Introduction to Computational Thinking and Programming for CFD

Module 13251

Dr. rer. nat. Marten Klein

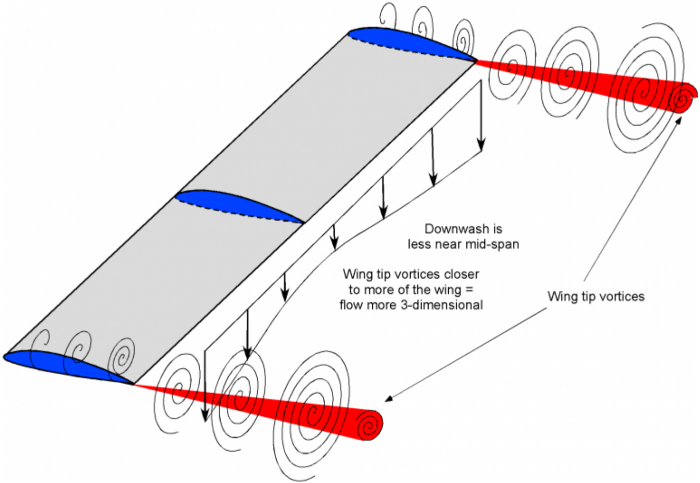
Numerical Fluid and Gas Dynamics, BTU Cottbus - Senftenberg

Brandenburg
University of Technology
Cottbus - Senftenberg

3 Meshing, gridded data, and interpolation

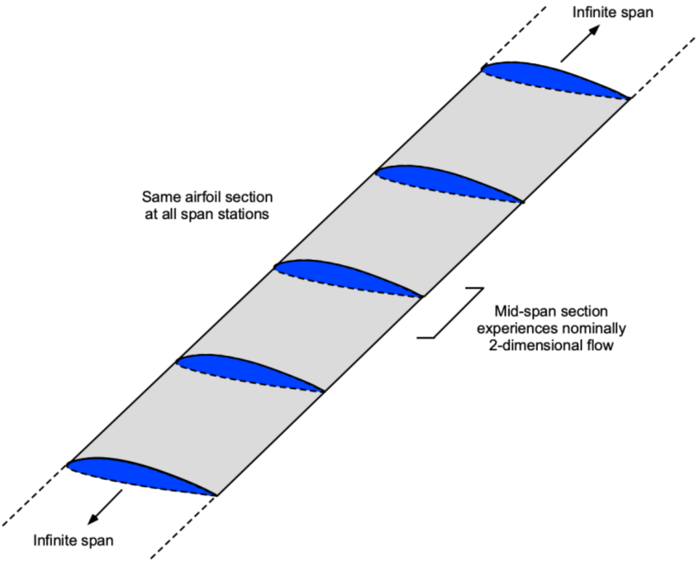
3.1 Overview and definitions

3-D airfoil



Source: `https://eaglepubs.erau.edu/app/uploads/sites/4/2022/07/WingVortices-768x529.png`

2-D airfoil



Source: `https://eaglepubs.erau.edu/app/uploads/sites/4/2021/08/Wing_infinite-768x613.png`

2-D structured grid (mesh) for the flow around an airfoil I

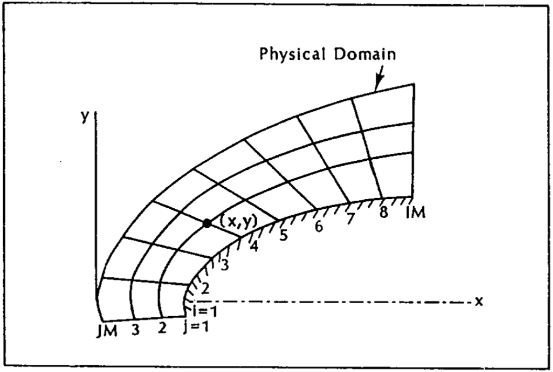


Figure 9-2. A typical 2-D domain for the axisymmetric blunt body problem.

From: K.A. Hoffmann & S.T. Chiang, *Computational Fluid Dynamics: Volume I*, Engineering Education System, 2000.

2-D structured grid (mesh) for the flow around an airfoil II

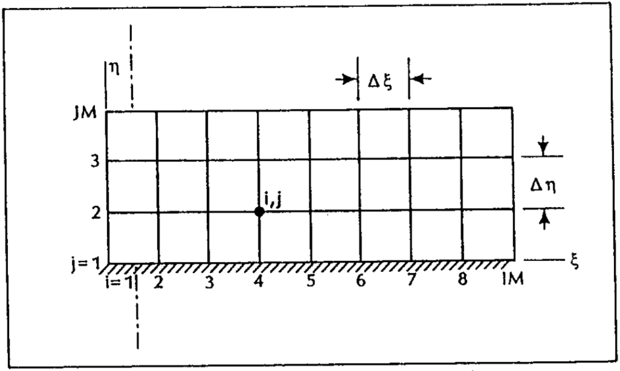


Figure 9-3. Computational domain with constant stepsizes $\Delta\xi$ and $\Delta\eta$.

