Introduction to Computational Fluid Dynamics: *Governing Equations, Turbulence Modeling Introduction and Finite Volume Discretization Basics.*

Joel Guerrero

April 22, 2020

Contents

Chapter 1

Notation and Mathematical Preliminaries

When presenting the fluid flow equations, as well as throughout this manuscript, we make use of the following vector/tensor notation and mathematical operators.

From now on we are going to refer to a zero rank tensor as a scalar. A first rank tensor will be referred as a vector. And a second rank tensor will associated to a tensor. Vectors will be denoted by minuscule bold letters, whereas tensors by majuscule bold letters or bold greek symbols. Scalars will be represented by normal letters or normal greek symbols.

Hereafter, the vector is almost always a column vector and a row vector is expressed as a transpose of a column vector indicated by the superscript $^{\mathrm{T}}$. Vectors $\mathbf{a} = a_1\mathrm{i} + a_2\mathrm{j} + a_3\mathrm{k}$ and $\mathbf{b} = b_1\mathrm{i} + b_2\mathrm{j} + b_3\mathrm{k}$ are expressed as follows

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

the transpose of the column vectors $\mathbf{a}$ and $\mathbf{b}$ are represented as follows

$$\mathbf{a}^{\mathrm{T}} = [a_1, a_2, a_3], \quad \mathbf{b}^{\mathrm{T}} = [b_1, b_2, b_3],$$

The magnitude of a vector $\mathbf{a}$ is defined as $|\mathbf{a}| = (\mathbf{a} \cdot \mathbf{a})^{\frac{1}{2}} = ({a_1}^2 + {a_2}^2 + {a_3}^2)^{\frac{1}{2}}$.

A tensor is represented as follows

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad \mathbf{A}^{\mathrm{T}} = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix}.$$

If $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$, the tensor is said to be symmetric, that is, its components are symmetric about the diagonal.

The dot product of two vectors $\mathbf{a}$ and $\mathbf{b}$ (also known as scalar product of two vectors), yields to a scalar quantity and is given by

$$\mathbf{a}^{\mathrm{T}} \cdot \mathbf{b} = \mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} a_1, a_2, a_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = a_1 b_1 + a_2 b_2 + a_3 b_3.$$

The dot product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is commutative ($\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$).

The cross product of two vectors $\mathbf{a} \times \mathbf{b}$ (also known as vector product of two vectors), is the vector normal to the plane of $\mathbf{a}$ and $\mathbf{b}$, and is defined by the determinant

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathrm{i} & \mathrm{j} & \mathrm{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix},$$

$\mathbf{a} \times \mathbf{b}$ and $\mathbf{b} \times \mathbf{a}$ result in two different vectors, pointing in opposite directions with the same magnitude ($\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$).

The tensor product (also known as dyadic product) of two vectors $\mathbf{a} \otimes \mathbf{b}$ produces a second rank tensor and is defined by

$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^{\mathrm{T}} = \mathbf{a}\mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} [b_1, b_2, b_3] = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix},$$

notice that unlike the dot product, the tensor product of two vectors is non-commutative ($\mathbf{a} \otimes \mathbf{b} \neq \mathbf{b} \otimes \mathbf{a}$).

The double dot product ($\mathbf{:}$) of two second rank tensors $\mathbf{A}$ and $\mathbf{B}$ (also known as scalar product of two second rank tensors)

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix},$$

produces a scalar $\phi = \mathbf{A}\mathbf{:}\mathbf{B}$, which can be evaluated as the sum of the 9 products of the tensor components

$$\begin{aligned} \phi = A_{ij} B_{ij} = {} & A_{11} B_{11} + A_{12} B_{12} + A_{13} B_{13} + \\ & A_{21} B_{21} + A_{22} B_{22} + A_{23} B_{23} + \\ & A_{31} B_{31} + A_{32} B_{32} + A_{33} B_{33}. \end{aligned}$$

The double dot product of two second rank tensors is commutative ($\mathbf{A}\mathbf{:}\mathbf{B} = \mathbf{B}\mathbf{:}\mathbf{A}$) .

The dot product of a tensor $\mathbf{A}$ and a vector $\mathbf{a}$, produces a vector $\mathbf{b} = \mathbf{A} \cdot \mathbf{a}$, whose components are

$$\mathbf{b} = b_i = A_{ij} a_j = \begin{bmatrix} A_{11} a_1 + A_{12} a_2 + A_{13} a_3 \\ A_{21} a_1 + A_{22} a_2 + A_{23} a_3 \\ A_{31} a_1 + A_{32} a_2 + A_{33} a_3 \end{bmatrix}.$$

The dot product of a non symmetric tensor $\mathbf{A}$ and a vector $\mathbf{a}$ is non-commutative ($A_{ij} a_j \neq a_i A_{ij}$). If the tensor $\mathbf{A}$ is symmetric then $\mathbf{b} = \mathbf{a} \cdot \mathbf{A} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{a}$.

The dot product of two tensors $\mathbf{A}$ and $\mathbf{B}$ (also known as single dot product or tensor product of two tensors), produces another second rank tensor $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$, whose components are evaluated as

$$\mathbf{C} = C_{ij} = A_{ik} B_{kj}$$

The dot product of two tensors is non-commutative ($\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$).

Note that our definitions of the tensor-vector dot product and tensor-tensor dot-product are consistent with the ordinary rules of matrix algebra.

The trace of a tensor $\mathbf{A}$ is a scalar, evaluated by summing its diagonal components

$$\operatorname{tr} \mathbf{A} = \mathbf{A}^{\mathrm{tr}} = A_{11} + A_{22} + A_{33}.$$

The gradient operator ∇ (read as nabla) in Cartesian coordinates is defined by

$$\nabla = \frac{\partial}{\partial x}\mathrm{i} + \frac{\partial}{\partial y}\mathrm{j} + \frac{\partial}{\partial z}\mathrm{k} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^{\mathrm{T}}.$$

The gradient operator ∇ when applied to a scalar quantity $\phi(x, y, z)$ (where x,y, z are the spatial coordinates), yields to a vector defined by

$$\nabla\phi = \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z}\right)^{\mathrm{T}}.$$

The notation *grad* for ∇ may be also used as the gradient operator, so that, $grad\,\phi \equiv \nabla\phi$. The gradient of a vector $\mathbf{a}$ produces a second rank tensor

$$grad\,\mathbf{a} = \nabla\mathbf{a} = \begin{bmatrix} \dfrac{\partial a_1}{\partial x} & \dfrac{\partial a_1}{\partial y} & \dfrac{\partial a_1}{\partial z} \\ \dfrac{\partial a_2}{\partial x} & \dfrac{\partial a_2}{\partial y} & \dfrac{\partial a_2}{\partial z} \\ \dfrac{\partial a_3}{\partial x} & \dfrac{\partial a_3}{\partial y} & \dfrac{\partial a_3}{\partial z} \end{bmatrix}.$$

The gradient can operate on any tensor field to produce a tensor field that is one rank higher.

The dot product of vector $\mathbf{a}$ and the operator ∇ is called the divergence (*div*) of the vector field; the output of this operator is a scalar and is defined as

$$div\,\mathbf{a} = \nabla \cdot \mathbf{a} = \frac{\partial a_1}{\partial x} + \frac{\partial a_2}{\partial y} + \frac{\partial a_3}{\partial z}.$$

The divergence of a tensor $\mathbf{A}$, *div* $\mathbf{A}$ or $\nabla \cdot \mathbf{A}$, yields to a vector and is defined as

$$div\,\mathbf{A} = \nabla \cdot \mathbf{A} = \begin{bmatrix} \dfrac{\partial A_{11}}{\partial x} + \dfrac{\partial A_{12}}{\partial y} + \dfrac{\partial A_{13}}{\partial z} \\ \dfrac{\partial A_{21}}{\partial x} + \dfrac{\partial A_{22}}{\partial y} + \dfrac{\partial A_{23}}{\partial z} \\ \dfrac{\partial A_{31}}{\partial x} + \dfrac{\partial A_{32}}{\partial y} + \dfrac{\partial A_{33}}{\partial z} \end{bmatrix}.$$

The divergence can operate on any tensor field of rank 1 and above to produce a tensor that is one rank lower.

The *curl* operator of a vector $\mathbf{a}$ produces another vector. This operator is defined by

$$curl\,\mathbf{a} = \nabla \times \mathbf{a} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \dfrac{\partial}{\partial x} & \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial z} \\ a_1 & a_2 & a_3 \end{vmatrix} = \left(\frac{\partial a_3}{\partial y} - \frac{\partial a_2}{\partial z}, \frac{\partial a_1}{\partial z} - \frac{\partial a_3}{\partial x}, \frac{\partial a_2}{\partial x} - \frac{\partial a_1}{\partial y}\right)^{\mathrm{T}}.$$

The divergence of the gradient is called the Laplacian operator and is denoted by Δ. The Laplacian of a scalar $\phi(x, y, z)$ yields to another scalar field and is defined as

$$div\, grad\, \phi = \nabla \cdot \nabla \phi = \nabla^2 \phi = \Delta \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}.$$

The Laplacian of a vector field $\mathbf{a}$ is defined as the diverge of the gradient just like in the scalar case, but in this case it yields to a vector field, such that

$$div\, grad\, \mathbf{a} = \nabla \cdot \nabla \mathbf{a} = \nabla^2 \mathbf{a} = \Delta \mathbf{a} = \begin{bmatrix} \frac{\partial^2 a_1}{\partial x^2} + \frac{\partial^2 a_1}{\partial y^2} + \frac{\partial^2 a_1}{\partial z^2} \\ \frac{\partial^2 a_2}{\partial x^2} + \frac{\partial^2 a_2}{\partial y^2} + \frac{\partial^2 a_2}{\partial z^2} \\ \frac{\partial^2 a_3}{\partial x^2} + \frac{\partial^2 a_3}{\partial y^2} + \frac{\partial^2 a_3}{\partial z^2} \end{bmatrix}.$$

The Laplacian transforms a tensor field into another tensor field of the same rank.

As previously discussed, a tensor is said to be symmetric if its components are symmetric about the diagonal, *i.e.*, $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$. A tensor is said to be skew or anti-symmetric if $\mathbf{A} = -\mathbf{A}^{\mathrm{T}}$ which intuitively implies that $A_{11} = A_{22} = A_{33} = 0$. Every second rank tensor can be decomposed into symmetric and skew parts by

$$\mathbf{A} = \underbrace{\frac{1}{2}\left(\mathbf{A} + \mathbf{A}^{\mathrm{T}}\right)}_{symmetric} + \underbrace{\frac{1}{2}\left(\mathbf{A} - \mathbf{A}^{\mathrm{T}}\right)}_{skew} = \mathrm{symm}\, \mathbf{A} + \mathrm{skew}\, \mathbf{A}.$$

The jacobian matrix of a vector field $\mathbf{a}$ is given by

$$\begin{bmatrix} \frac{\partial a_1}{\partial x} & \frac{\partial a_1}{\partial y} & \frac{\partial a_1}{\partial z} \\ \frac{\partial a_2}{\partial x} & \frac{\partial a_2}{\partial y} & \frac{\partial a_2}{\partial z} \\ \frac{\partial a_3}{\partial x} & \frac{\partial a_3}{\partial y} & \frac{\partial a_3}{\partial z} \end{bmatrix}.$$

The identity matrix or unit matrix, is a matrix whose diagonal entries are all 1 and the other entries are 0. The 3×3 identity matrix $\mathbf{I}$ is given by

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Hereafter we present some useful vector/tensor identities:

- $\nabla \cdot \nabla \times \mathbf{a} = 0$.
- $\nabla \times \nabla \alpha = 0$.
- $\nabla(\alpha\beta) = \alpha \nabla \beta + \beta \nabla \alpha$.
- $\nabla(\alpha \mathbf{a}) = \mathbf{a} \otimes \nabla \alpha + \alpha \nabla \mathbf{a}$.
- $(\nabla \mathbf{a})\mathbf{a} = \nabla \frac{\mathbf{a} \cdot \mathbf{a}}{2} - \mathbf{a} \times \nabla \times \mathbf{a}$.
- $\mathbf{a} \cdot (\nabla \mathbf{a})\mathbf{a} = \mathbf{a} \cdot \nabla \frac{\mathbf{a} \cdot \mathbf{a}}{2}$.
- $(\mathbf{a} \otimes \mathbf{b}) \cdot \nabla \mathbf{a} = \mathbf{b} \cdot \nabla \frac{\mathbf{a} \cdot \mathbf{a}}{2}$.
- $\nabla(\mathbf{a} \cdot \mathbf{b}) = \mathbf{b} \cdot \nabla \mathbf{a} + \mathbf{a} \cdot \nabla \mathbf{b} + \mathbf{a} \times \nabla \times \mathbf{b} + \mathbf{b} \times \nabla \times \mathbf{a}$.

- $\nabla \cdot (\alpha \mathbf{a}) = \alpha \nabla \cdot \mathbf{a} + \mathbf{a} \cdot \nabla \alpha$.
- $\nabla \cdot \nabla \mathbf{a} = \nabla(\nabla \cdot \mathbf{a}) - \nabla \times (\nabla \times \mathbf{a})$.
- $\nabla \cdot (\mathbf{a} \times \mathbf{b}) = \mathbf{b} \cdot \nabla \times \mathbf{a} - \mathbf{a} \cdot \nabla \times \mathbf{b}$.
- $\nabla \cdot (\mathbf{a} \otimes \mathbf{b}) = \mathbf{b} \cdot \nabla \mathbf{a} + \mathbf{a} \nabla \cdot \mathbf{b}$.
- $\mathbf{a} \cdot \nabla \cdot (\mathbf{b} \otimes \mathbf{c}) = (\mathbf{a} \cdot \mathbf{b}) \nabla \cdot \mathbf{c} + (\mathbf{a} \otimes \mathbf{b}) \cdot \nabla \mathbf{b}$.
- $\nabla \cdot (\alpha \mathbf{A}) = \mathbf{A} \nabla \alpha + \alpha \nabla \cdot \mathbf{A}$.
- $\nabla \cdot (\mathbf{A} \mathbf{b}) = (\nabla \cdot \mathbf{A}^{\mathrm{T}}) \cdot \mathbf{b} + \mathbf{A}^{\mathrm{T}} \cdot \nabla \mathbf{b}$.
- $\nabla \times (\alpha \mathbf{a}) = \alpha \nabla \times \mathbf{a} + \nabla \alpha \times \mathbf{a}$.
- $\nabla \times (\mathbf{a} \times \mathbf{b}) = \mathbf{a} \nabla \cdot \mathbf{b} + \mathbf{b} \cdot \nabla \mathbf{a} - (\nabla \cdot \mathbf{a}) \mathbf{b} - \mathbf{a} \cdot \nabla \mathbf{b}$.
- $\mathbf{a} \cdot (\mathbf{A} \mathbf{b}) = \mathbf{A} \cdot (\mathbf{a} \otimes \mathbf{b})$.
- $\mathbf{a} \cdot (\mathbf{A} \mathbf{b}) = (\mathbf{A} \mathbf{a}) \cdot \mathbf{b}$ if $\mathbf{A}$ is symmetric.
- $\mathbf{ab{:}A} = \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{A})$
- $\mathbf{A{:}ab} = (\mathbf{A} \cdot \mathbf{a}) \cdot \mathbf{b}$

where α and β are scalars; $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ are vectors; and $\mathbf{A}$ is a tensor.

Chapter 2

Governing Equations of Fluid Dynamics

The starting point of any numerical simulation are the governing equations of the physics of the problem to be solved. Hereafter, we present the governing equations of fluid dynamics and their simplification for the case of an incompressible viscous flow.

The equations governing the motion of a fluid can be derived from the statements of the conservation of mass, momentum, and energy [1, 2, 3]. In the most general form, the fluid motion is governed by the time-dependent three-dimensional compressible Navier-Stokes system of equations. For a viscous Newtonian, isotropic fluid in the absence of external forces, mass diffusion, finite-rate chemical reactions, and external heat addition; the conservation form of the Navier-Stokes system of equations in compact differential form and in primitive variable formulation (ρ, u, v, w, e_t) can be written as

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) &= -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{S}_u \\ \frac{\partial (\rho e_t)}{\partial t} + \nabla \cdot (\rho e_t \mathbf{u}) &= \nabla \cdot q - \nabla \cdot (p\mathbf{u}) + \boldsymbol{\tau} \mathbf{:} \nabla \mathbf{u} + \mathbf{S}_e \end{aligned} \tag{2.0.1}$$

where $\boldsymbol{\tau}$ is the viscous stress tensor and is given by

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}. \tag{2.0.2}$$

For the sake of completeness, let us recall that in the conservation form (or divergence form) [4], the momentum equation can be written as

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau}, \tag{2.0.3}$$

where the tensor product of the vectors $\mathbf{uu}$ in eq. 2.0.3 is equal to

$$\mathbf{u}\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \begin{bmatrix} u & v & w \end{bmatrix} = \begin{bmatrix} u^2 & uv & uw \\ vu & v^2 & vw \\ wu & wv & w^2 \end{bmatrix}. \tag{2.0.4}$$

Let us recall the following identity

$$\nabla \cdot (\mathbf{uu}) = \mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{u}(\nabla \cdot \mathbf{u}), \tag{2.0.5}$$

and from the divergence-free constraint ($\nabla \cdot \mathbf{u} = 0$) it follows that $\mathbf{u}(\nabla \cdot \mathbf{u})$ is zero, therefore $\nabla \cdot (\mathbf{uu}) = \mathbf{u} \cdot \nabla \mathbf{u}$. Henceforth, the conservation form of the momentum equation eq. 2.0.3 is equivalent to

$$\rho \left(\frac{\partial (\mathbf{u})}{\partial t} + \mathbf{u} \cdot \nabla (\mathbf{u}) \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau},$$

which is the non-conservation form or advective/convective form of the momentum equation.

The set of equations 2.0.1 can be rewritten in vector form as follows

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{e_i}}{\partial x} + \frac{\partial \mathbf{f_i}}{\partial y} + \frac{\partial \mathbf{g_i}}{\partial z} = \frac{\partial \mathbf{e_v}}{\partial x} + \frac{\partial \mathbf{f_v}}{\partial y} + \frac{\partial \mathbf{g_v}}{\partial z}, \tag{2.0.6}$$

where $\mathbf{q}$ is the vector of the conserved flow variables given by

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \end{bmatrix}, \tag{2.0.7}$$

and $\mathbf{e_i}$, $\mathbf{f_i}$ and $\mathbf{g_i}$ are the vectors containing the inviscid fluxes (or convective fluxes) in the x, y and z directions and are given by

$$\mathbf{e_i} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho e_t + p)\, u \end{bmatrix}, \qquad \mathbf{f_i} = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (\rho e_t + p)\, v \end{bmatrix}, \qquad \mathbf{g_i} = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (\rho e_t + p)\, w \end{bmatrix}, \tag{2.0.8}$$

where $\mathbf{u}$ is the velocity vector containing the u, v and w velocity components in the x, y and z directions and p, ρ and e_t are the pressure, density and total energy per unit mass respectively.

The vectors $\mathbf{e_v}$, $\mathbf{f_v}$ and $\mathbf{g_v}$ contain the viscous fluxes (or diffusive fluxes) in the x, y and z directions and are defined as follows

$$\begin{aligned}
\mathbf{e_v} &= \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix}, \\
\mathbf{f_v} &= \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix}, \\
\mathbf{g_v} &= \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix},
\end{aligned} \tag{2.0.9}$$

where the heat fluxes q_x, q_y and q_z are given by the Fourier's law of heat conduction as follows

$$\begin{aligned} q_x &= -k\,\frac{\partial T}{\partial x}, \\ q_y &= -k\,\frac{\partial T}{\partial y}, \\ q_z &= -k\,\frac{\partial T}{\partial z}, \end{aligned} \tag{2.0.10}$$

and the viscous stresses τ_{xx}, τ_{yy}, τ_{zz}, τ_{xy}, τ_{yx}, τ_{xz}, τ_{zx}, τ_{yz} and τ_{zy}, are given by the following relationships

$$\begin{aligned} \tau_{xx} &= \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right), \\ \tau_{yy} &= \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}\right), \\ \tau_{zz} &= \frac{2}{3}\mu\left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \\ \tau_{xy} &= \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \\ \tau_{xz} &= \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right), \\ \tau_{yz} &= \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right), \end{aligned} \tag{2.0.11}$$

In equations 2.0.9-2.0.11, T is the temperature, k is the thermal conductivity and μ is the molecular viscosity. In order to derive the viscous stresses in eq. 2.0.11 the Stokes hypothesis was used [5, 1, 6, 7].

