

# TransAT

Handbook Series

## 理论手册-算法

Version 5.7.1

© February 2020 Ascomp AG Switzerland

# Contents

|  |          |
|--|----------|
| <b>Contents</b>  | <b>i</b> |
| <b>1 Incompressible Navier–Stokes Equations</b>                    | <b>1</b> |
| 1.1 Single Phase Conservation Equations . . . . .                  | 1        |
| 1.2 The Scalar Equation . . . . .                                  | 2        |
| 1.3 Pressure Solver Implicit Formulation . . . . .                 | 2        |
| 1.3.1 Divergence-Free Flows . . . . .                              | 3        |
| 1.3.2 Discretisation . . . . .                                     | 4        |
| 1.4 Axisymmetric Flows . . . . .                                   | 4        |
| 1.4.1 Axisymmetry . . . . .  | 4        |
| <b>2 Compressible Flow Modelling</b>                               | <b>7</b> |
| 2.1 Single Phase Conservation Equations . . . . .                  | 7        |
| 2.2 N-Phase Conservation Equations . . . . .                       | 7        |
| 2.3 Level-Set Conservation Equations . . . . .                     | 8        |
| 2.4 Equations of state . . . . .                                   | 9        |
| 2.4.1 Ideal Gas equation of state . . . . .                        | 9        |
| 2.4.2 Stiffened Gas Equation of State . . . . .                    | 10       |
| 2.4.3 Tait Equation of State for Water . . . . .                   | 10       |
| 2.4.4 Tait Equation of State for Ammonia . . . . .                 | 11       |
| 2.4.5 Incompressible . . . . .                                     | 11       |
| 2.4.6 Peng-Robinson Equation of State for Water (Liquid) . . . . . | 11       |
| 2.4.7 Peng-Robinson Equation of State for Water (Vapour) . . . . . | 11       |
| 2.4.8 Sutherland’s law . . . . .                                   | 12       |
| 2.5 Pressure Solver Formulation . . . . .                          | 12       |
| 2.5.1 Mass conservation . . . . .                                  | 12       |
| 2.5.2 Pressure equation . . . . .                                  | 13       |
| 2.5.3 Limit of incompressible multiphase flow . . . . .            | 14       |
| 2.5.4 Limit of compressible single phase flow . . . . .            | 14       |

|          |  |           |
|----------|--|-----------|
| 2.5.5    | Pressure Solver Formulation for reactive flows . . . . .         | 15        |
| 2.6      | Numerical Schemes . . . . .                                      | 15        |
| 2.6.1    | Interpolation schemes for compressible density . . . . .         | 16        |
|          | Upwind scheme . . . . .  | 16        |
|          | Linear scheme . . . . .  | 17        |
| 2.6.2    | Unsteady simulation . . . . .                                    | 17        |
| <b>3</b> | <b>Numerical Methods</b>   | <b>19</b> |
| 3.1      | Governing Equations for Steady Flows . . . . .                   | 19        |
| 3.2      | Discretisation of Governing Equations . . . . .                  | 19        |
| 3.2.1    | Flux balance equations . . . . .                                 | 19        |
| 3.2.2    | Approximation of convection terms . . . . .                      | 20        |
| 3.2.3    | Approximation of diffusion terms . . . . .                       | 21        |
| 3.2.4    | Approximation of source terms . . . . .                          | 21        |
| 3.2.5    | Final form of discretised equations . . . . .                    | 22        |
| 3.3      | Boundary Conditions . . . . .                                    | 22        |
| 3.3.1    | Inflow planes . . . . .  | 22        |
| 3.3.2    | Outflow planes . . . . .   | 23        |
| 3.3.3    | Symmetry planes . . . . .  | 23        |
| 3.3.4    | Rigid wall . . . . .   | 23        |
| 3.3.5    | Total pressure boundary condition . . . . .                      | 24        |
|          | Ideal gas . . . . .  | 24        |
|          | Incompressible or weakly compressible fluids . . . . .           | 24        |
|          | Mixture of compressible fluids . . . . .                         | 25        |
|          | Linearization for pressure coefficient and flux update . . . . . | 27        |
| 3.4      | Solution Procedure . . . . .                                     | 27        |
| 3.4.1    | Convergence criterion . . . . .                                  | 27        |
| 3.4.2    | Calculation sequence . . . . .                                   | 28        |
| <b>4</b> | <b>Reactive Flow Modelling</b>                                   | <b>29</b> |
| 4.1      | Conservation equations . . . . .                                 | 29        |
| 4.1.1    | Species equation . . . . .                                       | 29        |
| 4.1.2    | Temperature equation . . . . .                                   | 36        |
| 4.2      | Turbulent reacting flow . . . . .                                | 37        |
| 4.2.1    | RANS . . . . .   | 37        |
| 4.2.2    | LES . . . . .  | 38        |
| 4.3      | Non-Premixed Reactive Flows . . . . .                            | 38        |
| 4.3.1    | Slow Chemistry . . . . .   | 40        |
| 4.3.2    | Infinitely Fast Chemistry . . . . .                              | 40        |

