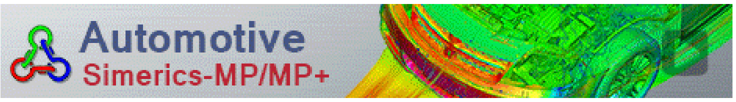
CFD Online
www.cfd-online.com



- Home
- News
- Forums
- Wiki
- Links
- Jobs
- Books
- Events
- Tools
- Feeds
- About
- Search

Home > Forums > Software User Forums > OpenFOAM > **OpenFOAM Running, Solving & CFD**

I need explanations about fixedFluxPressure

User Name User Name ☑ Remember Me
Password Log in

Register Blogs Community New Posts Updated Threads Search

133 Likes

Post Reply

LinkBack Thread Tools Search this Thread Display Modes

November 30, 2010, 05:15 **I need explanations about fixedFluxPressure** #1

Cyp
Senior Member

Cyprien
Join Date: Feb 2010
Location: Stanford University
Posts: 298
Rep Power: 14

Hello!

I had a look at the fixedFluxPressure boundary condition and I am not quite sure to well understand what this BC does.

I guess I have to use this BC if I want to evaluate the pressure according to a velocity input. Am I wrong ?

Furthermore, in the code source, one can read:

Code:

```
        gradient() = (phip - (patch().Sf() & Up))/patch().magSf()/rAp;
```

I do not succeed in linking this snippet with a mathematical formula...

All enlightments are welcome!
Regards,
Cyp

JR22, Guimloute, amuzeshi and 1 others like this.

Quote

August 19, 2013, 21:36 **buoyantPressure vs. fixedFluxPressure** #2

JR22
Senior Member

Jose Rey

I realize this is an old thread, but it is still very relevant.

I was having terrible problems trying to get buoyantBoussinesqSimpleFoam to converge. I changed the p_rgh BC's from buoyantPressure to fixedFluxPressure and the model started to converge.

Anybody?

Guimloute and amuzeshi like this.

Join Date: Oct 2012
Posts: 134
Rep Power: 14

Quote

August 1, 2014, 05:42 #3

Sherlock_1812
Senior Member

Srivathsan N
Join Date: Jan 2013
Location: India
Posts: 101
Rep Power: 9

Same question here. What is the difference between buoyantPressure and fixedFLuxPressure boundary conditions?

____

Regards,

Srivaths

Quote

October 16, 2014, 03:23

OpenFAOM 2.3.0 doesn't have buoyantPressure BC #4

zandi
Senior Member

Fatema Zandi Goharrizi
Join Date: Mar 2009
Posts: 158
Rep Power: 13

Hi
OpenFAOM version 2.3.0 doesn't have buoyantPressure libraries. Am I right?
what is the difference between buoyantPressure and fixedFluxPressure?
why fixedFluxPressure converge better? is it the only difference between them?
Please help☹

Quote

October 16, 2014, 04:46 #5

fabian_roesler
Senior Member

Fabian Roesler
Join Date: Mar 2009
Location: Germany
Posts: 212
Rep Power: 14

OpenFOAM 2.2.2 has both boundary conditions. In newer Version, buoyantPressure boundary was dismissed. For clarification have a look into the code or into Doxygen.

From buoyantPressureFvPatchScalarField we learn that:
This boundary condition sets the pressure gradient appropriately for buoyant flow. If the variable name is one of pd, p_rgh or ph_rgh, we assume that the pressure variable is $p - \rho(g \cdot h)$ and the gradient set using:

$$\nabla(p) = -\nabla_\perp(\rho)(g \cdot h)$$

Otherwise we assume that it is the static pressure, and the gradient calculated using:

$$\nabla(p) = \rho(g \cdot n)$$

From fixedFluxPressureFvPatchScalarField we see that:
This boundary condition adjusts the pressure gradient such that the flux on the boundary is that specified by the velocity boundary condition.