Examining closely equations 2.0.6-2.0.11 and counting the number of equations and unknowns, we clearly see that we have five equations in terms of seven unknown flow field variables u, v, w, ρ, p, T, and e_t. It is obvious that two additional equations are required to close the system. These two additional equations can be obtained by determining relationships that exist between the thermodynamic variables (p, ρ, T, e_i) through the assumption of thermodynamic equilibrium. Relations of this type are known as equations of state, and they provide a mathematical relationship between two or more state functions (thermodynamic variables). Choosing the specific internal energy e_i and the density ρ as the two independent thermodynamic variables, then equations of state of the form

$$p = p\left(e_i, \rho\right), \qquad T = T\left(e_i, \rho\right), \tag{2.0.12}$$

are required. For most problems in aerodynamics and gasdynamics, it is generally reasonable to assume that the gas behaves as a perfect gas (a perfect gas is defined as a gas whose intermolecular forces are negligible), *i.e.*,

$$p = \rho R_g T, \tag{2.0.13}$$

where R_g is the specific gas constant and is equal to $287\frac{m^2}{s^2 K}$ for air. Assuming also that the working gas behaves as a calorically perfect gas (a calorically perfect gas is defined as a perfect gas with constant specific heats), then the following relations hold

$$e_i = c_v T, \qquad h = c_p T, \qquad \gamma = \frac{c_p}{c_v}, \qquad c_v = \frac{R_g}{\gamma - 1}, \qquad c_p = \frac{\gamma R_g}{\gamma - 1}, \tag{2.0.14}$$

where γ is the ratio of specific heats and is equal to 1.4 for air, c_v the specific heat at constant volume, c_p the specific heat at constant pressure and h is the enthalpy. By using eq. 2.0.13 and eq. 2.0.14, we obtain the following relations for pressure p and temperature T in the form of eq. 2.0.12

$$p = (\gamma - 1)\,\rho e_i, \qquad T = \frac{p}{\rho R_g} = \frac{(\gamma - 1)\,e_i}{R_g}, \tag{2.0.15}$$

where the specific internal energy per unit mass $e_i = p/(\gamma - 1)\rho$ is related to the total energy per unit mass e_t by the following relationship,

$$e_t = e_i + \frac{1}{2}\left(u^2 + v^2 + w^2\right). \tag{2.0.16}$$

In our discussion, it is also necessary to relate the transport properties (μ, k) to the thermodynamic variables. Then, the molecular viscosity μ is computed using Sutherland's formula

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{(T + C_2)}, \tag{2.0.17}$$

where for the case of the air, the constants are $C_1 = 1.458 \times 10^{-6} \frac{kg}{ms\sqrt{K}}$ and $C_2 = 110.4K$.

The thermal conductivity, k, of the fluid is determined from the Prandtl number ($Pr = 0.72$ for air) which in general is assumed to be constant and is equal to

$$k = \frac{c_p \mu}{Pr}, \tag{2.0.18}$$

where c_p and μ are given by equations eq. 2.0.14 and eq. 2.0.17 respectively.

When dealing with high speed compressible flows, it is also useful to introduce the Mach number. The mach number is a non dimensional parameter that measures the speed of the gas motion in relation to the speed of sound a,

$$a = \left[\left(\frac{\partial p}{\partial \rho}\right)_S\right]^{\frac{1}{2}} = \sqrt{\gamma \frac{p}{\rho}} = \sqrt{\gamma R_g T}. \tag{2.0.19}$$

Then the Mach number M_∞ is given by,

$$M_\infty = \frac{U_\infty}{a} = \frac{U_\infty}{\sqrt{\gamma(p/\rho)}} = \frac{U_\infty}{\sqrt{\gamma R_g T}} \tag{2.0.20}$$

Another useful non dimensional quantity is the Reynold's number, this quantity represents the ratio of inertia forces to viscous forces and is given by,

$$Re = \frac{\rho_\infty U_\infty L}{\mu_\infty}, \tag{2.0.21}$$

where the subscript ∞ denotes freestream conditions, L is a reference length (such as the chord of an airfoil or the length of a vehicle), and μ_∞ is computed using the freestream temperature T_∞ according to eq 2.0.17.

The first row in eq. 2.0.6 corresponds to the continuity equation. Likewise, the second, third and fourth rows are the momentum equations, while the fifth row is the energy equation in terms of total energy per unit mass.

The Navier-Stokes system of equations 2.0.6-2.0.9, is a coupled system of nonlinear partial differential equations (PDE), and hence is very difficult to solve analytically. In fact, to the date there

is no general closed-form solution to this system of equations; hence we look for an approximate solution of this system of equation in a given domain $\mathcal{D}$ with prescribed boundary conditions $\partial\mathcal{D}$ and given initial conditions $\mathcal{D}\mathring{\mathbf{q}}$.

If in eq. 2.0.6 we set the viscous fluxes $\mathbf{e}_{\boldsymbol{v}} = 0$, $\mathbf{f}_{\boldsymbol{v}} = 0$ and $\mathbf{g}_{\boldsymbol{v}} = 0$, we get the Euler system of equations, which governs inviscid fluid flow. The Euler system of equations is a set of hyperbolic equations while the Navier-Stokes system of equations is a mixed set of hyperbolic (in the inviscid region) and parabolic (in the viscous region) equations. Therefore, time marching algorithms are used to advance the solution in time using discrete time steps.

2.1 Simplification of the Navier-Stokes System of Equations: Incompressible Viscous Flow Case

Equations 2.0.6-2.0.9, with an appropriate equation of state and boundary and initial conditions, governs the unsteady three-dimensional motion of a viscous Newtonian, compressible fluid. In many applications the fluid density may be assumed to be constant. This is true not only for liquids, whose compressibility may be neglected, but also for gases if the Mach number is below 0.3 [2, 8]; such flows are said to be incompressible. If the flow is also isothermal, the viscosity is also constant. In this case, the governing equations written in compact conservation differential form and in primitive variable formulation (u, v, w, p) reduce to the following set

$$\begin{aligned} \nabla \cdot (\mathbf{u}) &= 0, \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) &= \frac{-\nabla p}{\rho} + \nu \nabla^2 \mathbf{u}, \end{aligned} \tag{2.1.1}$$

where ν is the kinematic viscosity and is equal $\nu = \mu/\rho$. The previous set of equations in expanded three-dimensional Cartesian coordinates is written as follows

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0, \\ \frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right), \\ \frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial vw}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right), \\ \frac{\partial w}{\partial t} + \frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial w^2}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right). \end{aligned} \tag{2.1.2}$$

Equation 2.1.2 governs the unsteady three-dimensional motion of a viscous, incompressible and isothermal flow. This simplification is generally not of a great value, as the equations are hardly any simpler to solve. However, the computing effort may be much smaller than for the full equations (due to the reduction of the unknowns and the fact that the energy equation is decoupled from the system of equation), which is a justification for such a simplification. The set of equations 2.1.1 can be rewritten in vector form as follow

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{e}_{\boldsymbol{i}}}{\partial x} + \frac{\partial \mathbf{f}_{\boldsymbol{i}}}{\partial y} + \frac{\partial \mathbf{g}_{\boldsymbol{i}}}{\partial z} = \frac{\partial \mathbf{e}_{\boldsymbol{v}}}{\partial x} + \frac{\partial \mathbf{f}_{\boldsymbol{v}}}{\partial y} + \frac{\partial \mathbf{g}_{\boldsymbol{v}}}{\partial z}, \tag{2.1.3}$$

where $\mathbf{q}$ is the vector containing the primitive variables and is given by

$$\mathbf{q} = \begin{bmatrix} 0 \\ u \\ v \\ w \end{bmatrix}, \tag{2.1.4}$$

and $\mathbf{e_i}$, $\mathbf{f_i}$ and $\mathbf{g_i}$ are the vectors containing the inviscid fluxes (or convective fluxes) in the x, y and z directions and are given by

$$\mathbf{e_i} = \begin{bmatrix} u \\ u^2 + p \\ uv \\ uw \end{bmatrix}, \qquad \mathbf{f_i} = \begin{bmatrix} v \\ vu \\ v^2 + p \\ vw \end{bmatrix}, \qquad \mathbf{g_i} = \begin{bmatrix} w \\ wu \\ wv \\ w^2 + p \end{bmatrix}. \tag{2.1.5}$$

The viscous fluxes (or diffusive fluxes) in the x, y and z directions, $\mathbf{e_v}$, $\mathbf{f_v}$ and $\mathbf{g_v}$ respectively, are defined as follows

$$\mathbf{e_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \end{bmatrix}, \qquad \mathbf{f_v} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \end{bmatrix}, \qquad \mathbf{g_v} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \end{bmatrix}. \tag{2.1.6}$$

Since we made the assumptions of an incompressible flow, appropriate expressions for shear stresses must be used, these expressions are given as follows

$$\begin{aligned} \tau_{xx} &= 2\mu \frac{\partial u}{\partial x}, \\ \tau_{yy} &= 2\mu \frac{\partial v}{\partial y}, \\ \tau_{zz} &= 2\mu \frac{\partial w}{\partial z}, \\ \tau_{xy} = \tau_{yx} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \\ \tau_{xz} = \tau_{zx} &= \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right), \\ \tau_{yz} = \tau_{zy} &= \mu \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right), \end{aligned} \tag{2.1.7}$$

where we used Stokes hypothesis [5, 1, 6, 7] in order to derive the viscous stresses in eq. 2.1.7.

Equation 2.1.7 can be written in compact vector form as $\boldsymbol{\tau} = 2\mu\mathbf{D}$, where $\mathbf{D} = \frac{1}{2}\left[\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right]$ is the symmetric tensor of the velocity gradient tensor $\nabla\mathbf{u} = [\mathbf{D} + \mathbf{S}]$, and where $\mathbf{D}$ represents the strain-rate tensor and $\mathbf{S}$ represents the spin tensor (vorticity). The skew or anti-symmetric part of the velocity gradient tensor is given by $\mathbf{S} = \frac{1}{2}\left[\nabla\mathbf{u} - \nabla\mathbf{u}^{\mathrm{T}}\right]$.

Equations 2.1.3-2.1.6, are the governing equations of an incompressible, isothermal, viscous flow written in conservation form. Hence, we look for an approximate solution of this set of equations in a given domain $\mathcal{D}$ with prescribed boundary conditions $\partial\mathcal{D}$ and given initial conditions $\mathcal{D}\mathring{\mathbf{q}}$.

Chapter 3

Turbulence Modeling

All flows encountered in engineering applications, from simple ones to complex three-dimensional ones, become unstable above a certain Reynolds number ($Re = UL/\nu$ where U and L are characteristic velocity and length scales of the mean flow and ν is the kinematic viscosity). At low Reynolds numbers flows are laminar, but as we increase the Reynolds number, flows are observed to become turbulent. Turbulent flows are characterize by a chaotic and random state of motion in which the velocity and pressure change continuously on a broad range of time and length scales (from the smallest turbulent eddies characterized by Kolmogorov micro-scales, to the flow features comparable with the size of the geometry).

There are several possible approaches for the numerical simulation of turbulent flows. The first and most intuitive one, is by directly numerically solving the governing equations over the whole range of turbulent scales (temporal and spatial). This deterministic approach is referred as Direct Numerical Simulation (DNS) [9, 10, 11, 12, 13, 14]. In DNS, a fine enough mesh and small enough time-step size must be used so that all of the turbulent scales are resolved. Although some simple problems have been solved using DNS, it is not possible to tackle industrial problems due to the prohibitive computer cost imposed by the mesh and time-step requirements. Hence, this approach is mainly used for benchmarking, research and academic applications.

Another approach used to model turbulent flows is Large Eddy Simulation (LES) [15, 16, 9, 13, 14]. Here, large scale turbulent structures are directly simulated whereas the small turbulent scales are filtered out and modeled by turbulence models called subgrid scale models. According to turbulence theory, small scale eddies are more uniform and have more or less common characteristics; therefore, modeling small scale turbulence appears more appropriate, rather than resolving it. The computational cost of LES is less than that of DNS, since the small scale turbulence is now modeled, hence the grid spacing is much larger than the Kolmogorov length scale. In LES, as the mesh gets finer, the number of scales that require modeling becomes smaller, thus approaching DNS. Thanks to the advances in computing hardware and parallel algorithms, the use of LES for industrial problems is becoming practical.

Today's workhorse for industrial and research turbulence modeling applications is the Reynolds Averaged Navier-Stokes (RANS) approach [17, 18, 13, 12, 19, 14]. In this approach, the RANS equations are derived by decomposing the flow variables of the governing equations into time-mean (obtained over an appropriate time interval) and fluctuating part, and then time averaging the entire equations. Time averaging the governing equations gives rise to new terms, these new quantities must be related to the mean flow variables through turbulence models. This process introduces further assumptions and approximations. The turbulence models are primarily developed based on experimental data obtained from relatively simple flows under controlled conditions. This in turn limits the range of applicability of the turbulence models. That is, no

single RANS turbulence model is capable of providing accurate solutions over a wide range of flow conditions and geometries.

Two types of averaging are presently used, the classical Reynolds averaging which gives rise to the RANS equations and the mass-weighted averaging or Favre averaging which is used to derive the Favre-Averaged Navier-Stokes equations (FANS) for compressible flows applications. In both statistical approaches, all the turbulent scales are modeled, hence mesh and time-step requirements are not as restrictive as in LES or DNS. Hereafter, we limit our discussion to Reynolds averaging.

3.1 Reynolds Averaging

The starting point for deriving the RANS equations is the Reynolds decomposition [17, 3, 13, 12, 19, 14] of the flow variables of the governing equations. This decomposition is accomplished by representing the instantaneous flow quantity ϕ by the sum of a mean value part (denoted by a bar over the variable, as in $\bar{\phi}$) and a time-dependent fluctuating part (denoted by a prime, as in ϕ'). This concept is illustrated in figure 3.1 and is mathematically expressed as follows,

$$\phi(\mathbf{x},t) = \underbrace{\bar{\phi}(\mathbf{x})}_{mean\,value} + \underbrace{\phi'(\mathbf{x},t)}_{fluctuating\,part} \quad . \tag{3.1.1}$$

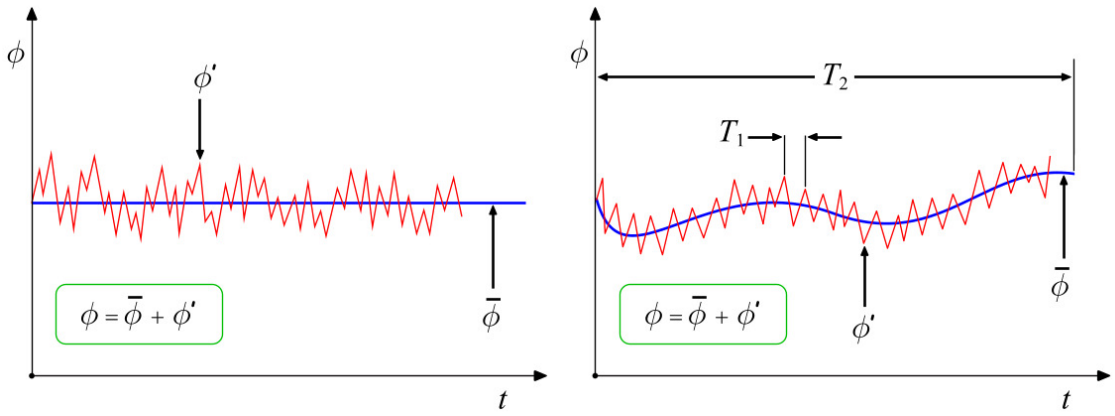

Figure 3.1: *Time averaging for a statistically steady turbulent flow (left) and time averaging for an unsteady turbulent flow (right).*

Hereafter, $\mathbf{x}$ is the vector containing the Cartesian coordinates x, y, and z in $\mathbb{N} = 3$ (where $\mathbb{N}$ is equal to the number of spatial dimensions). A key observation in eq. 3.1.1 is that $\bar{\phi}$ is independent of time, implying that any equation deriving for computing this quantity must be steady state.

In eq. 3.1.1, the mean value $\bar{\phi}$ is obtained by an averaging procedure. There are three different forms of the Reynolds averaging:

1. Time averaging: appropriate for stationary turbulence, *i.e.*, statically steady turbulence or a turbulent flow that, on average, does not vary with time.

$$\bar{\phi}(\mathbf{x}) = \lim_{T\to+\infty} \frac{1}{T} \int_t^{t+T} \phi(\mathbf{x},t)\,\mathrm{d}t, \tag{3.1.2}$$

here t is the time and T is the averaging interval. This interval must be large compared to the typical time scales of the fluctuations; thus, we are interested in the limit $T \to \infty$. As a consequence, $\bar{\phi}$ does not vary in time, but only in space.

2. Spatial averaging: appropriate for homogeneous turbulence.

$$\bar{\phi}(t) = \lim_{\mathcal{CV}\to\infty} \frac{1}{\mathcal{CV}} \int_{\mathcal{CV}} \phi(\mathbf{x}, t)\, \mathrm{d}\mathcal{CV}, \tag{3.1.3}$$

with $\mathcal{CV}$ being a control volume. In this case, $\bar{\phi}$ is uniform in space, but it is allowed to vary in time.

3. Ensemble averaging: appropriate for unsteady turbulence.

$$\bar{\phi}(\mathbf{x}, t) = \lim_{\mathcal{N}\to\infty} \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \bar{\phi}(\mathbf{x}, t), \tag{3.1.4}$$

where $\mathcal{N}$, is the number of experiments of the ensemble and must be large enough to eliminate the effects of fluctuations. This type of averaging can be applied to any flow (steady or unsteady). Here, the mean value $\bar{\phi}$ is a function of both time and space (as illustrated in figure 3.1).

We use the term Reynolds averaging to refer to any of these averaging processes, applying any of them to the governing equations yields to the Reynolds-Averaged Navier-Stokes (RANS) equations. In cases where the turbulent flow is both stationary and homogeneous, all three averaging are equivalent. This is called the ergodic hypothesis.

If the mean flow $\bar{\phi}$ varies slowly in time, we should use an unsteady approach (URANS); then, equations eq. 3.1.1 and eq. 3.1.2 can be modified as

$$\phi(\mathbf{x}, t) = \bar{\phi}(\mathbf{x}, t) + \phi'(\mathbf{x}, t), \tag{3.1.5}$$

and

$$\bar{\phi}(\mathbf{x}, t) = \frac{1}{T} \int_{t}^{t+T} \phi(\mathbf{x}, t)\mathrm{d}t, \quad T_1 << T << T_2, \tag{3.1.6}$$

where T_1 and T_2 are the characteristics time scales of the fluctuations and the slow variations in the flow, respectively (as illustrated in figure 3.1). In eq. 3.1.6 the time scales should differ by several order of magnitude, but in engineering applications very few unsteady flows satisfy this condition. In general, the mean and fluctuating components are correlated, *i.e.*, the time average of their product is non-vanishing. For such problems, ensemble averaging is necessary. An alternative approach to URANS is LES, which is out of the scope of this discussion but the interested reader should refer to references [15, 16, 9, 13, 14].

