Моделирование мелкой жидкости

Тенищев С.С.

Используются [уравнения записанные в консервативной форме](https://en.wikipedia.org/wiki/Shallow_water_equations#Conservative_form), без учета плотности жидкости.

u – скорость по оси Ox, v – скорость по оси Oy, h – высота столба жидкости.

Заметим, что уравнения можно переписать в векторном виде, как:

Так как:

Для решения уравнений вида:

Применим [метод Lax-Wedorff](https://en.wikipedia.org/wiki/Lax%E2%80%93Wendroff_method), в работе P.D Lax; B. Wendroff (1960). "Systems of conservation laws" показывается, что метод имеет точность *O(x3)*. Но он требует вычисление якобиана, поэтому воспользуемся его модификацией – двух-шаговый [метод Richtmyer](https://en.wikipedia.org/wiki/Lax%E2%80%93Wendroff_method#Richtmyer_method). На первом шаге функция *u* вычисляется в момент *ti+1/2* и решетку со смещением *xi+1/2*. Второй шаг вычисляет функцию *u* в момент *ti+1* используя значения в моменты *ti+1/2* и*ti*.

В нашем случае u – функция трех аргументов t, x и y.

Первый шаг:

Второй шаг:

Для программного моделирования будем использовать периодические граничные условия.

Для графического представления используется пакет OpenGL.

**Параллелизация и ускорение.**

Для ускорения и распараллеливания программы были рассмотрены пакеты [omp4j](http://www.omp4j.org/) (OpenMP for Java 6/7/8), [JOMP](https://www2.epcc.ed.ac.uk/computing/research_activities/jomp/index_1.html) (Java Open MP), [JaMPCuda](https://www2.informatik.uni-erlangen.de/EN/research/JavaOpenMP/index.html), [JPPF](http://www.jppf.org/).

Omp4j – в данный момент не может обрабатывать сложные многоклассовые проекты.

JOMP – устаревший пакет, не поддерживающий современную Java.

JaMPCuda – требует Nvidia Cuda определенных версий.

JPPF – рабочий пакет, но используется для распределения между компьютерами, требует реализации кода для клиентов и сервера задач.

В итоге для распараллеливания программы была написана ручная реализация того же кода что бы сгенерировал OpenMP для распараллеливания цикла. Реализация использует стандартные средства языка для поддержания пула вычисляющих процессов (ServiceExecutor) и барьеров(CyclicBarier).

К сожалению, в момент когда однопоточная реализация перестает успевать выдавать 25 расчетов в секунду (вычисление следующего состояния векторов u занимает более 40 мс), многопоточная реализация все еще дает не значительное ускорение. С другой стороны, размер матрицы, для которой однопоточная реализация работает более 40 мс, довольно большой и уже невозможно заметить незначительные колебания воды.