

ГИБРИДНЫЕ АЛГОРИТМЫ РЕШЕНИЯ ЛИНЕЙНЫХ СИСТЕМ ДЛЯ КОНЕЧНО-ЭЛЕМЕНТНОГО МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ФИЛЬТРАЦИИ

Аннотация. Представлен новый параллельный алгоритм решения систем линейных алгебраических уравнений с ленточной симметричной положительно-определенной матрицей, совмещающий вычисления на многоядерных процессорах и графических ускорителях. Приведены результаты моделирования пространственного режима фильтрации Черниговского месторождения подземных вод с помощью конечно-элементного решателя Надра-3D.

Ключевые слова: системы линейных алгебраических уравнений, ленточная матрица, гибридный параллельный алгоритм, графические процессоры, метод конечных элементов, фильтрация воды, Черниговское месторождение подземных вод.

ВВЕДЕНИЕ

Численное моделирование различных физических процессов, основанное на методе конечных элементов, в настоящее время применяется при проектировании и разработке в чрезвычайно широком диапазоне — начиная от проектирования корпуса мобильного телефона и заканчивая разработкой сейсмостойких объектов или ядерных реакторов. При этом большую часть времени моделирования занимает нахождение решения систем линейных алгебраических уравнений (СЛАУ), возникающих при дискретизации исходных систем дифференциальных уравнений. Следует заметить, что обычно моделирование предполагает многократное решение СЛАУ, поскольку интерес представляет динамика исследуемого процесса во времени, причем количество итераций может составлять сотни и тысячи. В случае нелинейных задач также предполагается многократное построение и решение СЛАУ в пределах одной итерации по времени. Таким образом, следует отметить чрезвычайно влияние скорости работы программы решения СЛАУ на общую скорость работы конечно-элементного решателя.

Еще одним важным фактором, определяющим применимость решателя СЛАУ для использования в современных задачах моделирования методом конечных элементов, является применяемая им схема размещения СЛАУ в оперативной памяти. Задачи проектирования сложных конструкций, задачи гидрогеологии часто приводят к построению СЛАУ с миллионами (или десятками миллионов) неизвестных. Например, для системы размерности $n = 1217610$, построенной для моделирования режима фильтрации Черниговского месторождения подземных вод, полуширина ленты матрицы после применения обратного алгоритма перенумерации Катхилла–Макки составляла $m = 3628$. Для ее хранения необходим объем оперативной памяти в 33 Gb (с учетом симметричности матрицы хранится только ее половина — 4 417 489 080 элементов).

При решении СЛАУ большой размерности ($10^5 - 10^8$) с полушириной ленты матрицы порядка $10^2 - 10^4$ количество арифметических операций с плавающей запятой (при использовании метода Холецкого) составляет порядка $10^9 - 10^{16}$. Общее количество хранимых элементов матрицы системы составляет $10^8 - 10^{12}$. Такие требования к вычислениям намного опережают возможности традиционных параллельных компьютеров, даже несмотря на многоядерность процессоров.

Решение проблемы ускорения вычислений на многоядерных компьютерах может достигаться использованием графических процессоров (GPU) для выпол-

нения больших объемов однородных арифметических операций. Ускорение вычислений в ближайшей перспективе, на наш взгляд, — это гибридные системы, которые совмещают многоядерные процессоры (CPU) с графическими ускорителями (GPU) (рейтинг Top500).

В настоящей статье представлен новый параллельный гибридный алгоритм решения СЛАУ с ленточной симметричной положительно-определенной матрицей, совмещающий вычисления на многоядерных процессорах и графических ускорителях. Приведены результаты моделирования пространственного режима фильтрации Черниговского месторождения подземных вод с помощью конечно-элементного решателя Надра-3D [10].

АЛГОРИТМ РЕШЕНИЯ СЛАУ С СИММЕТРИЧНОЙ ПОЛОЖИТЕЛЬНО-ОПРЕДЕЛЕННОЙ ЛЕНТОЧНОЙ МАТРИЦЕЙ

Выбор схемы распределения данных по процессорам. Проблему эффективного распараллеливания алгоритмов, в основе которых лежит факторизация исходной матрицы, удалось решить с использованием циклических схем хранения элементов матриц и векторов. Эти схемы были предложены независимо и практически одновременно несколькими группами исследователей [1, 2]. При использовании строчно-циклической схемы элементы матрицы располагаются циклически в p процессорах, участвующих в вычислениях: если в $(i-1)$ -м процессоре располагаются элементы r -й строки матрицы, то в следующем i -м процессоре располагаются элементы $(r+1)$ -й строки.