|                     |  |           |
|---------------------|--|-----------|
| <b>5</b>            | <b>Boussinesq Approximation</b>                            | <b>43</b> |
| <b>6</b>            | <b>Non-Newtonian Models</b>                                | <b>45</b> |
| <b>7</b>            | <b>Viscoelasticity</b>                                     | <b>49</b> |
| 7.1                 | Olroyd-B. model . . . . .                                  | 50        |
| 7.2                 | Exponential Phan-Thien-Tanner model (PTT) . . . . .        | 50        |
| 7.3                 | Linear PTT model . . . . .                                 | 50        |
| 7.4                 | Temperature correction using Arrhenius law . . . . .       | 51        |
| 7.5                 | Multiphase viscoelasticity . . . . .                       | 51        |
| 7.6                 | Logarithm of Conformation Tensor (LCT) . . . . .           | 51        |
| <b>8</b>            | <b>Radiation</b>   | <b>53</b> |
| <b>9</b>            | <b>Immersed Surfaces Technique (IST)</b>                   | <b>55</b> |
| 9.1                 | Introduction . . . . .                                     | 55        |
| 9.2                 | Equations and Implementation . . . . .                     | 56        |
| 9.2.1               | Immersed Level Set and solid Heaviside functions . . . . . | 56        |
| 9.2.2               | Solid temperature equation . . . . .                       | 57        |
| <b>10</b>           | <b>Smart Adaptive Run Parametrisation (SArP) algorithm</b> | <b>59</b> |
| 10.1                | Theory . . . . .   | 59        |
| 10.2                | Genetic Algorithm . . . . .                                | 60        |
| 10.2.1              | Structure . . . . .  | 61        |
| 10.2.2              | Triggers and stopping criteria . . . . .                   | 62        |
| Triggers            | . . . . .  | 62        |
| Stopping criteria   | . . . . .  | 63        |
| 10.2.3              | Genetic Operations . . . . .                               | 65        |
| Mutation Operation  | . . . . .  | 65        |
| Crossover Operation | . . . . .  | 66        |
| 10.2.4              | Selection Process . . . . .                                | 67        |
| Fitness function    | . . . . .  | 67        |
| Selection Operators | . . . . .  | 69        |
| 10.2.5              | Prediction and Fitness approximation . . . . .             | 70        |
| Strategy            | . . . . .  | 71        |
| Learning Methods    | . . . . .  | 72        |
| 10.2.6              | Speeding up the EGA . . . . .                              | 73        |
| <b>References</b>   |  | <b>76</b> |



# Chapter 1

## Incompressible Navier–Stokes Equations

### 1.1 Single Phase Conservation Equations

Considering the motion of a Newtonian fluid with constant properties (density and viscosity) velocity  $u_i = (u, v, w)$ , pressure  $p$  and temperature  $T$  are obtained by solving the incompressible Navier-Stokes equations which, in Cartesian–coordinates, are respectively:

$$\frac{\partial u_j}{\partial x_j} = 0 \quad (1.1)$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial(\sigma_{ij})}{\partial x_j} + \rho g_i; \quad i = 1, 2, 3 \quad (1.2)$$

