Поэтому для решения уравнений (1)-(4) был разработан параллельный алгоритм решения сеточных уравнений методом МПТМ, ориентированный на многопроцессорную вычислительную систему (МВС). Для повышения эффективности расчетов часть вычислительной нагрузки перекладывается на графический ускоритель, для чего разработан соответствующий алгоритм и его программная реализация на языке CUDA C.

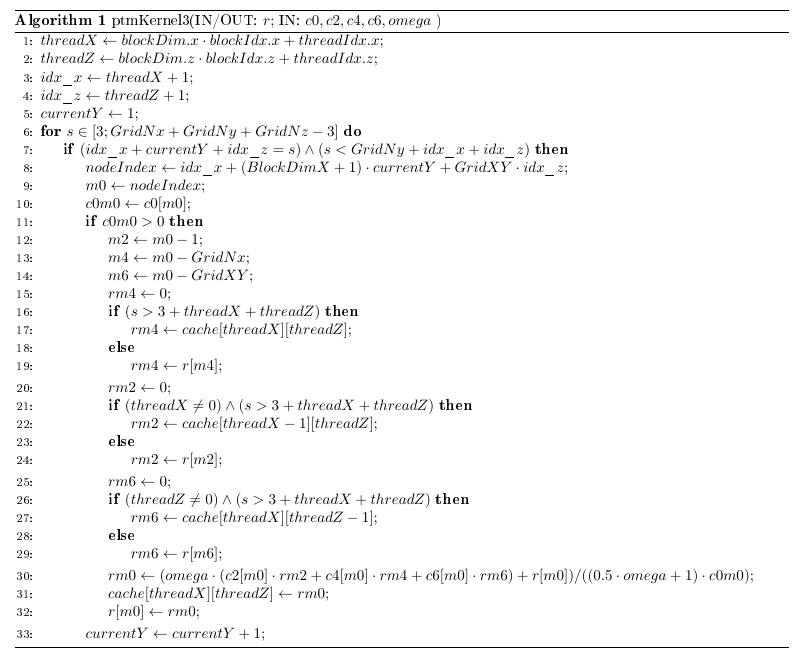


Рисунок 2 – Алгоритм реализации этапа МПТМ на CUDA C

На рисунке 2 приведен алгоритм поиска решения системы уравнений с нижнетреугольной матрицей (прямой прогон) на CUDA C.

Входными параметрами алгоритма являются векторы коэффициентов сеточных уравнений , константа . Выходной параметр – вектор скорости водного потока . Для вычисления вектора скорости водного потока необходимо: знать количество узлов расчетной сетки , размерности вычислительного блока CUDA , размерности расчетной сетки по пространственным координатам , вычислить индекс строки и индекс слоя, которую обрабатывает текущий поток, вычислить индекс центрального узла расчетной сетки и номера окрестных узлов .

В описанном выше алгоритме вычисления выполняются для каждого фиксированного слоя по координате .

??????????????

Проведенные исследования показывают существенную зависимость времени исполнения алгоритма вычисления предобуславливателя от соотношения нитей по пространственным координатам.

В процессе экспериментальных исследований использовался видеоадаптер GeForce МХ 250, объём видеопамяти которого составляет 4 Гб, тактовая частота ядра 1518 - 1582 МГц, тактовая частота памяти 7000 МГц, разрядность шины видеопамяти – 64 бита, число ядер CUDA – 384.

Цель эксперимента – определить распределение потоков по осям и  расчетной сетки при фиксированном значении узлов сетки по оси , равным 10000 таким образом, чтобы время вычисления на GPU одного шага ПТМ было минимальным.

В качестве факторов приняты две величины: – количество потоков по оси X, – количество потоков по оси Z. Целевая функция: – время вычисления одного шага ПТМ на GPU, мс.

Произведение потоков  не должно превышать 1024. Такое ограничение накладывает CUDA, т.к. 1024 – это количество потоков в одном блоке. В связи с вышеизложенным уровни варьирования факторов и были выбраны, как показано в таблице 1.

