

- [Home](#)
- [News](#)
- [Forums](#)
- [Wiki](#)
- [Links](#)
- [Jobs](#)
- [Books](#)
- [Events](#)
- [Tools](#)
- [Feeds](#)
- [About](#)
- [Search](#)

[Home](#) > [Forums](#) > [Software User Forums](#) > [OpenFOAM](#)

Differences in solution method for pisoFoam and buoyantBoussinesqPisoFoam

User Name ☒ Remember Me
Password

[REGISTER](#)[BLOGS](#) ▼[COMMUNITY](#) ▼[NEW POSTS](#) ▼[UPDATED THREADS](#) ▼[SEARCH](#) ▼ **11** Likes[LINKBACK](#) ▼[THREAD TOOLS](#) ▼[SEARCH THIS THREAD](#) ▼[DISPLAY MODES](#) ▼ December 30, 2009, 18:56 **Differences in solution method for pisoFoam and buoyantBoussinesqPisoFoam**

#1

[mchurchf](#)

Member

Matthew J. Churchfield

Join Date: Nov 2009

Location: Boulder,
Colorado, USA

Posts: 49

Rep Power: 15



To whom can help,

I am trying to reconcile some differences between pisoFoam and buoyantBoussinesqPisoFoam.

In the UEqn.H file of buoyantBoussinesqPisoFoam the following equation is set up to predict velocity:

UEqn

==

```
fvc::reconstruct((fvc::interpolate(rhok)*(g & mesh.Sf()) -  
fvc::snGrad(p)*mesh.magSf()))
```

However, in pisoFoam, this is the equation:

UEqn == **-fvc::grad(p)**

Aside from the inclusion of the gravity term in buoyantBoussinesqPisoFoam, why are the face values used to reconstruct the cell centered values, whereas in pisoFoam the cell center values are used directly?

The same situation occurs in setting up the pressure equation. In buoyantBoussinesqPisoFoam, the equation is:

```
volScalarField rUA("rUA", 1.0/UEqn.A());  
surfaceScalarField rUAF("(1|A(U))", fvc::interpolate(rUA));  
fvm::laplacian(rUAF, p) == fvc::div(phi)
```

whereas in pisoFoam, it is:

```
volScalarField rUA = 1.0/UEqn.A();  
fvm::laplacian(rUA, p) == fvc::div(phi)
```

buoyantBoussinesqPisoFoam is solving for pressure on the faces, and pisoFoam is solving for cell centered pressure. Does this make a difference? Why are the two codes different in this manner?

Thank you

[rajibroy](#) likes this.



January 10, 2010, 18:27

#2

[alberto](#)

Senior Member

Alberto Passalacqua

Join Date: Mar 2009

Location: Ames, Iowa,
United States

Posts: 1,912

Rep Power: 32



Hi,

in pisoFoam you have the standard implementation, in buoyantBoussinesqPisoFoam the solution algorithm is modified as follows:

- you reconstruct the gravity and the pressure gradient contributions from the corresponding contribution to the flux
- you solve a "pseudo-staggered" version of the pressure equation
- you correct the flux
- you obtain the velocity correction reconstructing from the flux again (remember the flux is always continuous)

This technique tries to mimic a staggered grid arrangement. It is applied to the gravity term too, since it is included in the pressure equation.

Best,
Alberto

[kaifu](#), [MPJ](#), [mechy](#) and [6 others](#) like this.

Alberto Passalacqua

[GeekoCFD](#) - A free distribution based on openSUSE 64 bit with CFD tools, including OpenFOAM. Available as in both physical and virtual formats (current status:

<http://albertopassalacqua.com/?p=1541>)

[OpenQBMM](#) - An open-source implementation of quadrature-based moment methods.

To obtain more accurate answers, please specify the version of OpenFOAM you are using.



Last edited by alberto; January 10, 2010 at 18:28. Reason: removed quote



March 12, 2012, 07:12

#3

[samiam1000](#)

Senior Member

Samuele Z

Join Date: Oct 2009

Location: Mozzate - Co -
Italy

Posts: 519

Quote:

Originally Posted by **alberto**

Hi,

in pisoFoam you have the standard implementation

Rep Power: 15

Dear Alberto,

I am trying to understand the different solvers. So, for simpleFoam I've found [this wiki](#).. As far as the pisoFoam is concerned, you wrote that it present the standard implementation.
What does this mean? Is there a reference page?

Thanks a lot,
Samuele



March 12, 2012, 07:53

#4

[alberto](#)

Senior Member

Alberto Passalacqua

Join Date: Mar 2009

Location: Ames, Iowa,
United States

Posts: 1,912

Rep Power: 32



Quote:

Originally Posted by **samiam1000**

Dear Alberto,

I am trying to understand the different solvers. So, for simpleFoam I've found [this wiki](#)..

As far as the pisoFoam is concerned, you wrote that it present the standard implementation.

What does this mean? Is there a reference page?