В настоящей статье, как и в работах [3, 4], описан алгоритм с циклическим распределением и обработкой данных при решении СЛАУ с ленточной симметричной матрицей, когда в каждом процессоре хранятся не отдельные строки, а полосы из s подряд идущих строк. Соответственно операции с отдельными элементами матрицы заменяются операциями с блоками размера $s \times s$.

Пусть гибридный параллельный алгоритм решения задачи $AX = B$ реализуется на p графических процессорах и таком же количестве CPU, виртуальные связи между которыми обеспечивают возможность передачи данных из одного вычислительного устройства в любое другое. Вычислительные устройства имеют логические номера $0, 1, \dots, p-1$. При этом CPU с логическим номером i отвечает за работу с GPU, который имеет такой же логический номер.

Пусть ленточная симметричная матрица A имеет порядок $n = qs$, а полуширина ленты $m = ls$, где s — порядок блока, который должен выбираться таким образом, чтобы блок полностью помещался в кэш-памяти как CPU, так и GPU. Поскольку матрица A симметричная, то для решения СЛАУ достаточно рассматривать только диагональные элементы матрицы и те элементы, которые находятся под ними. Таким образом, матрица A разбивается на квадратные блоки размера $s \times s$:

$$\begin{pmatrix} A_{11} & 0 & 0 & & 0 \\ A_{21} & A_{22} & 0 & & 0 \\ A_{31} & A_{32} & \ddots & & \\ \vdots & \vdots & & & \\ A_{(l+1)1} & A_{(l+1)2} & & & \\ 0 & A_{(l+2)2} & & & \\ 0 & 0 & & & \\ \vdots & \vdots & \dots & A_{(q-1)(q-1)} & 0 \\ 0 & 0 & \dots & A_{q(q-1)} & A_{qq} \end{pmatrix}. \quad (1)$$

Аналогичное распределение имеет место и для матрицы L ($A = LL^T$). При этом в ходе выполнения алгоритма факторизации на место элементов матрицы A будут помещаться элементы матрицы L . Элементы исходной матрицы распределяются по CPU

согласно следующей схеме хранения: блок A_{ij} хранится в CPU с логическим номером $(i-1) \bmod p$. Аналогичное распределение данных применяется и к плотной матрице B .

Схема решения СЛАУ. Пусть необходимо найти решение СЛАУ порядка n с q правыми частями и с ленточной (ширина ленты равна $2m+1$) симметричной положительно-определенной матрицей:

$$AX = B. \quad (2)$$

Наиболее эффективным прямым методом решения таких задач является, как известно, метод Холецкого [4], согласно которому решение системы состоит из следующих трех подзадач:

- факторизация исходной матрицы системы:

$$A = LL^T; \quad (3)$$

- решение СЛАУ с нижней треугольной матрицей:

$$LY = B; \quad (4)$$

- решение СЛАУ с верхней треугольной матрицей:

$$L^T X = Y. \quad (5)$$

Алгоритм факторизации матрицы. Алгоритм факторизации состоит из q шагов, на i -м шаге вычисляются блоки матрицы L в i -м столбце. Будем называть CPU и GPU ведущими на данном шаге, если блоки A_{ii} и L_{ii} находятся в их памяти.

На i -м шаге алгоритма факторизации требуются только те блоки, которые находятся в i -м столбце и в следующих за ним l -х столбцах [5]. Из разбиения матрицы (1) видно, что в каждом столбце находится не больше, чем $l+1$ блок. Таким образом, на каждом шаге алгоритма достаточно хранить в памяти GPU не более $(l+1)^2$ блоков. Для обозначения блоков матриц A и L , находящихся в памяти GPU, будем использовать верхний индекс d .

Перед началом работы алгоритма проводится копирование блоков матрицы A , которые находятся в первых $l+1$ столбцах, в память GPU. Блок A_{ij} копируется в GPU с номером $(i-1) \bmod p$.

Для каждого i -го столбца, $i = \overline{1, q}$, необходимо выполнить следующие шаги:

1. LL^T -факторизация блока A_{ii} на ведущем CPU, используя последовательный алгоритм:

$$A_{ii} = L_{ii}L_{ii}^T. \quad (6)$$

2. Пересылка блока L_{ii} из ведущего GPU в память всех остальных GPU.

