FINITE VOLUME METHOD [FVM] GRID GENERATION

Rolf-Henning Ulstein Klungsøyr, October 22nd, 2009

About FVM

- The Finite Volume Method is a CFD method developed to simulate fluid (or air) flow around an object
- Solves the same problems as FEM, but in quite a different way
- Used in FLUENT, one of the most popular comercial CFD applications for general purpose simulations.

About FVM (2)

- Based on dividing the domain into cells or control volumes (CV)
- This presentation focuses on grid generation
- The calculations done on the CV's are done in the centroids of the volumes.

FVM – Theoretical background

- Based on the following laws:
  - Conservation of fluid mass
  - Newton's 2nd law (Rate of change in momentum equals the sum of forces on a fluid particle)
  - 1st law of thermo dynamics (Rate of change in energy is equal to the sum of the rate of heat addition to and the rate of work done on a fluid particle)

FVM – Advantages

- Can be used on complex geometries if the gridding is done well (compared to i.e. FD)
- Due to its nature, with cell averaging, it has an automatic damping, and as such avoid simulation instability.
- Versatile
- Relatively easy to implement (vs FEM)

FVM – Drawbacks

- The damping also leads to loss of certain Physical phenomena

FVM – Governing equations

- As other CFD methods, FVM makes use of the Navier Stokes equations, here given for incompressible flow:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla\rho+\mu\nabla^2\mathbf{u}+\mathbf{f}$$

$$\nabla\cdot\mathbf{v}=0$$

FVM – Euler equations

- In most situations, these equations must be further simplified to reduce computational cost.
- Ignoring viscosity in NS yields the Euler equations:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla\rho+\mathbf{f}$$

- This often yields good results for simulations of gases, since the viscous forces are quite weak, and the CFD methods themselves induce viscous effects

FVM – Grid Generation

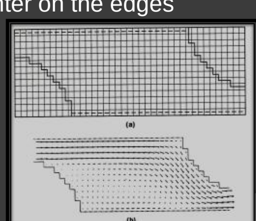
- Structured Grids
  - Cartesian Grids
  - Curvilinear Grids
- Block-Structured Grids
- Unstructured Grids

FVM – Structured Grids

- Structured grids are the most simple grids in CFD.
- Characteristics:
  - Grid points at intersecting coordinate lines
  - Fixed number of neighbours
  - Can be mapped into a matrix

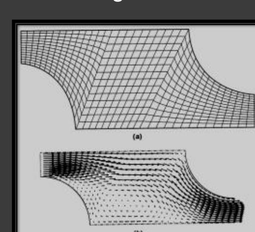
FVM – Structured Grids (2)

- Boundaries in Cartesian SG are straight lines going through the center of the cell and center on the edges



FVM – Structured Grids (3)

- Boundaries in Curvilinear SG are located on the edges of elements.

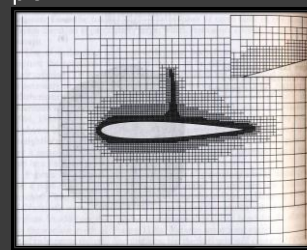


FVM – Block-Structured Grids

- Cartesian grids map very poorly to complex geometries. This can be improved by using Block-Structured grids
- Basically a structured grid which is further subdivided on critical areas of the modelled mesh.
- Easier to implement than curvilinear, so this is a way to get the accuracy of a curvilinear with the simplicity of cartesian.

FVM – Block-Structured Grids (2)

- Example:



FVM – Unstructured Grids

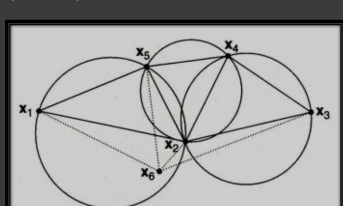
- For complex geometries the structured approach is inadequate.
- Unstructured Grids:
  - CV's can have any shape
  - No restriction on number of neighbors

FVM – Delaunay Triangulation (1)

- Delaunay Triangulation:
  - A triangulation of a set of nodes
  - Follows the circum-circle property
    - Leads to retriangulation when inserting new points

FVM – Delaunay Triangulation (2)

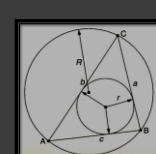
- Example of point inside one circumcircle



FVM – Delaunay Triangulation (3)

- Area and Aspect Ratio
  - $Area = \sqrt{s(s-a)(s-b)(s-c)}$
  - $AspectRatio = \frac{abc}{8(s-a)(s-b)(s-c)}$

$$\left[s = \frac{1}{2}(a+b+c) \right]$$



FVM – Delaunay Triangulation (4)

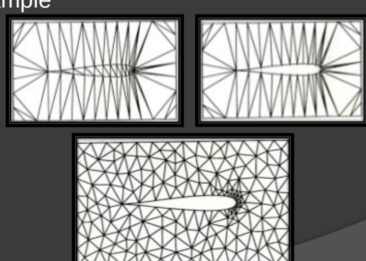
- Strategic point insertion
  - Insert point at triangle circumcentres
    - Largest area
    - Largest circumcircle radius
    - Target obtuse-angled and skinny triangles (use AR)
  - Using Voronoi segments

FVM – Delaunay Triangulation (5)

- Delaunay approach for generation:
  - Create initial Delaunay triangulation from boundary points
    - Remove faulty triangles (i.e. inside boundaries)
  - Insert points at strategical spots and retriangulate until aspect ratio and area of all triangles are under the max limit
    - Important to verify that boundaries are preserved

FVM – Delaunay Triangulation (6)

- Example



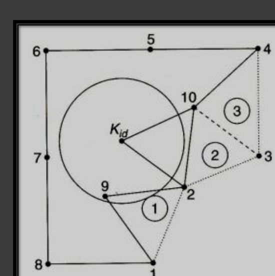
FVM – Advancing Front Technique (1)

- Different approach, makes the final triangulations intitially.
- Uses parameters δ and s from user to determine the grid
- Important functions (use is explained later):
  - $r = 0.8 * \partial'$
  - $\partial' = \begin{cases} 0.55 * l & if\ (\partial < 0.55 * l) \\ \partial & if\ (0.55 * l \le \partial \le 2.0 * l) \\ 2.0 * l & if\ (\partial > 2.0 * l) \end{cases}$

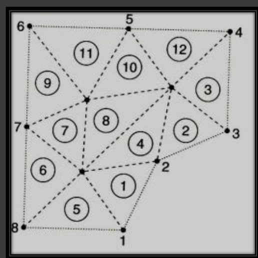
FVM – Advancing Front Technique (2)

- Algorithm
  - Initial grid front
  - Use shortest side in front
  - Compute position of ideal point Kideal
  - Construct a circle with Kideal as centre
  - Find active nodes within the circle
  - Check validity of triangle
  - If not valid, use the next-shortest side
  - If valid, re-order front
  - Reiterate

FVM – Advancing Front Technique (3)



FVM – Advancing Front Technique (4)



FVM – Parallelization

- I will not go into the solving of FVM, but it can be run in parallel by running the computations for each control volume in separate threads.

FVM – References [books]

- An introduction to CFD: The Finite Volume Method (Versteeg, Malalasekera)
- Basic Structured Grid Generation, with an intro to unstructured grid generation (Farashkhaluat, Miles)
- Computational Grids (Carey)