The predicted flux to be compensated by the pressure gradient is evaluated as ⚠ Invalid Equation, both of which are looked-up from the database, as is the pressure diffusivity used to calculate the gradient using:

$$\nabla(p) = \frac{\phi_{H/A} - \phi}{|Sf|D_p}$$

haghajani, wyldckat, zandi and 28 others like this.

Quote

October 16, 2014, 05:21 #6

fabian_roesler
Senior Member

Fabian Roesler
Join Date: Mar 2009
Location: Germany
Posts: 212
Rep Power: 14

Some additional remarks:

The fixedFluxPressure boundary is known do have a better convergence. This is in my opinion due to the more pressure related calculation of the gradient:

$$\nabla(p) = \frac{\phi_{H/A} - \phi}{|Sf| D_p}$$

phi and phiHbyA (predicted flux field) are directly linked to pressure equation. On a wall, the flux difference tends to zero and so the boundary condition turns to zeroGradient.
The buoyanPressure boundary is more the physical method to describe the pressure gradient with Archimedes' principle.
I hope my explanation is understandable and correct.

Cheers

Fabian

haghajani, wyldckat, zandi and 27 others like this.

Quote

November 5, 2014, 17:06 #7

santiagomarquezd
Senior Member

Santiago Marquez Damian
Join Date: Aug 2009
Location: Santa Fe, Santa Fe, Argentina
Posts: 452
Rep Power: 20

Hi folks, I'm dealing with fixedFluxPressure BCs also and this thread was helpful. Checking the code things have changed at least in FOAM 2.3.0. The class definition is now more simple

fixedFluxPressureFvPatchScalarField.C

Code:

```
125 void Foam::fixedFluxPressureFvPatchScalarField::updateCoeffs
126 (
127     const scalarField& snGradp
128 )
129 {
130     if (updated())
131     {
132         return;
133     }
134 
135     curTimeIndex_ = this->db().time().timeIndex();
136 
137     gradient() = snGradp;
138     fixedGradientFvPatchScalarField::updateCoeffs();
139 }
```

it only sets the proper gradient in line 137 and then calls the updateCoeffs() method from the fixedGradientFvPatchScalarField class (from which the present class inherits). This requires to set the value of snGradp which is done at solver level. For example, from pEqn.H of interFoam we have:

Code:

```
    // Update the fixedFluxPressure BCs to ensure flux consistency
    setSnGrad<fixedFluxPressureFvPatchScalarField>
    (
        p_rgh.boundaryField(),
        (
            phiHbyA.boundaryField()
          - (mesh.Sf().boundaryField() & U.boundaryField())
        )/(mesh.magSf().boundaryField()*rAUf.boundaryField())
    );
```

From the new code the gradient reads:

$$\vec{\nabla} p = \left(\vec{H}/a_P \cdot \vec{S}_f - \vec{U} \cdot \vec{S}_f\right) \frac{(a_P)_f}{||\vec{S}_f||}$$

Hope this helps for 2.3.0 users.

Regards.

wikstrom, fabian_roesler, haghajani and 23 others like this.

____

Santiago MÁRQUEZ DAMIÁN, Ph.D.
Research Scientist
Research Center for Computational Methods (CIMEC) - CONICET/UNL
Tel: 54-342-4511594 Int. 7032
Colectora Ruta Nac. 168 / Paraje El Pozo
(3000) Santa Fe - Argentina.
http://www.cimec.org.ar

Last edited by santiagomarquezd; November 7, 2014 at 06:25. Reason: Add equation

Quote

April 14, 2015, 17:46 #8

SSSS
Senior Member

anonymous
Join Date: Aug 2014
Posts: 205
Rep Power: 9

I would like to comment out a bit, why does OpenFOAM use the expression given by **santiagomarquezd** (gracias por el análisis del código fuente) for the pressure surface normal gradient in the boundaries.