3. Вычисление на соответствующих GPU блоков L_{ji}^d , где $j = \overline{i+1, \min(i+l, q)}$, по формуле

$$L_{ji}^d = A_{ji}^d (L_{ii}^d)^{-1}. \quad (7)$$

4. Асинхронное обновление блоков A_{jt}^d , используя соответствующие GPU, где $j = \overline{i+1, \min(i+l, q)}$, $t = \overline{j, \min(i+l, q)}$ и $t-j \equiv 0 \pmod{p}$, по формуле

$$A_{jt}^d \leftarrow A_{jt}^d - L_{ji}^d (L_{ii}^d)^T. \quad (8)$$

5. Каждый CPU выполняет рассылку вычисленных на третьем шаге блоков всем остальным вычислительным устройствам.

6. Обновление по формуле (8) блоков A_{jt}^d , где $j = \overline{i+1, \min(i+l, q)}$, $t = \overline{j, \min(i+l, q)}$ и $t-j$ не делится на p , на соответствующих GPU.

7. Если $i+l+2 < q$, то выполняем асинхронное копирование всех блоков матрицы, которые находятся в столбце $i+l+2$, в память соответствующих GPU.

8. CPU с логическим номером $i \bmod p$ выполняет синхронизацию вычисления блока A_{i+l+1}^d .

9. Копирование блока A_{i+1i+1}^d в память соответствующего CPU.

Заметим, что условие $t - j \equiv 0 \pmod{p}$ на шаге 4 алгоритма означает, что все блоки, необходимые для вычисления по формуле (8), находятся в одном и том же GPU. А на шаге (6) рассматриваются только те пары индексов, разность которых не делится на p ; значит, необходимые блоки были получены при рассылке на шаге (5).

Решение СЛАУ с нижней треугольной матрицей. Решение задач (4) и (5), как и факторизация, выполняется с использованием p процессоров и p графических ускорителей.

Вычислительная схема решения задачи (4) во многом аналогична вычислительной схеме алгоритма факторизации. При этом элементы матрицы Y будут записываться в соответствующие элементы матрицы B . Для каждого $i = 1, q$ выполняются следующие шаги.

1. Ведущий процессор выполняет рассылку диагонального блока L_{ii}^d всем GPU.

2. Выполняется решение СЛАУ с нижней треугольной матрицей блоков L_{ii}^d и правой частью матрицы B_{ik}^d , где $k = \overline{1, q}$, на всех GPU:

$$L_{ii}^d Y_{ik}^d = B_{ik}^d. \quad (9)$$

3. Каждый CPU выполняет рассылку принадлежащих ему блоков L_{ij}^d , где $j = \overline{i+1, \min(i+l, p)}$, всем остальным GPU.

4. Каждый CPU выполняет рассылку принадлежащих ему блоков Y_{ij}^d всем остальным GPU.

5. Обновление матриц B_{jk}^d , где $j = \overline{i+1, \min(i+l, p)}$, $k = \overline{1, q}$, на соответствующих GPU по формуле

$$B_{jk}^d \leftarrow B_{jk}^d - L_{ji}^d Y_{ik}^d.$$

Решение СЛАУ с верхней треугольной матрицей. Рассмотрим вычислительную схему задачи (5). Элементы матрицы X будут записываться в соответствующие элементы матрицы Y . Для каждого i от q до единицы выполняются следующие шаги.

1. Ведущий процессор выполняет рассылку диагонального блока L_{ii}^d всем GPU.

2. Выполняется решение СЛАУ с верхней треугольной матрицей $(L_{ii}^d)^T$ и правой частью Y_{ik}^d , где $k = \overline{1, q}$, на соответствующих GPU:

$$(L_{ii}^d)^T X_{ik}^d = Y_{ik}^d.$$

3. Каждый CPU выполняет пересылку полученного решения матрицы X_{ik}^d всем остальным GPU.

4. Обновление матриц Y_{kj}^d , где $j = \overline{\min(i-l, 1), i+1}$, $k = \overline{1, q}$, по формуле

$$Y_{kj}^d \leftarrow Y_{kj}^d - (L_{ij}^d)^T X_{ik}^d. \quad (10)$$

При этом вычисления на шаге 4 будут проводиться на всех GPU, а каждая матрица Y_{kj}^d будет изменяться только в GPU с логическим номером $(k-1) \bmod p$.