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p u_j \frac{\partial T}{\partial x_j} = \underbrace{\frac{\partial}{\partial x_j} \left( \lambda \frac{\partial T}{\partial x_j} \right)}_{-q''_j} + \dot{Q}''' \quad (1.3)$$

where  $\rho$  represents the density,  $g_i$  is the force of gravity,  $\mu$  is the molecular viscosity,  $\sigma_{ij}$  is the viscous stress tensor and  $\dot{Q}'''$  is a volumetric heat source.  $\rho$  and  $\mu$  are considered as constant for these incompressible equations. Note that the Einstein summation convention applies to repeated indices.

There is a need for closure laws for the viscous stress tensor  $\sigma_{ij}$ . This will in turn express the way a fluid environment is deformed (or strained) under a given stress and thus these constitutive equations are also known as the *stress-strain relations*. Making use of the Stokes' hypothesis yields:

$$\sigma_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad (1.4)$$

Fourier law is also used, which assumes that the thermal conductivity  $\lambda$  is a positive constant, which means that isotropic heat conductivity is assumed.

In some cases, solids or fluids do not have isotropic properties for heat conduction. In this case, thermal conductivity  $\lambda$  can differ according to the conduction direction. The energy equation 1.3 becomes:

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p u_j \frac{\partial T}{\partial x_j} = -\frac{\partial q_j}{\partial x_j} + \dot{Q}''' \quad (1.5)$$

where

$$q_j = -\lambda_{ij} \frac{\partial T}{\partial x_i} \quad (1.6)$$

takes into account the anisotropy of heat conduction.  $\lambda_{ij}$  is a tensor, based on the thermal conductivity along the principle axes.

## 1.2 The Scalar Equation

The evolution equation for the scalar  $C$  has the form:

$$\frac{\partial C}{\partial t} + \frac{\partial}{\partial x_j} (u_j C) = \frac{\partial}{\partial x_j} \left( \mathcal{D} \frac{\partial C}{\partial x_j} \right) + F_C \quad (1.7)$$

where  $\mathcal{D} = \nu/Sc$  is the molecular diffusivity of the scalar, and  $\rho$  is the density of the fluid.  $\nu$  is the kinematic viscosity, and  $Sc$  is the Schmidt number. The source term  $F_C$  represents any source term applied on the scalar, depending on what  $C$  is actually representing.

## 1.3 Pressure Solver Implicit Formulation

We start with the discretised version of the momentum equations given by,

$$A_{P,ij} u_{P,j}^{k+1} = \sum_{nb} A_{nb,ij} u_{nb,j}^{k+1} + S_u - \frac{\partial p_P^{k+1}}{\partial x_i} \Omega_P \quad (1.8)$$

where  $A_{ij}$  is the coefficient (diagonal) tensor with the diagonals being,  $A_u$ ,  $A_v$ , and  $A_w$ , subscript  $P$  denotes the point being discretised, and  $nb$  denotes its neighbours.  $\Omega_P$  is the cell volume. We split the unknown pressure ( $p^{k+1}$ ) into two parts,  $p^{k+1} = p^k + p'$  and introduce an intermediate velocity  $u_i^{k*}$  to obtain:

$$A_{P,ij}(u_{P,j}^{k*} + u_{P,j}^{k+1} - u_{P,j}^{k*}) = \sum_{nb} A_{nb,ij} u_{nb,j}^{k+1} + S_u - \frac{\partial p_P^k}{\partial x_i} \Omega_P - \frac{\partial p'_P}{\partial x_i} \Omega_P \quad (1.9)$$

