Grid Generation and Post-Processing for Computational Fluid Dynamics (CFD)

Tao Xing and Fred Stern
Outline

1. Introduction
2. Choice of grid
   2.1. Simple geometries
   2.2. Complex geometries
3. Grid generation
   3.1. Conformal mapping
   3.2. Algebraic methods
   3.3. Differential equation methods
   3.4. Commercial software
   3.5. Systematic grid generation for CFD UA
4. Post-processing
   4.1. UA (details in “Introduction to CFD”)
   4.2. Calculation of derived variables
   4.3. Calculation of integral variables
   4.4. Visualization
5. References and books
Introduction

• The numerical solution of partial differential equations requires some discretization of the field into a collection of points or elemental volumes (cells)
• The differential equations are approximated by a set of algebraic equations on this collection, which can be solved to produce a set of discrete values that approximate the solution of the PDE over the field
• **Grid generation** is the process of determining the coordinate transformation that maps the body-fitted non-uniform non-orthogonal physical space \(x, y, z, t\) into the transformed uniform orthogonal computational space, \(\xi, \eta, \zeta, \tau\).
• **Post-processing** is the process to examine and analyze the flow field solutions, including contours, vectors, streamlines, Iso-surfaces, animations, and CFD Uncertainty Analysis.
Choice of grid

• **Simple/regular geometries** (e.g. pipe, circular cylinder): the grid lines usually follow the coordinate directions.

• **Complex geometries** (Stepwise Approximation)
  1. Using Regular Grids to approximate solution domains with inclined or curved boundaries by staircase-like steps.
  2. Problems:
     1. Number of grid points (or CVs) per grid line is not constant, special arrays have to be created.
     2. Steps at the boundary introduce errors into solutions.
     3. Not recommended except local grid refinement near the wall is possible.

An example of a grid using stepwise approximation of an Inclined boundary.
Choice of grid, cont’d

• Complex geometries (Overlapping Chimera grid)
  1. **Definition:** Use of a set of grids to cover irregular solution domains
  2. **Advantages:**
     (1). Reduce substantially the time and efforts to generate a grid, especially for 3D configurations with increasing geometric complexity
     (2). It allows – without additional difficulty – calculation of flows around moving bodies
  3. **Disadvantages:**
     (1). The programming and coupling of the grids can be complicated
     (2). Difficult to maintain conservation at the interfaces
     (3). Interpolation process may introduce errors or convergence problems if the solution exhibits strong variation near the interface.
Choice of grid, cont’d

• Chimera grid (examples):

a. An example of a multi-zonal grid.

b. An example of a coarse-fine distribution in a multi-zonal grid.

c. An example of a chimera grid.

Different grid distribution approaches
Choice of grid, cont’d

- Chimera grid (examples):

  - Inlet
  - Far-field #1
  - No-slip
  - Symmetry
  - Exit
  - Far-field #2
Choice of grid, cont’d

• Complex geometries (*Boundary-Fitted Non-Orthogonal Grids*)

1. Types:
   (1). Structured
   (2). Block-structured
   (3). Unstructured

2. Advantages:
   (1). Can be adapted to any geometry
   (2). Boundary conditions are easy to apply
   (3). Grid spacing can be made smaller in regions of strong variable variation.

3. Disadvantages:
   (1). The transformed equations contain more terms thereby increasing both the difficulty of programming and the cost of solving the equations
   (2). The grid non-orthogonality may cause unphysical solutions.
Choice of grid, cont’d

- Complex geometries (Boundary-Fitted Non-Orthogonal Grids)

**Structured**

Fig. 2.1. Example of a 2D, structured, non-orthogonal grid, designed for calculation of flow in a symmetry segment of a staggered tube bank.

**Block-structured**

With matching interface

Fig. 2.2. Example of a 2D block-structured grid which matches at interfaces, used to calculate flow around a cylinder in a channel.

**Unstructured**

Block-structured

Without matching interface

Fig. 2.3. Example of a 2D block-structured grid which does not match at interfaces, designed for calculation of flow around a hydrofoil under a water surface.
Grid generation

- **Conformal mapping:** based on complex variable theory, which is limited to two dimensions.
- **Algebraic methods:**
  1. 1D: polynomials, Trigonometric functions, Logarithmic functions
  2. 2D: Orthogonal one-dimensional transformation, normalizing transformation, connection functions
  3. 3D: Stacked two-dimensional transformations, superelliptical boundaries
- **Differential equation methods:**
  Step 1: Determine the grid point distribution on the boundaries of the physical space.
  Step 2: Assume the interior grid point is specified by a differential equation that satisfies the grid point distributions specified on the boundaries and yields an acceptable interior grid point distribution.
- **Commercial software** (Gridgen, Gambit, etc.)
Grid generation (examples)