Эффективность алгоритма факторизации. Сложность факторизации матрицы системы (3) намного превышает сложность решения треугольных систем (4) и (5), поэтому остановимся на оценке эффективности алгоритма факторизации матрицы СЛАУ.

Обозначим t_C и t_G время выполнения одной арифметической операции соответственно на CPU и GPU. Пусть количество арифметических операций, кото-

рое может выполнять GPU одновременно, равно n_o . Поскольку GPU реализует архитектуру SIMD, то среднее время, затраченное одним GPU на выполнение Q однородных арифметических операций, можно оценить величиной Q/n_o .

Пусть общее количество операций алгоритма факторизации равно M на шаге 1, равно N на шаге 3, равно K на шаге 4, равно P на шаге 7. На остальных этапах алгоритма вычислений не проводится. Для этих величин справедливы следующие оценки:

$$M = \frac{s^3}{3} + O(s^2), \quad N = ms^2, \quad K = ms \frac{m+sp}{p}, \quad P = m^2 s \frac{p-1}{p}.$$

Предположим, что время пересылки одного блока между двумя GPU равно t_B . Если для рассылки данных от одного GPU всем остальным процессорам используется алгоритм «дерева» [7], то общее время мультирассылки на шаге 2 алгоритма факторизации можно оценить выражением $t_B \log_2 p$. Аналогично для шага 5 алгоритма общее время мультирассылки оценим как $lt_B \log_2 p$. На шагах 2 и 9 алгоритма факторизации выполняется копирование блока между CPU и GPU. Обозначим суммарное время этих операций t_I . Время на выполнение копирования на шаге 7 алгоритма можно не учитывать, так как это копирование будет проходить на фоне вычислений.

Пусть T_1 и T_p – время выполнения соответственно последовательного алгоритма и алгоритма с p графическими процессорами. Тогда имеем следующие оценки:

$$T_1 = q \left(\frac{s^3}{3} t_C + \frac{ms^2}{n_o} t_G + \frac{ms(m+s)}{n_o} t_G + t_I \right), \quad (11)$$

$$T_p = q \left(Mt_C + \frac{N}{pn_o} t_G + \max \left(\frac{K}{pn_o} t_G, lt_B \log_2 p \right) + \frac{P}{pn_o} t_G + t_I \right). \quad (12)$$

Как показывает практика, время обмена t_I и время факторизации $\frac{s^3}{3} t_C$ диагонального блока, как правило, намного меньше остальных слагаемых в формулах (11) и (12). Пренебрегая этими временами, получаем оценки ускорения и эффективности алгоритма:

$$S_p = \frac{T_1}{T_p} \cong \begin{cases} p, & \frac{K}{pn_o} \geq lt_B \log_2 p, \\ \frac{p(m+2s)}{s + m \frac{p-1}{p} + \frac{pn_o}{s^2} t_B \log_2 p}, & \frac{K}{pn_o} < lt_B \log_2 p, \end{cases}$$

$$E_p = \frac{S_p}{p} \cong \begin{cases} 1, & \frac{K}{pn_o} \geq lt_B \log_2 p, \\ \frac{(m+2s)}{s + m \frac{p-1}{p} + \frac{pn_o}{s^2} t_B \log_2 p}, & \frac{K}{pn_o} < lt_B \log_2 p. \end{cases}$$

Программная реализация и результаты численных экспериментов. Описанный параллельный алгоритм решения СЛАУ с ленточной симметричной матрицей реализован в программной среде MPI на кластере Института кибернетики имени В.М. Глушкова НАН Украины [11]. Для вычислений на GPU использовались технология CUDA и узлы кластера с многоядерными процессорами CPU Intel(R) Xeon(R) X5675 @ 3.07GHz и графическими ускорителями GPU Tesla M2050.

