

Численные методы решения задач динамики дисперсных систем

Среда моделирования OpenFOAM

К.И. Михайленко

ИМех УФИЦ РАН

<http://const.uimech.org/Q2rt67>



**ИНСТИТУТ
МЕХАНИКИ**
им. Р.Р. Мавлютова

УФИЦ РАН



Последовательность слайдов

1. Введение

- Историческая справка
- Достоинства и недостатки

2. Структура OpenFOAM

- Слои моделирования
- Ядро OpenFOAM
- Решатели СЛАУ
- Запись дискретного аналога
- Внешние программы

3. Пример расчёта в OpenFOAM

- Структура файлов
- Подготовка конечно-объёмной сетки
- Постановка начальных и граничных условий
- Управление расчётом



OpenFOAM: что и зачем

Назначение OpenFOAM

Open  FOAM

The Open Source CFD Toolbox

The Open Source **Field Operation And Manipulation** CFD ToolBox

Свободно распространяемый инструментарий вычислительной гидродинамики для операций с полями (скалярными, векторными и тензорными) – открытая интегрируемая платформа для численного моделирования задач механики сплошной среды.





Назначение OpenFOAM

- Задачи МСС и сопряженные задачи (жидкость–тело)
- МКО на базе многогранных ячеек, в том числе с поддержкой подвижных сеток
- Несжимаемые и сжимаемые, стационарные и нестационарные потоки
- Однофазные, двухфазные и многофазные потоки
- Лагранжева модель для расчета движения частиц
- URANS/LES/DNS моделирование турбулентности: $k - \epsilon$, $k - \omega$, $k - \omega$ SST и другие модели
- Модели конвекции, теплообмена, горения и распыления жидкости
- Неньютоновские жидкости

Историческая справка

Исходная разработка: Imperial College of Science. London. UK. 1991–2003

ГОСМЕН А. Д., ПАН В. М., РАНЧЕЛ А. К.,
СПОЛДИНГ Д. Б., ВОЛЬФШТЕЙН М.

ЧИСЛЕННЫЕ МЕТОДЫ
ИССЛЕДОВАНИЯ ТЕЧЕНИЙ
ВЯЗКОЙ ЖИДКОСТИ

*Перевод с английского
В. А. ХОХРЯКОВА
Под редакцией
Г. А. ТИРСКОГО*

ИЗДАТЕЛЬСТВО «МИР»
Москва 1972

Госмен А.Д., Пан В.М., Ранчел А.К., Сполдинг Д.Б., Вольфштейн М. Численные методы исследования течений вязкой жидкости. М.: Мир., 1972, 323 с.

Книга представляет собой значительный вклад в прикладную механику жидкости и газа, в ней излагается численный метод расчета стационарных двумерных однофазных ламинарных и турбулентных течений с учетом неоднородности свойств среды. На основе этого метода рассматриваются 11 конкретных задач. В книге приводятся рабочие программы, составленные на языке ФОРТРАН IV





A tensorial approach to computational continuum mechanics using object-oriented techniques

H. G. Weller and G. Tabor*

Department of Mechanical Engineering, Imperial College, London SW7 2BX, United Kingdom

H. Jasak

Computational Dynamics Limited, London W10 6BA, United Kingdom

C. Fureby

Department of Weapons and Protection, National Defence Research Establishment (FOA),

S-17200 Stockholm, Sweden

(Received 1 June 1998; accepted 11 August 1998)

In this article the principles of the field operation and manipulation (FOAM) C++ class library for continuum mechanics are outlined. Our intention is to make it as easy as possible to develop reliable and efficient computational continuum-mechanics codes: this is achieved by making the top-level syntax of the code as close as possible to conventional mathematical notation for tensors and partial differential equations. Object-orientation techniques enable the creation of data types that closely mimic those of continuum mechanics, and the operator overloading possible in C++ allows normal mathematical symbols to be used for the basic operations. As an example, the implementation of various types of turbulence modeling in a FOAM computational-fluid-dynamics code is discussed, and calculations performed on a standard test case, that of flow around a square prism, are presented. To demonstrate the flexibility of the FOAM library, codes for solving structures and magnetohydrodynamics are also presented with appropriate test case results given. © 1998 American Institute of Physics. [S0894-1866(98)01906-3]