From: K.A. Hoffmann & S.T. Chiang, *Computational Fluid Dynamics: Volume I*, Engineering Education System, 2000.

2-D structured grid (mesh) for the flow around an airfoil III

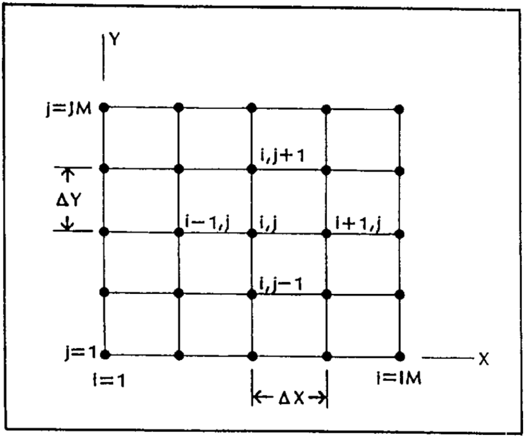
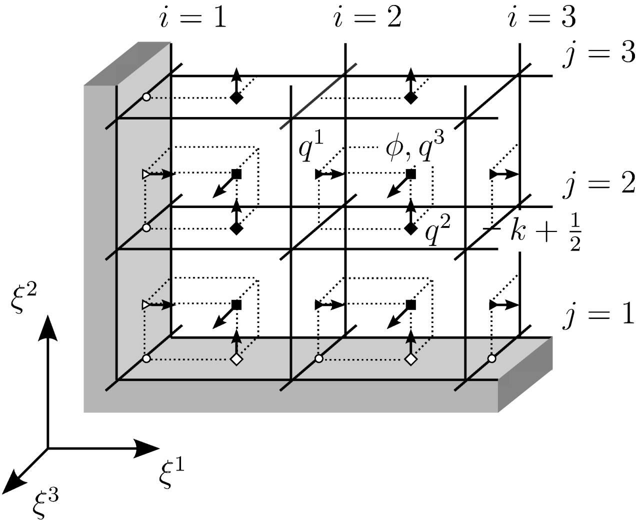


Figure 1-6. Sketch illustrating the nomenclature of computational space.

From: K.A. Hoffmann & S.T. Chiang, *Computational Fluid Dynamics: Volume I*, Engineering Education System, 2000.

Extension to 3-D airfoil geometry I



From: M. Klein, PhD thesis, BTU Cottbus-Senftenberg, 2016.

Extension to 3-D airfoil geometry II

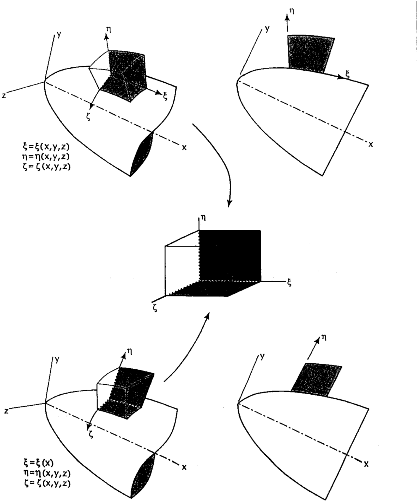


Figure 9-1. Nomenclature for the generalized coordinate system.

From: K.A. Hoffmann & S.T. Chiang, *Computational Fluid Dynamics: Volume I*, Engineering Education System, 2000.