*Thanks a lot,
Samuele*

Hi Samuele,

unfortunately I don't think there is a reference page. In pisoFoam you have the standard PISO algorithm you find in books, without body force term and without any particular treatment, except the Rhie-Chow interpolation. You can take a look at the icoFoam page on the wiki and you'll see many similarities.

My statement has to be read in the context of the comparison between the two solvers in the topic, where buoyantBoussinesqPisoFoam uses flux reconstruction to improve the solution procedure when body force terms are included.

Best,

Alberto Passalacqua

[GeekoCFD](#) - A free distribution based on openSUSE 64 bit with CFD tools, including OpenFOAM. Available as in both physical and virtual formats (current status:

<http://albertopassalacqua.com/?p=1541>)

[OpenQBMM](#) - An open-source implementation of quadrature-based moment methods.

To obtain more accurate answers, please specify the version of OpenFOAM you are using.



March 12, 2012, 07:57

#5

[samiam1000](#)

Senior Member

Samuele Z

Join Date: Oct 2009

Location: Mozzate - Co -
Italy

Posts: 519

Rep Power: 15

That's great, thanks.

And what about the pimpleFoam solver? Do you know which solver is embedded in such a solver?

Thanks a lot,
Samuele



March 12, 2012, 08:00

#6

[alberto](#)

Senior Member

Alberto Passalacqua

Join Date: Mar 2009

Location: Ames, Iowa,
United States

Posts: 1,912

Rep Power: 32



The "pimple" solvers use a "combination of PISO and SIMPLE", which is not that far from the flavors you find in other codes with different names (unsteady SIMPLE, iterative PISO, depending on the creativity of the authors :-)).

In short, it is an iterative solution method with sub-iterations over the set of equations to improve the robustness of the algorithm using under-relaxation, and to speed-up transient simulations or perform pseudo-transient simulations.

Best,

[samiam1000](#) likes this.

Alberto Passalacqua

[GeekoCFD](#) - A free distribution based on openSUSE 64 bit with CFD tools, including OpenFOAM. Available as in both physical and virtual formats (current status:

<http://albertopassalacqua.com/?p=1541>).

[OpenQBMM](#) - An open-source implementation of quadrature-based moment methods.

To obtain more accurate answers, please specify the version of OpenFOAM you are using.



September 11,
2013, 00:24

#7

[sharonyue](#)

Senior Member

Dongyue Li

Join Date: Jun

2012

Location:

Beijing, China

Posts: 776

Rep Power: 13



Quote:

Originally Posted by **alberto**

Hi,

in pisoFoam you have the standard implementation, in buoyantBoussinesqPisoFoam the solution algorithm is modified as follows:

- *you reconstruct the gravity and the pressure gradient contributions from the corresponding contribution to the flux*
- *you solve a "pseudo-staggered" version of the pressure equation*
- *you correct the flux*
- *you obtain the velocity correction reconstructing from the flux again (remember the flux is always continuous)*

This technique tries to mimic a staggered grid arrangement. It is applied to the gravity term too, since it is included in the pressure equation.

*Best,
Alberto*

Dear Alberto,

Can I say that I have to reconstruct it when there is a body force? Is it a must?

for example, in interFoam's UEqn:

Code:

```

solve
(
    UEqn
    ==
    fvc::reconstruct
    (
        (
            fvc::interpolate(interface.sigmaK())*fvc::snGrad(alpha1)
            - ghf*fvc::snGrad(rho)
            - fvc::snGrad(p_rgh)
        ) * mesh.magSf()
    )
);

```

Can I code it like:

Code:

```

solve
(
    UEqn
    ==
        interface.sigmaK()*fvc::grad(alpha1)
        - ghf*fvc::grad(rho)
        - fvc::grad(p_rgh)
);

```



« [Previous Thread](#) | [Next Thread](#) »

Posting Rules



You **may not** post new threads
 You **may not** post replies
 You **may not** post attachments
 You **may not** edit your posts

[BB code](#) is **On**
[Smilies](#) are **On**
[\[IMG\]](#) code is **On**
 HTML code is **Off**
[Trackbacks](#) are **Off**
[Pingbacks](#) are **On**
[Refbacks](#) are **On**

[Forum Rules](#)

Similar Threads



Thread	Thread Starter	Forum	Replies	Last Post
smoothSolver diverges - solution in using PBiCG solver?	makaveli_lcf	OpenFOAM Running, Solving & CFD	3	September 11, 2013 13:44
Solution Method & turbulent intensity in CFX 5.5.1	hamza albazzaz	CFX	7	June 29, 2011 13:39
Question about meshing / solution scheme of CFX	Coriolius	CFX	8	August 1, 2004 19:39

Similar Threads



FVM,FDM AND FEM	d	Main CFD Forum	4	May 30, 2003 04:19
Trouble w. tri. FV method and lam. backstep flow	Steve Reuss	Main CFD Forum	12	December 26, 2001 12:47

All times are GMT -4. The time now is 23:49.

[Contact Us](#) - [CFD Online](#) - [Privacy Statement](#) - [Top](#)



© CFD Online _