INTRODUCTION

Computational continuum mechanics (CCM) is the simulation of continua using computers. Fluid dynamics is a significant branch of continuum mechanics and covers a variety of cases, including compressible, incompressible, multiphase, and free-surface flows, as well as flows involving further physics such as chemical reactions, species transport, phase changes, and electromagnetic effects. All these flows can be described by systems of linked partial differential equations of the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{Q}) - \nabla \cdot \mu \nabla \mathbf{Q} = \mathbf{S}_v \mathbf{Q} + \mathbf{S}_s, \quad (1)$$

where \mathbf{U} is the fluid velocity, ρ is density, and \mathbf{Q} is any source-valued property of the flow, such as species concentration. These equations involve time derivatives ($\partial \mathbf{Q} / \partial t$), convective terms [$\nabla \cdot (\rho \mathbf{U} \otimes \mathbf{Q})$], diffusive terms ($\nabla \cdot \mu \nabla \mathbf{Q}$), and source terms ($\mathbf{S}_v \mathbf{Q}$ and \mathbf{S}_s). A simple example is that of incompressible flow as described by the Navier-Stokes equations ($\mathbf{Q} = [\mathbf{U}]$):

$$\nabla \cdot \mathbf{U} = 0, \quad (2)$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \otimes \mathbf{U}) - \nabla \cdot \mathbf{2} \mathbf{D} = -\frac{1}{\rho} \nabla p,$$

*Corresponding author; E-mail: g.tabor@ic.ac.uk

where

$$\mathbf{D} = \frac{1}{2} (\nabla \mathbf{U} + \nabla \mathbf{U}^T), \quad (3)$$

The effect of the nonlinearity embodied in these equations is significant; only in special cases can algebraic solutions be found. The vast majority of fluid-flow problems can only be properly studied by using computational methods involving discretization of the domain and the equations, followed by numerical solution of the resulting system of equations. The complexity of the problem is increased if effects such as turbulence, compressibility, multiphase, free surface, chemical reactions, and electromagnetism are included. The two predominant solution techniques are the finite-element method (FEM),¹ in which the functional form of the solution to these equations is expanded in terms of a predetermined basis set and its residual minimized, and the finite-volume method (FVM).^{2,3} In the latter technique, which is used in this article, the computational domain is divided into a set of discrete volumes δV , which fill the computational domain D without overlap, i.e., $\cup_j \delta V_j = D$ and $\cap_j \delta V_j = \emptyset$. The fluid-flow equations are then volume-integrated over each individual finite volume δV . Gauss's theorem is used to convert the divergence terms in Eqs. (2) into surface-integrated flux terms, reducing the problem of discretizing these terms to one of finding difference approximations for the fluxes at the surface of the control volume based on the known cell-center values. Other spatial derivatives are dealt with in a similar manner. This

Weller H.G., Tabor G., Jasak H., Fureby C.
A Tensorial Approach to CFD using Object
Orientated Techniques // Computers in Physics,
1998 v. 12 n. 6, pp 620–631
DOI: 10.1063/1.168744



Историческая справка

- Изначально: FOAM – коммерческое ПО, разрабатываемое компанией Nabla Ltd.
- В 2004 году компания Nabla прекратила свою деятельность и открыла FOAM с GNU General Public License под названием «OpenFOAM».
- Была создана OpenCFD Ltd, которая разрабатывала и поддерживала OpenFOAM.
- В 2011 OpenCFD Ltd была приобретена SGI. Ими был создан Фонд OpenFOAM для обеспечения развития OpenFOAM как ПО с открытым исходным кодом.
- В 2012 OpenCFD Ltd была приобретена ESI.

