

**ВВЕДЕНИЕ В РАБОТУ С НЕСТРУКТУРИРОВАННЫМИ СЕТКАМИ С ПОМОЩЬЮ
ПЛАТФОРМЫ INMOST И ДРУГИХ ИНСТРУМЕНТОВ**

Москва
2022

1 Установка INMOST

INMOST (www.inmost.org, <https://github.com/INMOST-DEV/INMOST>) – разрабатываемая в ИВМ платформа, предоставляющая необходимые инструменты и структуры данных для решения различных уравнений на неструктурированных сетках. Язык разработки – C++. Основной функционал, используемый в данном курсе – работа с неструктурированными сетками и решение линейных систем. Остальные возможности INMOST, такие как автоматическое дифференцирование для построения матриц Якоби и распараллеливание на основе MPI, остаются за рамками данного курса.

Перед скачиванием рекомендуется завести некоторую начальную директорию, в которой затем будут располагаться INMOST и связанные с ним проекты. Перейдя в эту директорию, можно скачать INMOST следующей командой:

```
git clone https://github.com/INMOST-DEV/INMOST.git
```

Затем надо перейти в новую директорию и создать папку со сборкой:

```
cd INMOST
mkdir build
cd build
```

Для сборки используется CMake. Запускаем сборку:

```
cmake ..
make
```

2 Установка пакета работы с сетками INMOST-GridTools

Пакет INMOST-GridTools используется, чтобы научиться собирать проекты, зависящие от INMOST. Этот пакет содержит некоторые полезные утилиты для работы с сетками, а также ряд генераторов неструктурированных сеток в единичном кубе. По большей части, эти генераторы создают сетки с разными кривыми ячейками, которые создают проблемы для методов дискретизации, и эти сетки используются для проверки их работы. Генераторы сохраняют сетки в разных форматах, среди которых мы будем использовать формат VTK (Visualisation ToolKit) – `.vtk`.

Установите INMOST-GridTools аналогично INMOST, адрес проекта – <https://github.com/INMOST-DEV/INMOST-GridTools>.

3 Установка ParaView

ParaView (<https://www.paraview.org/>) – повсеместно используемое средство визуализации, способное работать с файлами в формате VTK. Мы воспользуемся лишь малой частью возможностей ParaView. Необходимо скачать с официального сайта.

4 Генерация сеток и их визуализация в ParaView

. Научимся просматривать сетки в ParaView. Для этого в папке `build` проекта INMOST-GridTools запустите один из следующих исполняемых файлов:

- Азамат – kershaw_mesh
- Антон – acute_mesh
- Ирина – sinusoidal_mesh
- Лев – hex_mesh
- Михаил – shestakov_mesh
- Софья – nonconvex_mesh

Как правило, нужно подать 2 аргумента – число ячеек по горизонтальным осям и число ячеек по вертикальной оси.

Откройте `grid.vtk` в ParaView. Нажмите на зеленую кнопку **Apply**. Общий интерфейс представлен на рисунке 1. Основная работа с данными на сетках в ParaView делается с помощью *фильтров*. Среди **Filters** найдите **Clip**, задайте нормаль $[1\ 1\ 1]$ и отметьте галочку **Crinkle clip**, чтобы ячейки не резались. Примерный результат для одной из сеток представлен на рисунке 1. Использует формат отображения сетки **Surface with edges**, чтобы видеть отдельные ячейки. В данный момент на сетке (в большинстве случаев) нет никаких данных, так что раскрасить сетку можно только в **Solid Color**.