Before deriving the RANS equations, we recall the following averaging rules,

$$
\begin{aligned}
\bar{\phi'} &= 0, \\
\bar{\bar{\phi}} &= \bar{\phi}, \\
\overline{\bar{\phi}} = \overline{\bar{\phi} + \phi'} &= \bar{\phi}, \\
\overline{\phi + \varphi} &= \bar{\phi} + \bar{\varphi}\,, \\
\overline{\bar{\phi}\varphi} = \bar{\bar{\phi}}\bar{\varphi} &= \bar{\phi}\bar{\varphi}, \\
\overline{\bar{\phi}\varphi'} = \bar{\phi}\bar{\varphi'} &= 0, \\
\overline{\phi\varphi} &= \overline{(\bar{\phi} + \phi')(\bar{\varphi} + \varphi')} \\
&= \overline{\bar{\phi}\bar{\varphi} + \bar{\phi}\varphi' + \bar{\varphi}\phi' + \phi'\varphi'} \\
&= \overline{\bar{\phi}\bar{\varphi}} + \overline{\bar{\phi}\varphi'} + \overline{\bar{\varphi}\phi'} + \overline{\phi'\varphi'} \\
&= \bar{\phi}\bar{\varphi} + \overline{\phi'\varphi'}, \\
\overline{\phi'^2} &\neq 0, \\
\overline{\phi'\varphi'} &\neq 0, \\
\overline{\frac{\partial\phi}{\partial x}} &= \frac{\partial\bar{\phi}}{\partial x}, \\
\overline{\int \phi ds} &= \int \bar{\phi} ds
\end{aligned}
\tag{3.1.7}
$$

3.2 Incompressible Reynolds Averaged Navier-Stokes Equations

Let us recall the Reynolds decomposition for the flow variables of the incompressible Navier-Stokes equations eq. 2.1.1,

$$
\begin{aligned}
\mathbf{u}(\mathbf{x}, t) &= \bar{\mathbf{u}}(\mathbf{x}) + \mathbf{u}'(\mathbf{x}, t), \\
p(\mathbf{x}, t) &= \bar{p}(\mathbf{x}) + p'(\mathbf{x}, t),
\end{aligned}
\tag{3.2.1}
$$

we now substitute eq. 3.2.1 into the incompressible Navier-Stokes equations eq. 2.1.1 and we obtain for the continuity equation

$$
\nabla \cdot (\mathbf{u}) = \nabla \cdot \left(\bar{\mathbf{u}} + \mathbf{u}'\right) = \nabla \cdot (\bar{\mathbf{u}}) + \nabla \cdot \left(\mathbf{u}'\right) = 0. \tag{3.2.2}
$$

Then, time averaging this equation results in

$$
\nabla \cdot (\bar{\bar{\mathbf{u}}}) + \nabla \cdot \left(\bar{\mathbf{u}'}\right) = 0, \tag{3.2.3}
$$

and using the averaging rules stated in eq. 3.1.7, it follows that

$$
\nabla \cdot (\bar{\mathbf{u}}) = 0. \tag{3.2.4}
$$

We next consider the momentum equation of the incompressible Navier-Stokes equations eq. 2.1.1. We begin by substituting eq. 3.2.1 into eq. 2.1.1 in order to obtain,

$$
\frac{\partial\left(\bar{\mathbf{u}} + \mathbf{u}'\right)}{\partial t} + \nabla \cdot \left(\left(\bar{\mathbf{u}} + \mathbf{u}'\right)\left(\bar{\mathbf{u}} + \mathbf{u}'\right)\right) = \frac{-\nabla\left(\bar{p} + p'\right)}{\rho} + \nu\nabla^2\left(\bar{\mathbf{u}} + \mathbf{u}'\right), \tag{3.2.5}
$$

by time averaging eq. 3.2.5, expanding and applying the rules set in eq. 3.1.7, we obtain

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot \left(\bar{\mathbf{u}}\bar{\mathbf{u}} + \overline{\mathbf{u}'\mathbf{u}'}\right) = \frac{-\nabla \bar{p}}{\rho} + \nu \nabla^2 \bar{\mathbf{u}}. \tag{3.2.6}$$

or after rearranging,

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}}\bar{\mathbf{u}}) = \frac{-\nabla \bar{p}}{\rho} + \nu \nabla^2 \bar{\mathbf{u}} - \nabla \cdot \left(\overline{\mathbf{u}'\mathbf{u}'}\right) \tag{3.2.7}$$

By setting $\boldsymbol{\tau}^R = -\rho\left(\overline{\mathbf{u}'\mathbf{u}'}\right)$ in equation 3.2.7, and grouping with equation 3.2.4, we obtain the following set of equations,

$$\begin{aligned} \nabla \cdot (\bar{\mathbf{u}}) &= 0, \\ \frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}}\bar{\mathbf{u}}) &= \frac{-\nabla \bar{p}}{\rho} + \nu \nabla^2 \bar{\mathbf{u}} + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau}^R. \end{aligned} \tag{3.2.8}$$

The set of equations eq. 3.2.8 are the incompressible Reynolds-Averaged Navier-Stokes (RANS) equations. Notice that in eq. 3.2.8 we have retained the term $\partial \bar{\mathbf{u}}/\partial t$, despite the fact that $\bar{\mathbf{u}}$ is independent of time for statistically steady turbulence, hence this expression is equal to zero when time average. In practice, in all modern formulations of the RANS equations the time derivative term is included. In references [17, 13, 3, 10, 20], a few arguments justifying the retention of this term are discussed. For not statistically stationary turbulence or unsteady turbulence, a time-dependent RANS or unsteady RANS (URANS) approach is required, an URANS computation simply requires retaining the time derivative term $\partial \bar{\mathbf{u}}/\partial t$ in the computation. For unsteady turbulence, ensemble average is recommended and often necessary.

The incompressible Reynolds-Averaged Navier-Stokes (RANS) equations eq. 3.2.8 are identical to the incompressible Navier-Stokes equations eq. 2.1.1 with the exception of the additional term $\boldsymbol{\tau}^R = -\rho\left(\overline{\mathbf{u}'\mathbf{u}'}\right)$, where $\boldsymbol{\tau}^R$ is the so-called Reynolds-stress tensor (notice that by doing a check of dimensions, it will show that $\boldsymbol{\tau}^R$ it is not actually a stress; it must be multiplied by the density ρ, as it is done consistently in this manuscript, in order to have dimensions corresponding to the stresses. On the other hand, since we are assuming that the flow is incompressible, that is, ρ is constant, we might set the density equal to unity, thus obtaining implicit dimensional correctness. Moreover, because we typically use kinematic viscosity ν, there is an implied division by ρ in any case). The Reynolds-stress tensor represents the transfer of momentum due to turbulent fluctuations. In $3\mathbb{D}$, the Reynolds-stress tensor $\boldsymbol{\tau}^R$ consists of nine components

$$\boldsymbol{\tau}^R = -\rho\left(\overline{\mathbf{u}'\mathbf{u}'}\right) = -\begin{pmatrix} \rho\overline{u'u'} & \rho\overline{u'v'} & \rho\overline{u'w'} \\ \rho\overline{v'u'} & \rho\overline{v'v'} & \rho\overline{v'w'} \\ \rho\overline{w'u'} & \rho\overline{w'v'} & \rho\overline{w'w'} \end{pmatrix}. \tag{3.2.9}$$

However, since u, v and w can be interchanged, the Reynolds-stress tensor forms a symmetrical second order tensor containing only six independent components. By inspecting the set of equations eq. 3.2.8 we can count ten unknowns, namely; three components of the velocity (u, v, w), the pressure (p), and six components of the Reynolds stress $\left(\boldsymbol{\tau}^R = -\rho\left(\overline{\mathbf{u}'\mathbf{u}'}\right)\right)$, in terms of four equations, hence the system is not closed. The fundamental problem of turbulence modeling based on the Reynolds-averaged Navier-Stokes equations is to find six additional relations in order to close the system of equations eq. 3.2.8.

3.3 Boussinesq Approximation

The Reynolds averaged approach to turbulence modeling requires that the Reynolds stresses in eq. 3.2.8 to be appropriately modeled (however, it is possible to derive its own governing

equations, but it is much simpler to model this term). A common approach uses the Boussinesq hypothesis to relate the Reynolds stresses $\boldsymbol{\tau}^R$ to the mean velocity gradients such that

$$\boldsymbol{\tau}^R = -\rho\left(\overline{\mathbf{u}'\mathbf{u}'}\right) = 2\mu_T\bar{\mathbf{D}}^R - \frac{2}{3}\rho k\mathbf{I} = \mu_T\left[\nabla\overline{\mathbf{u}} + (\nabla\overline{\mathbf{u}})^{\mathrm{T}}\right] - \frac{2}{3}\rho k\mathbf{I}, \tag{3.3.1}$$

where $\bar{\mathbf{D}}^R$ denotes the Reynolds-averaged strain-rate tensor ($\frac{1}{2}(\nabla\overline{\mathbf{u}} + \nabla\overline{\mathbf{u}}^{\mathrm{T}})$), $\mathbf{I}$ is the identity matrix, μ_T is called the turbulent eddy viscosity, and

$$k = \frac{1}{2}\overline{\mathbf{u}'\cdot\mathbf{u}'}, \tag{3.3.2}$$

is the turbulent kinetic energy. Basically, we assume that this fluctuating stress is proportional to the gradient of the average quantities (similarly to Newtonian flows). The second term in eq. 3.3.1 ($\frac{2}{3}\rho k\mathbf{I}$), is added in order for the Boussinesq approximation to be valid when traced, that is, the trace of the right hand side in eq. 3.3.1 must be equal to that of the left hand side ($-\rho(\overline{\mathbf{u}'\mathbf{u}'})^{\mathrm{tr}} = -2\rho k$), hence it is consistent with the definition of turbulent kinetic energy (eq. 3.3.2). In order to evaluate k, usually a governing equation for k is derived and solved, typically two-equations models include such an option.

The turbulent eddy viscosity μ_T (in contrast to the molecular viscosity μ), is a property of the flow field and not a physical property of the fluid. The eddy viscosity concept was developed assuming that a relationship or analogy exists between molecular and turbulent viscosities. In spite of the theoretical weakness of the turbulent eddy viscosity concept, it does produce reasonable results for a large number of flows.

The Boussinesq approximation reduces the turbulence modeling process from finding the six turbulent stress components $\boldsymbol{\tau}^R$ to determining an appropriate value for the turbulent eddy viscosity μ_T.

One final word of caution, the Boussinesq approximation discussed here, should not be associated with the completely different concept of natural convection.

3.4 Two-Equations Models. The $k-\omega$ Model

In this section we present the widely used $k-\omega$ model. As might be inferred from the terminology (and the tittle of this section), it is a two-equation model. In its basic form it consist of a governing equation for the turbulent kinetic energy k, and a governing equation for the turbulent specific dissipation rate ω. Together, these two quantities provide velocity and length scales needed to directly find the value of the turbulent eddy viscosity μ_T at each point in a computational domain. The $k-\omega$ model has been modified over the years, new terms (such as production and dissipation terms) have been added to both the k and ω equations, which have improved the accuracy of the model. Because it has been tested more extensively than any other $k-\omega$ model, we present the Wilcox model [21].

Eddy Viscosity

$$\mu_T = \frac{\rho k}{\omega} \tag{3.4.1}$$

Turbulent Kinetic Energy

$$\rho\frac{\partial k}{\partial t} + \rho\nabla\cdot(\bar{\mathbf{u}}k) = \boldsymbol{\tau}^R{:}\nabla\bar{\mathbf{u}} - \beta^*\rho k\omega + \nabla\cdot\left[\left(\mu + \sigma^*\mu_T\right)\nabla k\right] \tag{3.4.2}$$

Specific Dissipation Rate

$$\rho\frac{\partial\omega}{\partial t}+\rho\nabla\cdot(\bar{\mathbf{u}}\omega)=\alpha\frac{\omega}{k}\boldsymbol{\tau}^{R}\mathbf{:}\nabla\bar{\mathbf{u}}-\beta\rho\omega^{2}+\nabla\cdot\left[(\mu+\sigma\mu_{T})\,\nabla\omega\right] \tag{3.4.3}$$

Closure Coefficients

$$\alpha=\frac{5}{9},\quad \beta=\frac{3}{40},\quad \beta^{*}=\frac{9}{100},\quad \sigma=\frac{1}{2},\quad \sigma^{*}=\frac{1}{2} \tag{3.4.4}$$

Auxiliary Relations

$$\epsilon=\beta^{*}\omega k\qquad and\qquad l=\frac{k^{\frac{1}{2}}}{\omega} \tag{3.4.5}$$

Equations eq. 3.2.8 and eq. 3.4.1-3.4.5, are the governing equations of an incompressible, isothermal, turbulent flow written in conservation form. Hence, we look for an approximate solution of this set of equations in a given domain $\mathcal{D}$, with prescribed boundary conditions $\partial\mathcal{D}$, and given initial conditions $\mathcal{D}\mathring{\mathbf{q}}$.

Chapter 4

Finite Volume Method Discretization

The purpose of any discretization practice is to transform a set of partial differential equations (PDEs) into a corresponding system of discrete algebraic equations (DAEs). The solution of this system produces a set of values which correspond to the solution of the original equations at some predetermined locations in space and time, provided certain conditions are satisfied. The discretization process can be divided into two steps, namely; the discretization of the solution domain and the discretization of the governing equation.

The discretization of the solution domain produces a numerical description of the computational domain (also known as mesh generation). The space is divided into a finite number of discrete regions, called control volumes ($\mathcal{CV}$s) or cells. For transient simulations, the time interval is also split into a finite number of time steps. The governing equations discretization step altogether with the domain discretization, produces an appropriate transformation of the terms of the governing equations into a system of discrete algebraic equations that can be solve using any direct or iterative method.

In this section, we briefly presents the finite volume method (FVM) discretization, with the following considerations in mind:

- The method is based on discretizing the integral form of the governing equations over each control volume of the discrete domain. The basic quantities, such as mass and momentum, will therefore be conserved at the discrete levels.
- The method is applicable to both steady-state and transient calculations.
- The method is applicable to any number of spatial dimensions ($1\mathbb{D}$, $2\mathbb{D}$ or $3\mathbb{D}$).
- The control volumes can be of any shape. All dependent variables share the same control volume and are computed at the control volume centroid, which is usually called the collocated or non-staggered variable arrangement.
- Systems of partial differential equations are treated in a segregated way, meaning that they are are solved one at a time in a sequential manner.

The specific details of the solution domain discretization, system of equations discretization practices and implementation of the FVM are far beyond the scope of the present discussion. Hereafter, we give a brief description of the FVM method. For a detailed discussion, the interested reader should refer to references [22, 3, 12, 23, 19, 24, 25].

4.1 Discretization of the Solution Domain

Discretization of the solution domain produces a computational mesh on which the governing equations are solved (mesh generation stage). It also determines the positions of points in space and time where the solution will be computed. The procedure can be split into two parts: temporal discretization and spatial discretization.

The temporal solution is simple obtained by marching in time from the prescribed initial conditions. For the discretization of time, it is therefore necessary to prescribe the size of the time-step that will be used during the calculation.

The spatial discretization of the solution domain of the FVM method presented in this manuscript, requires a subdivision of the continuous domain into a finite number of discrete arbitrary control volumes ($\mathcal{CV}$s). In our discussion, the control volumes do not overlap, have a positive finite volume and completely fill the computational domain. Finally, all variables are computed at the centroid of the control volumes (collocated arrangement).

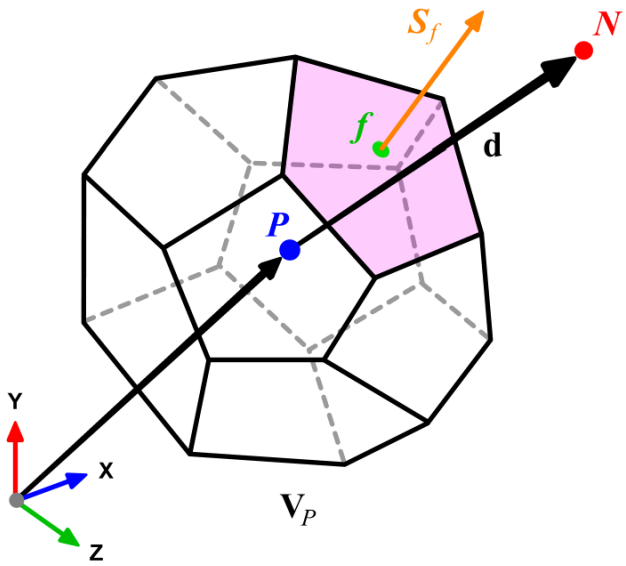

Figure 4.1: *Arbitrary polyhedral control volume V_P. The control volume has a volume V and is constructed around a point P (control volume centroid), therefore the notation V_P. The vector from the centroid of the control volume V_P (point P), to the centroid of the neighboring control volume V_N (point N), is defined as $\boldsymbol{d}$. The face area vector $\boldsymbol{S}_f$ points outwards from the surface bounding V_P and is normal to the face. The control volume faces are labeled as f, which also denotes the face center.*

A typical control volume is shown in figure 4.1. In this figure, the control volume V_P is bounded by a set of flat faces and each face is shared with only one neighboring control volume. The shape of the control volume is not important for our discussion, for our purposes it is a general polyhedron, as shown in figure 4.1. The control volume faces in the discrete domain can be divided into two groups, namely; internal faces (between two control volumes) and boundary faces, which coincide with the boundaries of the domain. The face area vector $\mathbf{S}_f$ is constructed for each face in such a way that it points outwards from the control volume, is located at the face centroid, is normal to the face and has a magnitude equal to the area of the face (*e.g.*, the shaded face in figure 4.1). Boundary face area vectors point outwards from the computational domain. In figure 4.1, the point P represents the centroid of the control volume V_P and the point N represents the centroid of the neighbor control volume V_N. The distance between the

point P and the point N is given by the vector $\mathbf{d}$. For simplicity, all faces of the control volume will be marked with f, which also denotes the face centroid (see figure 4.1).

A control volume V_P, is constructed around a computational point P. The point P, by definition, is located at the centroid of the control volume such that its centroid is given by

$$\int_{V_P} (\mathbf{x} - \mathbf{x}_P)\, dV = 0. \tag{4.1.1}$$

In a similar way, the centroid of the faces of the control volume V_P is defined as

$$\int_{S_f} (\mathbf{x} - \mathbf{x}_f)\, dS = 0. \tag{4.1.2}$$

Finally, let us introduce the mean value theorem for the transported quantity ϕ over the control volume V_P, such that

$$\overline{\phi} = \frac{1}{V_P} \int_{V_P} \phi(\mathbf{x}) dV. \tag{4.1.3}$$

In the FVM method discussed in this manuscript, the centroid value ϕ_P of the control volume V_P is represented by a piecewise constant profile. That is, we assume that the value of the transported quantity ϕ is computed and stored in the centroid of the control volume V_P and that its value is equal to the mean value of ϕ inside the control volume,

$$\phi_P = \overline{\phi} = \frac{1}{V_P} \int_{V_P} \phi(\mathbf{x}) dV. \tag{4.1.4}$$

This approximation is exact if ϕ is constant or vary linearly.

4.2 Discretization of the Transport Equation

The general transport equation is used throughout this discussion to present the FVM discretization practices. All the equations described in sections 2 and 3 can be written in the form of the general transport equation over a given control volume V_P (as the control volume shown in figure 4.1), as follows

$$\int_{V_P} \underbrace{\frac{\partial \rho\phi}{\partial t} dV}_{\text{temporal derivative}} + \int_{V_P} \underbrace{\nabla \cdot (\rho \mathbf{u} \phi)\, dV}_{\text{convective term}} - \int_{V_P} \underbrace{\nabla \cdot (\rho \Gamma_\phi \nabla \phi)\, dV}_{\text{diffusion term}} = \int_{V_P} \underbrace{S_\phi(\phi)\, dV}_{\text{source term}}. \tag{4.2.1}$$

Here ϕ is the transported quantity, *i.e.*, velocity, mass or turbulent energy and Γ_ϕ is the diffusion coefficient of the transported quantity. This is a second order equation since the diffusion term includes a second order derivative of ϕ in space. To represent this term with acceptable accuracy, the order of the discretization must be equal or higher than the order of the equation to be discretized. In the same order of ideas, to conform to this level of accuracy, temporal discretization must be of second order as well. As a consequence of these requirements, all dependent variables are assumed to vary linearly around the point P in space and instant t in time, such that

$$\phi(\mathbf{x}) = \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla \phi)_P \qquad \text{where} \qquad \phi_P = \phi(\mathbf{x}_P). \tag{4.2.2}$$

$$\phi(t + \delta t) = \phi^t + \delta t \left(\frac{\partial \phi}{\partial t} \right)^t \qquad \text{where} \qquad \phi^t = \phi(t). \tag{4.2.3}$$

Equations 4.2.2 and 4.2.3 are obtained by using Taylor Series Expansion (TSE) around the nodal point P and time t, and truncating the series in such a way to obtain second order accurate approximations.

