Instructional workshop on OpenFOAM programming
LECTURE # 4

Pavanakumar Mohanamuraly

April 26, 2014

Outline

ddt operator

- *fvc* and *fvm* versions of temporal differential operator exists
- *fvm* version works similar to the spatial *fvm* operators using *fvMatrix*
- FOAM stores previous solution information in $field.oldTime()$
- Euler time step in *fvm* is simply a diagonal solver $diag() = \frac{1}{\Delta t}$

$$\phi^{n+1} = \phi^n - \frac{1}{\Delta t} L[\phi^n] \tag{1}$$

ddt operator - dictionary

- Use the *ddtSchemes* subDict in *fvSchemes* dictionary to define scheme

```
ddtSchemes
{
  default Euler;
}
```

- Following *fvm* :: *ddt* schemes available in FOAM

Scheme	Description
`Euler`	First order, bounded, implicit
`localEuler`	Local-time step, first order, bounded, implicit
`CrankNicholson` ψ	Second order, bounded, implicit
`backward`	Second order, implicit
`steadyState`	Does not solve for time derivatives

Unsteady heat equation - without forcing

$$\frac{\partial \phi}{\partial t} - \kappa \frac{\partial^2 \phi}{\partial x^2} = 0 \quad 0 \leq x \leq L, \quad t \geq 0 \tag{2}$$

- Syntax for *fvm* :: *ddt* in combination with *fvm* :: *laplacian* slightly different
- Need to use the *solve* to combine both

```
/// Assumes zero RHS
solve( fvm::ddt(x) - fvm::laplacian(kappa , x) );
```

- Remember to make *kappa* consistent in dimensions

Hands on - Unsteady heat equation

- Using the *ddt* and *laplacian* operator write your own solver

Implementing Robins BC

- No need for two versions to be implemented *fvm* and *fvc*
- Need to read in three extra parameter ϕ', a and b

$$a\phi(0) + b\phi'(0) \; and/or \; a\phi(L) + b\phi'(L) \tag{3}$$

- This will introduce one extra *RHS* source term to the Dirichlet BC
- Makes sense to use the Dirichlet BC and modify it for Robin

Hands on - Setting up

- Copy the contents of *FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/fixedValue* to a folder named *MY_FOLDER/RobinBC*
- Rename all files having prefix *fixedValueFvPatchField* to *RobinFvPatchField*

```
for name in fixedValueFvPatchField*
do
    newname=RobinFvPatchField"$(echo "$name" | cut -
        c23-)"
    mv "$name" "$newname"
done
```

- Find and replace text *fixedValueFvPatchField* to *RobinFvPatchField* in all files

```
sed -i 's/fixedValueFvPatchField/RobinFvPatchField/g
    ' RobinFvPatchField*
```

Hands on - Compile code

- Create the *Make* folder with *files* and *options* as follows

files

```
RobinFvPatchFields.C

LIB = libRobinBC
```

options

```
EXE_INC = \
    -I$(LIB_SRC)/finiteVolume/lnInclude -g
EXE_LIBS = -lfiniteVolume
```

- *wmake libso* to create library *libRobinBC.so*

Hands on - Make changes

- Replace all *fixedValue* fields with *Robin*

```
sed -i 's/fixedValue/Robin/g' RobinFvPatchField*
```

- Runtime type information

```
TypeName("Robin");
```

- Runtime object selection

```
RobinFvPatchFields.C:37:makePatchFields(Robin);
RobinFvPatchFields.H:39:makePatchTypeFieldTypedefs(
    Robin)
RobinFvPatchFieldsFwd.H:40:
    makePatchTypeFieldTypedefs(Robin)
```

- *wclean* and
- *wmake libso* to create library *libRobinBC.so*

Hands on - More changes

- Make the *fixesValue*() boolean function return *false* in file *RobinFvPatchField.H*

```
virtual bool fixesValue() const
{
  return false;
}
```

Hands on - Preliminary testing I

- Go to the $1d$ case folder and add the following to *system/controlDict*

```
libs ("libRobinBC.so");
```

- Set the library environment search path to the *Make/linux* $*****$ folder (where the *libRobinBC.so* is created)
- Run the previous hands on example and check if you get errors
- If you get a warning as shown below

```
From function dlLibraryTable::open(const fileName&
    functionLibName)
in file db/dlLibraryTable/dlLibraryTable.C at line
    85
could not load dlopen(libRobinBC.so, 9): image not
    found
```

- Check your library path and see if the lib file exists

Hands on - Preliminary testing II

- In the fields file replace all *fixedValue* types to *Robin*
- Rerun the code and it should give the same results as run using *fixedValue*
- This ensures that the BC is compiled, loaded and setup correctly

End of Week 2 Day 2