При реализации гибридного алгоритма целесообразно использовать функции библиотеки cuBLAS. Для вычислений по формуле (6) можно использовать функ-

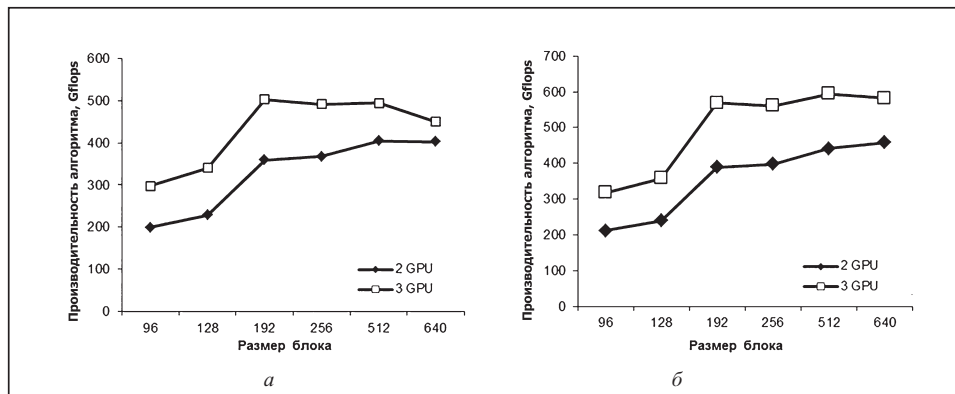


Рис. 1. Зависимость производительности алгоритма для 2 GPU и 3 GPU от размера блока

цию `cublasDtrsm`, а для формулы (8) — `cublasDgemm`. Для асинхронного запуска этих функций и лучшей утилизации вычислительных ресурсов GPU целесообразно вызывать эти функции, используя дополнительные потоки CUDA. Таким образом, операция синхронизации на шаге 8 алгоритма и вычисления на CPU на шаге 1 алгоритма будут выполняться одновременно с вычислениями по формуле (8).

В целях экономии оперативной памяти каждого CPU элементы матрицы L , которые хранятся в каждом CPU, будут перезаписывать соответствующие элементы исходной матрицы A , которые после факторизации не сохраняются.

На рис. 1, *а* показан график производительности алгоритма, измеряемой в Gflops, в зависимости от величины выбранного блока для разного количества GPU для матрицы СЛАУ порядка 1151112, полуширина ленты 5357, а на рис. 1, *б* — для матрицы СЛАУ порядка 1438890, полуширина ленты 6693.

Представляет также интерес сравнение гибридного алгоритма и параллельных алгоритмов для многоядерных компьютеров. На рис. 2 дан график зависимости ускорения (при вычислениях на двух и трех GPU) гибридного алгоритма по отношению к параллельному алгоритму, представленному в библиотеке Intel MKL, от количества ядер CPU для матрицы порядка 1151112 с полушириной ленты, равной 5357.

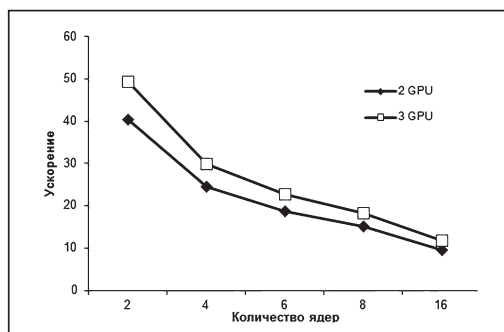


Рис. 2. График ускорения гибридного алгоритма

МОДЕЛИРОВАНИЕ РЕЖИМА ФИЛЬТРАЦИИ ЧЕРНИГОВСКОГО МЕСТОРОЖДЕНИЯ ПОДЗЕМНЫХ ВОД

Согласно выполнению работ программы «Стратегические минеральные ресурсы Украины» и работ общего проекта Национальной Академии наук Украины и ИАСА Институтом кибернетики им. В.М. Глушкова НАН Украины совместно с Институтом геологических наук НАН Украины было выполнено моделирование режима фильтрации Черниговского месторождения подземных вод (составляющая работ по уточнению запасов подземных вод). Для построения СЛАУ МКЭ был использован конечно-элементный решатель Надра-3D, а для нахождения ее решения использован описанный выше алгоритм для гибридных компьютеров с графическими ускорителями.

