма.
3. Примерно одинаковое количество сеточных точек в сетках одного уровня позволяет равномерно загрузить вычислительные ресурсы.

Как следует из изложенных факторов, все грубые сетки, начиная с некоторого уровня, можно разделить между необходимым количеством вычислителей, и осуществлять расчет, в котором не будет необходимости в обмене данными с другими вычислителями. Например, если у нас имеется 9 вычислителей, то сетку $\Omega(2,1)$ и все сетки, узлы которых являются ее подмножествами, можно расположить в памяти первого вычислителя, сетку $\Omega(2,2)$ и ее подсетки - в памяти второго вычислителя и т.д. Однако, как видно из блок-схемы на рисунке 3, при вычислении интегралов каждой из грубых сеток необходимо обращаться к данным самой мелкой сетки $\Omega(0,1)$, доступ к которой необходим всем вычислителям, что приводит к необходимости обмена данными. Для того чтобы избежать этого, для каждого вычислителя будем рассматривать как самую мелкую сетку того уровня, которая является самой мелкой среди тех, которые целиком располагаются в памяти только данного вычислителя. Данную сетку будем называть динамической мелкой сеткой данного вычислителя.

Пилообразный цикл УМТ на каждом вычислителе проходит между уровнем с самыми грубыми сетками до уровня с динамической грубой сеткой. При этом интегралы рассчитываются на основе данных динамической мелкой сетки. Этот процесс продолжается без обмена данными с другими вычислителями до тех пор, пока на каждом вычислителе процесс решения не сойдется. После этого можно произвести обмен данными между вычислителями и перейти к самой мелкой сетке для сглаживания. Примерная блок-схема данного алгоритма представлена на рисунке 5.

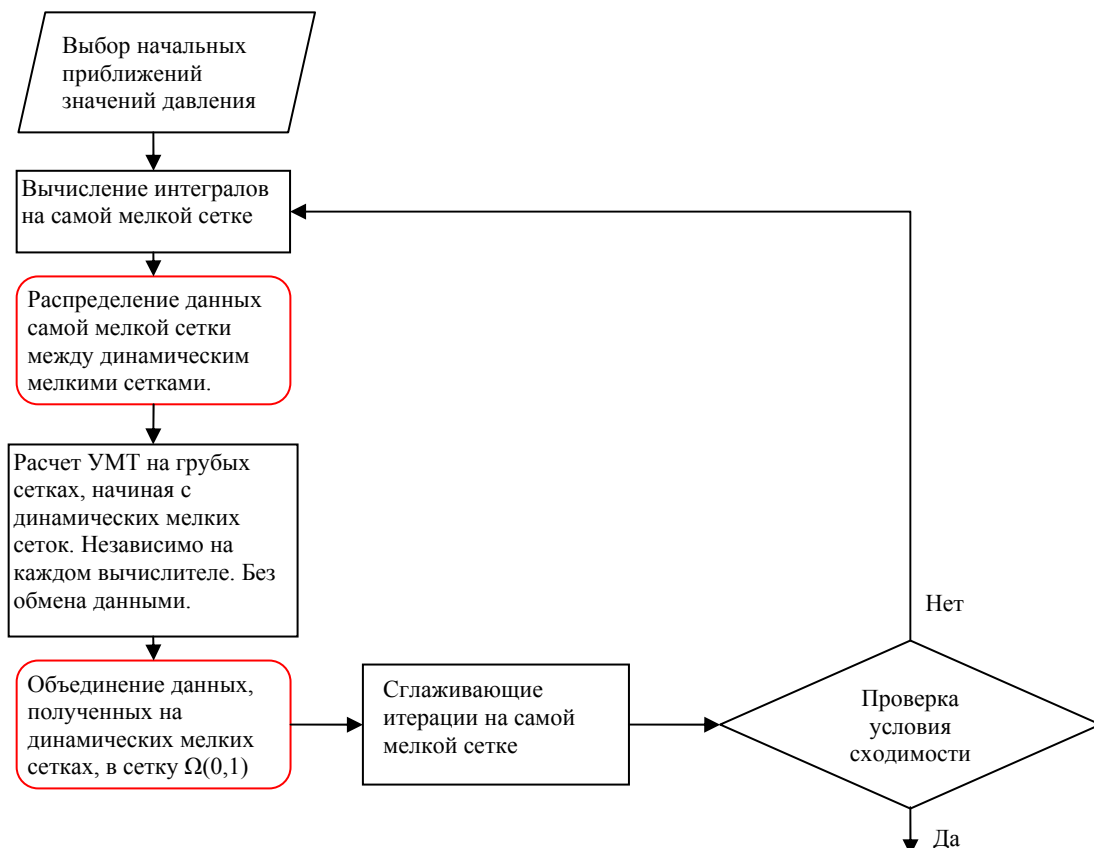


Рис. 5.

Как видно на рисунке 5, при распараллеливании УМТ с динамическими мелкими сетками обмен данными происходит только при переходе от уровня с самой мелкой сеткой к уровню с динамическими мелкими сетками. Расчет УМТ на грубых сетках включает в себя решение уравнения на самой грубых сетках и сглаживающие итерации на более мелких сетках. Он является процессом, который включает в себя множество итераций не связанных с обменом данными между вычислителями.