First of all we need to write down the momentum equation discretized using the Rhie-Chow interpolation method. This reads:

$$\vec{u}_f = \left(\frac{\vec{h}}{a_p}\right)_f - \left(\frac{1}{a_p}\right)_f \nabla p_m$$

Where the f subscript means interpolate to the face, p_m is the variable p_rgh whose value is $p_m = p - \rho\vec{g}\cdot\vec{h}$, and the vector u is the velocity. Now multiply the equation by the face surface vector $\vec{S}_f = ||S_f||\cdot\vec{n}$ where the vector n is the surface normal:

$$\vec{u}_f \cdot \vec{S}_f = \left(\frac{\vec{h}}{a_p}\right)_f \cdot \vec{S}_f - \left(\frac{1}{a_p}\right)_f \nabla p_m \cdot \vec{S}_f$$

Thus the following equation for the pressure surface gradient $\nabla p_m \cdot \vec{n}$ can be obtained:

$$\nabla p_m \cdot \vec{n} = \left(\left(\frac{\vec{h}}{a_p}\right)_f \cdot \vec{S}_f - \vec{u}_f \cdot \vec{S}_f\right)\frac{a_p|_f}{||S_f||}$$

Hope this is useful for you foamers

wikstrom, haghajani, zandi and 27 others like this.

Quote

October 14, 2015, 14:55 #9

angelmonsalve
New Member

Angel Monsalve
Join Date: Oct 2009
Posts: 6
Rep Power: 13

Thanks for the explanation!!

Quote

August 31, 2016, 15:35 **value uniform 0;** #10

Dear all, thank you for the above discussion

JonW
Member

Jon Elvar Wallevik
Join Date: Nov 2010
Location: Reykjavik, ICELAND
Posts: 94
Rep Power: 16

There is one thing I don't understand fully. In the damBreak example for interFoam (say of OF 4.0) the b.c. for p_rgh is given as...

```
leftWall
{
type fixedFluxPressure;
value uniform 0;
}
```

so the question is, what does "value uniform 0;" stand for?

There is one constructor in fixedFluxPressureFvPatchScalarField.C which has 3 arguments:

```
Foam::fixedFluxPressureFvPatchScalarField::fixedFl uxPressureFvPatchScalarField
(
const fvPatch& p,
const DimensionedField<scalar, volMesh>& iF,
const dictionary& dict
)
```

... and in that function body, there is

```
if (dict.found("value") && dict.found("gradient"))
```

But in interFoam, this constructor is not used, since it is the constructor with two arguments that is used (actually the first constructor), c.f. in pEqn.H

```
setSnGrad<fixedFluxPressureFvPatchScalarField>
(
p_rgh.boundaryField(),
(
phiHbyA.boundaryField()
- fvOptions.relative(mesh.Sf().boundaryField() & U.boundaryField())
)/(mesh.magSf().boundaryField()*rAUf.boundaryField() )
);
```

So am I understanding this correctly: When using interFoam, then the "value uniform 0;" is actually not used?

Any comment would be helpful
J.

Quote

November 23, 2016, 18:45 #11

decah
Member

Declan
Join Date: Oct 2016
Location: Ireland
Posts: 40
Rep Power: 6

Hi Jon,

I had a similar question about flowRateInletVelocity which requires an input value that doesn't appear to do anything. As explained by Roman in the below thread some derived boundary conditions necessarily inherit a placeholder value like this one because of their basic structure.

http://www.cfd-online.com/Forums/ope...tvelocity.html

I think the value uniform 0; you asked about is like this and could be assigned any value you like without affecting your simulation.

JonW, amolrajan, mizzou and 1 others like this.

Quote

January 18, 2017, 05:46 #12

nw_ds
New Member

Mido
Join Date: Mar 2011
Posts: 24
Rep Power: 11

So the bottom line, The difference between this bc and zeroGradient is in term of the convergence (HOW MANY ITER TO SOLVE THE PRESSURE EQN) not the accuracy or whether the solver will converge or not.

