OpenFOAM's basic solvers for linear systems of equations

Solvers, preconditioners, smoothers

What are we going to do?

- Look at the structure of lduMatrix
- Compare DIC/FDIC preconditioner
- Set up a multi grid solver

$FOAM_SRC/OpenFOAM/matrices/lduMatrix/

- **lduAddressing/**
- **lduMatrix/**
- **preconditioners/**
- **smoothers/**
- **solvers/**

$FOAM_SRC/OpenFOAM/matrices/lduMatrix/solvers

- **diagonalSolver** - diagonal solver for both symmetric and asymmetric problems
- **GAMG** - Geometric agglomerated algebraic multigrid solver
- **PBiCG** - Preconditioned bi-conjugate gradient solver for asymmetric lduMatrices using a run-time selectable preconditioner
- **PCG** - Preconditioned conjugate gradient solver for symmetric lduMatrices using a run-time selectable preconditiioner
- **smoothSolver** - Iterative solver using smoother for symmetric and asymmetric matrices which uses a run-time selected smoother

BICCG, ICCG obsolete

$FOAM_SRC/OpenFOAM/matrices/lduMatrix/solvers

For description of the other folders and more information see the report. There is also an example of implementing your own version of the PBiCG solver *myPBiCG*.

Preconditioner test

Copy cavity test case and change directory:

```
cp -r $FOAM_TUTORIALS/icoFoam/cavity $FOAM_RUN/cavity
cd $FOAM_RUN/cavity
```

Preconditioner test

Change blockMeshDict:

```
blocks
(
    hex (0 1 2 3 4 5 6 7) (150 150 1) simpleGrading (1 1 1)
);
```

Generate mesh

```
blockMesh
```

Preconditioner test

Change the controlDict to:

```
      endTime         0.4;
      deltaT          0.005;
      writeControl    timeStep;
      writeInterval   20;
```

and run icoFoam

```
icoFoam > logDIC &
```

Preconditioner test

Now changing DIC to FDIC in fvSolution file

```
      preconditioner   FDIC;
```

and running icoFoam

```
icoFoam > logFDIC &
```

Compare run-time (Execution time in log files), difference in percentage?

Generalised geometric-algebraic multi-grid

Look at some GAMG options

preconditioner	select preconditioner
mergeLevels	1/2/3, coarsen levels faster, for simple geometric grids;
smoother	select smoother (GaussSeidel recommended)
agglomerator	faceAreaPair (recommended) or algebraicPair;
nCellsInCoarsestLevel	number of cells the coarse level solver has to calculate (important!)

Table 1: GAMG options

Generalised geometric-algebraic multi-grid

Change the fvSolution to:

```
p GAMG
{
    preconditioner    FDIC;
    mergeLevels       1;
    smoother          GaussSeidel;
    agglomerator      faceAreaPair;
    nCellsInCoarsestLevel   100;
    tolerance         1e-05;
    relTol            0;
};
```

Generalised geometric-algebraic multi-grid

Change the fvSolution to:

```
U GAMG
{
    preconditioner    DILU;
    mergeLevels       1;
    smoother          GaussSeidel;
    agglomerator      faceAreaPair;
    nCellsInCoarsestLevel   100;
    tolerance         1e-05;
    relTol            0;
};
```

Generalised geometric-algebraic multi-grid

and running icoFoam

```
icoFoam > logGAMG &
```

Speed-up compared to same calculation with PCG/PBiCG: about 3 times faster

Thank you for your attention

Questions?