

МАТЕМАТИЧЕСКОЕ
МОДЕЛИРОВАНИЕ
ПРОВЕТРИВАЕМОСТИ
ГОРОДСКОЙ ЗАСТРОЙ
КИ НА НЕРЕГУЛЯРНОЙ
СЕТКЕ

СТУДЕНТ 4-ГО КУРСА
ТАИПОВА Д.Д.

Начиная с середины XX в. процесс урбанизации общества происходит высокими темпами, охватывая все большее число развивающихся стран. Наиболее интенсивно растут крупные города, которые, полнее удовлетворяют многообразные запросы людей. Если в начале XX в. население так называемых городов-мегаполисов не превышало 5 %, то к началу XXI в. его доля превысила треть населения мира. Вместе с тем опыт даже наиболее успешно развивающихся мегаполисов свидетельствует о трудностях решения целого ряда проблем, связанных с обеспечением благополучной в социально-экономическом отношении среды обитания горожан.

Основными источниками загрязнения воздуха жилых территорий являются промышленные предприятия и автомобильный транспорт.



Метод конечных объемов

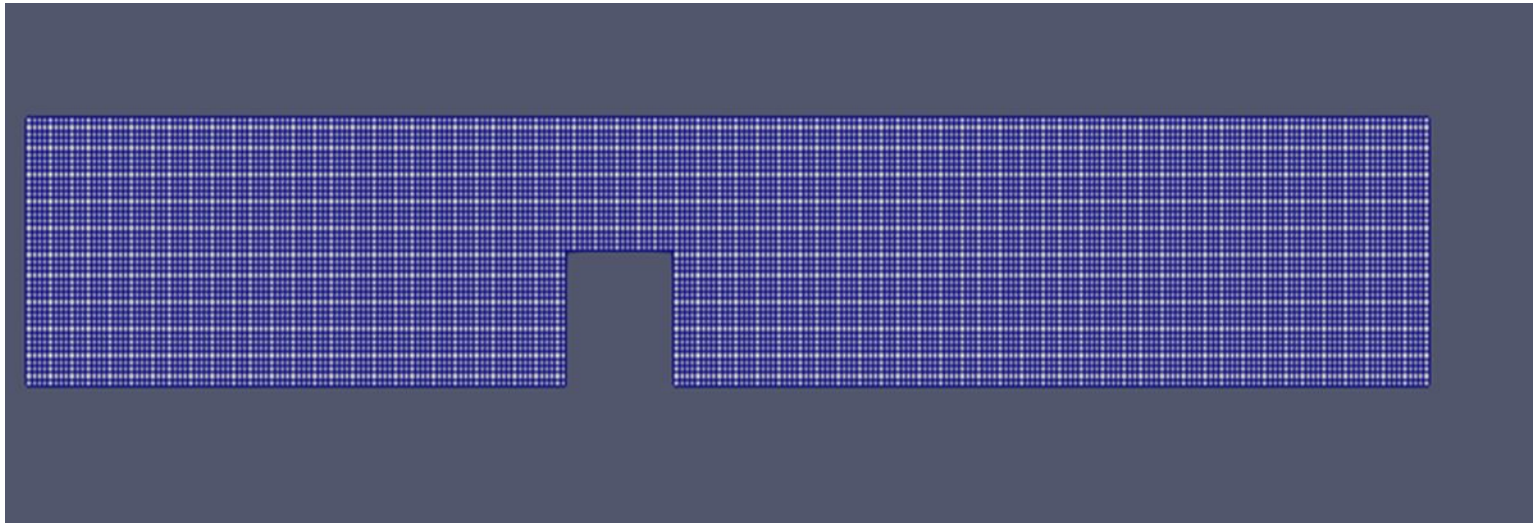
Метод конечных объемов (МКО) – универсальный способ построения консервативных схем для неравномерных, криволинейных и даже неструктурированных сеток. Этот метод применяется, в частности, при моделировании задач гидрогазодинамики в свободном пакете OpenFOAM

С помощью метода конечных объемов я решала одномерное и двумерное уравнение конвекции и диффузии. Поняв суть метода, переходим к решению задачи в пакете OpenFOAM.

Строим расчетную область

```
vertices  
(  
    (-5 0 0)  
    (0 0 0)  
    (1 0 0)  
    (8 0 0)  
    (-5 1 0)  
    (0 1 0)  
    (1 1 0)  
    (8 1 0)  
    (-5 2 0)  
    (0 2 0)  
    (1 2 0)  
    (8 2 0)  
  
    (-5 0 -0.01)  
    (0 0 -0.01)  
    (1 0 -0.01)  
    (8 0 -0.01)  
    (-5 1 -0.01)  
    (0 1 -0.01)  
    (1 1 -0.01)  
    (8 1 -0.01)  
    (-5 2 -0.01)  
    (0 2 -0.01)  
    (1 2 -0.01)  
    (8 2 -0.01)
```

```
blocks  
(  
    hex (12 13 17 16 0 1 5 4) (100 20 1) simpleGrading (1 1 1)  
    hex (14 15 19 18 2 3 7 6) (140 20 1) simpleGrading (1 1 1)  
    hex (16 17 21 20 4 5 9 8) (100 20 1) simpleGrading (1 1 1)  
    hex (17 18 22 21 5 6 10 9) (20 20 1) simpleGrading (1 1 1)  
    hex (18 19 23 22 6 7 11 10) (140 20 1) simpleGrading (1 1 1)  
);
```



Задаем начальные и граничные условия

```
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (1 0 0);
    }

    outlet
    {
        type            zeroGradient;
    }
}
```

```
top
{
    type            fixedValue;
    value           uniform (0 0 0);
}

bottom
{
    type            fixedValue;
    value           uniform (0 0 0);
}

cube
{
    type            fixedValue;
    value           uniform (0 0 0);
}

frontAndBack
{
    type            empty;
}
```