Достоинства



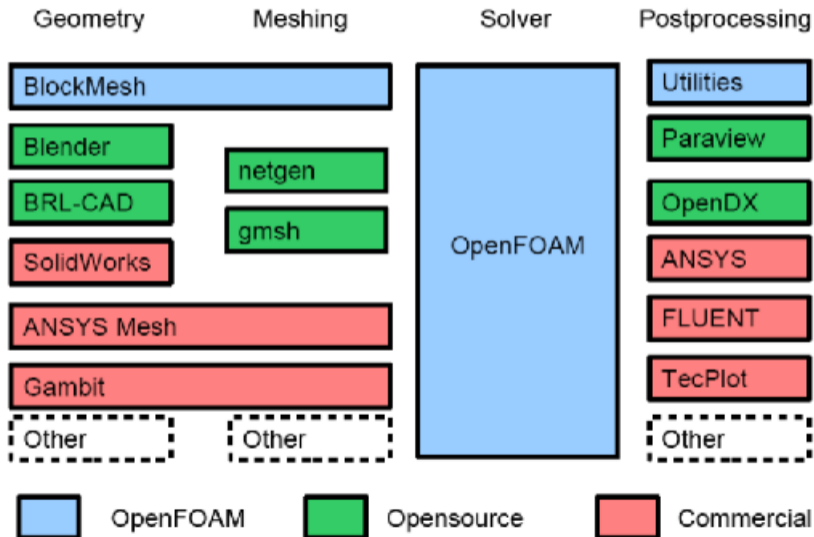
- Полностью доступная информация и возможность выбора методов дискретизации, алгоритмов решения систем диф.уравнений, решателей СЛАУ.
- Перспективное и динамично развивающееся открытое программное обеспечение для моделирования задач механики сплошной среды.
- В его разработке и развитии принимают участие десятки организаций и сотни разработчиков по всему миру.
- Обладает большой функциональностью и удовлетворяет всем основным требованиям, предъявляемым к современному программному обеспечению для расчета задач CFD.
- Конвертация между различными форматами входных и выходных данных: ANSYS, Fluent, STAR-CD, EnSight, Fieldview-UNS, GMV, Tecplot...

Недостатки



- Сложная идеология OpenFOAM.
- Пользователю необходимо быть квалифицированным программистом на C++ для того, чтобы развивать расчетные модули для реальных задач.
- Разрозненная документация.

Слои моделирования



Ядро OpenFOAM (solver)



- Объекты для объявления скалярных, векторных и тензорных полей (массивы физических параметров).
- Объекты и модели для дискретизации дифференциалов указанных полей (отображение дифференциалов на конечно-объемную сетку).
- Объекты для записи дискретного аналога исходных дифференциальных уравнений (запись решаемых в рамках используемого метода дифференциальных уравнений, в том числе для промежуточных и предварительных значений).
- Библиотека методов решения систем линейных алгебраических уравнений (решение полученных из предыдущего пункта СЛАУ подходящим алгоритмом)

Решатели СЛАУ: солверы



- DiagonalSolver
- GAMGSolver (Geometric agglomerated algebraic multigrid solver)
- PCG (Preconditioned conjugate gradient solver)
- SmoothSolver



Решатели СЛАУ: предобуславливатели

- diagonalPreconditioner
- DICPreconditioner (simplified **D**iagonal-based **I**ncomplete **C**holesky **P**reconditioner for symmetric matrices)
- DILUPreconditioner (simplified **D**iagonal-based **I**ncomplete **L**U **P**reconditioner for asymmetric matrices)
- FDICPreconditioner (**F**aster version of the **DIC**Preconditioner)
- GAMGPreconditioner (**G**eometric **A**gglomerated algebraic **M**ulti**G**rid **P**reconditioner)
- noPreconditioner



Решатели СЛАУ: сглаживатели

- DICSmoothing (simplified **D**agonal-based **I**ncomplete **C**holesky **S**moother for symmetric matrices)
- DICGaussSeidelSmoothing (combined **DIC**/GaussSeidel smoother for symmetric matrices)
- DILUSmoothing (simplified **D**agonal-based **I**ncomplete **LU** **S**moother for asymmetric matrices)
- GaussSeidelSmoothing
- NonBlockingGaussSeidelSmoothing
- SymGaussSeidelSmoothing



Решатели СЛАУ: запись дискретного аналога

Запишем уравнение импульсов в виде

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \phi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p$$



Решатели СЛАУ: запись дискретного аналога

Запишем уравнение импульсов в виде

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \phi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p$$

Явная схема

```
solve  
(  
    fvm::ddt(rho,U)  
    + fvc::div(phi,U)  
    - fvc::laplacian(mu,U)  
    ==  
    - fvc::grad(p)  
);
```



Решатели СЛАУ: запись дискретного аналога

Запишем уравнение импульсов в виде

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \phi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p$$

Явная схема

```
solve  
(  
    fvm::ddt(rho,U)  
  + fvc::div(phi,U)  
  - fvc::laplacian(mu,U)  
  ==  
  - fvc::grad(p)  
);
```

Неявная схема

```
solve  
(  
    fvm::ddt(rho,U)  
  + fvm::div(phi,U)  
  - fvm::laplacian(mu,U)  
  ==  
  - fvc::grad(p)  
);
```