Applying operator splitting we obtain two equations,

$$A_{P,ij} u_{P,j}^{k*} = \sum_{nb} A_{nb,ij} u_{nb,j}^{k*} + S_u - \frac{\partial p_P^k}{\partial x_i} \Omega_P \quad (1.10)$$

$$A_{P,ij}(u_{P,j}^{k+1} - u_{P,j}^{k*}) = \sum_{nb} A_{nb,ij} (u_{nb,j}^{k+1} - u_{nb,j}^{k*}) - \frac{\partial p'_P}{\partial x_i} \Omega_P \quad (1.11)$$

If the first term in the RHS is approximated as,

$$\frac{\sum_{nb} A_{nb,ij} (u_{nb,j}^{k+1} - u_{nb,j}^{k*})}{\sum_{nb} A_{nb,ij}} = (u_{P,j}^{k+1} - u_{P,j}^{k*}) \quad (1.12)$$

based on the idea that the cell–centre velocity correction is a coefficient weighted average of the velocity corrections at the neighbouring cells, we obtain the SIMPLEC method. If this term is neglected then we obtain the SIMPLE method.

$$(A_{P,ij} - \sum_{nb} A_{nb,ij})(u_{P,j}^{k+1} - u_{P,j}^{k*}) = -\frac{\partial p'_P}{\partial x_i} \Omega_P \quad (1.13)$$

Denoting  $A'_{P,ij} = (A_{P,ij} - \sum_{nb} A_{nb,ij})$  and multiplying Eq. (1.13) by  $A'^{-1}_{P,ij}$  and taking the divergence, we obtain the pressure–correction equation as,

$$\frac{\partial u_{P,j}^{k+1}}{\partial x_j} - \frac{\partial u_{P,j}^{k*}}{\partial x_j} = -\frac{\partial}{\partial x_j} \left( A'^{-1}_{P,ij} \frac{\partial p'_P}{\partial x_i} \Omega_P \right) \quad (1.14)$$

### 1.3.1 Divergence-Free Flows

For incompressible single–phase flows, two–phase flows using interface tracking, or for homogeneous flow models, we set the expectation on the divergence of  $\mathbf{u}^{k+1}$  to be zero, thus obtaining the pressure correction equation as,

$$\frac{\partial u_{P,j}^{k*}}{\partial x_j} = \frac{\partial}{\partial x_j} \left( A'^{-1}_{P,ij} \frac{\partial p'_P}{\partial x_i} \Omega_P \right) \quad (1.15)$$

### 1.3.2 Discretisation

The integral form of Eq. (1.15) is solved. Integrating using the divergence theorem we get the following discretised equation,

$$\sum_f \tilde{u}_{f,j}^{k*} n_{f,j} = \sum_f n_{f,j} A_{f,ij}^{-1} \frac{\partial p'_P}{\partial x_i} \Omega_f \quad (1.16)$$

where, the velocity at the faces  $\tilde{u}$  is calculated with the Rhie–Chow method,  $n_{f,j}$  is the area vector of the cell face  $f$ ,  $A_{f,ij}^{-1}$  is the interpolated coefficient tensor at the face and  $\Omega_f$  is the volume of dual control volume centered at the face.

## 1.4 Axisymmetric Flows

### 1.4.1 Axisymmetry

For axisymmetric geometries, the continuity equation is given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial r}(\rho v) + \frac{\rho v}{r} = 0 \quad (1.17)$$

or

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{1}{r} \frac{\partial}{\partial r}(r \rho v) = 0 \quad (1.18)$$

where  $x$  is the axial coordinate,  $r$  is the radial coordinate,  $u$  is the axial velocity, and  $v$  is the radial velocity.

The axial and radial momentum conservation equations are given by

$$\begin{aligned} \frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho uu) + \frac{1}{r} \frac{\partial}{\partial r}(r \rho vu) &= -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left[ \mu \left( 2 \frac{\partial u}{\partial x} - \frac{2}{3} (\nabla \cdot \vec{v}) \right) \right] \\ &\quad + \frac{1}{r} \frac{\partial}{\partial r} \left[ r \mu \left( \frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right) \right] + \end{aligned} \quad (1.19)$$