A key theorem in the FVM method is the Gauss theorem (also know as the divergence or Ostrogradsky's theorem), which will be used throughout the discretization process in order to reduce the volume integrals in eq. 4.2.1 to their surface equivalents.

The Gauss theorem states that the volume integral of the divergence of a vector field in a region inside a volume, is equal to the surface integral of the outward flux normal to the closed surface that bounds the volume. For a vector $\mathbf{a}$, the Gauss theorem is given by,

$$\int_V \nabla \cdot \mathbf{a} dV = \oint_{\partial V} \mathbf{n} dS \cdot \mathbf{a}, \tag{4.2.4}$$

where ∂V is the surface bounding the volume V and dS is an infinitesimal surface element with the normal $\mathbf{n}$ pointing outward of the surface ∂V. From now on, $d\mathbf{S}$ will be used as a shorthand for $\mathbf{n}dS$.

By using the Gauss theorem, we can write eq. 4.2.1 as follows

$$\frac{\partial}{\partial t}\int_{V_P} (\rho\phi)\, dV + \oint_{\partial V_P} \underbrace{d\mathbf{S}\cdot(\rho\mathbf{u}\phi)}_{\text{convective flux}} - \oint_{\partial V_P} \underbrace{d\mathbf{S}\cdot(\rho\Gamma_\phi\nabla\phi)}_{\text{diffusive flux}} = \int_{V_P} S_\phi(\phi)\, dV. \tag{4.2.5}$$

Equation 4.2.5 is a statement of conservation. It states that the rate of change of the transported quantity ϕ inside the control volume V_P is equal to the rate of change of the convective and diffusive fluxes across the surface bounding the control volume V_P, plus the net rate of creation of ϕ inside the control volume. Notice that so far we have not introduce any approximation, equation 4.2.5 is exact.

In the next sections, each of the terms in eq. 4.2.1 will be treated separately, starting with the spatial discretization and concluding with the temporal discretization. By proceeding in this way we will be solving eq. 4.2.1 by using the Method of Lines (MOL). The main advantage of the MOL, is that it allows us to select numerical approximations of different accuracy for the spatial and temporal terms. Each term can be treated differently to yield to different accuracies.

4.2.1 Approximation of Surface Integrals and Volume Integrals

In eq. 4.2.5, a series of surface and volume integrals need to be evaluated over the control volume V_P. These integrals must be approximated to at least second order accuracy in order to conform to the same level of accuracy of eq. 4.2.1.

To calculate the surface integrals in eq. 4.2.5 we need to know the value of the transported quantity ϕ on the faces of the control volume. This information is not available, as the variables are calculated on the control volume centroid, so an approximation must be introduced at this stage.

We now make a profile assumption about the transported quantity ϕ. We assume that ϕ varies linearly over each face f of the control volume V_P, so that ϕ may be represented by its mean value at the face centroid f. We can now approximate the surface integral as a product of the transported quantities at the face center f (which is itself an approximation to the mean value

over the surface) and the face area. This approximation to the surface integral is known as the midpoint rule and is of second-order accuracy.

It is worth mentioning that a wide range of choices exists with respect to the way of approximating the surface integrals, *e.g.*, midpoint rule, trapezoid rule, Simpson's rule, Gauss quadrature. Here, we have used the simplest method, namely, the midpoint rule.

For illustrating this approximation, let us consider the term under the divergence operator in eq. 4.2.4 and recalling that all faces are flat (that is, all vertexes that made up the face are contained in the same plane), eq. 4.2.4 can be converted into a discrete sum of integrals over all faces of the control volume V_P as follows,

$$\begin{aligned}\int_{V_P} \nabla \cdot \mathbf{a} dV &= \oint_{\partial V_P} d\mathbf{S} \cdot \mathbf{a}, \\ &= \sum_f \left(\int_f d\mathbf{S} \cdot \mathbf{a} \right), \\ &\approx \sum_f \left(\mathbf{S}_f \cdot \overline{\mathbf{a}}_f \right) = \sum_f \left(\mathbf{S}_f \cdot \mathbf{a}_f \right).\end{aligned} \tag{4.2.6}$$

Using the same approximations and assumptions as in eq. 4.2.6, the surface integrals (or fluxes) in eq. 4.2.5 can be approximate as follow

$$\oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\rho \mathbf{u} \phi)}_{\text{convective flux}} = \sum_f \int_f d\mathbf{S} \cdot (\rho \mathbf{u} \phi)_f \approx \sum_f \mathbf{S}_f \cdot \left(\overline{\rho \mathbf{u} \phi} \right)_f = \sum_f \mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f . \tag{4.2.7}$$

$$\oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\rho \Gamma_\phi \nabla \phi)}_{\text{diffusive flux}} = \sum_f \int_f d\mathbf{S} \cdot (\rho \Gamma_\phi \nabla \phi)_f \approx \sum_f \mathbf{S}_f \cdot \left(\overline{\rho \Gamma_\phi \nabla \phi} \right)_f = \sum_f \mathbf{S}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f . \tag{4.2.8}$$

To approximate the volume integrals in eq. 4.2.5, we make similar assumptions as for the surface integrals, that is, ϕ varies linearly over the control volume and $\overline{\phi} = \phi_P$. Integrating eq. 4.2.2 over a control volume V_P, it follows

$$\begin{aligned}\int_{V_P} \phi(\mathbf{x}) \, dV &= \int_{V_P} \left[\phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla \phi)_P \right] dV, \\ &= \phi_P \int_{V_P} dV + \left[\int_{V_P} (\mathbf{x} - \mathbf{x}_P) \, dV \right] \cdot (\nabla \phi)_P , \\ &= \phi_P V_P .\end{aligned} \tag{4.2.9}$$

The second integral in the RHS of eq. 4.2.9 is equal to zero because the point P is the centroid of the control volume (recall eq. 4.1.1). This quantity is easily calculated since all variables at the centroid of V_P are known, no interpolation is needed. The above approximation becomes exact if ϕ is either constant or varies linearly within the control volume; otherwise, it is a second order approximation.

Introducing equations 4.2.7-4.2.9 into eq. 4.2.5 we obtain,

$$\frac{\partial}{\partial t} \rho \phi V_P + \sum_f \mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f - \sum_f \mathbf{S}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f = S_\phi V_P . \tag{4.2.10}$$

Let us recall that in our formulation of the FVM, all the variables are computed and stored at the control volumes centroid. The face values appearing in eq. 4.2.10; namely, the convective flux $F^C = \mathbf{S}\cdot(\rho\mathbf{u}\phi)$ through the faces, and the diffusive flux $F^D = \mathbf{S}\cdot(\rho\Gamma_\phi\nabla\phi)$ through the faces, have to be calculated by some form of interpolation from the centroid values of the neighboring control volumes located at both sides of the faces, this issue is discussed in the following section.

4.2.2 Convective Term Spatial Discretization

The discretization of the convective term in eq. 4.2.1 is obtained as in eq. 4.2.7, *i.e.*,

$$\begin{aligned}\int_{V_P} \nabla\cdot(\rho\mathbf{u}\phi)\,dV &= \sum_f \mathbf{S}_f\cdot(\rho\mathbf{u}\phi)_f\,,\\ &= \sum_f \mathbf{S}_f\cdot(\rho\mathbf{u})_f\,\phi_f,\\ &= \sum_f \mathring{F}\phi_f,\end{aligned} \tag{4.2.11}$$

where $\mathring{F}$ in eq. 4.2.11 represents the mass flux through the face,

$$\mathring{F} = \mathbf{S}_f\cdot(\rho\mathbf{u})_f\,. \tag{4.2.12}$$

Obviously, the flux $\mathring{F}$ depends on the face value of ρ and $\mathbf{u}$, which can be calculated in a similar fashion to ϕ_f (as it will be described in the next section), with the caveat that the velocity field from which the fluxes are derived must be such that the continuity equation is obeyed, *i.e.*,

$$\int_{V_P}\nabla\cdot\mathbf{u}dV = \oint_{\partial V_P} d\mathbf{S}\cdot\mathbf{u} = \sum_f\left(\int_f d\mathbf{S}\cdot\mathbf{u}\right) = \sum_f \mathbf{S}_f\cdot(\rho\mathbf{u})_f = \sum_f \mathring{F} = 0. \tag{4.2.13}$$

Before we continue with the formulation of the interpolation scheme or convection differencing scheme used to compute the face value of the transported quantity ϕ; it is necessary to examine the physical properties of the convection term. Irrespective of the distribution of the velocity in the domain, the convection term does not violate the bounds of ϕ given by its initial condition. If for example, ϕ initially varies between 0 and 1, the convection term will never produce values of ϕ that are lower than zero or higher that one. Considering the importance of boundedness in the transport of scalar properties, it is essential to preserve this property in the discretized form of the term.

4.2.2.1 Convection Interpolation Schemes

The role of the convection interpolation schemes is to determine the value of the transported quantity ϕ on the faces f of the control volume V_P. Therefore, the value of ϕ_f is computed by using the values from the neighbors control volumes. Hereafter, we present two of the most widely used schemes. For a more detailed discussion on the subject and a presentation of more convection interpolation schemes, the interested reader should refer to references [22, 3, 12, 23, 24, 25, 26].

- **Central Differencing (CD) scheme.** In this scheme (also known as linear interpolation), linear variation of the dependent variables is assumed. The face centered value

is found from a simple weighted linear interpolation between the values of the control volumes at points P and N (see figure 4.2), such that

$$\phi_f = f_x \phi_P + (1 - f_x)\, \phi_N. \tag{4.2.14}$$

In eq. 4.2.14, the interpolation factor f_x, is defined as the ratio of the distances fN and PN (refer to figure 4.2), *i.e.*,

$$f_x = \frac{fN}{PN} = \frac{\mid \mathbf{x}_f - \mathbf{x}_N \mid}{\mid \mathbf{d} \mid}. \tag{4.2.15}$$

A special case arises when the face is situated midway between the two neighboring control volumes V_P and V_N (uniform mesh), then the approximation reduces to an arithmetic average

$$\phi_f = \frac{(\phi_P + \phi_N)}{2}. \tag{4.2.16}$$

This practice is second order accurate, which is consistent with the requirement of overall second order accuracy of the method. It has been noted however, that CD causes non-physical oscillations in the solution for convection dominated problems, thus violating the boundedness of the solution ([22, 3, 12, 23, 24, 25, 26]).

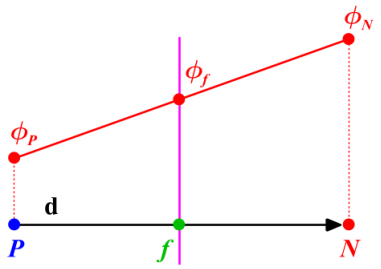

Figure 4.2: *Face interpolation. Central Differencing (CD) scheme.*

- **Upwind Differencing (UD) scheme.** An alternative discretization scheme that guarantees boundedness is the Upwind Differencing (UD). In this scheme, the face value is determined according to the direction of the flow (refer to figure 4.3),

$$\phi_f = \begin{cases} \phi_f = \phi_P & \text{for} \quad \mathring{F} \geq 0, \\ \phi_f = \phi_N & \text{for} \quad \mathring{F} < 0. \end{cases} \tag{4.2.17}$$

This scheme guarantees the boundedness of the solution ([22, 3, 12, 23, 24, 25, 26]). Unfortunately, UD is at most first order accurate, hence it sacrifices the accuracy of the solution by implicitly introducing numerical diffusion.

In order to circumvent the numerical diffusion inherent of UD and unboundedness of CD, linear combinations of UD and CD, second order variations of UD and bounded CD schemes has been developed in order to conform to the accuracy of the discretization and maintain the boundedness and stability of the solution [22, 3, 12, 23, 24, 25, 26].

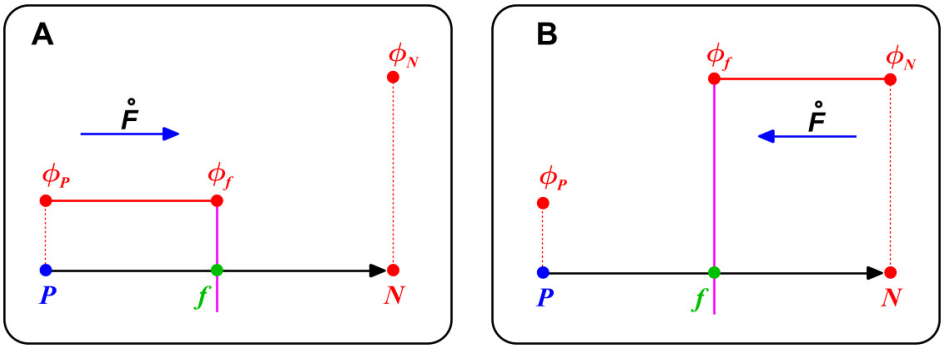

Figure 4.3: *Face interpolation. Upwind Differencing (UD) scheme. A)* $\mathring{F} \geq 0$. *B)* $\mathring{F} < 0$.

4.2.3 Diffusion Term Spatial Discretization

Using a similar approach as before, the discretization of the diffusion term in eq. 4.2.1 is obtained as in eq. 4.2.8, *i.e.*,

$$\begin{aligned} \int_{V_P} \nabla \cdot (\rho \Gamma_\phi \nabla \phi)\, dV &= \sum_f \mathbf{S}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f\,, \\ &= \sum_f (\rho \Gamma_\phi)_f\, \mathbf{S}_f \cdot (\nabla \phi)_f\,, \end{aligned} \tag{4.2.18}$$

4.2.3.1 The Interface Conductivity

In eq. 4.2.18, Γ_ϕ is the diffusion coefficient. If Γ_ϕ is uniform, its value is the same for all control volumes. The following discussion is, of course, not relevant to situations where the Γ_ϕ is uniform. For situations of non-uniform Γ_ϕ, the interface conductivity $(\Gamma_\phi)_f$ can be found by using linear interpolation between the control volumes V_P and V_N (see figure 4.4),

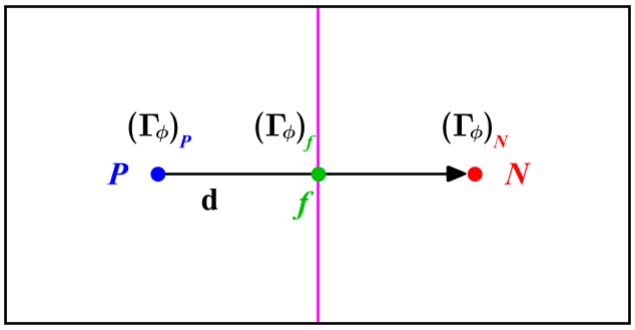

Figure 4.4: *Diffusion coefficient* Γ_ϕ *variation in neighboring control volumes.*

$$(\Gamma_\phi)_f = f_x (\Gamma_\phi)_P + (1 - f_x)\,(\Gamma_\phi)_N \quad \text{where} \quad f_x = \frac{fN}{PN} = \frac{|\,\mathbf{x}_f - \mathbf{x}_N\,|}{|\,\mathbf{d}\,|}. \tag{4.2.19}$$

If the control volumes are uniform (the face f is midway between V_P and V_N), then f_x is equal to 0.5, and $(\Gamma_\phi)_f$ is equal to the arithmetic mean.

However, the method above described suffers from the drawback that if $(\Gamma_\phi)_N$ is equal to zero, it is expected that there would be no diffusive flux across face f. But in fact, eq. 4.2.19 approximates a value for $(\Gamma_\phi)_f$, namely

$$(\Gamma_\phi)_f = f_x(\Gamma_\phi)_P, \tag{4.2.20}$$

where we normally would have expected zero. Similarly, if $(\Gamma_\phi)_N$ is much less that $(\Gamma_\phi)_P$, there would be relatively little resistance to the diffusive flux between V_P and face f, compared to that between V_N and the face f. In this case it would be expected that $(\Gamma_\phi)_f$ would depend on $(\Gamma_\phi)_N$ and inversely on f_x.

A better model for the variation of Γ_ϕ between control volumes is to use the harmonic mean, which is expressed as follows,

$$(\Gamma_\phi)_f = \frac{(\Gamma_\phi)_N(\Gamma_\phi)_P}{f_x(\Gamma_\phi)_P + (1-f_x)(\Gamma_\phi)_N} \quad \text{where} \quad f_x = \frac{fN}{PN} = \frac{\mid \mathbf{x}_f - \mathbf{x}_N \mid}{\mid \mathbf{d} \mid}. \tag{4.2.21}$$

This formulation gives $(\Gamma_\phi)_f$ equal to zero if either $(\Gamma_\phi)_N$ or $(\Gamma_\phi)_P$ is zero. For $(\Gamma_\phi)_P >> (\Gamma_\phi)_N$ gives

$$(\Gamma_\phi)_f = \frac{(\Gamma_\phi)_N}{f_x}, \tag{4.2.22}$$

as required.

4.2.3.2 Numerical Approximation of the Diffusive Term

From the spatial discretization process of the diffusion terms a face gradient arise, namely $(\nabla\phi)_f$ (see eq. 4.2.18). This gradient term can be computed as follows. If the mesh is orthogonal, *i.e.*, the vectors **d** and **S** in figure 4.5 are parallel, it is possible to use the following expression

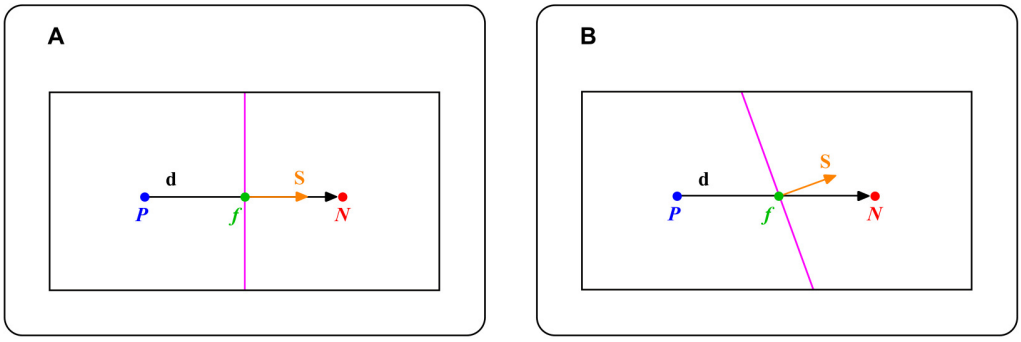

Figure 4.5: *A) Vector **d** and **S** on an orthogonal mesh. B) Vector **d** and **S** on a non-orthogonal mesh.*

$$\mathbf{S} \cdot (\nabla\phi)_f = |\mathbf{S}| \frac{\phi_N - \phi_P}{|\mathbf{d}|}. \tag{4.2.23}$$

By using eq. 4.2.23, the face gradient of ϕ can be calculated from the values of the control volumes straddling face f (V_P and V_N), so basically we are computing the face gradient by using a central difference approximation of the first order derivative in the direction of the vector **d**. This method is second order accurate, but can only be used on orthogonal meshes.

An alternative to the previous method, would be to calculate the gradient of the control volumes at both sides of face f by using Gauss theorem, as follows

$$(\nabla\phi)_P = \frac{1}{V_P}\sum_f (\mathbf{S}_f\phi_f)\,. \tag{4.2.24}$$

After computing the gradient of the neighboring control volumes V_P and V_N, we can find the face gradient by using weighted linear interpolation.

Although both of the previously described methods are second order accurate; eq. 4.2.24 uses a larger computational stencil, which involves a larger truncation error and can lead to unbounded solutions. On the other hand, spite of the higher accuracy of eq. 4.2.23, it can not be used on non-orthogonal meshes.