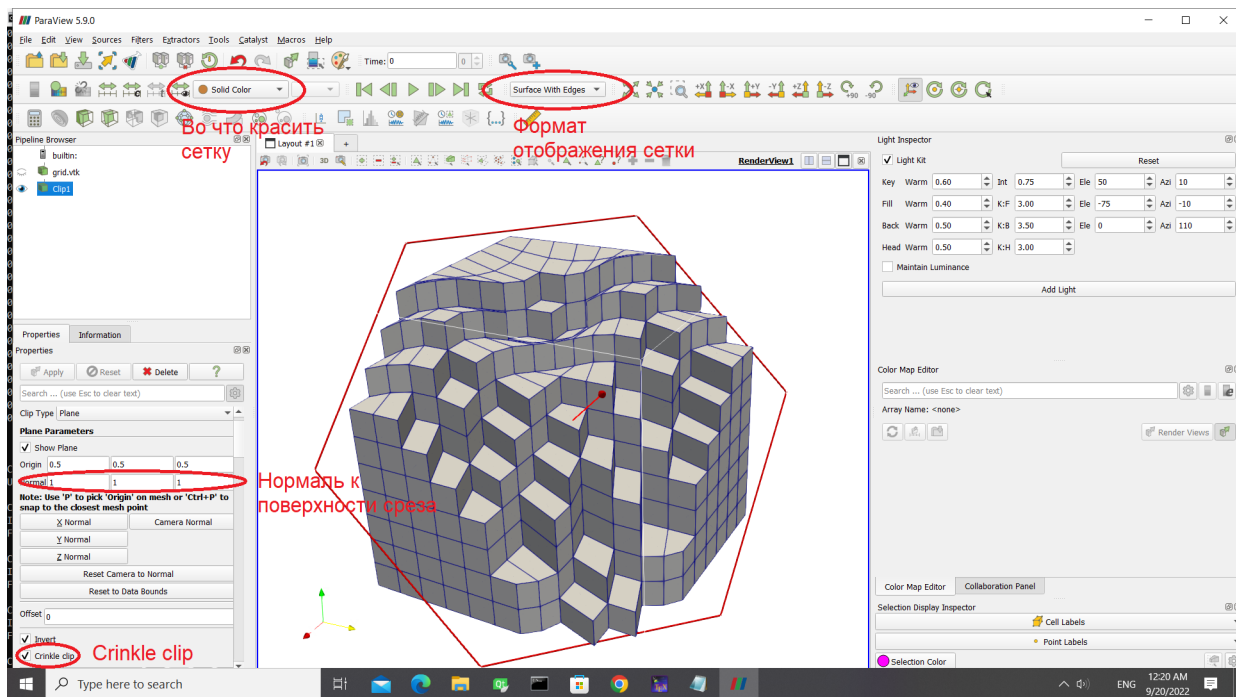


Рис. 1: Интерфейс ParaView

5 Генерация сеток в Gmsh

Gmsh (<https://gmsh.info/>) – один из самых популярных генераторов сеток, способный строить двумерные и трехмерные сетки и сохранять их в формате VTK.

5.1 Построение треугольных сеток в единичном квадрате

Воспользуемся малой частью функционала Gmsh, чтобы построить треугольные сетки в единичном квадрате. В директории **data** находится файл для Gmsh под названием **square.geo**. Откроем его в Gmsh (см. рис. 2).