Решатели СЛАУ: запись дискретного аналога



Рассмотрим произвольное уравнение

$$\Delta\psi + k^2\psi = f$$



Решатели СЛАУ: запись дискретного аналога

Рассмотрим произвольное уравнение

$$\Delta\psi + k^2\psi = f$$

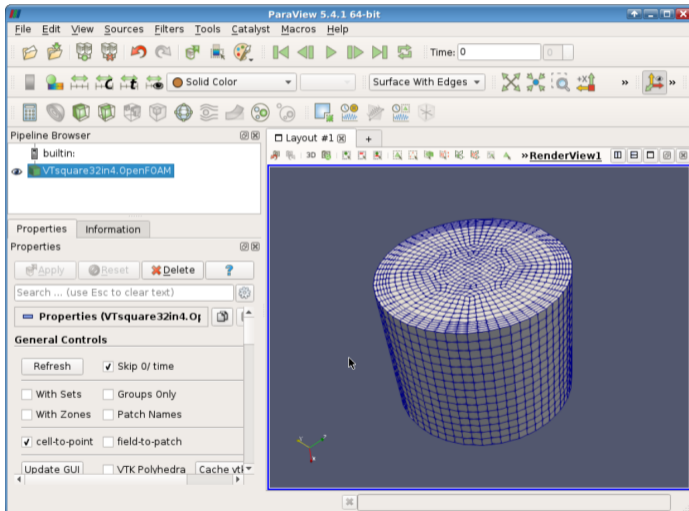
```
solve  
(  
    fvm::laplacian(psi) + k * k * fvm::Sp(1,psi) == f  
);
```

Внешние программы

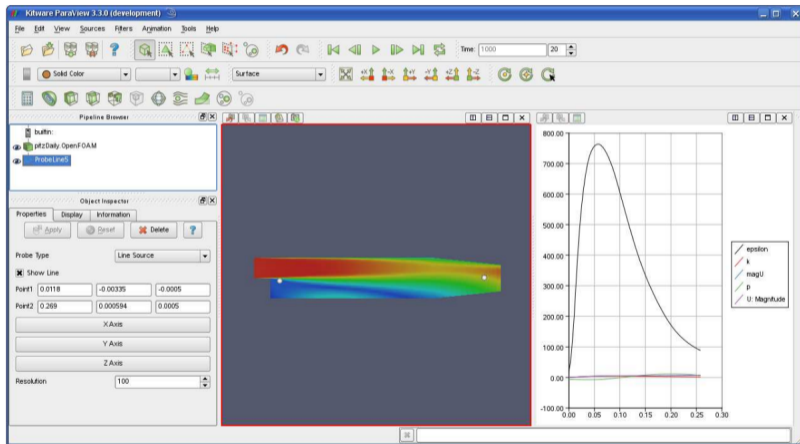
- gcc
- openmpi
- ParaView
- конвертеры данных
- ...