Unfortunately, mesh orthogonality is more an exception than a rule. In order to make use of the higher accuracy of eq. 4.2.23, the product $\mathbf{S}\cdot(\nabla\phi)_f$ is split in two parts

$$\mathbf{S}\cdot(\nabla\phi)_f = \underbrace{\Delta_\perp\cdot(\nabla\phi)_f}_{\text{orthogonal contribution}} + \underbrace{\mathbf{k}\cdot(\nabla\phi)_f}_{\text{non-orthogonal contribution}}\,. \tag{4.2.25}$$

The two vectors introduced in eq. 4.2.25, namely; $\Delta_\perp$ and $\mathbf{k}$, need to satisfy the following condition

$$\mathbf{S} = \Delta_\perp + \mathbf{k}. \tag{4.2.26}$$

If the vector $\Delta_\perp$ is chosen to be parallel with $\mathbf{d}$, this allows us to use eq. 4.2.23 on the orthogonal contribution in eq. 4.2.25, and the non-orthogonal contribution is computed by linearly interpolating the face gradient from the centroid gradients of the control volumes at both sides of face f, obtained by using eq. 4.2.24. The purpose of this decomposition is to limit the error introduced by the non-orthogonal contribution, while keeping the second order accuracy of eq. 4.2.23.

To handle the mesh orthogonality decomposition within the constraints of eq. 4.2.26, let us study the following approaches ([26, 27, 12]), with $\mathbf{k}$ calculated from eq. 4.2.26:

- Minimum correction approach (figure 4.6). This approach attempts to minimize the non-orthogonal contribution by making $\Delta_\perp$ and $\mathbf{k}$ orthogonal,

$$\Delta_\perp = \frac{\mathbf{d}\cdot\mathbf{S}}{\mathbf{d}\cdot\mathbf{d}}\mathbf{d}. \tag{4.2.27}$$

  In this approach, as the non-orthogonality increases, the contribution from ϕ_P and ϕ_N decreases.

- Orthogonal correction approach (figure 4.7). This approach attempts to maintain the condition of orthogonality, irrespective of whether non-orthogonality exist,

$$\Delta_\perp = \frac{\mathbf{d}}{|\mathbf{d}|}|\mathbf{S}|. \tag{4.2.28}$$

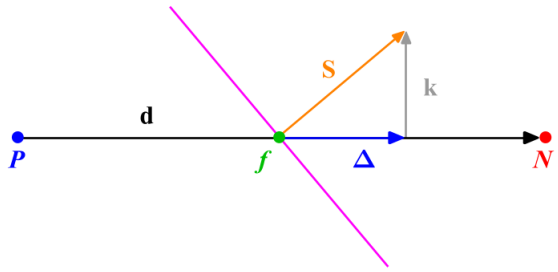

Figure 4.6: *Non-orthogonality treatment in the minimum correction approach.*

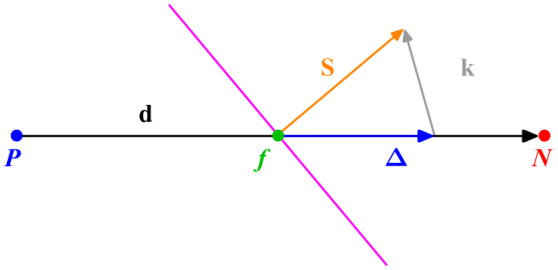

Figure 4.7: *Non-orthogonality treatment in the orthogonal correction approach.*

- Over-relaxed approach (figure 4.8). In this approach, the contribution from ϕ_P and ϕ_N increases with the increase in non-orthogonality, such as

$$\Delta_\perp = \frac{\mathbf{d}}{\mathbf{d}\cdot\mathbf{S}}|\mathbf{S}|^2. \tag{4.2.29}$$

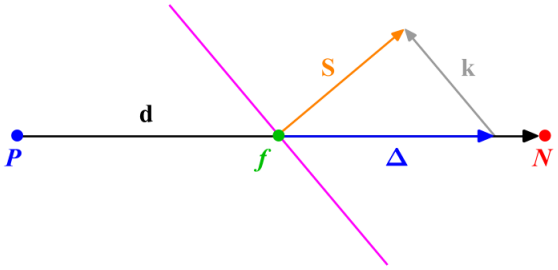

Figure 4.8: *Non-orthogonality treatment in the over-relaxed approach.*

All of the approaches described above are valid, but the so-called over-relaxed approach seems to be the most robust, stable and computationally efficient.

Non-orthogonality adds numerical diffusion to the solution and reduces the accuracy of the numerical method. It also leads to unboundedness, which in turn can conduct to nonphysical results and/or divergence of the solution.

The diffusion term, eq. 4.2.18, in its differential form exhibits a bounded behavior. Hence, its discretized form will preserve only on orthogonal meshes. The non-orthogonal correction potentially creates unboundedness, particularly if mesh non-orthogonality is high. If the preservation of boundedness is more important than accuracy, the non-orthogonal correction has got to be limited or completely discarded, thus violating the order of accuracy of the discretization. Hence

care must be taken to keep mesh orthogonality within reasonable bounds.

The final form of the discretized diffusion term is the same for all three approaches. Since eq. 4.2.23 is used to compute the orthogonal contribution, meaning that $\mathbf{d}$ and $\Delta_\perp$ are parallel, it follows

$$\Delta_\perp \cdot (\nabla\phi)_f = |\Delta_\perp| \frac{\phi_N - \phi_P}{|\mathbf{d}|}, \tag{4.2.30}$$

then eq. 4.2.25 can be written as

$$\mathbf{S} \cdot (\nabla\phi)_f = \underbrace{|\Delta_\perp| \frac{\phi_N - \phi_P}{|\mathbf{d}|}}_{\text{orthogonal contribution}} + \underbrace{\mathbf{k} \cdot (\nabla\phi)_f}_{\text{non-orthogonal contribution}} . \tag{4.2.31}$$

In eq. 4.2.31, the face interpolated value of $\nabla\phi$ of the non-orthogonal contribution is calculated as follows

$$(\nabla\phi)_f = f_x (\nabla\phi)_P + (1 - f_x) (\nabla\phi)_N . \tag{4.2.32}$$

where the gradient of the control volumes V_P and V_N are computed using eq. 4.2.24.

4.2.4 Evaluation of Gradient Terms

In the previous section, the face gradient arising from the discretization of the diffusion term was computed by using eq. 4.2.23 (central differencing) in the case of orthogonal meshes, and a correction was introduced to improve the accuracy of this face gradient in the case of non-orthogonal meshes (eq. 4.2.31).

By means of the Gauss theorem, the gradient terms of the control volume V_P arising from the discretization process or needed to compute the face gradients are calculated as follows,

$$\begin{aligned} \int_{V_P} \nabla\phi dV &= \oint_{\partial V_P} d\mathbf{S}\phi, \\ (\nabla\phi)_P V_P &= \sum_f (\mathbf{S}_f \phi_f) , \\ (\nabla\phi)_P &= \frac{1}{V_P} \sum_f (\mathbf{S}_f \phi_f) , \end{aligned} \tag{4.2.33}$$

where the value ϕ_f on face f can be evaluated using the convection central differencing scheme.

After computing the gradient of the control volumes at both sides of face f by using eq. 4.2.33, we can find the face gradient by using weighted linear interpolation,

$$(\nabla\phi)_f = f_x (\nabla\phi)_P + (1 - f_x) (\nabla\phi)_N , \tag{4.2.34}$$

and dot it with $\mathbf{S}$. This method is often referred to as Green-Gauss cell based gradient evaluation and is second order accurate.

The Green-Gauss cell based gradient evaluation uses a computational stencil larger than the one used by eq. 4.2.23; hence the truncation error is larger and it might lead to oscillatory

solutions (unboundedness), which in turns can lead to nonphysical values of ϕ and divergence, The advantage of this method is that it can be used in orthogonal and non-orthogonal meshes; whereas eq. 4.2.23, can be only used in orthogonal meshes.

Another alternative, is by evaluating the face gradient by using a Least-Square fit (LSF). This method assumes a linear variation of ϕ (which is consistent with the second order accuracy requirement), and evaluates the gradient error at each neighboring control volume N using the following expression,

$$\epsilon_N = \phi_N - (\phi_P + \mathbf{d} \cdot (\nabla\phi)_P) \,. \tag{4.2.35}$$

The objective now is to minimize the least-square error at P given by

$$\epsilon_P^2 = \sum_N w_N^2 \epsilon_N^2, \tag{4.2.36}$$

where the weighting function w is given by $w_N = 1/|\mathbf{d}|$. Then, the following expression is used to evaluate the gradient at the centroid of the control volume V_P,

$$\begin{aligned} (\nabla\phi)_P &= \sum_N w_N^2 \mathbf{G}^{-1} \cdot \mathbf{d} \, (\phi_N - \phi_P) \,. \\ \mathbf{G} &= \sum_N w_N^2 \mathbf{d}\mathbf{d}. \end{aligned} \tag{4.2.37}$$

After evaluation the neighbor control volumes gradient, they can be interpolated to the face. Note that $\mathbf{G}$ is a symmetric $\mathbb{N} \times \mathbb{N}$ matrix and can easily be inverted (where $\mathbb{N}$ is the number of spatial dimensions). This formulation leads to a second order accurate gradient approximation which is independent of the mesh topology.

4.2.5 Source Terms Spatial Discretization

All terms of the transport equation that cannot be written as convection, diffusion or temporal contributions are here loosely classified as source terms. The source term, $S_\phi(\phi)$, can be a general function of ϕ. When deciding on the form of the discretization for the source term, its interaction with other terms in the equation and its influence on boundedness and accuracy should be examined. Some general comments on the treatment of source terms are given in references [22, 3, 12, 24, 25]. But in general and before the actual discretization, the source terms need to be linearized (for instance by using Picard's method), such that,

$$S_\phi \, (\phi) = S_c + S_p \phi, \tag{4.2.38}$$

where S_c is the constant part of the source term and S_p depends on ϕ. For instance, if the source term is assume to be constant, eq. 4.2.38 reduces to $S_\phi(\phi) = S_u$.

Following eq. 4.2.9, the volume integral of the source terms is calculated as

$$\int_{V_P} S_\phi \, (\phi) \, dV = S_c V_P + S_p V_P \phi_P. \tag{4.2.39}$$

4.2.6 Temporal Discretization

In the previous sections, the discretization of the spatial terms was presented. Let us now consider the temporal derivative of the general transport eq. 4.2.1, integrating in time we get

$$\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{V_P}\rho\phi dV+\int_{V_P}\nabla\cdot\left(\rho\mathbf{u}\phi\right)dV-\int_{V_P}\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)dV\right]dt = \int_t^{t+\Delta t}\left(\int_{V_P}S_\phi\left(\phi\right)dV\right)dt. \tag{4.2.40}$$

Using equations 4.2.7-4.2.9 and 4.2.39, eq. 4.2.40 can be written as,

$$\int_t^{t+\Delta t}\left[\left(\frac{\partial\rho\phi}{\partial t}\right)_P V_P+\sum_f\mathbf{S}_f\cdot\left(\rho\mathbf{u}\phi\right)_f-\sum_f\mathbf{S}_f\cdot\left(\rho\Gamma_\phi\nabla\phi\right)_f\right]dt = \int_t^{t+\Delta t}\left(S_cV_P+S_pV_P\phi_P\right)dt. \tag{4.2.41}$$

The above expression is usually called the *semi-discretized* form of the transport equation. It should be noted that the order of the temporal discretization of the transient term in eq. 4.2.41 does not need to be the same as the order of the discretization of the spatial terms (convection, diffusion and source terms). Each term can be treated differently to yield different accuracies. As long as the individual terms are second order accurate, the overall accuracy of the solution will also be second order.

4.2.6.1 Time Centered Crank-Nicolson

Keeping in mind the assumed variation of ϕ with t (eq. 4.2.3), the temporal derivative and time integral can be calculated as follows,

$$\begin{aligned}\left(\frac{\partial\rho\phi}{\partial t}\right)_P &= \frac{\rho_P^n\phi_P^n-\rho_P^{n-1}\phi_P^{n-1}}{\Delta t},\\ \int_t^{t+\Delta t}\phi(t)dt &= \frac{1}{2}\left(\phi_P^{n-1}+\phi^n\right)\Delta t,\end{aligned} \tag{4.2.42}$$

where $\phi^n=\phi(t+\Delta t)$ and $\phi^{n-1}=\phi(t)$ represent the value of the dependent variable at the new and previous times respectively. Equation 4.2.42 provides the temporal derivative at a centered time between times $n-1$ and n. Combining equations 4.2.41 and 4.2.42 and assuming that the density and diffusivity do not change in time, we get

$$\begin{aligned}\frac{\rho_P\phi_P^n-\rho_P\phi_P^{n-1}}{\Delta t}V_P &+ \frac{1}{2}\sum_f\mathring{F}\phi_f^n-\frac{1}{2}\sum_f\left(\rho\Gamma_\phi\right)_f\mathbf{S}\cdot\left(\nabla\phi\right)_f^n\\ &+ \frac{1}{2}\sum_f\mathring{F}\phi_f^{n-1}-\frac{1}{2}\sum_f\left(\rho\Gamma_\phi\right)_f\mathbf{S}\cdot\left(\nabla\phi\right)_f^{n-1}\\ &= S_uV_P+\frac{1}{2}S_pV_P\phi_P^n+\frac{1}{2}S_pV_P\phi_P^{n-1}.\end{aligned} \tag{4.2.43}$$

This form of temporal discretization is called Crank-Nicolson (CN) method and is second order accurate in time. It requires the face values of ϕ and $\nabla\phi$ as well as the control volume values for

both old $(n-1)$ and new (n) time levels. The face values are calculated from the control volume values on each side of the face, using the appropriate differencing scheme for the convection term and eq. 4.2.31 for the diffusion term. The CN method is unconditionally stable, but does not guarantee boundedness of the solution.

4.2.6.2 Backward Differencing

Since the variation of ϕ in time is assumed to be linear, eq. 4.2.42 provides a second order accurate representation of the time derivative at $t + \frac{1}{2}\Delta t$ only. Assuming the same value for the derivative at time t or $t + \Delta t$ reduces the accuracy to first order. However, if the temporal derivative is discretized to second order, the whole discretization of the transport equation will be second order without the need to center the spatial terms in time. The scheme produced is called Backward Differencing (BD) and uses three time levels,

$$\begin{aligned} \phi^{n-2} &= \phi^{t-\Delta t}, \\ \phi^{n-1} &= \phi^{t}, \\ \phi^{n} &= \phi^{t+\Delta t}, \end{aligned} \tag{4.2.44}$$

to calculate the temporal derivative. Expressing time level $n-2$ as a Taylor expansion around n we get

$$\phi^{n-2} = \phi^{n} - 2\left(\frac{\partial \phi}{\partial t}\right)^{n}\Delta t + 2\left(\frac{\partial^2 \phi}{\partial t^2}\right)^{n}\Delta t^2 + \mathcal{O}\left(\Delta t^3\right), \tag{4.2.45}$$

doing the same for time level $n-1$ we obtain

$$\phi^{n-1} = \phi^{n} - \left(\frac{\partial \phi}{\partial t}\right)^{n}\Delta t + \frac{1}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^{n}\Delta t^2 + \mathcal{O}\left(\Delta t^3\right). \tag{4.2.46}$$

Combining this equation with eq. 4.2.45 produces a second order approximation of the temporal derivative at the new time n as follows

$$\left(\frac{\partial \phi}{\partial t}\right)^{n} = \frac{\frac{3}{2}\phi^{n} - 2\phi^{n-1} + \frac{1}{2}\phi^{n-2}}{\Delta t}. \tag{4.2.47}$$

By neglecting the temporal variation in the face fluxes and derivatives, eq. 4.2.47 produces a fully implicit second order accurate discretization of the general transport equation,

$$\begin{aligned} \frac{\frac{3}{2}\rho_P\phi^{n} - 2\rho_P\phi^{n-1} + \frac{1}{2}\rho_P\phi^{n-2}}{\Delta t}V_P + \sum_f \mathring{F}\phi_f^n - \sum_f \left(\rho\Gamma_\phi\right)_f \mathbf{S}\cdot\left(\nabla\phi\right)_f^n \\ = S_u V_P + S_p V_P \phi_P^n. \end{aligned} \tag{4.2.48}$$

In the CN method, since the flux and non-orthogonal component of the diffusion term have to be evaluated using values at the new time n, it means that it requires inner-iterations during each time step. Coupled with the memory overhead due to the large number of stored variables, this implies that the CN method is expensive compared to the BD described before. The BD method, although cheaper and considerably easier to implement than the CN method, results in a truncation error larger than the latter. This is due to the assumed lack of temporal variation in face fluxes and derivatives. This error manifests itself as an added diffusion. However, if we restrict the Courant number (CFL) to a value below 1, the time step will tend to be very small, keeping temporal diffusion errors to a minimum.

4.2.7 System of Algebraic Equations

At this point, after spatial and temporal discretization and by using equations 4.2.43 or 4.2.48 in every control volume of the domain, a system of algebraic equations of the form

$$[A]\,[\phi] = [R]\,, \tag{4.2.49}$$

is assembled. In eq. 4.2.49, $[A]$ is a sparse matrix, with coefficients a_P on the diagonal and a_N off the diagonal, $[\phi]$ is the vector of ϕ for all control volumes and $[R]$ is the source term vector. When this system is solved, it gives a new set of $[\phi]$ values (the solution for the new time step n). The coefficients a_P include the contribution from all terms corresponding to $[\phi_P^n]$, that is, the temporal derivative, convection and diffusion terms, as well as the linear part of the source term. The coefficients a_N include the corresponding terms for each of the neighboring control volumes. The summation is done over all the control volumes that share a face with the current control volume. The source term includes all terms that can be evaluated without knowing the new $[\phi]$, namely, the constant part of the source term and the parts of the temporal derivative, convection and diffusion terms corresponding to the old time level $n-1$. This system of equations can be solved either by direct or iterative methods. Direct methods give the solution of the system of algebraic equations in a finite number of arithmetic operations. Iterative methods start with an initial guess and then continue to improve the current approximation of the solution until some solution tolerance is met. While direct methods are appropriate for small systems, the number of operations necessary to reach the solution raises with the number of equations squared, making them prohibitively expensive for large systems. Iterative methods are more economical, but they usually pose some requirements on the matrix.

4.2.8 Boundary Conditions and Initial Conditions

Each control volume provides one algebraic equation. Volume integrals are calculated in the same way for every interior control volume, but fluxes through control volume faces coinciding with the domain boundary require special treatment. These boundary fluxes must either be known, or be expressed as a combination of interior values and boundary data. Since they do not give additional equations, they should not introduce additional unknowns. Since there are no nodes outside the boundary, these approximations must be based on one-sided differences or extrapolations.

Mainly, there are three boundary conditions which are used to close the system of equations, namely:

- Zero-gradient boundary condition, defining the solution gradient to be zero. This condition is known as a Neumann type boundary condition.
- Fixed-value boundary condition, defining a specified value of the solution. This is a Dirichlet type condition.
- Symmetry boundary condition, treats the conservation variables as if the boundary was a mirror plane. This condition defines that the component of the solution gradient normal to this plane should be fixed to zero. The parallel components are extrapolated from the interior cells,

For example, for an external aerodynamics simulation we might set the following boundary conditions. At the inflow boundary the velocity is defined as fixed-value and the pressure as zero-gradient. At the outflow boundary, the pressure is defined as a fixed-value and the velocity as a zero-gradient. If symmetry is a concern, symmetrical boundary conditions are used at fixed

boundaries. On a fixed wall, we need to ensure a zero flux through the wall or non penetrating condition. In the case of a no-slip wall, a fixed-value is specified for the velocity ($\mathbf{u} = 0$) in combination with a zero-gradient for the pressure. If the boundary of the wall moves, then the proper boundary condition is a moving-wall velocity which introduces an extra velocity in order to maintain the no-slip condition and ensures a zero flux through the moving boundary.

Together with suitable boundary conditions we need to impose initial conditions. The initial conditions determine the initial state of the governing equations at the initial time for an unsteady problem (usually at $t = 0$), or at the first iteration for an iterative scheme. The better the initial conditions are (the closer to the real solution), the stable and robust the numerical scheme will be and the faster a converged solution will be reached (locally or globally). A common practice in external aerodynamics consist in setting the freestream values of velocity and pressure as initial conditions in the whole domain.

4.3 Discretization Errors

The discretization errors related to the FVM formulation previously presented, results mainly from two sources. The first source of errors is linked to the truncation errors associated with the second order approximation of the temporal and spatial terms (profile assumptions). And the second source of errors is related to mesh quality issues, where the most important quality metrics to consider are non-orthogonality and skewness. In this section, we are going to study the discretization errors due to the profile assumptions and mesh quality.