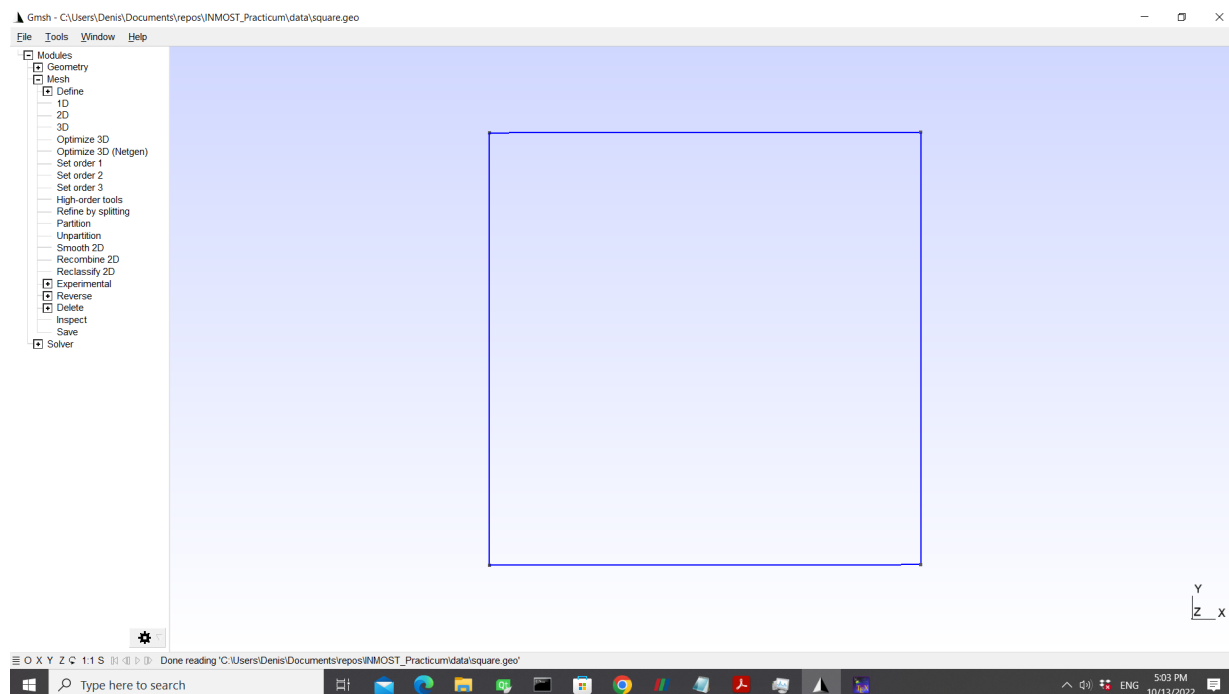


Рис. 2: Интерфейс Gmsh

Построение сеток в Gmsh происходит следующим образом. Сначала разбиваются одномерные объекты (ребра), затем двумерные, а если область трехмерная – то и трехмерные. Нам сначала нужно разбить ребра, для чего в меню **Mesh** выберем **Define** и в нем нажмем на **1D**. Далее строим двумерную сетку, выбрав **2D**. Результат представлен на рисунке 3. Сетка содержит 4 треугольника. Граница квадрата фактически не разбивалась, поскольку в настройках по умолчанию стоит разбиение линий на отрезки длины 1.

Как получить более мелкие сетки? Во-первых, можно измельчить имеющуюся сетку, выбрав **Mesh** -> **Define** -> **Refine by splitting**. После трехкратного применения этого действия сетка выглядит, как показано на рисунке 4.

Но такая сетка не является достаточно интересной, все треугольники в ней одинаковые (с точностью до поворота), имеется некоторая структура. Поэтому применим также второй способ. Вернемся к **Mesh** -> **Define** -> **1D**. И сделаем **Refine by splitting** в этот момент. Применив 3 раза, получим разбиение отрезков на $2^3 = 8$ частей (этого пока не видно). Затем применим **Mesh** -> **Define** -> **2D** и получим результат, представленный на рисунке 5.

5.2 Сохранение в формате VTK

Нужно выбрать **File** -> **Export** или просто нажать **Ctrl+E** и задать имя файла, оканчивающегося на **.vtk**.