Definitions

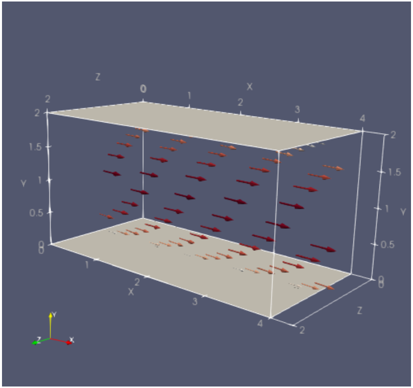
- A *mesh* (grid) denotes the **dicretization of the flow domain** into a finite number, $N_x \cdot N_y \cdot N_z$, of *mesh* (grid) *cells* that store physical properties as **nodal values** at corresponding **nodal points**.
- Each *mesh cell* (grid cell) is defined by a certain number of **vertices** that are numbered points $\boldsymbol{x}_{ijk} = (x_{ijk}, y_{ijk}, z_{ijk})^{\mathsf{T}}$, where $i \in [0, N_x]$, $j \in [0, N_y]$, $k \in [0, N_z]$
- In a Cartesian or **cuboidal grid**, $N_x \cdot N_y \cdot N_z$ mesh cells are built from $(N_x + 1) \cdot (N_y + 1) \cdot (N_z + 1)$ vertices that define the *cell corners* of the *structured grid*.
- *Mesh generation* (grid generation) denotes the **process** of constructing a mesh (grid). This involves a **numerical algorithm**.
- *Gridded data* is obtained when some spatial data is **interpolated** to the *nodal points* of the grid, resulting in a set of *nodal values*.

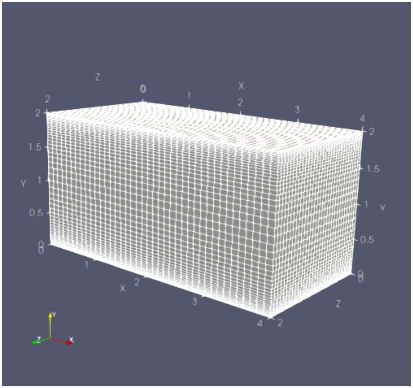
3 Meshing, gridded data, and interpolation

3.2 Algebraic grid generation for channel flow

Channel flow configuration and reference grid

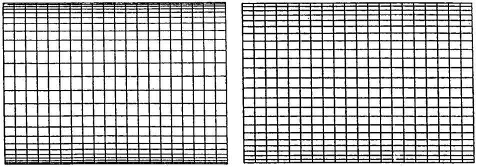
- **Left:** Sketch of the channel flow configuration
- **Right:** Mesh for the *channel395* OpenFOAM[1] LES tutorial case algebraically generated with the built-in `blockMesh` application





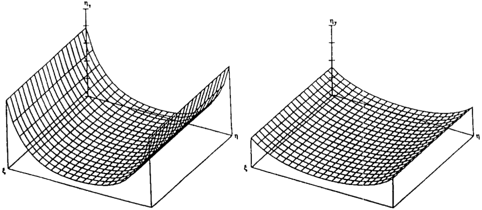
[1] Open-spurce CFD software *OpenFOAM*: `https://openfoam.org/`

Grid stretching for near-wall refinement (→ *Exercise*)



(a) $\beta = 1.05$ (b) $\beta = 1.2$

Figures 9-9a, b. Grid system generated by the transformation function (9-42).



Figures 9-10a, b. Metric distributions for the domains shown in Figure 9-9.

From: K.A. Hoffmann & S.T. Chiang, ***Computational Fluid Dynamics**: Volume I*, Engineering Education System, 2000.

1-D stretching function for the wall-normal direction

$\xi = \dfrac{i}{N}$	**equidistant grid coordinate**	*auxiliary*
$x = x(\xi)$	**'stretched' spatial coordinate**	*physical*

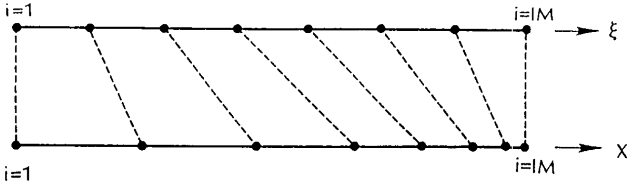


Figure P3-9b. Physical and the corresponding computational grid points distributions.

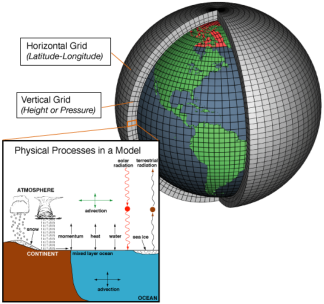
From: K.A. Hoffmann & S.T. Chiang, *Computational Fluid Dynamics: Volume I*, Engineering Education System, 2000.

