Simulation and Scientific Computing 2

Assignment 3: Interpolation between Unstructured Grids

Benjamin Mann
Chair for System Simulation, Friedrich-Alexander University of Erlangen-Nuremberg
June 9, 2022

Outline

Motivation

Grids

- Uniform structured grid
- Unstructured grid

Interpolation between grids

- Structured to unstructured grid
- Unstructured to structured grid
- Unstructured to unstructured grid

Motivation

Motivation

- To use experimental data as initial condition, comparison or for boundary conditions.
    - Eg: Comparison between wind tunnel results and numerical simulation [1].

Motivation

- To use experimental data as initial condition, comparison or for boundary conditions.
  - Eg: Comparison between wind tunnel results and numerical simulation [1].
- To restart simulation from a stored state with different grids.

Motivation

- To use experimental data as initial condition, comparison or for boundary conditions.
  - Eg: Comparison between wind tunnel results and numerical simulation [1].
- To restart simulation from a stored state with different grids.
- To compare simulation results from different simulations, which uses different grids or methods.
  - Eg: To solve fluid-structure interface problem [1].

Motivation

- To use experimental data as initial condition, comparison or for boundary conditions.
  - Eg: Comparison between wind tunnel results and numerical simulation [1].
- To restart simulation from a stored state with different grids.
- To compare simulation results from different simulations, which uses different grids or methods.
  - Eg: To solve fluid-structure interface problem [1].
- To combine different discretization techniques

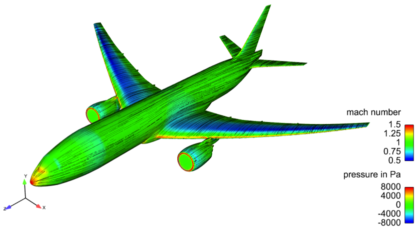


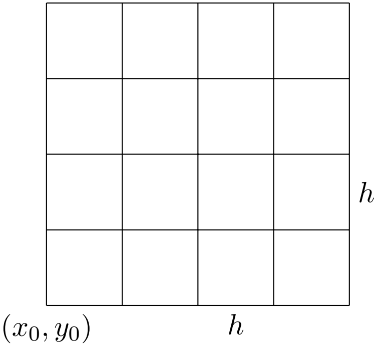
Figure: Numerical simulation of aircraft [2]

Figure: Wind tunnel testing of aircraft [3]

Grids

Uniform structured grid

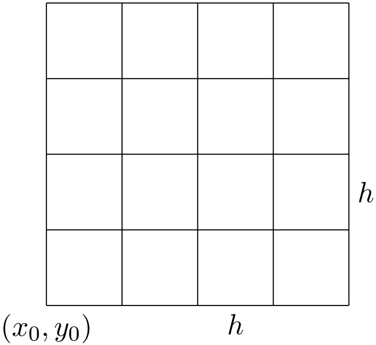
Grid definition



- $\Omega_h = \{(ih, jh) + (x_0, y_0) | i, j = 0, \ldots, n\}$

Uniform structured grid

Grid definition



- $\Omega_h = \{(ih, jh) + (x_0, y_0) | i, j = 0, \ldots, n\}$
- Values are stored at nodes.
- From here on (x_0, y_0) will be considered to be origin.

Uniform structured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_h$. Then to interpolate

Uniform structured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_h$. Then to interpolate

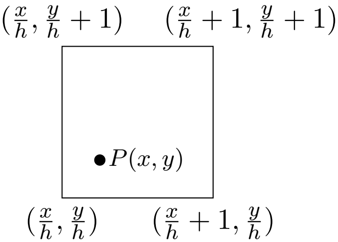
1. Find the containing cell

Uniform structured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_h$. Then to interpolate

1. Find the containing cell
   - No search is required to find the cell containing the point $P(x, y)$



Uniform structured grid

Interpolation procedure for a point

2. Apply suitable interpolation.

Uniform structured grid

Interpolation procedure for a point

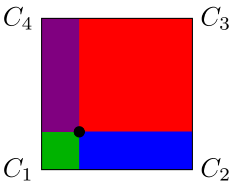
2. Apply suitable interpolation.

    - Nearest neighbor.
    - Bi-linear Interpolation.

Uniform structured grid

Interpolation procedure for a point

2. Apply suitable interpolation.
   - Nearest neighbor.
   - Bi-linear Interpolation.



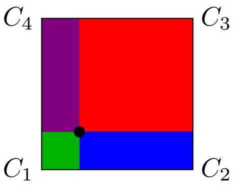
$$\lambda_1 = \frac{A(P, C_3)}{A(C_1, C_3)} \quad \lambda_2 = \frac{A(P, C_4)}{A(C_1, C_3)} \quad \lambda_3 = \frac{A(P, C_1)}{A(C_1, C_3)} \quad \lambda_4 = \frac{A(P, C_2)}{A(C_1, C_3)}$$

Uniform structured grid

Interpolation procedure for a point

2. Apply suitable interpolation.

   - Nearest neighbor.
   - Bi-linear Interpolation.

