



Московский Государственный Университет им. М.В. Ломоносова
факультет Вычислительной математики и кибернетики
кафедра Суперкомпьютеров и Квантовой информатики



GMSH: Инструменты генерации сетки и постобработки в численном моделировании

Выполнила:
студентка группы 538, Ши Хуэй

Руководитель:
доцент кафедры СКИ, к.т.н., Русол А.В.

What is **gmsh**?

для генерации сеток методом конечных элементов

Создание и визуализация трехмерных сеток

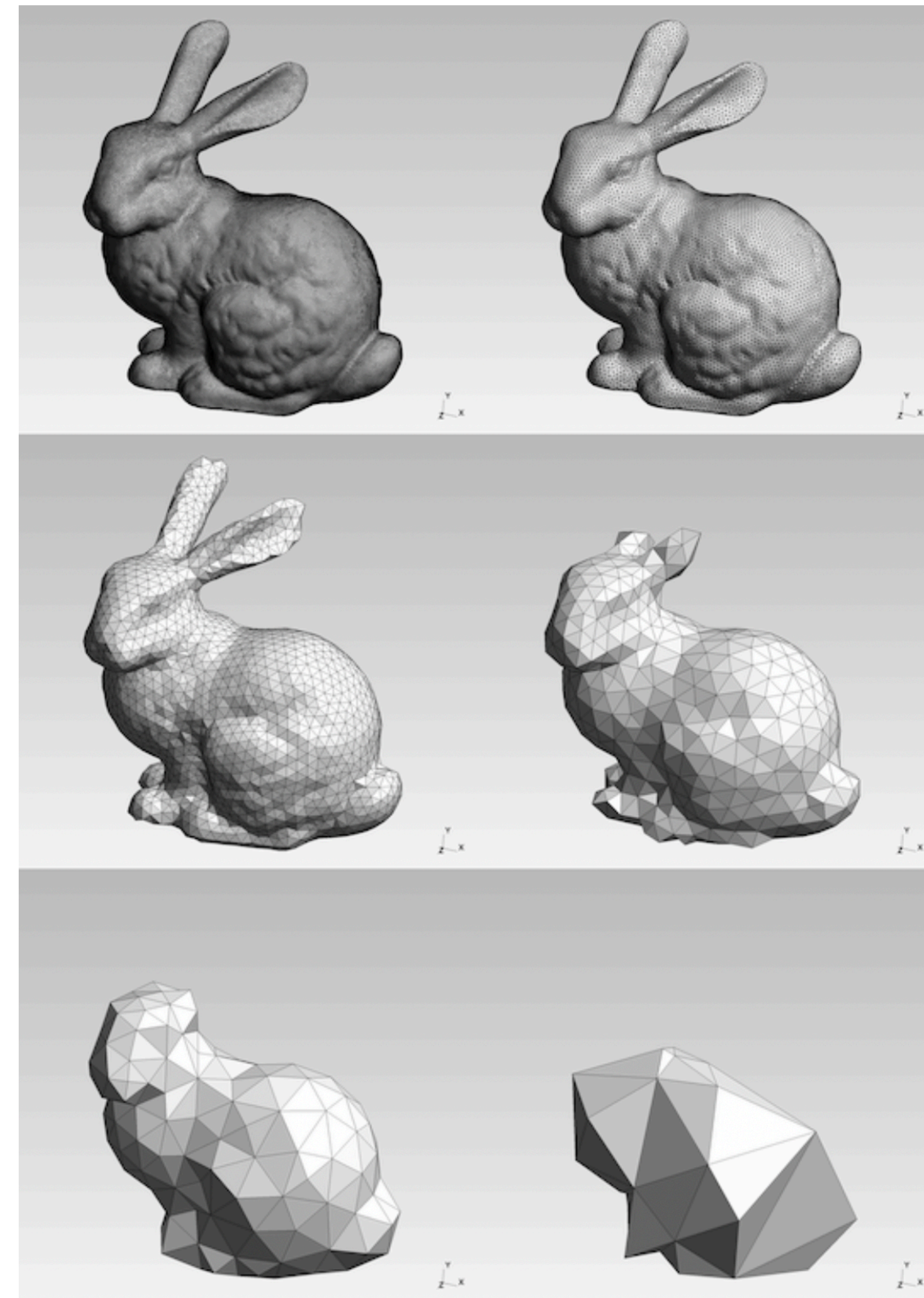
Численное моделирование (конечный элементный анализ, вычислительная гидродинамика и др.)