Orthogonal one-dimensional transformation

Superelliptical transformations: (a) symmetric; (b) centerbody; (c) asymmetric
Grid generation (commercial software, gridgen)

- Commercial software GRIDGEN will be used to illustrate typical grid generation procedure
Grid generation (Gridgen, 2D pipe)

- **Geometry:** two-dimensional axisymmetric circular pipe
- **Creation of connectors:** “connectors” → “create” → “2 points connectors” → “input x,y,z of the two points” → “Done”.
- **Dimension of connectors:** “Connectors” → “modify” → “Redimension” → “40” → “Done”.
- Repeat the procedure to create C2, C3, and C4
Grid generation (Gridgen, 2D pipe, cont’d)

- **Creation of Domain:** “domain”→“create”→“structured”→“Assemble edges”→“Specify edges one by one”→“Done”.

- **Redistribution of grid points:** Boundary layer requires grid refinement near the wall surface. “select connectors (C2, C4)”→“modify”→“redistribute”→“grid spacing(start+end)” with distribution function

- For turbulent flow, the first grid spacing near the wall, i.e. “matching point”, could have different values when different turbulent models applied (near wall or wall function).
Grid generation (3D NACA12 foil)

- **Geometry:** two-dimensional NACA12 airfoil with 60 degree angle of attack
- **Creation of geometry:** unlike the pipe, we have to import the database for NACA12 into Gridgen and create connectors based on that (only half of the geometry shape was imported due to symmetry).
  - “input”→”database”→”import the geometry data”→”connector”→”create”→”on DB entities”→”delete database”
- **Creation of connectors C1 (line), C2(line), C3(half circle)**

Half of airfoil surface
Grid generation (3D NACA12 airfoil, cont’d)

- Redimensions of the four connectors and create domain
- Redistribute the grid distribution for all connectors. Especially refine the grid near the airfoil surface and the leading and trailing edges
Grid generation (3D NACA12 airfoil, cont’d)

- **Duplicate the other half of the domain:** "domain”→”modify”→”mirror respect to y=0”→”Done”.
- **Rotate** the whole domain with angle of attack 60 degrees: 
  "domain”→”modify”→”rotate”→”using z-principle axis”→”enter rotation angle: -60”→”Done”.
Grid generation (3D NACA12 airfoil, cont’d)

- Create 3D block: “blocks”→”create”→”extrude from domains”→specify “translate distance and direction”→”Run N”→”Done”.
- Split the 3D block to be four blocks: “block”→”modify”→”split”→”in ξ direction”→”ξ =?”→”Done”.
- Specify boundary conditions and export Grid and BCS.
Systematic grid generation for CFD UA

- CFD UA analysis requires a series of meshes with uniform grid refinement ratio, usually start from the fine mesh to generate coarser grids.
- A tool is developed to automate this process, i.e., each fine grid block is input into the tool and a series of three, 1D interpolation is performed to yield a medium grid block with the desired non-integer grid refinement ratio.
- 1D interpolation is the same for all three directions.

Consider 1D line segment with \( N_1 \) and \( N_2 = 1 + (N_1 - 1)/r_G \) points for the fine and medium grids, respectively.

**step 1**: compute the fine grid size at each grid node:

\[
\Delta x_{1_i} = x_{1_i} - x_{1_{i-1}}
\]

**step 2**: compute the medium grid distribution:

\[
\Delta x_{2_i} = r_G \Delta x_{1_i}
\]

where \( \Delta x_{1_i} \) from the first step is interpolated at location \( x_{2_i} \).

**step 3**: The medium grid distribution \( x_{2_i} \) is scaled so that the fine and medium grid line segments are the same (i.e.,

\[
x_{2_{N_2}} = x_{1_{N_1}}
\]

**step 4**: The procedure is repeated until it converges.
Post-Processing

- **Uncertainty analysis**: estimate order of accuracy, correction factor, and uncertainties (for details, CFD Lecture 1, introduction to CFD).
- **MPI functions** required to combine data from different blocks if parallel computation used
- **Calculation of derived variables** (vorticity, shear stress)
- **Calculation of integral variables** (forces, lift/drag coefficients)
- **Calculation of turbulent quantities**: Reynolds stresses, energy spectra
- **Visualization**
  1. XY plots (time/iterative history of residuals and forces, wave elevation)
  2. 2D contour plots (pressure, velocity, vorticity, eddy viscosity)
  3. 2D velocity vectors
  4. 3D Iso-surface plots (pressure, vorticity magnitude, Q criterion)
  5. Streamlines, Pathlines, streaklines
  6. Animations
- **Other techniques**: Fast Fourier Transform (FFT), Phase averaging
Post-Processing (visualization, XY plots)