Модель геологической среды Черниговского месторождения подземных вод. Следуя предложенной специалистами-геологами модели из [8], массив геологической среды Черниговского месторождения подземных вод рассматривает-

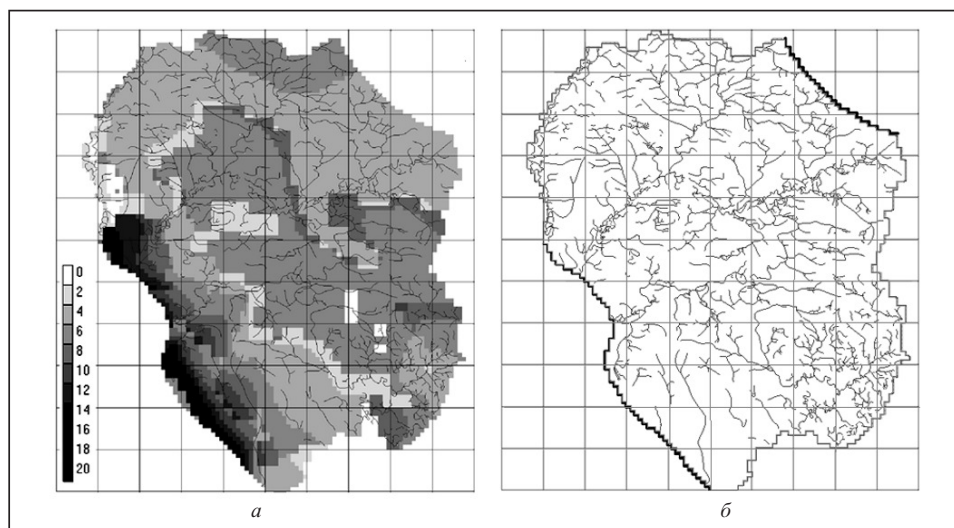


Рис. 3. Распределение коэффициентов фильтрации для сеноманского водоносного горизонта (а); сетка поверхностных рек Черниговского региона (б)

ся как этажная водоносная система, состоящая из пяти слоев — трех водоносных горизонтов, разделенных двумя водоупорами (слабопроницаемыми слоями). Размеры модели в плоскости составляют 184×222 км.

Каждый слой пород является многокомпонентным массивом, представленным в модели объединением областей с одинаковыми коэффициентами фильтрации. На рис. 3, а показано распределение таких областей для сеноманского водоносного горизонта. На кровле четвертичного водоносного горизонта заданы значения инфильтрации, а также условия 1-го рода в узлах, отвечающих руслам разветвленной речной сети (рис. 3, б). В качестве краевых условий на боковых поверхностях водоносных слоев заданы постоянные во времени значения пьезометрических напоров. На северо-восточной границе расчетной области эти значения приняты 140 м, на юго-западной границе — 110 м.

На боковых сторонах водоупоров и части боковых поверхностей водоносных горизонтов заданы краевые условия 2-го рода $Q = 0$ (условия отсутствия расхода через поверхность), на подошве расчетной области также принято условие непроницаемости.

Математическая модель фильтрации воды. Представим геологическую среду в виде объединения областей Ω_i с постоянными коэффициентами фильтрации k_i (м/сут): $\Omega = \bigcup_i \Omega_i$. Примем гипотезу, согласно которой в каждой из этих областей фильтрация подчиняется закону Дарси; тогда для каждой Ω_i справедливо уравнение

$$\frac{\partial}{\partial x} \left(k_{xi} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_{yi} \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_{zi} \frac{\partial h}{\partial z} \right) + f_i = \eta_i \frac{\partial h}{\partial t}, \quad (\bar{x}, t) \in \Omega_i \times (0, T],$$

где $\bar{x} = (x, y, z)$, $h(\bar{x}, t)$ — пьезометрический напор (м), k_{xi}, k_{yi}, k_{zi} — коэффициенты фильтрации области Ω_i (м/сут); f_i — водоотбор/водосброс на единицу объема за единицу времени (1/сут); η_i — коэффициент упругости породы (1/м) (значение η можно также определить из соотношения $\eta = k / \alpha$, α — коэффициент пьезопроводности (м²/сут)).

На границе контакта двух таких областей $\gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, $i \neq j$, справедливы условия идеального контакта:

$$[h] = 0, \quad (\bar{x}, t) \in \gamma_{ij} \times (0, T];$$

$$\left[\sum_{m=1}^3 k_m \frac{\partial h}{\partial x_m} \cos(\bar{n}, \bar{x}_m) \right] = 0, \quad (\bar{x}, t) \in \gamma_{ij} \times (0, T],$$

где \bar{n} — нормаль к границе контакта γ_{ij} , \bar{x}_m — m -я ось декартовой системы координат.