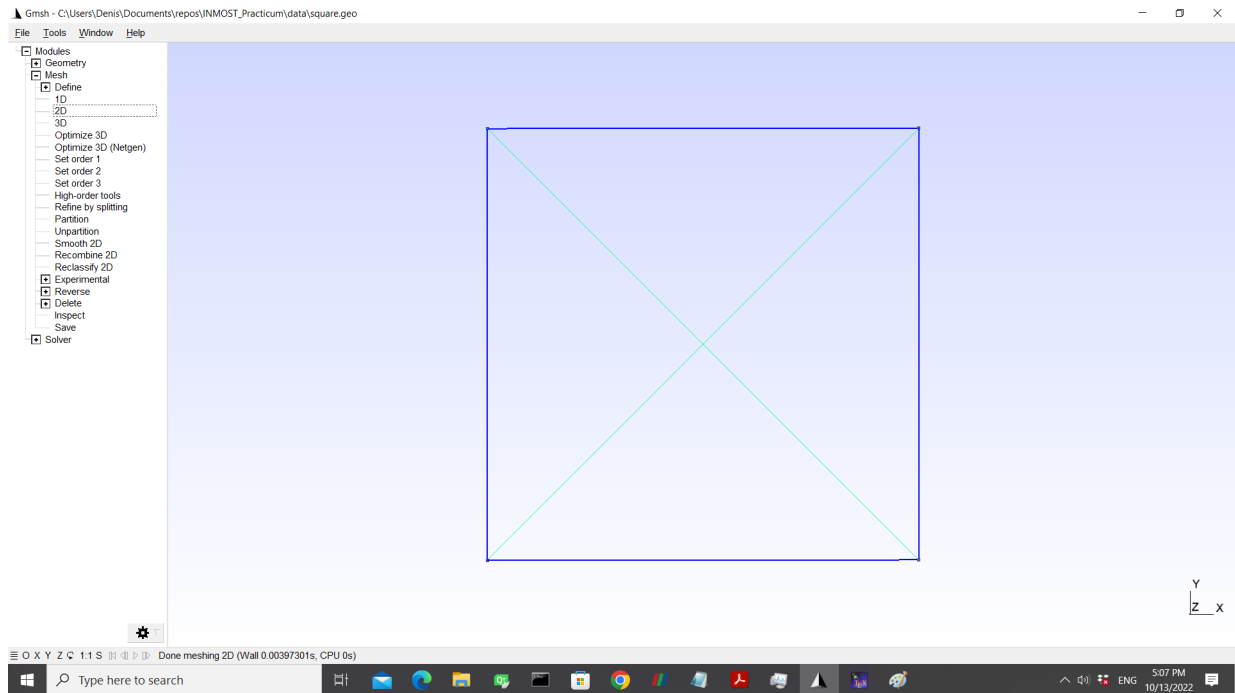


Рис. 3: Простейшая треугольная сетка в Gmsh

6 Работа с данными на сетках в INMOST

6.1 Создайте свой проект

Создайте директорию для своего проекта рядом с директориями `INMOST`, `INMOST-GridTools`. Поместите туда приложенные `CMakeLists.txt`, `main.cpp`. Создайте папку `build` и проверьте, что сборка идет успешно.

6.2 Понятие тега

Для записи данных на сетке в `INMOST` используется понятие *тега* (ярлыка). При наличии объекта класса тег считается, что на некоторых сеточных элементах (узлах, ячейках, гранях) хранится информация, которую можно запросить для этих сеточных элементов. При создании тега указываются

- имя тега,
- тип данных (как правило, целые или действительные числа),
- тип элементов, на которых определен тег, при этом для сразу нескольких типов запись имеет вид `CELL|FACE`,
- тип элементов, где тег является разреженным, эта возможность нам не нужна, так что будет писать `NONE`,
- размер тега: для скаляров это 1, для векторов – 2 или 3; мы пока ограничимся скалярами.

В приложенной программе на сетке создаются теги `Cell_count` и `X_Coord`. Первый тег определен на узлах и для каждого узла содержит число окружающих его ячеек. Второй тег определен на ячейках и содержит x-координату центроидов ячеек.

