OpenFOAM Phd course 2009 - project

Deformable mesh in a two stroke engine

Outline

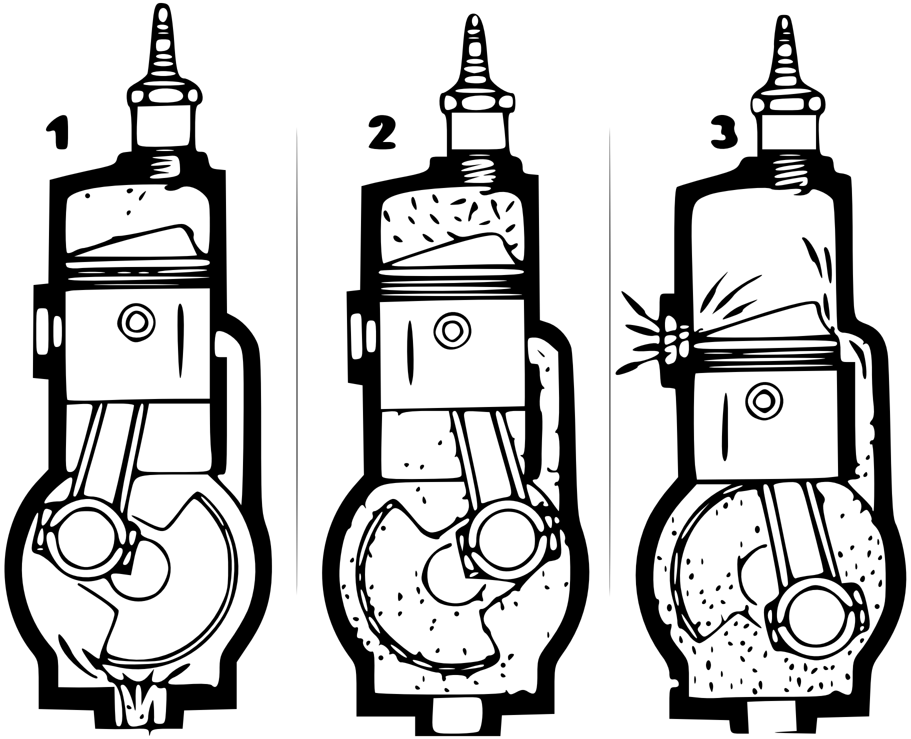
- Introduction.
- Problem.
- Mesh generation.
- Mesh motion.
- Next step.