Сглаживающие итерации на самой мелкой сетке можно реализовывать на одном вычислителе, однако, предпочтительнее использовать для этого распараллеливание аналогичное тому, что используется при разбиении области на части. Таким образом, самую мелкую сетку целесообразно хранить не в памяти одного вычислителя, а распределять ее между всеми имеющимися вычислителями. Отсюда следует, что обмены данными в данном алгоритме представляют собой обмены типа “каждый-каждому”. Учитывая возможность наличия большого количества скважин и граничных точек пласта в области моделирования, сам процесс подготовки данных для пересылки является достаточно затратным.

Подведем небольшой итог. При реализации распараллеливания при помощи разбиения области пересылка данных происходит очень часто, но представляет собой пересылку небольших сообщений типа “точка-точка”. При распараллеливании УМТ при помощи динамических мелких сеток пересылка данных происходит редко, но объем пересылаемых данных очень велик, и подготовка их занимает много времени.

Для оценки эффективности распараллеливания УМТ предложенный алгоритм был реализован на многопроцессорном вычислительном комплексе кластерного типа. В таблице 2 выборочно предложены результаты замеров времени счета уравнения (1) в рамках одного временного шага. Замеры проводились для сеток с различным количеством узлов на 3, 9 и 27 ядрах. Ядра вычислительного комплекса выбирались таким образом, чтобы ни какие два ядра не имели общей кэш-памяти второго уровня.

Узлы ядра	100	500	1000	10000	16000	25000
3	0,0546992	0,0543534	0,0557733	0,0784771	0,0690047	0,106561
9	0,15766	0,146378	0,00743507	0,00480217	0,0343705	0,0473372
27	0,47703	0,434253	0,0154813	0,00676618	0,0240099	0,0292537

Табл. 2.

По результатам замеров видна четкая тенденция увеличения целесообразности распараллеливания с ростом размера задачи. Это говорит о том, что УМТ следует распараллеливать для больших задач, однако это не говорит ничего о преимуществах параллельной реализации УМТ или параллельного алгоритма с разбиением области.

Как было отмечено в предыдущем разделе, УМТ целесообразно применять в тех случаях, когда для нахождения решения обычными итерационными методами потребовалось бы сделать большое количество итераций. Для параллельного варианта УМТ хотелось бы отметить, как влияет время, затрачиваемое на пересылку данных между процессорами, на эффективность применения данного метода. Для ответа на этот вопрос необходимо провести сравнение времени, которое затрачивается на пересылку данных в УМТ, со временем проведения одной итерации вместе с пересылкой данных при использовании разбиения области. В таблице 3 представлены результаты численного эксперимента, который проводился для оценки отношения времени пересылки данных в УМТ ко времени проведения одной итерации методом разбиения области на части. Измерения проводились для сеток с количеством узлов от 100 до 50 000. Для задач разного размера это соотношение принимало разные значения. В таблице приведены диапазоны, соответствующие тому или иному количеству параллельных ветвей алгоритма.

Количество ядер	отношение пересылки времени итерации	времени данных ко
3	3 – 5 раз	
9	10 – 19 раз	
27	20 – 60 раз	

Табл. 3.

Из таблицы 3 видно, что с ростом числа вычислителей предпочтительность УМТ в сравнении с методом разбиения на подобласти падает. Очевидно, что нет необходимости использовать УМТ, если задачу можно решить методом разбиения на подобласти за время меньшее, чем займет пересылка данных в параллельном варианте УМТ. При этом следует учитывать, что сама по себе УМТ предпочтительная для случаев, когда решение требует большого количества итераций.

5. Применение УМТ в многомерных случаях.

До сих пор мы рассматривали применение УМТ для одномерных моделей двухфазной фильтрации. Существуют методики применения УМТ в многомерных случаях, однако, в связи с особенностями моделей нефтяного месторождения их прямое применение является затруднительным. К таким особенностям относится наличие в области моделирования большого количества точечных границ, которые представляют собой нагнетательные и добывающие скважины. Для того чтобы обойти эти сложности, в двумерном случае, целесообразно использовать многосеточный метод совместно с методом переменных направлений. Во-первых, это позволяет свести двумерную задачу к параллельному решению в два этапа множества одномерных задач, во-вторых, метод переменных направлений сам по себе является экономичным методом, т.е. позволяет найти решение на каждом временном слое за известное число операций и, при этом, безусловно устойчив.