На участке границы $\Gamma_1 \in \partial\Omega$ известны значения $\varphi(\bar{x}, t)$ (в метрах) пьезометрических напоров (заданы краевые условия первого рода):

$$h = \varphi(\bar{x}, t), \quad (\bar{x}, t) \in \Gamma_1 \times (0, T].$$

На участке границы $\Gamma_2 \in \partial\Omega$ известны значения расхода β (м/сут) через единицу площади (заданы краевые условия второго рода):

$$\sum_{m=1}^3 k_m \frac{\partial h}{\partial x_m} \cos(\bar{n}, \bar{x}_m) = \beta, \quad (\bar{x}, t) \in \Gamma_2 \times (0, T].$$

На участках границы $\Gamma_3 \in \partial\Omega$, занятых водоемами, примем условия третьего рода (условия Ньютона):

$$\sum_{m=1}^3 k_m \frac{\partial h}{\partial x_m} \cos(\bar{n}, \bar{x}_m) = -\alpha h + \beta, \quad (\bar{x}, t) \in \Gamma_3 \times (0, T],$$

где $\alpha = \frac{k_u}{m_u}$, $\beta = \frac{k_u}{m_u} h_e$, k_u — коэффициент фильтрации донных иловых отложений водоема, m_u — мощность (толщина) слоя донных иловых отложений, h_e — уровень воды в водоеме.

Для решения задачи также необходимо задать начальные условия — распределение пьезометрических напоров на момент начала моделирования:

$$h(\bar{x}, 0) = \bar{\varphi}(\bar{x}), \quad \bar{x} \in \bar{\Omega}, \quad t = 0.$$

Для практических задач в качестве начального распределения обычно выступает решение соответствующей стационарной задачи по исходным данным на некоторый момент времени, когда внешними нагрузками можно пренебречь. Для Черниговского месторождения подземных вод — это данные на 1942 г., когда промышленные водоотборы можно считать несущественными.

Численное моделирование. Задача моделирования режима фильтрации для Черниговского месторождения подземных вод решалась методом конечных элементов [9] с помощью конечно-элементного решателя Надра-3D [10]. Расчетные сетки тетраэдральных конечных элементов строились для трехмерной модели региона (рис. 4) с учетом геометрии залегания слоев пород, зон изменения параметров в пределах каждого слоя, конфигурации сети поверхностных рек.

Особенность построенной модели — более детальный, чем позволяли конечно-разностные схемы, учет разветвленной речной сети региона. При моделировании это позволяет точнее учитывать влияние различных факторов на режим питания подземных вод, а соответственно — на их запасы. Моделирование выполнялось для