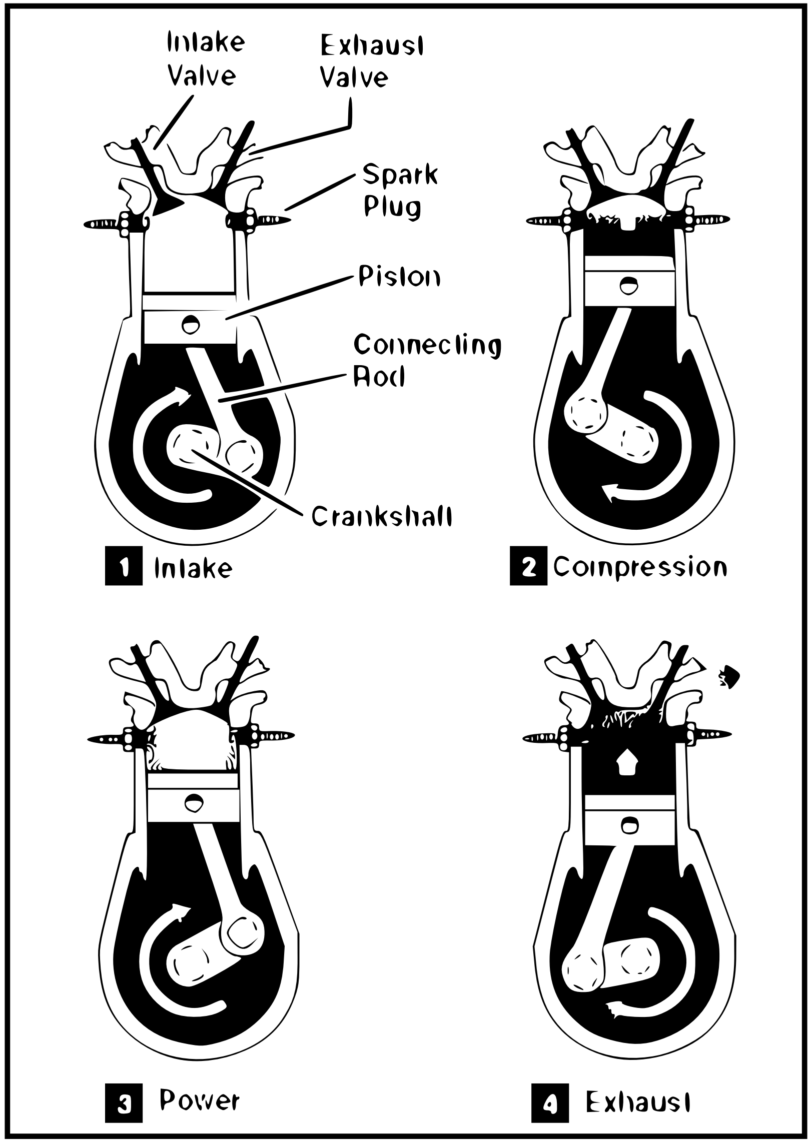
Introduction

Two stroke engine

- Four stroke vs two stroke
- Mesh challenges
- B.C. challenges
- Top dead center (TDC)
- Con rod
- Cylinder head
- Ports
- Liner

Two stroke vs. fourStroke





Deformable mesh

- Mesh shape changes throughout the simulation.
- Domain topology changes.
- Enlarge cells?
- Simple but less accurate. Large cells, skew cells and so on.

Dynamic mesh layering

- Points move. Cells grows larger or shrink.
- Split/add or remove cells.
- Rules

Create mesh

- Start with cylinder.
- Use `blockMesh` to create the cylinder. In this case a cube.
- Define patches.
- `mkdir caseTS` and we go there and copy three folders from the `jonssonmProject`-directory.

```
cp -r ../jonssonmProject/cylinderBm .
cp -r ../jonssonmProject/intPortBm .
cp -r ../jonssonmProject/exhPortBm .
```

- Have a look in `cylinderBm/constant/polyMesh/blockMeshDict`
- Based on `icoFoam`

Merge the meshes

- Run `blockMesh` and `checkMesh` on `cylinderBm`, `intPortBm` and `exhPortBm`.
- Now the three parts are to be merged into one, using the `mergeMeshes`-utility.

```
cp -r cylinderBm mergedBm
mergeMeshes . mergedBm . intPortBm
cd mergedBm
rm -r constant/polyMesh
cp -r 0.005/polyMesh constant/
rm -r 0.005
mergeMeshes . mergedBm . exhPortBm
```

- Now the mesh is put together. Run `checkMesh -time 0.005` .
- Then make a new directory `caseTS/compCase`. And move the the mesh there

  ```
  cd compCase
  cd mkdir constant
  cd cp -r ../mergedBm/0.005/polyMesh constant/
  ```

Coordinate system

- The piston must be centered by and move along the z-axis.
- First center the mesh in the current coordinate system so the origin is in the middle of the piston bowl.

  ```
  transformPoints -translate "(-0.015 0 -0.015)"
  ```
- The we have to roll the mesh.

  ```
  transformPoints -rollPitchYaw "(90 0 0)"
  ```
- Check the results using paraview or `checkMesh` .

Sets

- Two point sets, two cell sets and one face set is required to move the mesh.
- The `setSet`-utility is used.
- Creates a set from a dicitonary. In this case using some meshtools.
- Create a dictionary `vi makePointSet.setSet`

  ```
  pointSet pistonPoints new boxToPoint (-0.021 -0.021 -0.0001)/
  (0.021 0.021 0.0026)
  ```

  and in the same file

  ```
  pointSet headPoints new boxToPoint (-0.021 -0.021 0.0074)/
  (0.021 0.021 0.0101)
  ```

- Now create the two sets. `setSet -batch makePointSet.setSet`
- `boxToPoint` can also be used to remove points from a set.
- Have a look in `constant/polyMesh/sets/`

More sets

- Two cell sets and one face set are still needed.
  `cp -r ../../jonssonmProject/*.setSet .`
- Lets have a look in them. Then run `setSet -batch makeCellSet.setSet` and so on.
- The mesh is now created.

Dictionaries

- The engine parameters and layering rules must be set as well as the `controlDict`.
- Copy the `engineGeometry`-dictionary to the `constant/`-directory.
- And the `controlDict`-dictionary to the `system/`-directory.

  ```
  cp -r ../../jonssonmProject/engineGeometry constant/
  ```
- Lets have a look at the dicitonaries.
- Two more diciotnaries needed.

  ```
  cp -r $FOAM_TUTORIALS/icoFoam/cavity/system/fv* system/
  ```
- The mesh setup is now done.