Для решения начально-краевой задачи с расщеплением по координатам используется метод переменных направлений Писмена-Рекфорда с разбиением шага по времени на два полушага. Решение системы уравнений (1) – (2) подразумевает решение задачи вида

$$\frac{\partial u}{\partial t} = Lu + f(x_1, x_2, t),$$

$$Lu = \frac{\partial(k(x,t)u)}{\partial x_1} + \frac{\partial(k(x,t)u)}{\partial x_2} + \frac{\partial}{\partial x_1} \left(k(x,t) \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k(x,t) \frac{\partial u}{\partial x_2} \right).$$

Здесь u выступает в качестве функции распределения давления P , в случае уравнения (1), и в качестве функции распределения насыщенности s , в случае уравнения (2). Оператор L можно представить в виде: $Lu = L_1u + L_2u$, где L_α – оператор:

$$L_\alpha u = \frac{\partial(k(x,t)u)}{\partial x_\alpha} + \frac{\partial}{\partial x_\alpha} \left(k(x,t) \frac{\partial u}{\partial x_\alpha} \right), \quad \alpha = 1, 2. \quad (5)$$

Для решения задачи каждый временной шаг делится на два. На первом дробном временном слое первый дифференциальный оператор (5) аппроксимируется неявно (производится решение по направлению координаты x_1), а второй – явно. На втором дробном слое неявно аппроксимируется второй дифференциальный оператор (производится решение по направлению координаты x_2), а первый – явно.

Такой двухступенчатый подход предполагает на каждом полушаге решение множества одномерных задач, которые можно решать при помощи УМТ.

Если на каждом временном слое рассматривать иерархию, в соответствии с которой происходит разделение данных между вычислителями, а, следовательно, организуются ветви параллельного алгоритма, то можно отметить следующее:

- на вершине иерархии находятся пространственная сетка Ω в целом;
- на втором уровне в группы выделяются узлы, находящиеся на одной линии параллельной одной из осей, необходимые для решения соответствующей одномерной подзадачи;
- на третьем и последующих уровнях сетки одномерных задач могут делиться на множество грубых подсеток.

Таким образом, на каждом из временных полушагов решается от N до $N \cdot Q$ параллельных задач, где Q – количество уровней грубости сеток в многосеточном методе. Здесь следует отметить, что выбор числа Q целесообразно ограничить сверху, поскольку при разбиении задачи на слишком большое количество ветвей приводит к неэффективному использованию вычислительных ресурсов.

Для эффективного распараллеливания вычислительных алгоритмов в трехмерных моделях можно обобщить подходы, применяемые к двумерным моделям. Один из базовых методов – это использовать трехмерную многосеточную схему, однако этот метод является достаточно трудоемким, и в случае трех измерений требует больших затрат. Эти затраты связаны с организацией обмена данными между ветвями алгоритма при переходе от грубых сеток к более мелким сеткам.

Предпочтительным является использование многосеточного метода совместно с локально-одномерным методом. Это позволяет свести трехмерную задачу к последовательному решению множества одномерных задач, для которых многосеточные методы зарекомендовали себя с лучшей стороны. При таком подходе каждый шаг по времени разбивается на три шага. Схемы, записанные для каждого из промежуточных слоев, имеют только суммарную аппроксимацию. На промежуточных слоях они вообще не аппроксимируют дифференциальные уравнения модели. Но погрешности аппроксимации гасят друг друга так, что на целом слое аппроксимация есть. При этом разностное решение следует сравнивать с точным только на целых слоях, не придавая промежуточным слоям самостоятельного смысла.

Как показывает практика, недостатком метода переменных направлений и локально одномерного метода является необходимость пересылки большого объема данных при переходе к каждому новому дробному шагу. Целесообразность их применения имеет общие корни с целесообразностью применения УМТ: если решение задачи методом разбиения на подобласти потребует большого объема вычислений, то применение экономичных методов целесообразно, в противном случае – нет.

6. Заключение

При распараллеливании экономичных методов решения ДУ, таких как УМТ, метод переменных направлений и локально одномерный метод, для решения задач двухфазной фильтрации взаимодействие между параллельными ветвями алгоритма имеет три характерные особенности: большой объем пересылаемых данных, сложность организации взаимодействия и относительную редкость передачи данных. Это приводит к тому, что использование этих методов целесообразно только в тех случаях, когда решение классическими итерационными методами требует реализации большого количества итераций. В таких условиях для моделирования процессов разработки нефтяных месторождений целесообразно реализовать адаптивный алгоритм, который будет на разных временных этапах, в зависимости от необходимости, использовать предложенные экономичные или классические итерационные методы.

СПИСОК ЛИТЕРАТУРЫ

1. *Генри Б. Кричлоу*. Современная разработка нефтяных месторождений – проблемы моделирования. Москва: Недра, 1979.
2. *Мартыненко С.И.* Универсальная многосеточная технология для численного решения дифференциальных уравнений в частных производных на структурированных сетках. / Вычислительные методы и программирование, 2000. 1, №1 с. 83-102.
3. *Федоренко Р.П.* Введение в вычислительную физику. Долгопрудный: Интеллект, 2008.
4. *Анисимова Н.А., Гребенник О.С., Ярошенко Е.А.* Моделирование процессов двухфазной фильтрации воды и нефти с использованием многопроцессорных вычислительных комплексов. / Труды Института проблем управления им. В.А. Трапезникова РАН. Том XXVIII. М.: 2008. С. 84-91.
5. *Дж. Деммель*. Вычислительная линейная алгебра. Теория и приложения. М: Мир, 2001.