4.3.1 Taylor Series Expansions

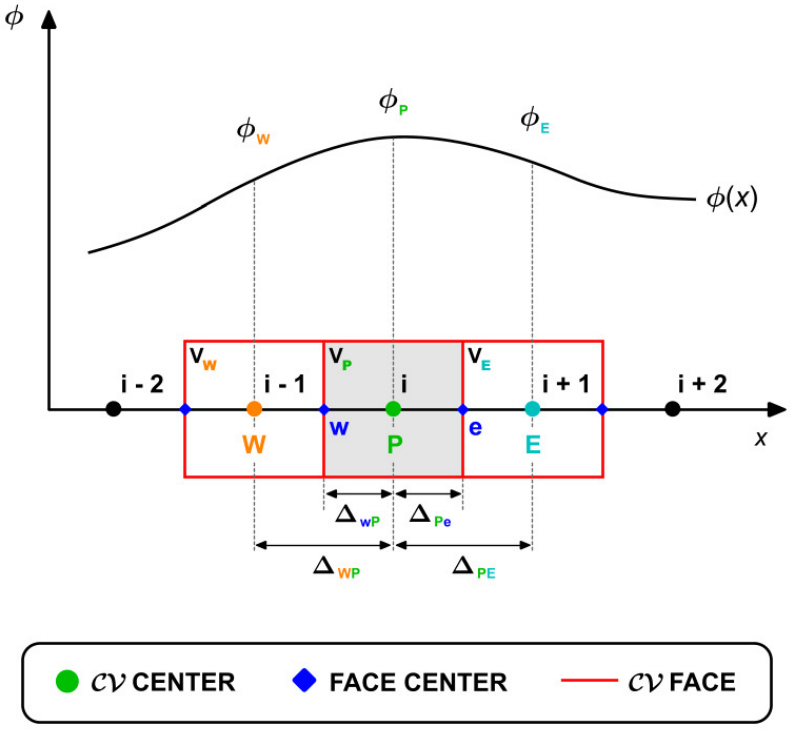

Figure 4.9: *Variation of $\phi(x)$ in a uniform mesh.*

Let us first introduce Taylor Series Expansions (TSE), which we are going to use to determine the order of the truncation error $\mathcal{O}$ of the various approximations presented in the previous sections.

Consider the equally spaced mesh shown in figure 4.9, where $\Delta_{\mathbf{WP}} = \Delta_{\mathbf{PE}} = \Delta x$. According to TSE, any continuous differentiable function can be expressed as an infinite sum of terms that are calculated from the values of the function derivatives at a single point. For $\phi(x)$ the TSE at $\phi(x + \Delta x)$ is equal to

$$\phi(x + \Delta x) = \phi(x) + \Delta x \left(\frac{\partial \phi}{\partial x}\right)_x + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_x + \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_x + \mathcal{HOT}, \tag{4.3.1}$$

where $\mathcal{HOT}$ are the higher order terms. Equation 4.3.1 can be written in a more compact way as

$$\phi(x + \Delta x) = \phi(x) + \sum_{n=1}^{\infty} \frac{(\Delta x)^n}{n!} \frac{\partial^n \phi}{\partial x^n}. \tag{4.3.2}$$

Similarly, the TSE of $\phi(x)$ at $\phi(x - \Delta x)$ is equal to

$$\phi(x - \Delta x) = \phi(x) - \Delta x \left(\frac{\partial \phi}{\partial x}\right)_P + \frac{1}{2}\Delta x^2 \left(\frac{\partial^2 \phi}{\partial x^2}\right)_x - \frac{1}{3!}\Delta x^3 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_x + \mathcal{HOT}, \tag{4.3.3}$$

which can be written in a more compact way as

$$\phi(x - \Delta x) = \phi(x) + \sum_{n=1}^{\infty} \left[(-1)^n \frac{(\Delta x)^n}{n!}\right] \frac{\partial^n \phi}{\partial x^n}. \tag{4.3.4}$$

By using TSE, we can obtain approximate expressions for the first and higher derivatives at a point located in the direction x in terms of the function values at neighboring points.

Let us consider the discrete points W, P and E shown in figure 4.9. The TSE of ϕ_E around point P (for P located midway between points W and E such that $\Delta_{\mathbf{WP}} = \Delta_{\mathbf{PE}} = \Delta x$ and $\Delta_{\mathbf{wP}} = \Delta_{\mathbf{Pe}} = \Delta_{\mathbf{PE}}/2 = \Delta x/2$), is given by

$$\phi_E = \phi_P + \Delta x \left(\frac{\partial \phi}{\partial x}\right)_P + \frac{1}{2!}\Delta x^2 \left(\frac{\partial^2 \phi}{\partial x^2}\right)_P + \frac{1}{3!}\Delta x^3 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}. \tag{4.3.5}$$

Equation 4.3.5 may be rearranged to give

$$\left(\frac{\partial \phi}{\partial x}\right)_P = \frac{\phi_E - \phi_P}{\Delta x} - \frac{1}{2!}\Delta x \left(\frac{\partial^2 \phi}{\partial x^2}\right)_P - \frac{1}{3!}\Delta x^2 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}. \tag{4.3.6}$$

By summing all terms which involve the multiplying factor Δx and higher and representing them as $\mathcal{O}(\Delta x)$, yields to

$$\left(\frac{\partial \phi}{\partial x}\right)_P = \frac{\phi_E - \phi_P}{\Delta x} + \mathcal{O}(\Delta x), \tag{4.3.7}$$

which is an approximation for the first derivative of ϕ with respect to x at the discrete point P. The term $\mathcal{O}(\Delta x)^n$ represents the truncation error of the approximation and determines the rate at which the error decreases as the spacing between the points is reduced. The smaller Δx is, the smaller the error.

Equation 4.3.7 is known as the forward difference approximation of the first derivative and is first order accurate because the truncation error is of order one or $\mathcal{O}(\Delta x)$.

In our notation we used discrete values ϕ_P, ϕ_E and ϕ_W, by using subscript index notation i to represent the discrete points (this notation might be more amenable to follow for some readers), eq. 4.3.7 is written as

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_i}{\Delta x} + \mathcal{O}(\Delta x). \tag{4.3.8}$$

Let us consider the TSE of ϕ_W around the discrete point P which is given by

$$\phi_W = \phi_P - \Delta x \left(\frac{\partial \phi}{\partial x}\right)_P + \frac{1}{2}\Delta x^2 \left(\frac{\partial^2 \phi}{\partial x^2}\right)_P - \frac{1}{3!}\Delta x^3 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}. \tag{4.3.9}$$

Rearranging and grouping eq. 4.3.9 we obtain

$$\left(\frac{\partial \phi}{\partial x}\right)_P = \frac{\phi_P - \phi_W}{\Delta x} + \mathcal{O}(\Delta x), \tag{4.3.10}$$

or

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_i - \phi_{i-1}}{\Delta x} + \mathcal{O}(\Delta x). \tag{4.3.11}$$

Equation 4.3.10 (or eq. 4.3.11) is known as the backward difference approximation of the first derivative and is first order accurate because the truncation error is of order one or $\mathcal{O}(\Delta x)$.

Now consider the TSE in eq. 4.3.5 and eq. 4.3.9, which are repeated here for convenience

$$\phi_E = \phi_P + \Delta x \left(\frac{\partial \phi}{\partial x}\right)_P + \frac{1}{2!}\Delta x^2 \left(\frac{\partial^2 \phi}{\partial x^2}\right)_P + \frac{1}{3!}\Delta x^3 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}, \tag{4.3.12}$$

and

$$\phi_W = \phi_P - \Delta x \left(\frac{\partial \phi}{\partial x}\right)_P + \frac{1}{2}\Delta x^2 \left(\frac{\partial^2 \phi}{\partial x^2}\right)_P - \frac{1}{3!}\Delta x^3 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}. \tag{4.3.13}$$

Subtracting eq. 4.3.13 from eq. 4.3.12, we obtain

$$\phi_E - \phi_W = 2\Delta x \left(\frac{\partial \phi}{\partial x}\right)_P + 2\frac{1}{3!}\Delta x^3 \left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}. \tag{4.3.14}$$

Rearranging and grouping eq. 4.3.14, we obtain

$$\left(\frac{\partial \phi}{\partial x}\right)_P = \frac{\phi_E - \phi_W}{2\Delta x} + \mathcal{O}(\Delta x)^2, \tag{4.3.15}$$

or

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x)^2. \tag{4.3.16}$$

Equation 4.3.15 (or eq. 4.3.16) is known as the central difference approximation of the first derivative and is second order accurate because the truncation error is of order two or $\mathcal{O}(\Delta x)^2$. Approximations for the derivatives in the other directions are obtained in exactly the same fashion.

4.3.2 Accuracy of the Upwind Scheme and Central Differencing Scheme

Let us study the truncation error associated with the upwind and central differencing schemes presented in section 4.2.2. Consider the equally spaced mesh shown in figure 4.9, such that $\Delta_{\mathbf{WP}} = \Delta_{\mathbf{PE}} = \Delta x$ and $\Delta_{\mathbf{Pe}} = \Delta_{\mathbf{eE}} = \Delta_{\mathbf{PE}}/2 = \Delta x/2$. Using TSE about face e, we obtain

$$\phi_P = \phi_e - \left(\frac{\Delta x}{2}\right)\left(\frac{\partial \phi}{\partial x}\right)_e + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 \phi}{\partial x^2}\right)_e - \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 \phi}{\partial x^3}\right)_e + \mathcal{HOT}, \tag{4.3.17}$$

$$\phi_E = \phi_e + \left(\frac{\Delta x}{2}\right)\left(\frac{\partial \phi}{\partial x}\right)_e + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2 \phi}{\partial x^2}\right)_e + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3 \phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.18}$$

Truncating equations 4.3.17-4.3.18 at the first order derivative we get

$$\phi_P = \phi_e + \mathcal{O}(\Delta x), \tag{4.3.19}$$

$$\phi_E = \phi_e + \mathcal{O}(\Delta x), \tag{4.3.20}$$

and recalling eq. 4.2.17, which we rewrite here for convenience

$$\phi_e = \begin{cases} \phi_e = \phi_P & \text{for} \quad \mathring{F} \geq 0, \\ \phi_e = \phi_N & \text{for} \quad \mathring{F} < 0. \end{cases} \tag{4.3.21}$$

From equations 4.3.19-4.3.21 we see that the upwind differencing scheme is first order accurate because the truncation error is of order one or $\mathcal{O}(\Delta x)$.

Adding equations 4.3.17-4.3.18, rearranging and manipulating we obtain

$$\phi_e = \frac{\phi_P + \phi_E}{2} - \frac{(\Delta x)^2}{8}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_e + \mathcal{HOT}, \tag{4.3.22}$$

which is a central differencing approximation of ϕ at face e. Truncating eq. 4.3.22 at the second order derivative yields to

$$\phi_e = \frac{\phi_P + \phi_E}{2} + \mathcal{O}(\Delta x)^2, \tag{4.3.23}$$

hence the central differencing scheme is second order accurate because the truncation error is of order two or $\mathcal{O}(\Delta x)^2$.

Equation 4.3.23 is second order accurate only on uniform meshes (when $\Delta_{\mathbf{WP}} = \Delta_{\mathbf{PE}} = \Delta x$ and $\Delta_{\mathbf{Pe}} = \Delta_{\mathbf{eE}} = \Delta_{\mathbf{PE}}/2 = \Delta x/2$). On non uniform meshes we need to use equations 4.2.14-4.2.15.

Consider the mesh shown in figure 4.9 and let us say that $\Delta_{\mathbf{WP}} \neq \Delta_{\mathbf{PE}}$ and $\Delta_{\mathbf{Pe}} \neq \Delta_{\mathbf{eE}}$ (non-uniform mesh). The interpolated face value e can be found by using a weighted linear interpolation as follows

$$\phi_e = e_x \phi_P + (1 - e_x)\, \phi_E, \tag{4.3.24}$$

where the interpolation factor e_x, is defined as the ratio of the distances eE and PE, *i.e.*,

$$e_x = \frac{eE}{PE} = \frac{\mid x_e - x_E \mid}{\mid \Delta_{\mathbf{PE}} \mid}. \tag{4.3.25}$$

Let us find the truncation error of eq. 4.3.24. The TSE of ϕ_P around the discrete point E is given by

$$\phi_P = \phi_E - (x_E - x_P)\left(\frac{\partial \phi}{\partial x}\right)_E + \frac{(x_E - x_P)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_E - \frac{(x_E - x_P)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_E + \mathcal{HOT}. \quad (4.3.26)$$

Solving for $\partial\phi/\partial x$ in eq. 4.3.26 and truncating in the second order derivative, we obtain

$$\left(\frac{\partial \phi}{\partial x}\right)_E = \frac{\phi_E - \phi_P}{(x_E - x_P)}. \quad (4.3.27)$$

The TSE of ϕ_e about the discrete point E is expressed as follows

$$\phi_e = \phi_E - (x_E - x_e)\left(\frac{\partial \phi}{\partial x}\right)_E + \frac{(x_E - x_e)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_E - \frac{(x_E - x_e)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_E + \mathcal{HOT}. \quad (4.3.28)$$

Substituting eq. 4.3.27 into eq. 4.3.28 in order to eliminate the first order derivative, we obtain

$$\phi_e = \phi_E - (x_E - x_e)\frac{\phi_E - \phi_P}{(x_E - x_P)} + \frac{(x_E - x_e)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_E - \frac{(x_E - x_e)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_E + \mathcal{HOT}. \quad (4.3.29)$$

Notice that the truncation error of the first order derivative in eq. 4.3.26 and eq. 4.3.28 is of the same order. Rearranging and manipulating eq. 4.3.29 yields to

$$\phi_e = (1 - e_x)\,\phi_E + e_x \phi_P + \frac{(x_E - x_e)(x_e - x_P)}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_E - \frac{(x_E - x_e)^3 (x_e - x_P)}{3!(x_E - x_e)}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_E + \mathcal{HOT}. \quad (4.3.30)$$

where the interpolation factor e_x is given by

$$e_x = \frac{x_E - x_e}{x_E - x_P} \quad (4.3.31)$$

The truncation error in eq. 4.3.30 is proportional to the product of the mesh spacing, hence the scheme is second order accurate on uniform and non uniform meshes. Notice that when the face e is situated midway between the two neighboring control volumes (uniform mesh), e_x is equal to 0.5 and eq. 4.3.30 reduces to eq. 4.3.23.

4.3.3 Mean Value Approximation

Consider the variation of the function $\phi(x)$ within the control volume V_P (as shown in figure 4.9). The TSE of $\phi(x)$ about point P is equal to

$$\phi(x) = \phi_P + (x - x_P)\left(\frac{\partial \phi}{\partial x}\right)_P + \frac{(x - x_P)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_P + \frac{(x - x_P)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}. \quad (4.3.32)$$

Integrating eq. 4.3.32 over the control volume V_P, yields

$$\int_{V_P} \phi\,(x)\, dV = \int_{V_P} \left[\phi_P + (x - x_P)\left(\frac{\partial \phi}{\partial x}\right)_P + \frac{(x - x_P)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_P + \frac{(x - x_P)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}\right]. \quad (4.3.33)$$

Assuming that $\phi(x)$ varies linearly over the control volume, all derivatives of order higher than $\partial\phi/\partial x$ in eq. 4.3.33 are zero. Also, the term containing the derivative $\partial\phi/\partial x$ is equal to zero since the point P is the centroid of the control volume, which is given by

$$\int_{V_P} (x - x_P)\, dV = 0. \tag{4.3.34}$$

Hence eq. 4.3.33 becomes

$$\int_{V_P} \phi\,(x)\, dV = \phi_P V_P, \tag{4.3.35}$$

dividing eq. 4.3.35 by V_P we obtain

$$\overline{\phi} = \frac{1}{V_P} \int_{V_P} \phi\,(x)\, dV = \phi_P. \tag{4.3.36}$$

Thus, the centroid value ϕ_P represents the mean value $\overline{\phi}$. Equation 4.3.36 is easily calculated since all variables at the centroid of V_P are known, no interpolation is needed. The above approximation becomes exact if ϕ is either constant or varies linearly within the control volume; otherwise, it is a second order approximation. The above analysis can be applied to any variable being represented by its volume or face centroid value.

4.3.4 Gradient Approximation

Consider the equally spaced mesh shown in figure 4.9, such that $\Delta_{\mathbf{WP}} = \Delta_{\mathbf{PE}} = \Delta x$ and $\Delta_{\mathbf{Pe}} = \Delta_{\mathbf{eE}} = \Delta_{\mathbf{PE}}/2 = \Delta x/2$. Let us study the truncation error associated in representing the face gradient $(\partial\phi/\partial x)_e$ as

$$\left(\frac{\partial\phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{\Delta x}. \tag{4.3.37}$$

Using TSE about face e, we obtain

$$\phi_E = \phi_e + \left(\frac{\Delta x}{2}\right)\left(\frac{\partial\phi}{\partial x}\right)_e + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2\phi}{\partial x^2}\right)_e + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3\phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.38}$$

$$\phi_P = \phi_e - \left(\frac{\Delta x}{2}\right)\left(\frac{\partial\phi}{\partial x}\right)_e + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2\left(\frac{\partial^2\phi}{\partial x^2}\right)_e - \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3\left(\frac{\partial^3\phi}{\partial x^3}\right)_e + \mathcal{HOT}, \tag{4.3.39}$$

Subtracting eq. 4.3.39 from eq. 4.3.38 , rearranging and manipulating we obtain

$$\left(\frac{\partial\phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{\Delta x} - \frac{\Delta x^2}{48}\left(\frac{\partial^3\phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.40}$$

Truncating eq. 4.3.40 at the third order derivative we get

$$\left(\frac{\partial\phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{\Delta x} + \mathcal{O}(\Delta x)^2. \tag{4.3.41}$$

Therefore the assumption of linear variation in eq. 4.3.41 leads to a second order accurate approximation of $(\partial\phi/\partial x)_e$ because the truncation error is of order two or $\mathcal{O}(\Delta x)^2$.

4.3.5 Spatial and Temporal Linear Variation

Let us study the truncation error associated with the assumption of spatial and temporal linear variation of ϕ. Using TSE around the discrete point P (refer to figure 4.9) and time t yields to

$$\phi(x) = \phi_P + (x - x_P)\left(\frac{\partial \phi}{\partial x}\right)_P + \frac{(x - x_P)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_P + \frac{(x - x_P)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + \mathcal{HOT}, \tag{4.3.42}$$

$$\phi(t + \Delta t) = \phi^t + \Delta t\left(\frac{\partial \phi}{\partial t}\right)^t + \frac{\Delta t^2}{2!}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^t + \frac{\Delta t^3}{3!}\left(\frac{\partial^3 \phi}{\partial t^3}\right)^t + \mathcal{HOT}. \tag{4.3.43}$$

Truncating equations 4.3.42-4.3.43 at the second order derivative we get

$$\phi(x) = \phi_P + (x - x_P)\left(\frac{\partial \phi}{\partial x}\right)_P + \mathcal{O}(\Delta x)^2, \tag{4.3.44}$$

$$\phi(t + \Delta t) = \phi^t + \Delta t\left(\frac{\partial \phi}{\partial t}\right)^t + \mathcal{O}(\Delta t)^2. \tag{4.3.45}$$

From equations 4.3.44-4.3.45 we see that the assumption of spatial and temporal linear variation is second order accurate because the truncation error is of order two or $\mathcal{O}(\Delta x)^2$ in space, and $\mathcal{O}(\Delta t)^2$ in time.

4.3.6 Mesh Induced Errors

The influence of mesh non-orthogonality on the solution accuracy has been described in section 4.2.3. When the mesh is orthogonal (figure 4.10), the face gradient of the transported quantity ϕ in eq. 4.2.18 can be calculated by using eq. 4.2.23. This equation uses the ϕ values of the control volumes straddling the face f and is second order accurate only on orthogonal meshes.