Geometry – Геометрия

Mesh – Сетка

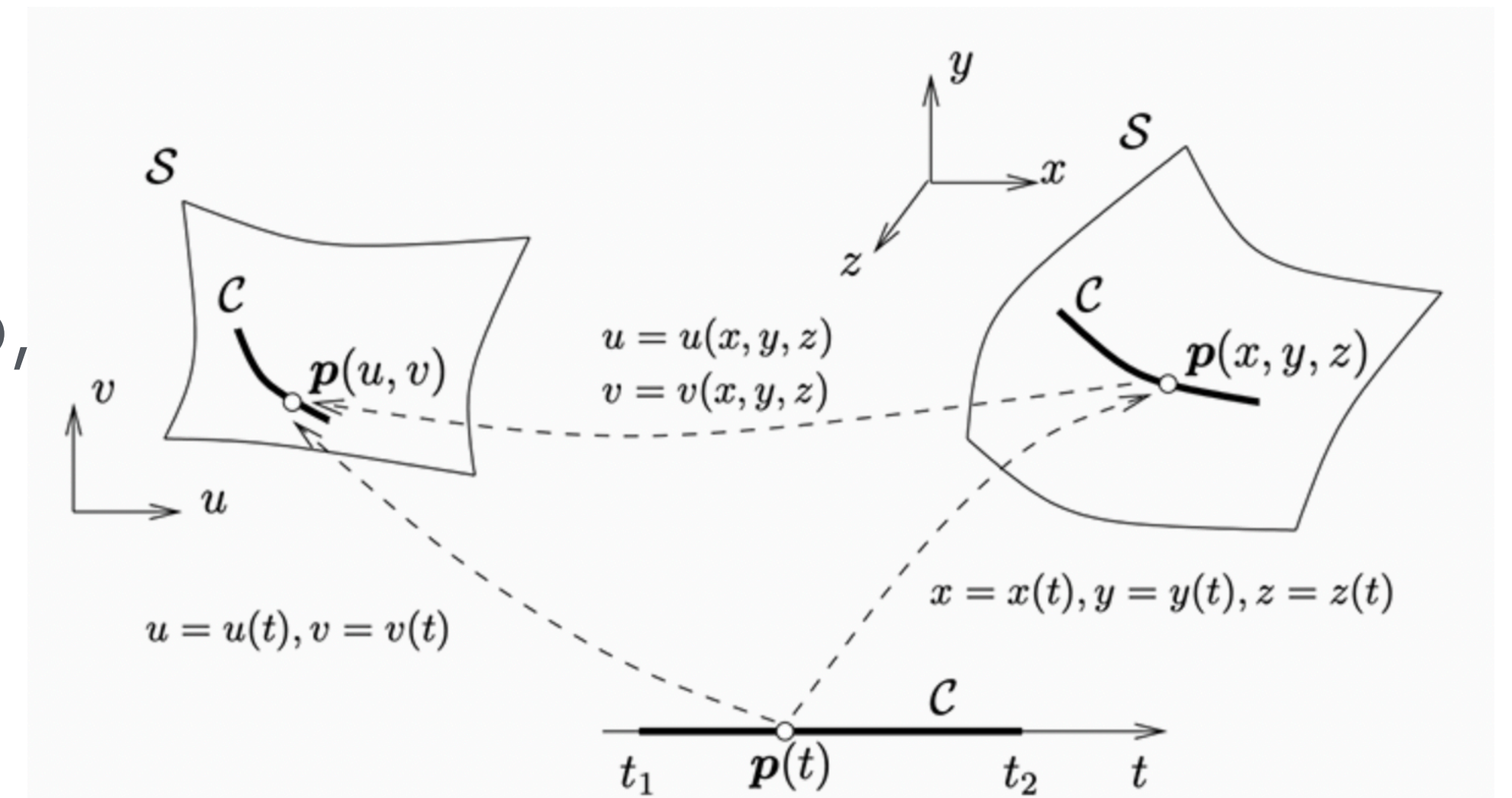
Solver – Решатель

Post-processing – Постобработка



Geometry – Геометрия

1. Определение геометрического объекта: точка, отрезок, поверхность и тело (например, прямоугольный параллелепипед, цилиндр, сфера и т.д.).