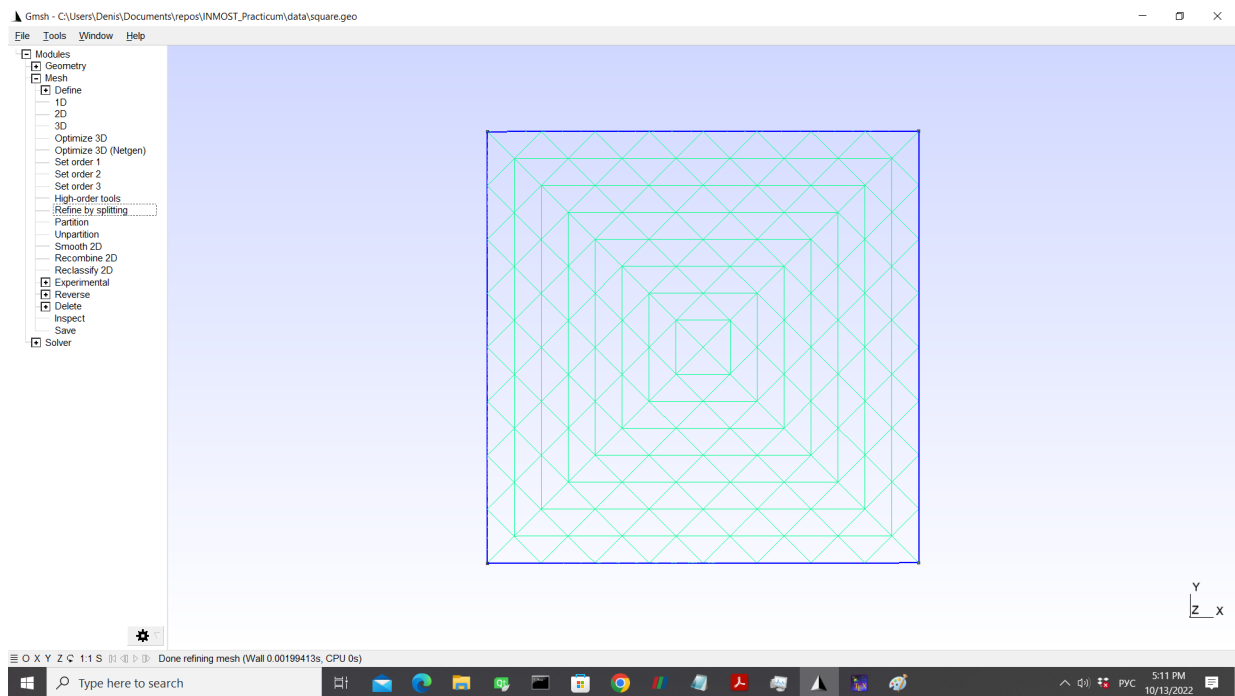


Рис. 4: Сетка, полученная измельчением простейшей

6.3 Задания: вычисление норм ошибки, подсчет числа узлов в ячейках

Даны сетки с тегами U и U_{approx} , определенными в ячейках. Считается, что функции кусочно-постоянны на сетке. Найдите C - и L_2 -нормы ошибки.

Создайте на ячейках тег и запишите для каждой ячейки число узлов.

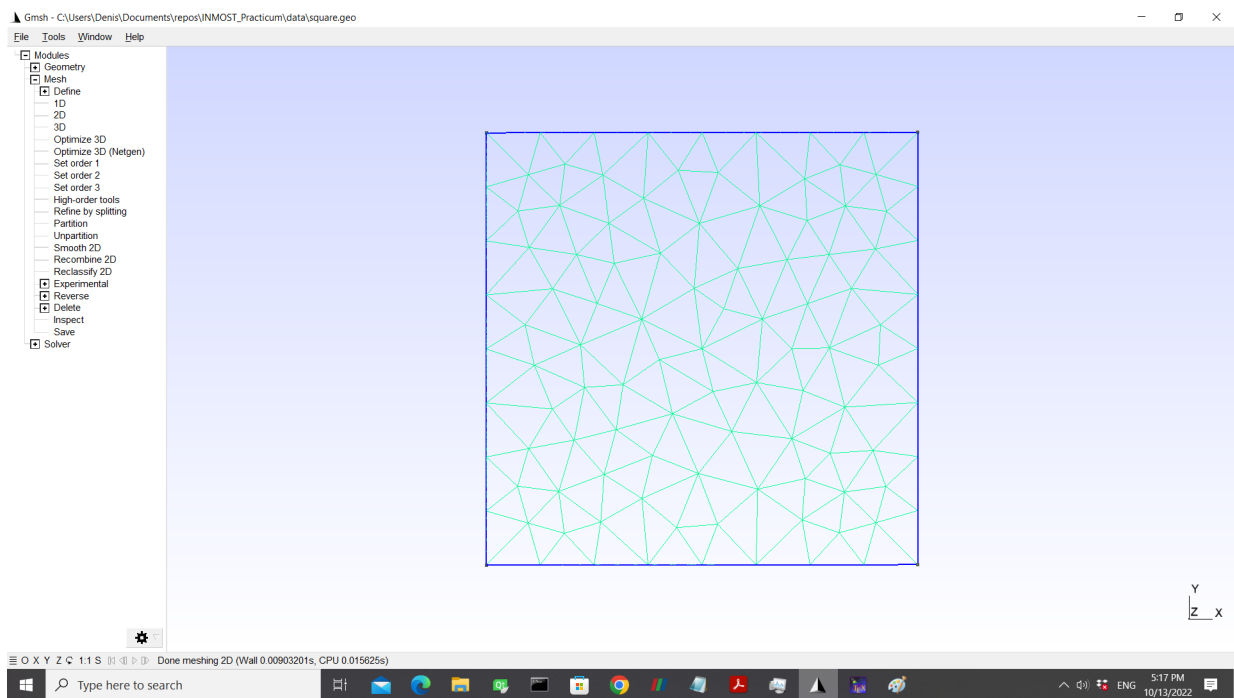


Рис. 5: Неструктурированная сетка, полученная предварительным измельчением разбиения границы