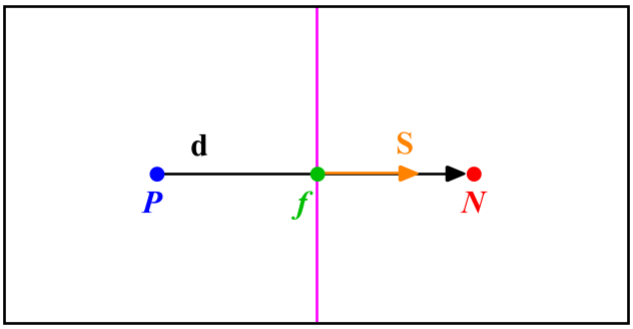

Figure 4.10: *Orthogonal and non-skew mesh. Notice that the vectors* $\boldsymbol{d}$ *and* $\boldsymbol{S}$ *are parallel.*

For non-orthogonal meshes (figure 4.11), we computed the face gradient of the transported quantity ϕ using eq. 4.2.31. In this equation we introduced a correction to improve the accuracy of the face gradient in the case of non-orthogonality. The non-orthogonality affect the solution by adding numerical diffusion to the solution, hence reducing the accuracy. Non-orthogonality can also lead to oscillatory solutions (unboundedness), which in turn can lead to nonphysical values and divergence. The higher the non-orthogonal angle (the angle between the face area vector **S**

and the vector **d** in figure 4.11), the higher the numerical diffusion, and this in fact reduces the accuracy of the numerical method.

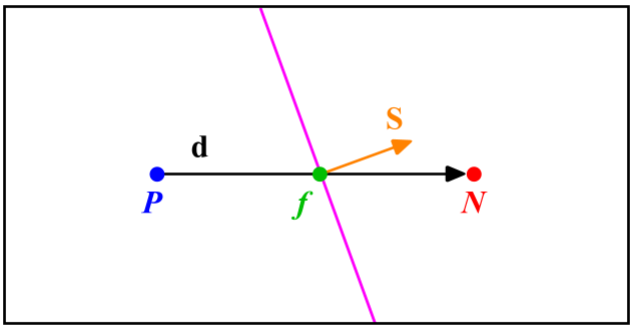

Figure 4.11: *Non-orthogonal and non-skew mesh. Notice that the vectors* $\boldsymbol{d}$ *and* $\boldsymbol{S}$ *are not parallel. The angle between the vector* $\boldsymbol{d}$ *and* $\boldsymbol{S}$ *is the non-orthogonal angle.*

Let us now study the influence of skewness on the solution accuracy. Skewness can be defined as the deviation of the face centroid f from the point where the vector **d** intercepts the face. This situation is shown in figure 4.12, where f is the face centroid, f_i is the point where the vector **d** intersects the face f, and Δ_i is the vector that represents the deviation of f_i from f. Under these conditions, the face values linearly interpolated from the control volumes V_P and V_N, no longer accurately represent the value of the face center.

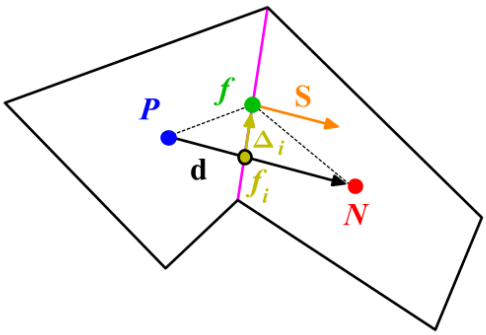

Figure 4.12: *Skewness error in neighboring control volumes; where* f_i *represents the face interpolated value,* f *the face centroid, and the vector* Δ_i *is the deviation of* f_i *from* f*.*

With reference to figure 4.12, the degree of skewness can be measured ass follows,

$$\psi_{skew} = \frac{|\,\Delta_i\,|}{|\,\mathbf{d}\,|}. \tag{4.3.46}$$

Skewness adds numerical diffusion to the solution and reduces the accuracy of the numerical method. It also leads to unboundedness, which in turn can conduct to nonphysical results and/or divergence of the solution.

In some situations, it can happen that the point f_i falls outside face f, which leads to even higher truncation errors and more severe unboundedness. This type of scenarios is usually found when dealing with sharp edges or the intersection of two or more surfaces. In general, it is highly

advisable to keep the skewness to a minimum.

The error due to mesh skewness can be reduced by correcting the error introduced due to the deviation of the face interpolated value f_i from the face centroid f, by using the following second order approximation,

$$\phi_f = \phi_{f_i} + \Delta_i \cdot (\nabla\phi)_{f_i} \tag{4.3.47}$$

where ϕ_{f_i} and $(\nabla\phi)_{f_i}$ are the interpolated values of ϕ and $\nabla\phi$ at the point where the vector $\mathbf{d}$ intersects the face f (point f_i), as shown in figure 4.12. In eq. 4.3.47 ϕ_{f_i} and $(\nabla\phi)_{f_i}$ can be evaluated by using linear interpolation as follows

$$\phi_{f_i} = f_x\phi_P + (1 - f_x)\,\phi_N, \tag{4.3.48}$$

$$\nabla\phi_{f_i} = f_x\nabla\phi_P + (1 - f_x)\,\nabla\phi_N. \tag{4.3.49}$$

where the interpolation factor f_x, is defined as the ratio of the distances f_iN and PN (refer to figure 4.12), *i.e.*,

$$f_x = \frac{f_iN}{PN} = \frac{\mid \mathbf{x}_{f_i} - \mathbf{x}_N \mid}{\mid \mathbf{d} \mid}. \tag{4.3.50}$$

Problems can be encountered in the evaluation of the gradients of ϕ in eq. 4.3.49, as the calculation of the gradient of the control volumes V_P and V_N requires the knowledge of the face centroid gradient at point f_i or $(\nabla\phi)_{f_i}$, by using the Gauss theorem (eq. 4.2.33) we obtain,

$$(\nabla\phi)_P = \frac{1}{V_P}\sum_f \mathbf{S}_f\phi_f = \frac{1}{V_P}\sum_f \mathbf{S}_f\left[\phi_{f_i} + \Delta_i \cdot (\nabla\phi)_{f_i}\right]. \tag{4.3.51}$$

One way to circumvent this problem, is by computing an initial approximation of $(\nabla\phi)_{f_i}$ by using centered differences as follows,

$$\nabla\phi_{f_i} = \frac{(\phi_N - \phi_P)}{|\mathbf{d}|}. \tag{4.3.52}$$

Then eq. 4.3.51 is used to compute $(\nabla\phi)_P$ and $(\nabla\phi)_N$ based on the initial approximation of $(\nabla\phi)_{f_i}$. Once we obtain the new face gradient $(\nabla\phi)_{f_i}$ by using eq. 4.3.49, we can improve this initial approximation by iterating again using the newly computed value. At the end of the iterative process, the corrected value of ϕ_f is computed by using eq. 4.3.47 and the corrected value of $\nabla\phi_f$ is approximated by linearly interpolating the gradient of the neighboring control volumes.

Alternatively, we can compute the gradient of the control volumes straddling face f by using Gauss theorem (eq. 4.2.33), which we repeat here for convenience

$$(\nabla\phi)_P = \frac{1}{V_P}\sum_f (\mathbf{S}_f\phi_f), \tag{4.3.53}$$

where the value ϕ_f on face f can be evaluated by using linear interpolation (eq. 4.2.14). Once we have obtained the gradient of the neighboring control volumes V_P and V_N (by using eq. 4.3.53), we can obtain an initial approximation to the face gradient by using arithmetic average as follows,

$$\nabla\phi_f = \frac{(\nabla\phi_P + \nabla\phi_N)}{2}. \tag{4.3.54}$$

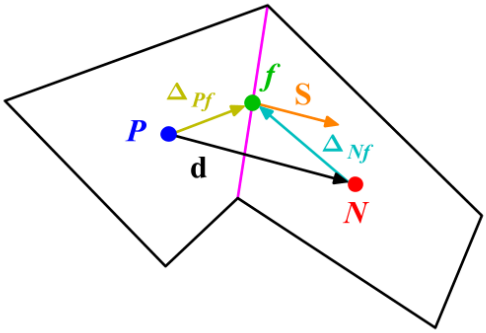

Figure 4.13: *Orthogonal and skew mesh; where Δ_{Pf} is the vector connecting the centroid of the control volume V_P with the face center f, and Δ_{Nf} is the vector connecting the centroid of the control volume V_N with the face center f.*

This initial approximation of the face interpolated values is then improved by doing a linear reconstruction from the control volumes centroid values to the face f (linear extrapolation). By looking at figure 4.13, this approximation is given by

$$\phi_f = \frac{(\phi_P + \Delta_{Pf} \cdot \nabla\phi_P) + (\phi_N + \Delta_{Nf} \cdot \nabla\phi_N)}{2}. \tag{4.3.55}$$

Equation 4.3.55 is a second order approximation to the face value ϕ_f. We now can use this new approximation to compute a new control volume gradient value by using eq. 4.3.53. Finally, we find the face corrected gradient by using linear interpolation as follows,

$$(\nabla\phi)_f = f_x \, (\nabla\phi)_P + (1 - f_x) \, (\nabla\phi)_N \, . \tag{4.3.56}$$

Both of the previously presented approaches to treat the skewness suggest an iterative approach for computing successively better approximations to the face values. In practice, only two or three iterations are used.

Skewness and non-orthogonality can be presented together, reducing significatively the accuracy of the numerical scheme (refer to figure 4.14). Whenever one of these mesh induced errors are presented (figures 4.11-4.14), corrections should be applied in order to avoid numerical diffusion, unboundedness, and to maintain second order accuracy. In spite of the fact that the methods previously presented to handle non-orthogonality and skewness are second order accurate, they use a large computational stencil, which implies larger truncation errors, and can also lead to potential unboundedness.

To approximate the convective fluxes with the highest accuracy by using the approximations presented in the previous sections (*e.g.*, CD and midpoint rule), the vector $\mathbf{d}$ connecting the centroid of two neighboring control volumes should pass through the center of the common face f (non-skew mesh). Maximum accuracy for the diffusive flux is obtained when the vector $\mathbf{d}$ connecting the centroid of two neighboring control volumes is orthogonal to the face f and passes through the face center (orthogonal and non-skew mesh). Unfortunately, this type of meshes are more an exception than a rule. Hence, mesh generation requires careful user input and good meshing practices in order to avoid highly skewed and/or awful non-orthogonal meshes.

From our discussion, we have seen the importance of mesh quality in the solution accuracy. Highly skewed and/or awful non-orthogonal meshes, will substantially reduce the accuracy of the numerical method and will add numerical diffusion to the solution, which in turn will smear the gradients of the transported quantity ϕ. Bad quality meshes can also lead to oscillatory

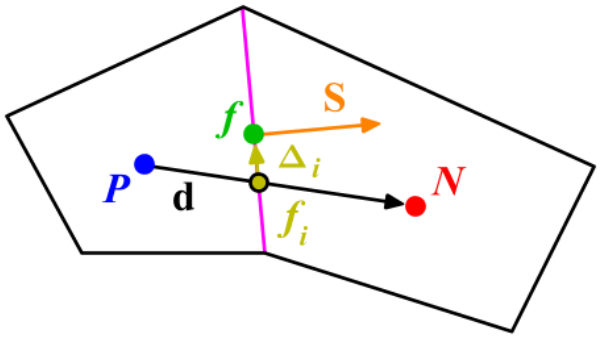

Figure 4.14: *Non-orthogonal and skew mesh; where f_i represents the face interpolated value, f the face centroid and the vector Δ_i is the deviation of f_i from f.*

solutions (unboundedness), which in turn can conduct to nonphysical values and therefore divergence. In practice, in order to avoid unboundedness when reconstructing the face gradients we use gradient limiters (also known as slope limiters), to bound the face gradients so as to avoid undershoots and overshoots on the solution. In this manuscript we will not discuss gradient limiters, but the interested reader should refer to references [3, 22, 27, 24, 12, 25, 26, **?**, 28].

4.3.7 Mesh Spacing

In the previous sections, we employed TSE to determine the accuracy of the approximations used to find the face values of ϕ and $\nabla\phi$. It was found that the weighted linear interpolation eq. 4.3.24 is valid on uniform meshes (see figure 4.15) and non-uniform meshes (see figure 4.16). On uniform meshes, eq. 4.3.24 reduces to an arithmetic average between the two neighboring control volumes.

As we assumed that the values of the transported quantity ϕ are computed and stored in the centroid of the control volume V_P, and at the centroid of the faces of the same control volume; the mean value approximation (eq. 4.3.36), is valid on both uniform and non-uniform meshes.

However, the value of the face gradient approximation, eq. 4.3.37, is only valid on uniform meshes (refer to section 4.3.4). For non-uniform meshes (see figure 4.16), some of the terms in the TSE do not cancel, leaving a formal truncation error of order one or $\mathcal{O}(\Delta x)$, this scenario will be studied hereafter.

Let us consider the unequally spaced mesh shown in figure 4.16, such that $\Delta_{\mathbf{WP}} < \Delta_{\mathbf{PE}}$, $\Delta_{\mathbf{Ww}} < \Delta_{\mathbf{wP}}$, and $\Delta_{\mathbf{Pe}} < \Delta_{\mathbf{eE}}$. Using TSE about face e, we obtain

$$\phi_E = \phi_e + \Delta_{\mathbf{eE}}\left(\frac{\partial\phi}{\partial x}\right)_e + \frac{1}{2!}\Delta^2_{\mathbf{eE}}\left(\frac{\partial^2\phi}{\partial x^2}\right)_e + \frac{1}{3!}\Delta^3_{\mathbf{eE}}\left(\frac{\partial^3\phi}{\partial x^3}\right)_e + \mathcal{HOT}, \tag{4.3.57}$$

$$\phi_P = \phi_e - \Delta_{\mathbf{Pe}}\left(\frac{\partial\phi}{\partial x}\right)_e + \frac{1}{2!}\Delta^2_{\mathbf{Pe}}\left(\frac{\partial^2\phi}{\partial x^2}\right)_e - \frac{1}{3!}\Delta^3_{\mathbf{Pe}}\left(\frac{\partial^3\phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.58}$$

Subtracting eq. 4.3.58 from eq. 4.3.57, we obtain

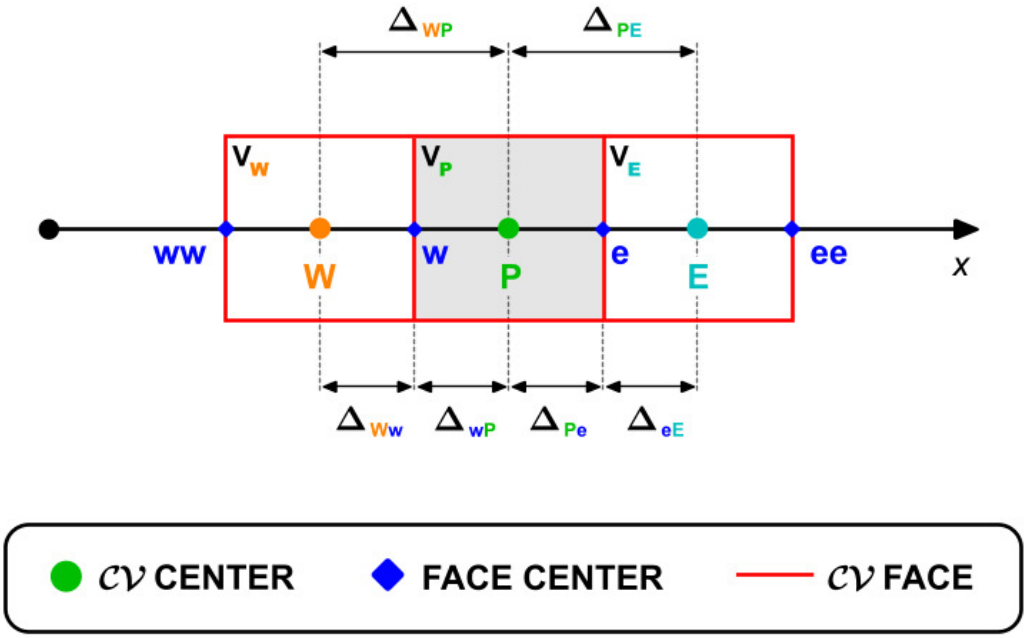

Figure 4.15: *Uniform mesh. Notice that* $\Delta_{WP} = \Delta_{PE} = \Delta x$, $\Delta_{wP} = \Delta_{Pe}$, *and* $\Delta_{Ww} = \Delta_{Pe} = \Delta_{eE} = \Delta_{PE}/2 = \Delta x/2$.

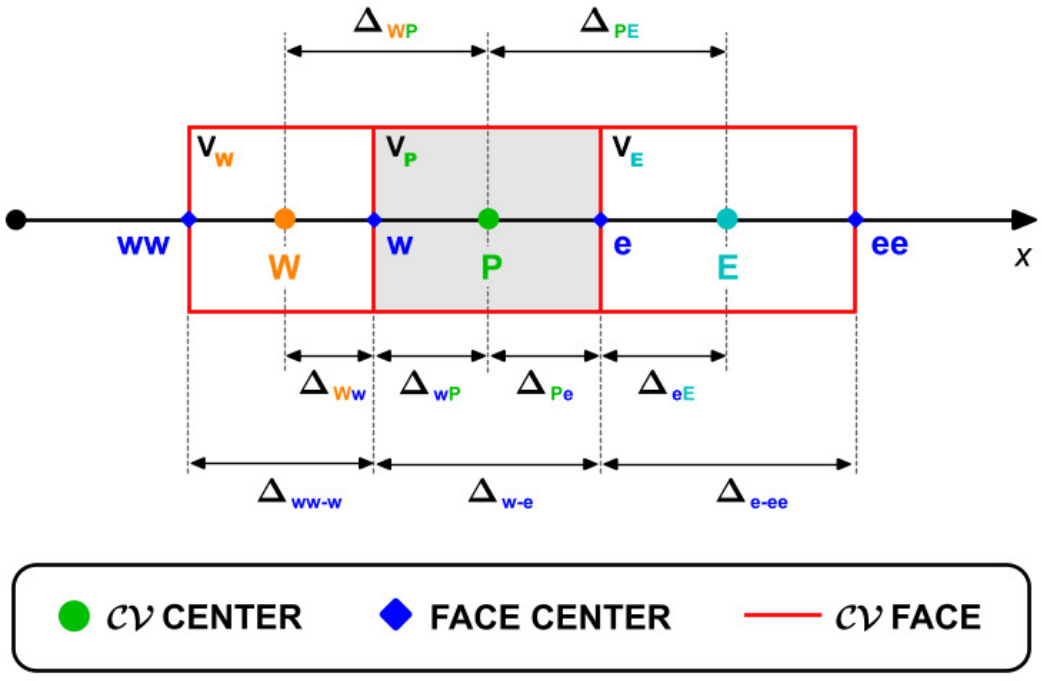

Figure 4.16: *Non-uniform mesh. Notice that* $\Delta_{WP} < \Delta_{PE}$, $\Delta_{Ww} < \Delta_{wP}$, *and* $\Delta_{Pe} < \Delta_{eE}$.

$$\phi_E - \phi_P = (\Delta_{\mathbf{eE}} + \Delta_{\mathbf{Pe}}) \left(\frac{\partial \phi}{\partial x}\right)_e + \frac{1}{2} (\Delta_{\mathbf{eE}}^2 - \Delta_{\mathbf{Pe}}^2) \left(\frac{\partial^2 \phi}{\partial x^2}\right)_e + \frac{1}{6} (\Delta_{\mathbf{eE}}^3 + \Delta_{\mathbf{Pe}}^3) \left(\frac{\partial^3 \phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.59}$$

Solving for $(\partial\phi/\partial x)_e$ in eq. 4.3.59, yields to

$$\left(\frac{\partial \phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{(\Delta_{\mathbf{eE}} + \Delta_{\mathbf{Pe}})} - \frac{1}{2} \frac{(\Delta_{\mathbf{eE}}^2 - \Delta_{\mathbf{Pe}}^2)}{(\Delta_{\mathbf{eE}} + \Delta_{\mathbf{Pe}})} \left(\frac{\partial^2 \phi}{\partial x^2}\right)_e - \frac{1}{6} \frac{(\Delta_{\mathbf{eE}}^3 + \Delta_{\mathbf{Pe}}^3)}{(\Delta_{\mathbf{eE}} + \Delta_{\mathbf{Pe}})} \left(\frac{\partial^3 \phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.60}$$

In eq. 4.3.60 the truncation error ϵ_t is given by

$$\epsilon_t = -\frac{1}{2} \frac{(\Delta_{\mathbf{eE}}^2 - \Delta_{\mathbf{Pe}}^2)}{(\Delta_{\mathbf{eE}} + \Delta_{\mathbf{Pe}})} \left(\frac{\partial^2 \phi}{\partial x^2}\right)_e - \frac{1}{6} \frac{(\Delta_{\mathbf{eE}}^3 + \Delta_{\mathbf{Pe}}^3)}{(\Delta_{\mathbf{eE}} + \Delta_{\mathbf{Pe}})} \left(\frac{\partial^3 \phi}{\partial x^3}\right)_e + \mathcal{HOT}. \tag{4.3.61}$$

By inspecting eq. 4.3.61, we notice that if $\Delta_{eE} = \Delta_{Pe}$ (uniform mesh), the leading term of ϵ_t is equal to zero and we obtain a second order accurate approximation to $(\partial\phi/\partial x)_e$. We can also infer that the larger the difference between Δ_{eE} and Δ_{Pe}, the larger the error. Hence, it becomes clear that if we keep the difference between Δ_{eE} and Δ_{Pe} small, the error of the leading term in eq. 4.3.61 will tend to zero.