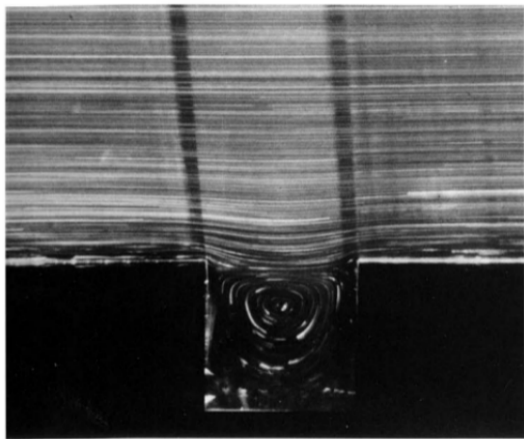
paraFoam



paraFoam



Задача о каверне

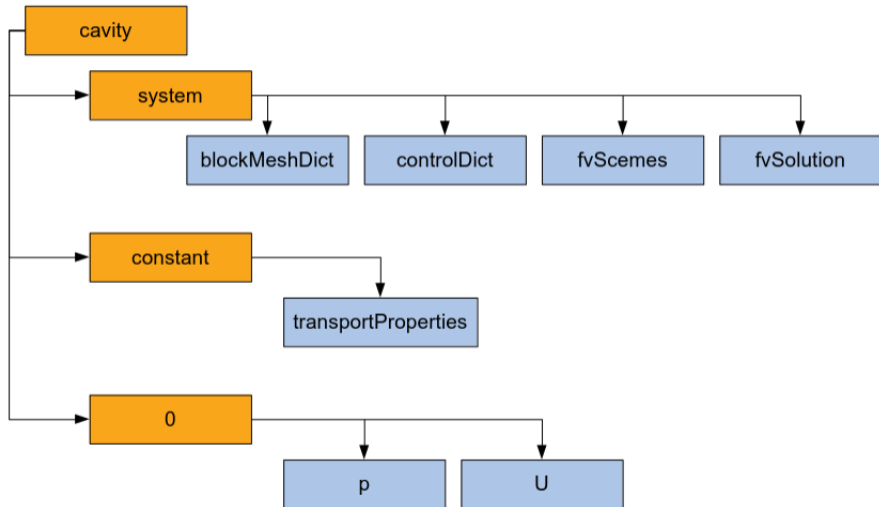


$$b/h = 1$$

Ван-Дайк М. Альбом течений жидкости и газа. М. : Мир, 1986. 184 с.



Подготовка файлов задачи



Заголовок словаря



```
1 /*----- C++ -----*/
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: v1606+ |
5 | \ \ / A n d | Web: www.OpenFOAM.com |
6 | \ \ M a n i p u l a t i o n | |
7 /*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        blockMeshDict;
14 }
15 // * * * * * //
```

Подготовка конечно-объёмной сетки



```
blockMeshDict 31 blocks
32 (
33     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34 );

17 scale 0.1;
18
19 vertices
20 (
21     (0 0 0)
22     (1 0 0)
23     (1 1 0)
24     (0 1 0)
25     (0 0 0.1)
26     (1 0 0.1)
27     (1 1 0.1)
28     (0 1 0.1)
29 );

36 edges
37 (
38 );
39
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;

53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
```

Постановка начальных и граничных условий



```
0/p 17 dimensions      [0 2 -2 0 0 0 0];
    18
    19 internalField  uniform 0;
    20
    21 boundaryField
    22 {
    23     movingWall
    24     {
    25         type          zeroGradient;
    26     }
    27
    28     fixedWalls
    29     {
    30         type          zeroGradient;
    31     }
    32
    33     frontAndBack
    34     {
    35         type          empty;
    36     }
    37 }
```

Постановка начальных и граничных условий



0/U

```
17 dimensions      [0 1 -1 0 0 0 0];
18
19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     movingWall
24     {
25         type      fixedValue;
26         value      uniform (1 0 0);
27     }
28
29     fixedWalls
30     {
31         type      noSlip;
32     }
33
34     frontAndBack
35     {
36         type      empty;
37     }
38 }
```



Контроль размерностей

Единицы измерения (СИ):

[килограмм – метр – секунда – кельвин – моль – ампер – кандела]

Пример для задания скорости:

...

```
dimensions [0 1 -1 0 0 0 0];
```

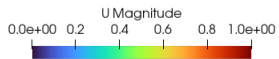
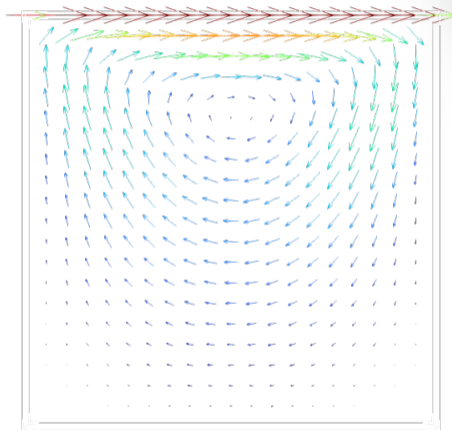
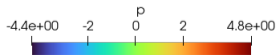
...

Управление расчётом



```
controlDict 17 application      icoFoam ;
              18
              19 startFrom      startTime ;
              20
              21 startTime      0 ;
              22
              23 stopAt        endTime ;
              24
              25 endTime       0.5 ;
              26
              27 deltaT        0.005 ;
              28
              29 writeControl   timeStep ;
              30
              31 writeInterval  20 ;
              32
              33 purgeWrite     0 ;
              34
              35 writeFormat    ascii ;
              36
              37 writePrecision 6 ;
              38
              39 writeCompression off ;
              40
              41 timeFormat     general ;
              42
              43 timePrecision  6 ;
              44
              45 runTimeModifiable true ;
```

Некоторые результаты



Некоторые результаты