3 Meshing, gridded data, and interpolation

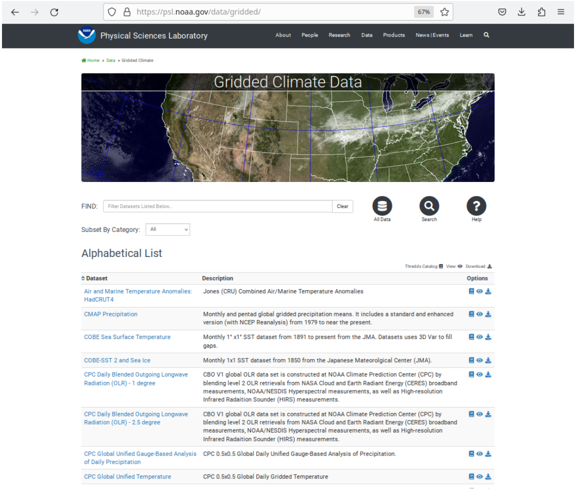
3.3 Gridded data and interpolation

Gridded data – Application in Earth system science I

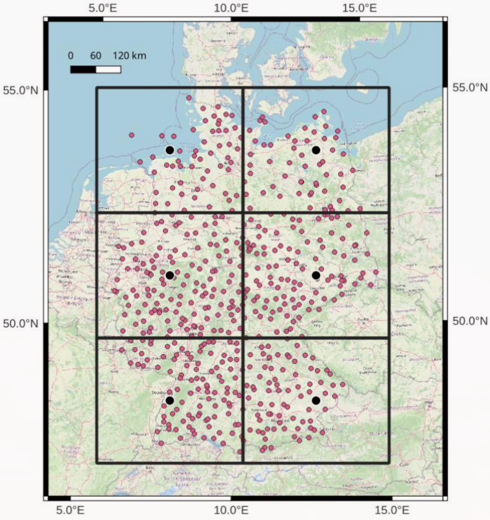
- **Acquire data** from various sources: satellites, balloon soundings, ...
- **Interpolate data** to various grids as needed by Earth system models
- **Provide gridded data** interpolated to latitude-longitude grid



Source: NOAA
https://celebrating200years.noaa.gov/breakthroughs/climate_model/modeling_schematic.html



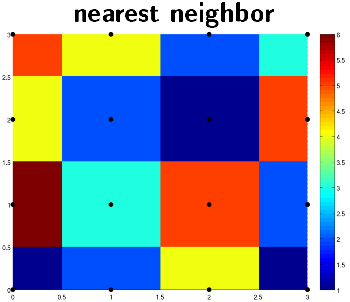
Gridded data – Application in Earth system science II

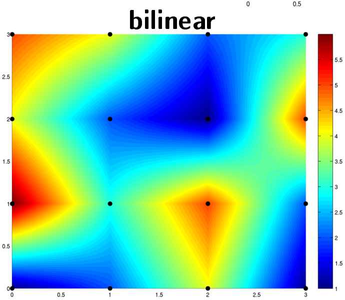


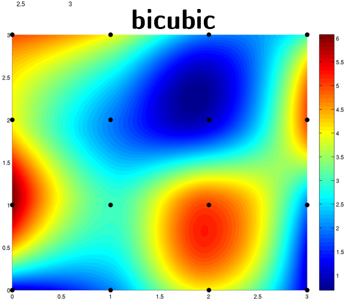
© 2021 Kai Lochbihler
https://lochbihler.nl/a-gridded-data-set-of-cloud-types-for-germany

Map showing the definition of the used **grid** (black polygons). The **black circles** (•) mark the *centroids* (nodal points) of the *grid cells*. The amount of data (station *density*) in each cell differs and depends on the location of the grid cell. Corner cells have the lowest coverage of Germany which reduces the number of stations in them.

2-D interpolation (for surface data)