Let us introduce the mesh growth factor $\mathcal{G}_f$, such that

$$\Delta_{eE} = \mathcal{G}_f \Delta_{Pe}, \tag{4.3.62}$$

$\mathcal{G}_f$ determines how fast or how slow the mesh expands or contracts between adjacent control volumes. Substituting eq. 4.3.62 into 4.3.61, yields to

$$\epsilon_t = -\frac{1}{2} \frac{\left(\mathcal{G}_f^2 \Delta_{\mathbf{Pe}}^2 - \Delta_{\mathbf{Pe}}^2\right)}{(\mathcal{G}_f \Delta_{\mathbf{Pe}} + \Delta_{\mathbf{Pe}})} \left(\frac{\partial^2 \phi}{\partial x^2}\right)_e, \tag{4.3.63}$$

where we only show the leading term of the truncation error ϵ_t. Manipulating eq. 4.3.63 yields to

$$\epsilon_t = \frac{(1 - \mathcal{G}_f)\, \Delta_{\mathbf{Pe}}}{2} \left(\frac{\partial^2 \phi}{\partial x^2}\right)_e. \tag{4.3.64}$$

From eq. 4.3.64 we can see that the truncation error of the centered difference approximation is of order one and proportional to the mesh spacing $\Delta_{\mathbf{Pe}}$. Similarly, the truncation error of the backward difference approximation (eq. 4.3.58) is of order one and proportional to the mesh spacing $\Delta_{\mathbf{Pe}}$ and is given by

$$\epsilon_t = \frac{\Delta_{\mathbf{Pe}}}{2} \left(\frac{\partial^2 \phi}{\partial x^2}\right)_e. \tag{4.3.65}$$

However, if we set the value of $\mathcal{G}_f$ close to one in eq. 4.3.64, the truncation error of the centered difference approximation is significatively smaller than the truncation error of the backward difference approximation (eq. 4.3.65). It is clear that in order to keep small as possible the error of the leading term in eq. 4.3.61, we should use a growth factor $\mathcal{G}_f$ close to unity.

From the previous discussion, it seems that uniform meshes are desirable. The use of uniform meshes to represent complex geometries is not an easy task and it is computational expensive, as it will use the same mesh resolution in areas of high gradients (where we concentrate more control volumes in order to better resolve steep gradients or local features), and areas where the solution change slowly. In practice, we refine the mesh (or concentrate more control volumes), close to walls where we expect boundary layers, in areas of strong gradients, and in zones where we want to better resolve some local features. Far from the walls and areas of steep gradients or zones interest, we use a coarse mesh. Non-uniform meshes are the rule rather than the exception when dealing with complex geometries.

The only thing that we should keep in mind when using non-uniform meshes is that the mesh should be smooth, *i.e.*, there should be no large spacing differences or fast volume transitions between neighboring control volumes (otherwise the solution accuracy will be compromise), and this is achieved by using local refinement an a suitable value of growth factor $\mathcal{G}_f$.

Chapter 5

The Finite Volume Method for Diffusion and Convection-Diffusion Problems

5.1 Steady One-Dimensional Diffusion

WORK IN PROGRESS

5.2 Steady One-Dimensional Convection-Diffusion

In the absence of source terms, steady convection and diffusion of a property ϕ in a given one-dimensional flow field is governed by

$$\frac{\partial}{\partial x}\left(\rho \mathbf{u}\phi\right) = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) \tag{5.2.1}$$

The flow must also satisfy continuity

$$\frac{\partial\left(\rho \mathbf{u}\right)}{\partial x} = 0 \tag{5.2.2}$$

We consider the one-dimensional control volume shown in figure 5.1. Our attention is focused in a general node P, the neighboring nodes are identified by W and E and the control volume faces by w and e.

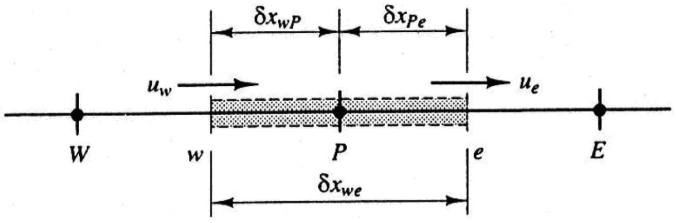

Figure 5.1: *A general nodal point is identified by P and its neighbors in a one-dimensional geometry, the nodes to the west and east, are identified by W and E respectively. The west side face of the control volume is referred to by w and the east side control volume face by e. The distances between the nodes W and P, and between nodes P and E, are identified by Δx_{WP} and Δx_{PE} respectively. Similarly distances between face w and point P and between P and face e are denoted by Δx_{wP} and Δx_{Pe} respectively. Figure 6 shows that the control volume width is $\Delta x = \Delta x_{we}$.*

Integration of eq. 5.2.1 over the control volume shown in figure 5.1 gives

$$(\rho \mathbf{u}\mathbf{S}\phi)_e - (\rho \mathbf{u}\mathbf{S}\phi)_w = \left(\Gamma \mathbf{S}\frac{\partial \phi}{\partial x}\right)_e - \left(\Gamma \mathbf{S}\frac{\partial \phi}{\partial x}\right)_w \tag{5.2.3}$$

and integration of eq. 5.2.2 yields to

$$(\rho \mathbf{u}\mathbf{S})_e - (\rho \mathbf{u}\mathbf{S})_w = 0 \tag{5.2.4}$$

To obtain discretized equations for the convection-diffusion problem we must approximate the terms in eq. 5.2.3. It is convenient to define two variables F and D to represent the convective mass flux per unit area and diffusion conductance at cell faces

$$\begin{aligned} F &= \rho \mathbf{u}, \\ D &= \frac{\Gamma}{\Delta x} \end{aligned} \tag{5.2.5}$$

The cell face values of the variables F and D can be written as

$$\begin{aligned} F_w = (\rho \mathbf{u})_w \quad F_e = (\rho \mathbf{u})_e \,, \\ D_w = \frac{\Gamma}{\Delta x_{WP}} \quad D_e = \frac{\Gamma}{\Delta x_{PE}} \end{aligned} \tag{5.2.6}$$

By employing the central differencing approach to represent the contribution of the diffusion terms on the right hand side of eq. 5.2.3, we obtain

$$\begin{aligned} \left(\Gamma \mathbf{S}\frac{\partial \phi}{\partial x}\right)_e &= \Gamma_e \mathbf{S}_e \left(\frac{\phi_E - \phi_P}{\Delta x_{PE}}\right) \\ \left(\Gamma \mathbf{S}\frac{\partial \phi}{\partial x}\right)_w &= \Gamma_w \mathbf{S}_w \left(\frac{\phi_P - \phi_W}{\Delta x_{WP}}\right) \end{aligned} \tag{5.2.7}$$

for the diffusive terms.

Integrating the convection-diffusion eq. 5.2.3, we obtain

$$F_e \phi_e - F_w \phi_w = D_e \left(\phi_E - \phi_P\right) - D_w \left(\phi_P - \phi_W\right) \tag{5.2.8}$$

and the integrated continuity eq. 5.2.4 becomes

$$F_e - F_w = 0 \tag{5.2.9}$$

In the previous we assumed that $\mathbf{S}_w = \mathbf{S}_e = \mathbf{S}$, so we can divide the left and right hand sides of eq. 5.2.3 by the area $\mathbf{S}$.

The central differencing approximation has been used to represent the diffusion terms which appear on the right hand side of eq. 5.2.8, and it seems logical to try linear interpolation to compute the cell face values for the convective terms on the left hand side of this equation. For a uniform mesh we can write the cell face values of the quantity ϕ as

$$\begin{aligned} \phi_e &= \frac{(\phi_P + \phi_E)}{2} \\ \phi_w &= \frac{(\phi_W + \phi_P)}{2} \end{aligned} \tag{5.2.10}$$

Substitution of the above equations into the convection terms of eq. 5.2.8 yields to

$$\frac{F_e}{2}(\phi_P + \phi_E) - \frac{F_w}{2}(\phi_W + \phi_P) = D_e(\phi_E - \phi_P) - D_w(\phi_P - \phi_W) \tag{5.2.11}$$

Rearranging and grouping eq. 5.2.11 yields to

$$a_P\phi_P = a_W\phi_W + a_E\phi_E \tag{5.2.12}$$

where

$$\begin{aligned} a_W &= D_w + \frac{F_w}{2} \\ a_E &= D_e - \frac{F_e}{2} \\ a_P &= a_W + a_E + (F_e - F_w) \end{aligned} \tag{5.2.13}$$

To solve a one-dimensional convection-diffusion problem we write discretized equations of the form of eq. 5.2.13 for all mesh nodes. This yields to a set of algebraic equations that is solved to obtain the distribution of the transported property ϕ. We also need to defined the initial and boundary condition in order to have a well posed problem. Let us now present the previously discussed concepts by means of a working example.

5.3 Steady one-dimensional convection-diffusion working example

A property ϕ is transported by means of convection and diffusion through the one-dimensional domain sketched in figure 5.2. The governing equation is eq. 5.2.1 and eq. 5.2.2; the boundary conditions are $\phi_0 = 1$ at $x = 0$ and $\phi_L = 0$ at $x = L$. Using five equally spaced cells and the central differencing scheme for the convection and diffusion terms, calculate the distribution of ϕ as a function of x for $u = 0.1\,m/s$ and $u = 2.5\,m/s$, and compare the results with the analytical solution

$$\frac{\phi - \phi_0}{\phi_L - \phi_0} = \frac{e^{\frac{\rho \mathbf{u} x}{\Gamma}} - 1}{e^{\frac{\rho \mathbf{u} L}{\Gamma}} - 1} \tag{5.3.1}$$

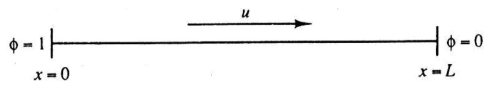

Figure 5.2: *Domain with initial and boundary conditions.*

Let us explain step by step the solution method by using the mesh illustrated in figure 5.3. The domain has been divided into five control volumes, so as $\Delta_x = 0.2\,m$. Note that $L = 1.0\,m$ (length), $\rho = 0.1\,kg/m^3$, $\Gamma = 0.1\,kg/m.s$.

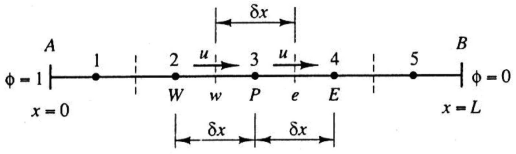

Figure 5.3: *Grid used for discretization.*

The discretized eq. 5.2.12 and its coefficients eq. 5.2.13 apply to all internal control volumes (2, 3 and 4). Control volumes 1 and 5 need special treatment since they are boundary cells. Integrating the convection-diffusion equation eq. 5.2.1 and using central differences for both the convective and diffusive terms, and by applying the boundary and initial conditions we can obtain the solution to our model equation.

The value of ϕ is given at the west face of cell 1 ($\phi_w = \phi_A = 1$) so we do not need to make any approximation in the convective flux term at this boundary. This yields the following equation for node 1,

$$\frac{F_e}{2}(\phi_P + \phi_E) - F_A\phi_A = D_e(\phi_E - \phi_P) - D_A(\phi_P - \phi_A) \tag{5.3.2}$$

For the control volume 5, the ϕ value at the east face is known ($\phi_e = \phi_B = 0$). As before, we obtain the following equation

$$F_B\phi_B - \frac{F_w}{2}(\phi_P + \phi_W) = D_E(\phi_B - \phi_P) - D_w(\phi_P - \phi_W) \tag{5.3.3}$$

Rearranging and grouping equations 5.3.2 and 5.3.3, we get the discretized equations at boundaries nodes,

$$a_P\phi_P = a_W\phi_W + a_E\phi_E + S_u \tag{5.3.4}$$

where

$$a_P = a_W + a_E + (F_e - F_w) - S_P \tag{5.3.5}$$

Note that $D_A = D_B = 2\Gamma/\Delta x = 2D$ and $F_A = F_B = F$.

To introduce the boundary conditions we have suppressed the link to the boundary side and entered the boundary flux as source terms.

Table 5.1: *Nodes discretization.*

Node	a_W	a_E	S_P	S_u
1	0	$D - F/2$	$-(2D + F)$	$(2D + F)\phi_A$
2, 3, 4	$D + F/2$	$D - F/2$	0	0
5	$D + F/2$	0	$-(2D - F)$	$(2D - F)\phi_B$

For $u = 0.1\,m/s$, $F = \rho u = 0.1$, $D = \Gamma/\Delta x = 0.5$, the coefficient are summarized in table 2.

Table 5.2: *Coefficients summary.*

Node	a_W	a_E	S_P	S_u	$a_P = a_W + a_E - S_P$
1	0	0.45	-1.1	$1.1\phi_A$	1.55
2	0.55	0.45	0	0	1.0
3	0.55	0.45	0	0	1.0
4	0.55	0.45	0	0	1.0
5	0.55	0	-0.9	$0.9\phi_B$	1.45

By setting now $\phi_A = 1$ and $\phi_B = 0$, we get the following system of equations,

$$\begin{bmatrix} 1.55 & -0.45 & 0.0 & 0.0 & 0.0 \\ -0.55 & 1.0 & -0.45 & 0.0 & 0.0 \\ 0.0 & -0.55 & 1.0 & -0.45 & 0.0 \\ 0.0 & 0.0 & -0.55 & 1.0 & -0.45 \\ 0.0 & 0.0 & 0.0 & -0.55 & 1.45 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} 1.1 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \tag{5.3.6}$$

The solution of the previous system yields to

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} 0.9421 \\ 0.8006 \\ 0.6276 \\ 0.4163 \\ 0.1579 \end{bmatrix} \tag{5.3.7}$$

The numerical and analytical solutions are compared in table 5.3 and in figure 5.4. The analytical solution for this problem is,

$$\phi(x) = \frac{2.7183 - e^x}{1.7183} \tag{5.3.8}$$

From the results, it can be seen that regardless the coarseness of the mesh, the central differencing (CD) scheme gives reasonable agreement with the analytical solution.

Table 5.3: *Comparison of the numerical and analytical solutions.*

Node	Position	FVM solution	Analytical solution	Difference	Percentage error
1	0.1	0.9421	0.9387	-0.003	-0.36
2	0.3	0.8006	0.7963	-0.004	-0.53
3	0.5	0.6276	0.6224	-0.005	-0.83
4	0.7	0.4163	0.4100	-0.006	-1.53
5	0.9	0.1579	0.1505	-0.007	-4.91

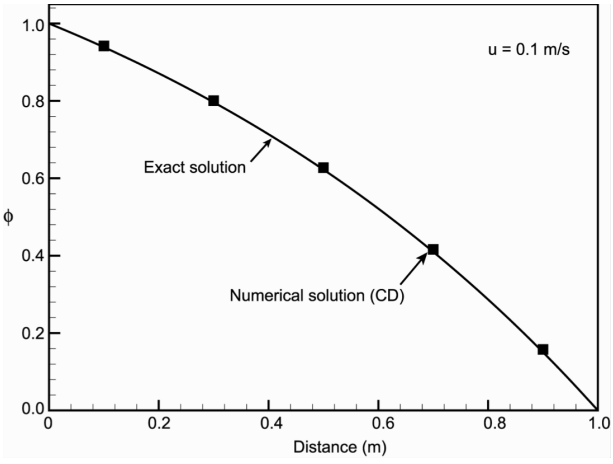

Figure 5.4: *Comparison of the numerical and analytical solutions for* $u = 0.1\,m/s$.

The cell Peclet number (or cell Reynolds number), is defined as,

$$Pe_{cell} = \frac{\rho u \Delta x}{\Gamma} \tag{5.3.9}$$

gives a relation between convection and diffusion. If the local Peclet number is less than 2, it is sufficient for boundedness of the solution by using CD for computing the convective terms. But

when the Peclet number is higher than 2, the solution obtained by using CD for the convective terms shows an oscillatory behavior (unboundedness).

As an exercise, try to compute the cell Peclet number for the previous example.

In the next case ($u = 2.5\,m/s$), the cell Peclet number is higher than 2. Let us see the solution, this is let to you as an exercise. You just need to proceed in exactly the same way as we did before.

5.4 Unsteady One-Dimensional Diffusion

WORK IN PROGRESS

5.5 Unsteady One-Dimensional Convection-Diffusion

WORK IN PROGRESS

Chapter 6

Finite Volume Method Algorithms for Pressure-Velocity Coupling

WORK IN PROGRESS

Bibliography

[1] J. D. Anderson. *Computational Fluid Dynamics: The Basics with Applications.* McGraw-Hill, 1995.

[2] J. D. Anderson. *Fundamentals of Aerodynamics.* McGraw-Hill, 3 edition, 2010.

[3] D. Drikakis and W. Rider. *High-Resolution Methods for Incompressible and Low-Speed Flows.* Springer, 2005.

[4] P. M. Gresho. Incompressible fluid dynamics: some fundamental formulation issues. *Annual Review of Fluid Mechanics*, 23:413–453, 1991.

[5] F. M. White. *Viscous Fluid Flow.* McGraw-Hill, 2005.

[6] P. Kundu, I. Cohen, D. Dowling, and G. Tryggvason. *Fluid Mechanics.* Academic Press, 2015.

[7] P. Tucker. *Advanced Computational Fluid and Aerodynamics.* Cambridge University Press, 2016.

[8] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics: Volume 2: Specific Techniques for Different Flow Categories.* Springer, 1996.

[9] X. Jiang and C-H. Lai. *Numerical Techniques for Direct and Large-Eddy Simulation.* CRC Numerical Analysis and Scientific Computing Series, 2009.

[10] P. A. Durbin and B. A. Pettersson-Reif. *Statistical Theory and Modeling for Turbulent Flows.* Wiley, 2010.

[11] A. Dewan. *Tackling Turbulent Flows in Engineering.* Springer, 2011.

[12] H. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Pearson, 2007.

[13] S. B. Pope. *Turbulent Flows.* Cambridge University Press, 2000.

[14] K. A. Hoffmann and S. T. Chiang. *Computational Fluid Dynamics. Volume 3.* EES, 2000.

[15] P. Sagaut. *Large Eddy Simulation for Incompressible Flows: An Introduction.* Springer, 2005.

[16] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows.* Springer, 2010.

[17] D. C. Wilcox. *Turbulence Modeling for CFD.* DCW Industries, 2006.

[18] T. Cebeci. *Turbulence Models and Their Application: Efficient Numerical Methods with Computer Programs.* Springer, 2004.

[19] R. H. Pletcher, J. C. Tannehill, and D. A. Anderson. *Computational Fluid Mechanics and Heat Transfer*. Taylor and Francis, 1997.

[20] U. Frisch. *Turbulence, the legacy of A. N. Kolmogorov*. Cambridge University Press, 1995.

[21] D. C. Wilcox. Re-assessment of the scale-determining equation for advanced turbulence model. *AIAA Journal*, 26(11):1299–1310, 1988.

[22] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 2002.

[23] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, 1980.

[24] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 2009.

[25] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics, 2002.

[26] H. Jasak. Error analysis and estimation for the finite volume method with application for fluid flows. Technical report, Imperial College London, 1996.

[27] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2003.

[28] C. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998.