PROJECT 1

OpenFoam Development

Om Mihani

10/05/2022

Project Description:
Use the icoFoam solver in the applications/solvers/incompressible/icoFoam directory to create your own solver - scalarFoam.

The scalar transport equation is defined as:

$$\frac{\partial s}{\partial s} + \nabla \cdot (\mathbf{u}s) = 0,$$

where s is the scalar quantity being transported by the velocity u.

Changes in the solver:
• Replace the main working equation with
• What change will you make in the createFields.H file (Hint: the p field is originally created here)
• What change will you make in the UEqn.H file? (Hint: replace the solve function)
• Compile scalarFoam using wmake. How are the Make/files modified?

Changes in the case:
• Use the $FOAM_TUTORIALS/incompressible/icoFoam/cavity case to test out your solver. Rename it scalarCavity.
• The scalar transport equation has no pressure p to solve, instead a scalar s. What changes will you make in the 0 folder?
• You will have to change the interpolation scheme in fvSchemes to account for the ∇ · (us) term. How will you do this?
• In fvSolutions, a solver has to be specified for the quantity s. Can you just replace this with the solver used for p?
• Run this case by making the appropriate change in controlDict

---

Please turn to the next page for the solution

SOLUTION

Code: [GitHub link]

Report:

Target

To develop our own solver using the icoFoam template.

Procedure:

1) Copy the aforementioned solver into your user directory in the folder application.
2) Rename the solver to scalarFoam
3) Now open the folder and rename icoFoam.C to scalarFoam.C
4) Open the Make/file and change the .C file name and executable name to scalarFoam
5) Change the address to FOAM_USER_APPBIN (Note: This step is prone to mistakes)
6) Open the scalarFoam.C file and do the following changes in the given order:
    a) Covert PISO loop to a SIMPLE loop since it is much easier to deal with
    b) We do not need a p corrector equation anymore so comment out that entire region.
    c) Here, our working variable is s, not U, so rename the equation as sEqn instead of the UEqn.
    d) Also, phi is U not s. So the sEqn changes to the following:
       fvm::ddt(s) + fvm::div(phi,s)
    e) We do not need any fvOptions file here, so do changes accordingly.
    f) Save and exit
7) Open the createFields.H and do the following changes:
    a) We don't need to read transport properties anymore. Remove that section
    b) Similarly, remove nu
    c) Change p to s
8) Now execute wmake. The executable is saved in the bin of the user directory.
9) Now copy the case from the given directory
10) Take only the cavity and delete the other folders. Rename it to scalarCavity
11) Open 0 folder:
    a) Rename p as s
    b) S can be of any dimensions. I have considered metres
    c) Set an initial value and boundary conditions. I have used the original case.
    d) U file is unaltered
12) Open constant folder:
    a) Delete the transportProperties file as we do not need it
13) Open the system folder:
    a) No change in the blockMeshDict folder
    b) fvSchemes:
        i) We do not need a laplacian and grad but the program doesn't work without them.

ii) Change the div from (phi, U) to (phi, s).
iii) Let other things be identical

c) fvSolution:
   i) Change from PISO to SIMPLE. Make the other changes likewise.
   ii) Remove the p_final parameter
   iii) Change from p to s
   iv) Change the solver for s to PBiCGStab and tol to 0

d) controlDict:
   i) Change solver name to scalarFoam and endTime to 1s

14) Run blockMesh
15) Run scalarFoam
16) The code converges pretty quickly since in the initial few steps itself, the DE is satisfied.

Results:

It is observed that in the initial few iterations itself, the differential equation is satisfied and the code runs pretty fast. Let's see how.