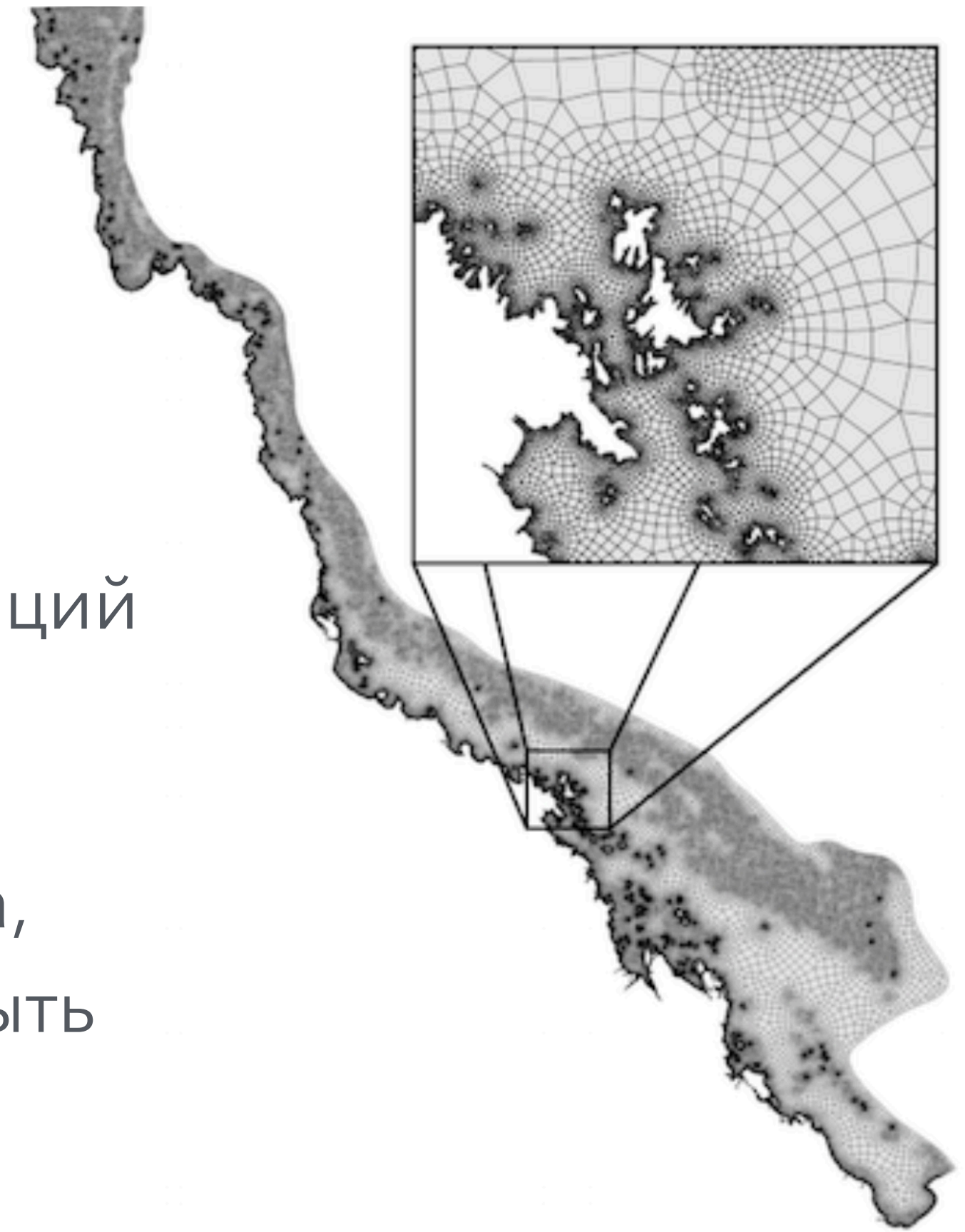
2. Геометрические преобразования: можно выполнять операции вращения, масштабирования, сдвига и другие преобразования геометрических объектов.