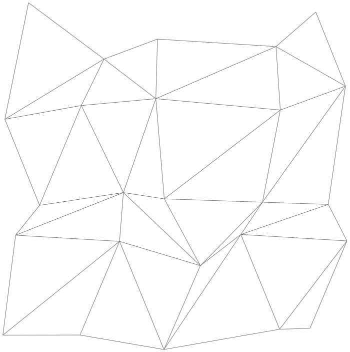


$$\lambda_1 = \frac{A(P, C_3)}{A(C_1, C_3)} \quad \lambda_2 = \frac{A(P, C_4)}{A(C_1, C_3)} \quad \lambda_3 = \frac{A(P, C_1)}{A(C_1, C_3)} \quad \lambda_4 = \frac{A(P, C_2)}{A(C_1, C_3)}$$

$V(P) = \sum_{i=1}^{4} V(C_i) * \lambda_i$, where V is value at corresponding point/node.

Unstructured grid

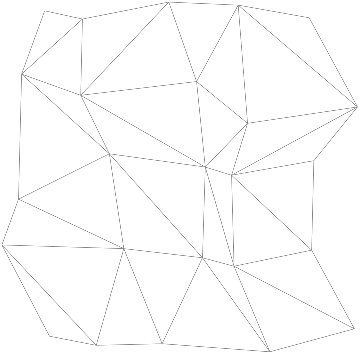
Grid definition



- $\Omega_u = \bigcup_i T_i$

Unstructured grid

Grid definition



- $\Omega_u = \bigcup_i T_i$
- It consist of array of points and an array of triangles.
- Values are stored at nodes.

Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_u$. Then to interpolate

1. Find the containing triangle

Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_u$. Then to interpolate

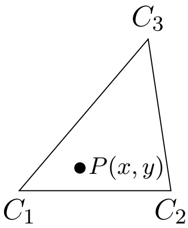
1. Find the containing triangle
    - Loop over all triangles and check whether P is contained inside

Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_u$. Then to interpolate

1. Find the containing triangle
   - Loop over all triangles and check whether P is contained inside
   - To check use barycentric coordinates

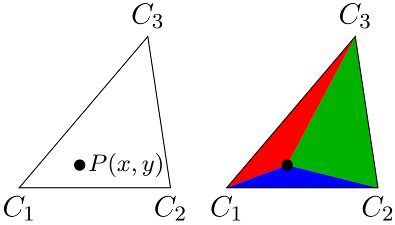


Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_u$. Then to interpolate

1. Find the containing triangle
   - Loop over all triangles and check whether P is contained inside
   - To check use barycentric coordinates



$$\lambda_1 = \frac{A(P, C_2, C_3)}{A(C_1, C_2, C_3)}$$

$$\lambda_2 = \frac{A(P, C_3, C_1)}{A(C_1, C_2, C_3)}$$

$$\lambda_3 = \frac{A(P, C_1, C_2)}{A(C_1, C_2, C_3)}$$

$$\sum_{i=1}^{3} \lambda_i = 1$$

Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_u$. Then to interpolate

1. Find the containing triangle
   - Loop over all triangles and check whether P is contained inside
   - To check use barycentric coordinates
   - The point $P(x, y)$ is contained if and only if $\lambda_i \geqslant 0 \,\forall\, i$

Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x,y) \in \Omega_u$. Then to interpolate

1. Find the containing triangle
   - Loop over all triangles and check whether P is contained inside
   - To check use barycentric coordinates
   - The point $P(x,y)$ is contained if and only if $\lambda_i \geqslant 0 \,\forall\, i$
2. Apply suitable interpolation.

Unstructured grid

Interpolation procedure for a point

Let us consider a random point $P(x, y) \in \Omega_u$. Then to interpolate

1. Find the containing triangle
   - Loop over all triangles and check whether P is contained inside
   - To check use barycentric coordinates
   - The point $P(x, y)$ is contained if and only if $\lambda_i \geqslant 0 \; \forall \; i$
2. Apply suitable interpolation.
   - Linear interpolation using barycentric coordinates:

$$V(P) = \sum_{i=1}^{3} V(C_i) * \lambda_i$$

   - where V is value at corresponding point/node.

Interpolation between grids

Interpolation between grids

Interpolation from structured to unstructured grid

1. Loop over all the nodes of unstructured grid.

Interpolation between grids

Interpolation from structured to unstructured grid

1. Loop over all the nodes of unstructured grid.
    - And for every node/point perform interpolation of point as discussed in structured grid.

Interpolation between grids

Interpolation from structured to unstructured grid

1. Loop over all the nodes of unstructured grid.
   - And for every node/point perform interpolation of point as discussed in structured grid.
2. This algorithm is quite fast with $O(N)$.
   - N is number of nodes in unstructured grid.

Interpolation between grids

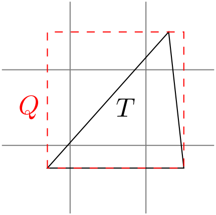
Interpolation from unstructured to structured grid

1. Loop over all the triangles of unstructured grid (Ω_u).
    - And for each triangle T form a bounding box Q.

Interpolation between grids

Interpolation from unstructured to structured grid

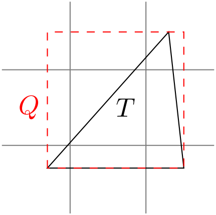
1. Loop over all the triangles of unstructured grid (Ω_u).
   - And for each triangle T form a bounding box Q.