October 11, 2017, 12:22 #13

beh_zadi
New Member

Join Date: Jan 2016
Posts: 3
Rep Power: 6
🟩

Hi openfoam users,

I'm new to openfoam and I need your help about fixedFluxPressure BC. I have made a new solver derived from icoFoam and I didn't find the following code in it while it exists in interFoam.
I specified fixedValue BC for velocity field and fixedFluxPressure for pressure field. Does it need to add the following code in new solver to ensure flux consistency or not?

Code:

```
// Update the fixedFluxPressure BCs to ensure flux consistency
setSnGrad<fixedFluxPressureFvPatchScalarField>
(
    p_rgh.boundaryField(),
    (
        phiHbyA.boundaryField()
      - (mesh.Sf().boundaryField() & U.boundaryField())
    )/(mesh.magSf().boundaryField()*rAUf.boundaryField())
);
```

Any comment would be helpful

December 14, 2017, 09:58 #14

roenby
Member

Johan Roenby
Join Date: May 2011
Location: Denmark
Posts: 89
Rep Power: 17
🟩

Just for the record:

If you are using a newer version of OpenFOAM (e.g. 5.x), you will find that the referred code snippet from interFoam's pEqn.H file has been replaced by the function call:

constrainPressure(p_rgh, U, phiHbyA, rAUf, MRF);

This is also the case for icoFoam.

In the source code for this function, you'll find a similar code snippet:

https://github.com/OpenFOAM/OpenFOAM...Pressure.C#L34

Best,
Johan

Andrea1984, tiam, amolrajan and 3 others like this.

April 30, 2020, 09:21 #15

Andrea1984
Senior Member

Andrea
Join Date: Feb 2012
Location: Leeds, UK
Posts: 168
Rep Power: 12
🟩

Apologies for digging up and old thread.

I have noticed that the call to constrainPressure does not take place in the Eulerian multiphase solvers (I am using OpenFOAM 6.0). I take this means that zeroGradient and fixedFluxPressure are equivalent for this class of solvers. Would be useful is some can confirm this.

Cheers,
Andrea

April 30, 2020, 10:00 #16

Ok so to answer my previous question, in multiphase Eulerian solvers this is done within pEqn without

Andrea1984
Senior Member

Andrea
Join Date: Feb 2012
Location: Leeds, UK
Posts: 168
Rep Power: 12

calling the constrainPressure function (probably because this is only suitable for one-fluid solvers).

In twoPhaseEulerFoam's pEqn.H this is done by

Code:

```
// Update the fixedFluxPressure BCs to ensure flux consistency
setSnGrad<fixedFluxPressureFvPatchScalarField>
(
    p_rgh.boundaryFieldRef(),
    (
        phiHbyA.boundaryField()
      - (
            alphaf1.boundaryField()*phi1.boundaryField()
          + alphaf2.boundaryField()*phi2.boundaryField()
        )
    )/(mesh.magSf().boundaryField()*rAUf.boundaryField())
);
```

The same is done in the pressure equation of the reactingEulerFoam family.
So to conclude the answer to my question in the previous post is no: fixedFluxPressure and zeroGradient are NOT the same thing in multiphase Eulerian solvers.

Quote

Post Reply

« 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
Rotating objects	Marcus Gellert (Gellert)	OpenFOAM Running, Solving & CFD	17	December 14, 2010 16:06
muSgsWallFunctionFvPatchField, explanations or bug ?	fgal	OpenFOAM Bugs	0	July 22, 2010 15:19
FixedFluxPressure cannot find field 1%7cAU	anger	OpenFOAM Running, Solving & CFD	3	November 10, 2008 06:50
New to CFX-need explanations about mesh parameters	Cyril	CFX	3	November 24, 2006 07:33
Rotor/stator tutorial, and how to...	gilberto	CFX	5	January 21, 2002 10:41

All times are GMT -4. The time now is 09:32.

Contact Us - **CFD Online** - **Privacy Statement** - **Top**

twitter facebook