Таблица 1 – Результаты эксперимента

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | X | Z | TGPU, мс |
|  | 16 | 64 | 64 |
|  | 32 | 32 | 65 |
|  | 64 | 16 | 81 |
|  | 128 | 8 | 109 |
|  | 256 | 4 | 100 |
|  | 512 | 2 | 103 |

В результате обработки экспериментальных данных было получено регрессионное уравнение (5) и по нему построен график (рисунок 1).

 (5)

где  – время вычисления одного шага ПТМ на GPU, мс;

– количество потоков по оси Z.

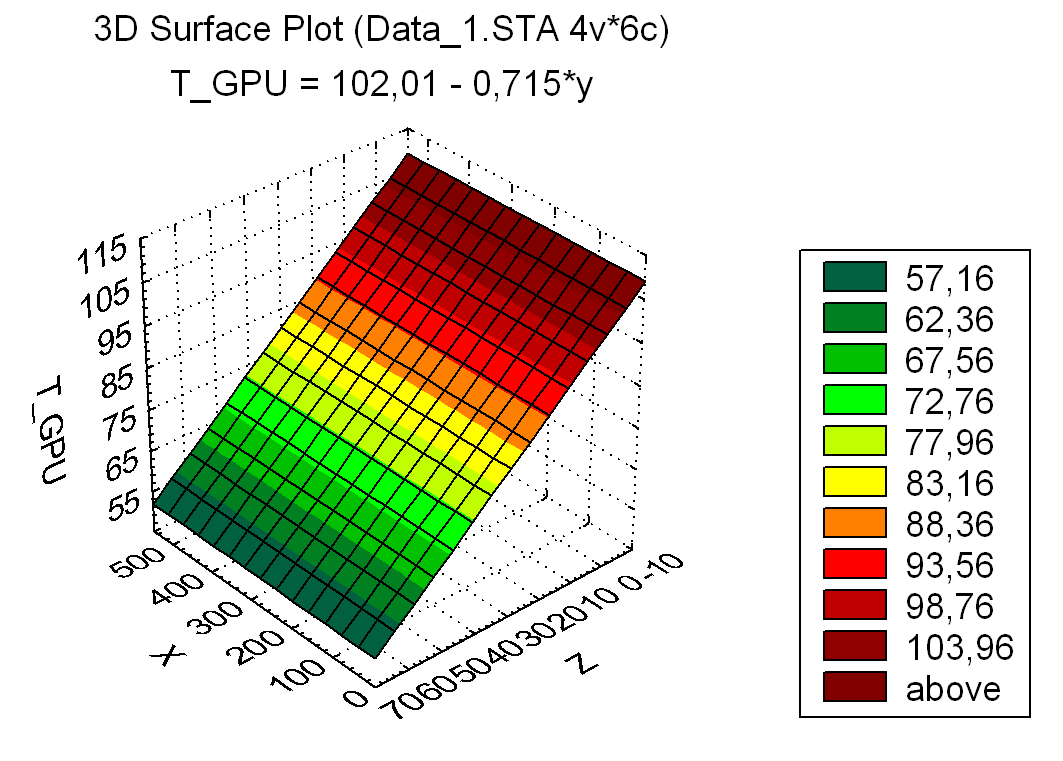


Рисунок 1 – Поверхность функции отклика , построенная по регрессионному уравнению

В результате анализа экспериментальных данных установлено, что на время вычисления одного шага ПТМ на GPU влияет только количество потоков по оси . Время вычисления одного шага ПТМ на GPU обратно пропорционально количеству узлов расчетной сетки по оси , т.е. с увеличением числа узлов по оси  время вычисления уменьшается по линейному закону. Поэтому целесообразно выполнять декомпозицию расчетной области в виде параллелепипедов, у которых размер по оси  максимален, а значит по оси минимален.