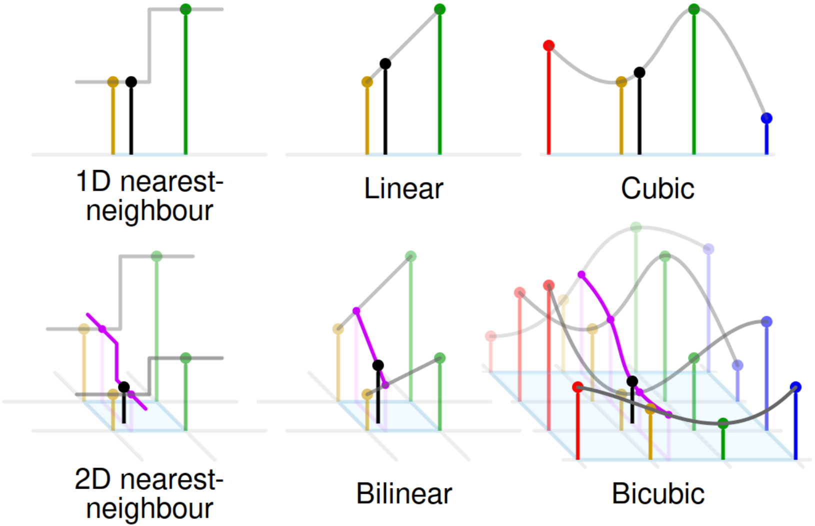






Public domain. Source: Wikipedia, `https://en.wikipedia.org/wiki/Interpolation`

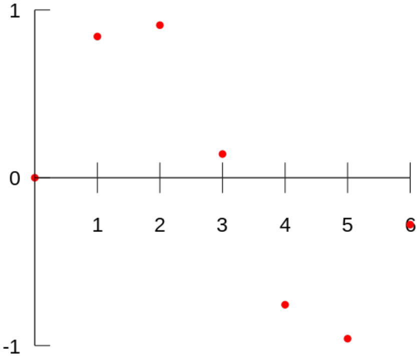
2-D *vs.* 1-D interpolation



Public domain. Source: Wikipedia, `https://en.wikipedia.org/wiki/Interpolation`

1-D interpolation I

Data

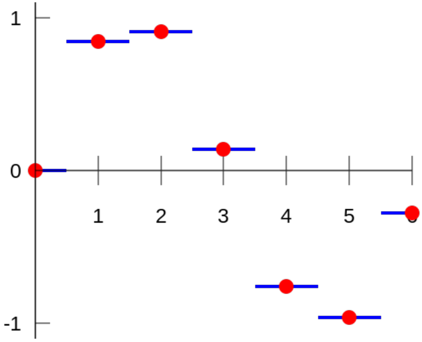


Public domain. Source: Wikipedia
https://en.wikipedia.org/wiki/Interpolation

- **Given:**
  gridded data or
  unevenly sampled point data
- **Question:**
  How to fill in the gaps?

1-D interpolation II

Piecewise-*constant* interpolation



Public domain. Source: Wikipedia
https://en.wikipedia.org/wiki/Interpolation

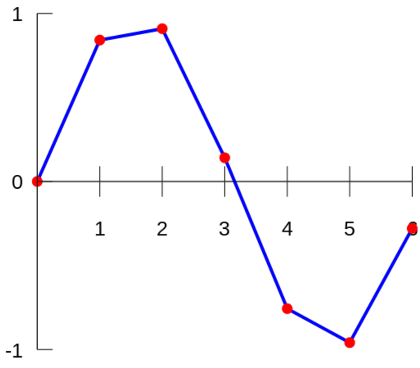
- Used in finite-volume-based (FVM) CFD solvers
- Equivalent name: **nearest neighbor interpolation**

$$f(x) = y_i$$

for $x_{i-1/2} \le x < x_{i+1/2}$
interior points i

1-D interpolation III

Piecewise-*linear* interpolation



Public domain. Source: Wikipedia
https://en.wikipedia.org/wiki/Interpolation

- Used in finite-element-based (FEM) solvers
- Application of the **two-point equation**

The interpolating property yields a linear system of equations (LSE) for the nodal points.

Looking at $i = 0$ and $i + 1 = 1$:

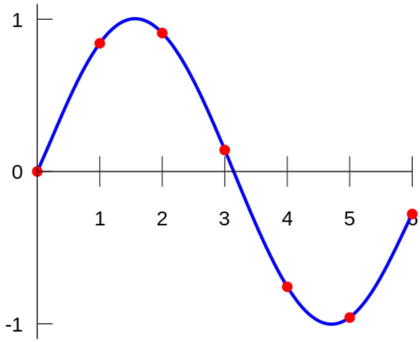
$$f(x) = Ax + B$$

$$y_0 = Ax_0 + B$$
$$y_1 = Ax_1 + B$$

$$\Rightarrow \quad f(x) = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0$$

1-D interpolation IV

Higher-order interpolation schemes



Public domain. Source: Wikipedia
https://en.wikipedia.org/wiki/Interpolation

- Used in finite-difference (FDM) and spectral-element (SEM) solvers
- **Polynomials** yield an extended LSE that couples more nodal points in a non-local manner

The specific C-spline for the left-hand example:

$$f(x) = \begin{cases} -0.152x^3 + 0.994x \\ -0.013x^3 - 0.419x^2 + 1.413x - 0.140 \\ 0.140x^3 - 1.336x^2 + 3.247x - 1.362 \\ 0.158x^3 - 1.495x^2 + 3.723x - 1.838 \\ 0.054x^3 - 0.245x^2 - 1.276x + 4.826 \\ -0.187x^3 + 3.367x^2 - 19.34x + 34.93 \end{cases}$$

Keywords

- Discretization
- Mesh (grid) generation
- Structured grid
- Gridded data
- Interpolation