We had set initial condition as uniform s = 2. Let's substitute in the differential equation.

$$given: \frac{\partial s}{\partial t} + \nabla.(\mathbf{U}s) = 0$$

$$\implies \frac{\partial 2}{\partial t} + \nabla.(2\mathbf{U}) = 0$$

$$\implies \nabla.(\mathbf{U}) = 0$$

which is essentially the mass conservation equation

The condition of zero divergence for U can be achieved quite quickly. When visualised in paraView, the figures look like this:

	
Plot for s	Plot for U

Following are some notes on procedure which describe the details of the challenges faced and the methods used to tackle them:

Notes on procedure:

1. *While copying, the complete path needs to be specified. This can be done by going to the particular solver and using the pwd command to get the path.*
2. *Copy the icoFoam solver in the applications folder and name it scalarFoam.*
3. *Make changes in the Make/files*
    a. *Change .C file to scalarFoam*
    b. *Change the path to FOAM_USER_APPBIN*
    c. *Change the executable name to scalarFoam*
4. *Now we need to change the main equation. Considerations:*
    a. *Piso Loop is no longer needed*
    b. *UEqn should give me fvm::ddt(s) + fvm::div(phi,U)*
    c. *This should be equated to zero*
    d. *Rename icoFoam.C to scalarFoam.C*
5. *Now we move on to createFields.H*
    a. *We are not using nu. So comment it*
    b. *Change p to s*
6. *Note that since we copied an icoFoam solver, there is no UEqn.H file created. The role of the UEqn file is done by the .C file itself.*
7. *There was an error in using the PISO loop. Trying to get some motivation from a SIMPLE loop and laplacianFoam*
8. *Observe that since s is a scalar, ddt of s won't give a vector matrix*
9. *Also, since a PISO loop is not required, using the SIMPLE loop of LaplacianFoam*
    a. *Used simpleControl.H instead of pisoControl.H*
    b. *While loop based on simple parameters instead of PISO parameters*
    c. *Name UEqn changed to sEqn*
10. *Even the changes in the createFields.H now are copied from laplacianFoam:*
    a. *volScalarField s is read instead of transport properties*
    b. *Instance changed to runTime.timeName()*
    c. *Read changed to MUST_READ and write to AUTO_WRITE*
    d. *nu field is commented out*
    e. *sRefCell and all related commands are commented out*

Even after doing this and a few other combinations, I was unable to form the correct .C file
So, I discarded the entire folder and started over. *Turn to next page for notes*

Attempt #2

1. *Try and take motivation from the scalar Transport foam. Observe that in this case, phi is U and not s in the TEqn (and not UEqn)*
2. *While copying, the complete path needs to be specified. This can be done by going to the particular solver and using the pwd command to get the path.*
3. *Copy the icoFoam solver in the applications folder and name it scalarFoam.*
4. *Change the icoFoam.C file name to scalarFoam.C*
5. *In the Make/files:*
    a. *Change .C File name*
    b. *Change the address to FOAM_USER_APPBIN*
    c. *Change the executable name to scalarFoam*
6. *In the scalarFoam.C file:*
    a. *Switch from piso to Simple*
    b. *Switch from fvVectorMatric to fvScalarMatrix*
    c. *Remove the pCorrecter loop*
7. *In the CreateFields.H file:*
    a. *Change p to s*
    b. *Do not read transport properties*
    c. *Remove nu*
8. *Now try wmake*
9. *It worked …*

Now trying to edit the case file:

1. *Copy the case from the above mentioned path*
2. *Delete the other two cavity cases. Rename p to s in the zero folder*
3. *S can have any dimensions. Let's assume it has dimensions of length. Change the file accordingly*
4. *No change in the U file*
5. *In the constant file, we don't need anything. So delete transportProperties file*
6. *In the system folder:*
    a. *blockMeshDict: No change*
    b. *fvSchemes:*
        i. *We don't need laplacian. So comment it out*
        ii. *We don't need a grad scheme as well. So lets comment it out*
        iii. *Let ddt scheme be Euler only*
        iv. *divSchemes : Let's move ahead with Gauss Linear only, but for div(phi,s) not div(phi,U)*
    c. *fvSolution:*
        i. *Change from PISO to Simple: Change arguments likewise*
        ii. *Not changing anything for U solver*
        iii. *For s, using PBiCGStab (inspired from scalarTransportFoam)*
    d. *controlDict:*
        i. *Change solver to scalarFoam*
        ii. *Change endTime to 1 s*
7. *It flagged an error. Says grad scheme undefined. Let's put that back and retry*
8. *Same for laplacian.*
9. *It worked…*
10. *Solution: uniform value of s*