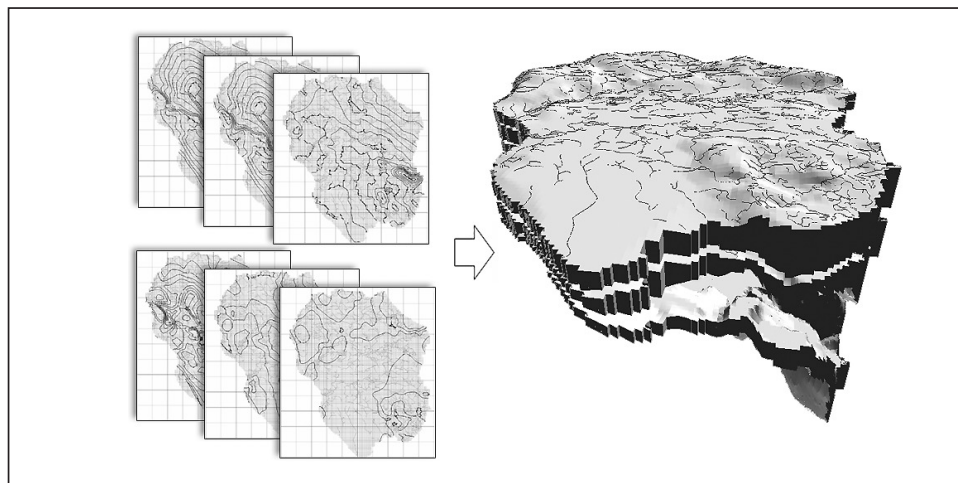


Рис. 4. Геометрическая модель геологической среды региона

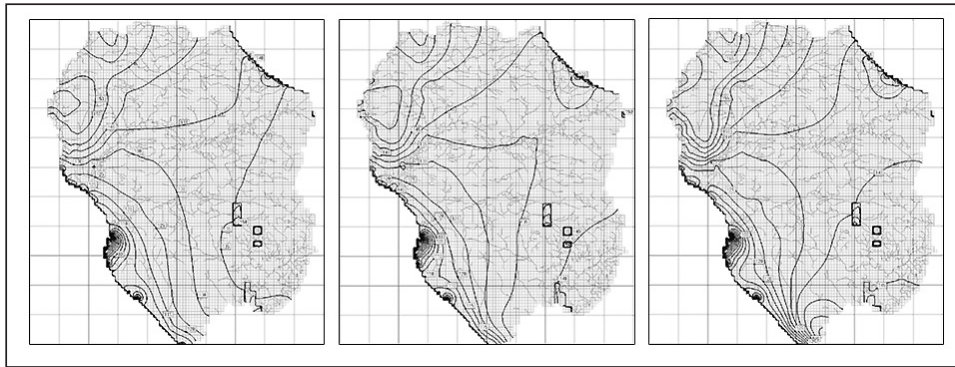


Рис. 5. Изолинии пьезометрических напоров для сеноманского водоносного горизонта

расчетных сеток различной детальности (от 400 тыс. до 1.5 млн неизвестных) с использованием кластерных вычислительных комплексов СКИТ Института кибернетики имени В.М. Глушкова НАН Украины. Целью моделирования является более точное, чем существующее в настоящее время описание влияния инфильтрации на кровле региона на режим фильтрации подземных вод месторождения. На рис. 5 приведены изолинии пьезометрических напоров для сеноманского водоносного горизонта, рассчитанные для различных значений инфильтрации. Как видим, модель позволяет оценить зависимость роста пьезометрических напоров внутри водоносного слоя от увеличения инфильтрационного питания.

ЗАКЛЮЧЕНИЕ

Предложенный гибридный алгоритм решения СЛАУ с ленточной симметричной положительно-определенной матрицей на многоядерных компьютерах с графическими ускорителями позволяет значительно (в десятки раз) уменьшить время моделирования методом конечных элементов на больших расчетных сетках с использованием прямых методов решения СЛАУ (с количеством неизвестных порядка $10^6 - 10^{12}$). Это особенно существенно для решения практических задач регионального моделирования пространственных процессов в многокомпонентных средах.

СПИСОК ЛИТЕРАТУРЫ

1. Михалеви́ч В.С., Молчанов И.Н., Сергиенко И.В. и др. Численные методы для многопроцессорного вычислительного комплекса ЕС / Под ред. И.Н. Молчанова. — М.: ВВИА им. Н.Е. Жуковского, 1986. — 401 с.
2. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. — М.: Мир, 1991. — 368 с.
3. Химич А.Н., Попов А.В., Полянко В.В. Алгоритмы параллельных вычислений для задач линейной алгебры с матрицами нерегулярной структуры // Кибернетика и системный анализ. — 2011. — № 6. — С. 159–174.
4. Попов А.В., Химич А.Н. Параллельный алгоритм решения систем линейных алгебраических уравнений с ленточной симметричной матрицей // Компьютерная математика. — 2005. — № 2. — С. 52–59.
5. Уилкинсон Дж. Х., Райнш К. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. — М.: Машиностроение, 1976. — 389 с.
6. Хіміч О.М., Баранов А.Ю. Гібридний алгоритм розв'язування лінійних систем із стрічковими матрицями прямими методами // Компьютерная математика. — 2013. — № 2. — С. 80–87.
7. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.
8. Шесто́палов В.М., Руденко Ю.Ф. та ін. Звіт: Оцінка стану прогнозних ресурсів та експлуатаційних запасів підземних вод Київської та Чернігівської областей методом математичного моделювання. — Київ: Фонди НІЦ РПД, 2007. — 104 с.
9. Дейнека В.С., Сергиенко И.В., Скопецкий В.В. Модели и методы решения задач с условиями сопряжения. — К.: Наук. думка, 1998. — 616 с.
10. Білоус М.В. Скінченно-елементний розв'язувач Надра-3D // Матеріали II Міжнародної конференції «Кластерні обчислення-2013», Львів, 3-5 червня 2013 року. — С. 40–47.
11. <http://icybcluster.org.ua>

Поступила 29.12.2014