3. Геометрическое построение: поддержка более сложных конструкций геометрических объектов, включая создание новых форм путем операций с геометрическими телами (например, сложение, вычитание).

Mesh – Сетка

Генерация сетки: Модуль сетки GMSH в основном используется для преобразования геометрической модели в сетку, которая состоит из дискретных малых элементов (например, треугольников, тетраэдров, гексэдров и т.д.) и предназначена для последующих численных симуляций (например, конечном элементном анализе).

Разбиение сетки: В зависимости от формы геометрического объекта, GMSH автоматически генерирует разбиение сетки, которое может быть структурированным, неструктурированным или гибридным.



Seam Edges – Швы (или Краевые линии шва)
Degenerate Edges – Вырожденные ребра

CAD?

FEA – Метод конечных элементов (МКЭ)
CFD – Вычислительная гидродинамика (ВГД)

Solver – Решатель

1. **Интерфейс ONELAB:** GMSH не включает в себя собственный решатель, а интегрируется с внешними решателями через интерфейс ONELAB. ONELAB выполняет роль модели “клиент-сервер”, где серверная часть управляет данными, а клиент решателя (например, GetDP) выполняет конкретные численные расчеты.
2. **Клиент-сервер:** GMSH действует как сервер, который хранит все параметры симуляции и взаимодействует с клиентом через память или TCP/IP-сокеты. Например, GetDP – это клиент GMSH, который отвечает за выполнение конечного элементного решения. Клиент передает параметры через интерфейс ONELAB на сервер и получает результаты расчетов.
3. **Управление параметрами:** Через интерфейс ONELAB пользователи могут непосредственно управлять параметрами решателя через графический интерфейс GMSH, например, изменять условия решения, загружать модель и т.д. Все параметры централизованно управляются, что делает изменение и распространение данных более эффективным.

Post-processing – Постобработка

1. **Виды и визуализация данных:** Объяснение концепции видов, как хранятся данные и параметры отображения.
2. **Функции визуализации:** Основное внимание уделяется применению различных инструментов визуализации, таких как сечения, изоповерхности, векторные поля, линии тока и другие.
3. **Временные ряды и интерполяция:** Описание того, как модуль постобработки обрабатывает данные с несколькими шагами и адаптивное уточнение сетки, обеспечивая детализированные визуализационные результаты.

Why **gmsh**?

1. GMSH используется **для генерации и отображения сеток**, помогая создавать и визуализировать модель системы.
2. Сетки и модельные данные, созданные с помощью GMSH, могут быть использованы **для отображения результатов вычислений**, позволяя понять, как модель влияет на выполнение задачи.
3. Сетки, созданные в GMSH, подходят для вычислений, обеспечивая более быстрый процесс и простоту в использовании.
4. Можно запускать GMSH с помощью команд программирования на Python.

How to use **gmsh**?

Инициализировать : `gmsh.initialize()`

Определите форму физической проблемы :

`gmsh.model.occ.addPoint(x, y, z)`

`gmsh.model.occ.addLine(p1, p2)`

`gmsh.model.occ.addRectangle(x, y, z, dx, dy)`

`gmsh.model.occ.addCircle(x, y, z, radius)`

`gmsh.model.occ.addBox(x, y, z, dx, dy, dz)`

Сгенерировать сетку : `gmsh.model.mesh.generate()`

Геометрическое преобразование :

`gmsh.model.occ.translate`

`gmsh.model.occ.rotate`

Запустите графический интерфейс, чтобы просмотреть результаты визуализации сетки :

`gmsh.fltk.run()`

Экспорт сетевых данных : `gmsh.write("example.msh") #.stl`