Lift and drag coefficients of NACA12 with 60° angle of attack (CFDSHIP-IOWA, DES)

Wave profile of surface-piercing NACA24, Re=1.52e6, Fr=0.37 (CFDSHIP-IOWA, DES)
Different colors illustrate different blocks (6)

Re=$10^5$, DES, NACA12 with angle of attack 60 degrees
Post-Processing (NACA12, 2D contour plots, vorticity)

- Define and compute new variable: “Data”→“Alter”→“Specify equations”→“vorticity in x,y plane: v10”→“compute”→“OK”.
Post-Processing (NACA12, 2D contour plot)

- Extract 2D slice from 3D geometry: “Data” → “Extract” → “Slice from plane” → “z=0.5” → “extract”
Post-Processing (NACA12, 2D contour plots)

- **2D contour plots** on $z=0.5$ plane (vorticity and eddy viscosity)

![Vorticity $\omega_z$](image1)

![Eddy viscosity](image2)
Post-Processing (NACA12, 2D contour plots)

- **2D contour plots** on \( z=0.5 \) plane (pressure and streamwise velocity)

![Pressure Contour Plot](image1.png)

![Streamwise Velocity Contour Plot](image2.png)
Post-Processing (2D velocity vectors)

- **2D velocity vectors** on \( z=0.5 \) plane: turn off “contour” and activate “vector”, specify the vector variables.

![Select Variables dialog box showing U: V4 : U and V: V5 : V]
Post-Processing (3D Iso-surface plots, cont’d)

• 3D Iso-surface plots: pressure, $p=$constant
• 3D Iso-surface plots: vorticity magnitude

- $\Omega = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$

• 3D Iso-surface plots: $\lambda_2$ criterion

  Second eigenvalue of

- $\frac{1}{2\rho} \nabla^2 p$

• 3D Iso-surface plots: $Q$ criterion

- $Q = \frac{1}{2} \left( \Omega_{ij} \Omega_{ij} - S_{ij} S_{ij} \right)$

- $\Omega_{ij} = \left( u_{i,j} - u_{j,i} \right) / 2$

- $S_{ij} = \left( u_{i,j} + u_{j,i} \right) / 2$
Post-Processing (3D Iso-surface plots)

- **3D Iso-surface plots**: used to define the coherent vortical structures, including pressure, vorticity magnitude, Q criterion, $\lambda_2$, etc.
Post-Processing (streamlines)

- **Streamlines (2D):**

  Streamlines with contour of pressure

  - Streaklines and pathlines (not shown here)
Post-Processing (Animations)

- **Animations** (3D): animations can be created by saving CFD solutions with or without skipping certain number of time steps and playing the saved frames in a continuous sequence.
- Animations are important tools to study time-dependent developments of vortical/turbulent structures and their interactions.
Other Post-Processing techniques

- Fast Fourier Transform
  1. A signal can be viewed from two different standpoints: the **time domain** and the **frequency domain**
  2. The **time domain** is the trace on a signal (forces, velocity, pressure, etc.) where the vertical deflection is the signal's amplitude, and the horizontal deflection is the time variable.
  3. The **frequency domain** is like the trace on a spectrum analyzer, where the deflection is the frequency variable and the vertical deflection is the signal's amplitude at that frequency.
  4. We can go between the above two domains using *(Fast)* Fourier Transform
- Phase averaging (next two slides)
Other Post-Processing techniques (cont’d)

- **Phase averaging**
  - **Assumption:** the signal should have a coherent dominant frequency.
  - **Steps:**
    1. a filter is first used to smooth the data and remove the high frequency noise that can cause errors in determining the peaks.
    2. once the number of peaks determined, zero phase value is assigned at each maximum value.
    3. Phase averaging is implemented using the triple decomposition.

\[
\begin{align*}
z(t) &= \overline{z}(t) + \tilde{z}(t) + z'(t) = \langle z(t) \rangle + z'(t) \\
\overline{z}(t) &= \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} z(t) dt \\
\tilde{z}(t) &= \text{organized oscillating component} \\
z'(t) &= \text{random fluctuating component} \\
\tau &= \text{is the time period of the dominant frequency} \\
\langle z(t) \rangle &= \lim_{N \to \infty} \sum_{n=0}^{N-1} z(t + n\tau)
\end{align*}
\]

\[\langle z(t) \rangle \text{ is the phase average associated with the coherent structures}\]
Other Post-Processing techniques (cont’d)

- FFT and Phase averaging (example)

Original, phase averaged, and random fluctuations of the wave elevation at one point

FFT of wave elevation time histories at one point
References and books

• User Manual for GridGen
• User Manual for Tecplot
• Numerical recipes: http://www.library.cornell.edu/nr/