Interpolation between grids

Interpolation from unstructured to structured grid

1. Loop over all the triangles of unstructured grid (Ω_u).
   - And for each triangle T form a bounding box Q.

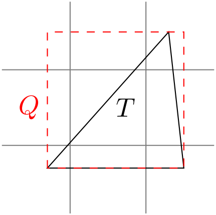


   - Loop over grid points $P \in \Omega_h \cap Q$, interpolate if $P \in T$ as discussed in unstructured grid

Interpolation between grids

Interpolation from unstructured to structured grid

1. Loop over all the triangles of unstructured grid (Ω_u).
   - And for each triangle T form a bounding box Q.



   - Loop over grid points $P \in \Omega_h \cap Q$, interpolate if $P \in T$ as discussed in unstructured grid
2. This algorithm is fast with $O(N)$.
   - N is number of nodes in structured grid.

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})
2. Store all triangles in the input mesh that intersect with a cell $C \in \Omega_h$ in a list:
    - For each triangle $T \in \Omega_{u1}$ (input grid):
        - Find all $C \in \Omega_h$ where $C \cap T \neq \emptyset$, e.g., by using the bounding box of T.

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})
2. Store all triangles in the input mesh that intersect with a cell $C \in \Omega_h$ in a list:
   - For each triangle $T \in \Omega_{u1}$ (input grid):
     - Find all $C \in \Omega_h$ where $C \cap T \neq \emptyset$, e.g., by using the bounding box of T.
     - Add T to the lists corresponding to those cells C

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})
2. Store all triangles in the input mesh that intersect with a cell $C \in \Omega_h$ in a list:
   - For each triangle $T \in \Omega_{u1}$ (input grid):
     - Find all $C \in \Omega_h$ where $C \cap T \neq \emptyset$, e.g., by using the bounding box of T.
     - Add T to the lists corresponding to those cells C
3. For every vertex $P \in (\Omega_{u2})$ (output grid):
   - Find the cell $C \in \Omega_h$ that contains P.

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})
2. Store all triangles in the input mesh that intersect with a cell $C \in \Omega_h$ in a list:
   - For each triangle $T \in \Omega_{u1}$ (input grid):
     - Find all $C \in \Omega_h$ where $C \cap T \neq \emptyset$, e.g., by using the bounding box of T.
     - Add T to the lists corresponding to those cells C
3. For every vertex $P \in (\Omega_{u2})$ (output grid):
   - Find the cell $C \in \Omega_h$ that contains P.
   - Find the triangle $T \in \Omega_{u1}$ where $P \in T$ in the list of triangles intersecting with C.

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})
2. Store all triangles in the input mesh that intersect with a cell $C \in \Omega_h$ in a list:
   - For each triangle $T \in \Omega_{u1}$ (input grid):
     - Find all $C \in \Omega_h$ where $C \cap T \neq \emptyset$, e.g., by using the bounding box of T.
     - Add T to the lists corresponding to those cells C
3. For every vertex $P \in (\Omega_{u2})$ (output grid):
   - Find the cell $C \in \Omega_h$ that contains P.
   - Find the triangle $T \in \Omega_{u1}$ where $P \in T$ in the list of triangles intersecting with C.
   - Perform the interpolation from T to P as discussed in unstructured grid.

Interpolation between grids

Interpolation from one unstructured to another unstructured grid

1. Construct a structured grid Ω_h such that both unstructured grids lie within the bounds of Ω_h and mesh width $\approx$ mesh width of input unstructured grid (Ω_{u1})
2. Store all triangles in the input mesh that intersect with a cell $C \in \Omega_h$ in a list:
   - For each triangle $T \in \Omega_{u1}$ (input grid):
     - Find all $C \in \Omega_h$ where $C \cap T \neq \emptyset$, e.g., by using the bounding box of T.
     - Add T to the lists corresponding to those cells C
3. For every vertex $P \in (\Omega_{u2})$ (output grid):
   - Find the cell $C \in \Omega_h$ that contains P.
   - Find the triangle $T \in \Omega_{u1}$ where $P \in T$ in the list of triangles intersecting with C.
   - Perform the interpolation from T to P as discussed in unstructured grid.
4. This algorithm has the complexity $O(NM)$.
   - N is number of nodes in unstructured grid Ω_{u2}.
   - M is number of triangles stored in a cell.

Take a look at assignment sheet.

How to use paraview ?

Thanks for listening.

Any questions?

References

References I

[1] E. Rank, A. Halfmann, D. Scholz, *et al.*, *Wind loads on lightweight structures : Numerical simulation and wind tunnel tests*. Germany: WILEY-VCH, 2004.

[2] https://www.bionicsurface.com/wp-content/uploads/2016/01/Boeing777_pressure_01.png.

[3] http://www.boeingimages.com/archive/777-Wind-Tunnel-Testing-at-QinetiQ-2F3XC5OETU4.html.