Cell layer addition and removal

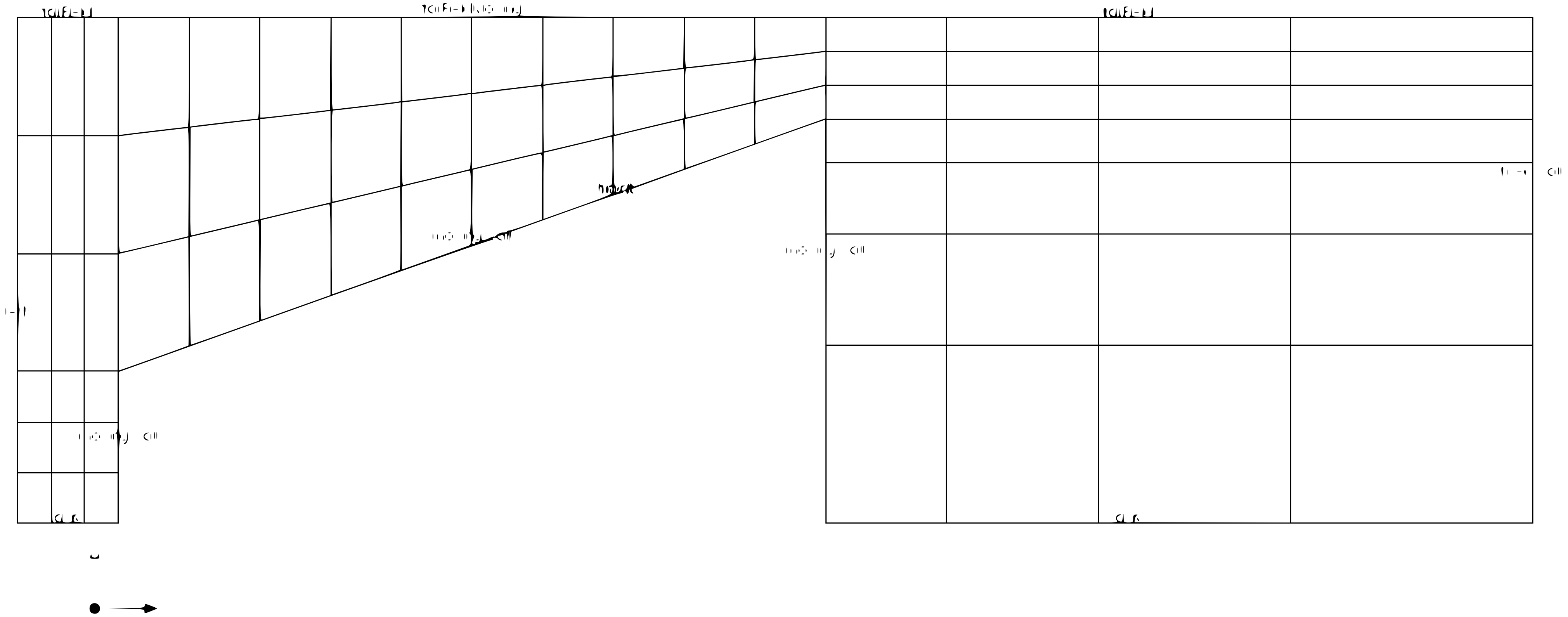
- Layer addition and removal based on `movingConeTopo`-tutorial.

```
cp -r $FOAM_TUTORIALS/icoDyMFoam/movingConeTopo .
blockMesh
icoDyMFoam
paraFoam
```

dynamicMeshDict

- Compare `motionVelPeriod` with `start`- and `endtime` in the `controlDict`.
- Uses `layerAdditionRemoval`-library. Which is based on `addCellLayer` and `removeCellLayer`.
- Adds and removes cells as a function of the `max`-and `minThickness` variables.
- Determined by `leftEdge` , `leftObstacleEdge` and `rigthObstacleEdge` .

layerAdditionRemoval



Alternative to blockMesh

- Create each patch separately in CAD-software
- Export to `.stl`-format. Triangulated surfaces.
- Use the mesh generation utility `snappyHexMesh`
- Problem with multiple volumes at the same time, `locationInMesh`.
- Either first or third time step.

The next step

- Implement boundary conditions.
- B.C. at ports.
- Solving the equations.