(1.20)

and

$$\begin{aligned} \frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho uv) + \frac{1}{r} \frac{\partial}{\partial r}(r \rho vv) &= -\frac{\partial p}{\partial r} + \frac{\partial}{\partial x} \left[ \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial r} \right) \right] \\ &\quad + \frac{1}{r} \frac{\partial}{\partial r} \left[ r \mu \left( 2 \frac{\partial v}{\partial r} - \frac{2}{3} (\nabla \cdot \vec{v}) \right) \right] - 2\mu \frac{v}{r^2} + \frac{2\mu}{3r} (\nabla \cdot \vec{v}) \end{aligned} \quad (1.21)$$

where

$$\nabla \cdot \vec{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r} \quad (1.22)$$

or

$$\nabla \cdot \vec{v} = \frac{\partial u}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (rv) \quad (1.23)$$



## Chapter 2

# Compressible Flow Modelling

This section describes the compressible flow modelling capabilities of TransAT.

### 2.1 Single Phase Conservation Equations

Equations that are solved for compressible flows are the following ([Poinsot & Veynante, 2001](#)):

$$\frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x_j} (\rho u_j) = 0 \quad (2.1)$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i \quad (2.2)$$

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p u_j \frac{\partial T}{\partial x_j} = \frac{\partial}{\partial x_j} \left( \lambda \frac{\partial T}{\partial x_j} \right) + \left( \frac{\partial p}{\partial t} + u_j \frac{\partial p}{\partial x_j} \right) + \Phi + \dot{Q}''' \quad (2.3)$$

where

$$\sigma_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad (2.4)$$

and  $\Phi$  is the viscous dissipation of kinetic energy. Thermodynamical properties  $\rho$ ,  $p$  and  $T$  are linked using a specified equation of state

$$\rho = \rho(p, T) \quad (2.5)$$

### 2.2 N-Phase Conservation Equations

Equations that are solved for compressible n-phase flows are given belows. In the current version, compressible flow is only available for the homogeneous model (no drift velocity), without phase change.

$$\frac{\partial}{\partial t} \rho^m + \frac{\partial}{\partial x_j} (\rho^m u_j^m) = 0 \quad (2.6)$$

$$\frac{\partial \rho_k \alpha_k}{\partial t} + \frac{\partial}{\partial x_j} (\rho_k \alpha_k u_j^m) = 0 \quad (2.7)$$

$$\frac{\partial}{\partial t} (\rho^m u_i^m) + \frac{\partial}{\partial x_j} (\rho^m u_i^m u_j^m) = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}^m}{\partial x_j} + \rho g_i \quad (2.8)$$

$$\rho^m C_p^m \left( \frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left( \lambda \frac{\partial T}{\partial x_j} \right) + \frac{Dp}{Dt} + \Phi + \dot{Q}''' \quad (2.9)$$

$$\sum_{k=1}^N \alpha_k = 1 \quad (2.10)$$

with the subscript  $k$  denoting the phase index and the superscript  $m$  denoting a mixture variable, and  $\alpha$  being the volume fraction. Mixture variables are defined as follows

$$\text{Mixture density} \quad \rho^m = \sum_{k=1}^N \alpha_k \rho_k \quad (2.11)$$

$$\text{Mixture velocity} \quad u^m = \frac{\sum_{k=1}^N \alpha_k \rho_k u_k}{\rho^m} \quad (2.12)$$

$$\text{Mixture heat capacity} \quad \rho^m C_p^m = \sum_{k=1}^N \alpha_k \rho_k C_{p,k} \quad (2.13)$$

$$\text{Mixture viscosity} \quad \mu^m = \sum_{k=1}^N \alpha_k \mu_k \quad (2.14)$$

$$\text{Mixture thermal conductivity} \quad \lambda^m = \sum_{k=1}^N \alpha_k \lambda_k \quad (2.15)$$

## 2.3 Level-Set Conservation Equations

The equations used to model compressible flows with Level-Set method are the following. In the current version, phase change is not available for compressible flows.

$$\frac{\partial}{\partial t}\phi + u_j^I \frac{\partial}{\partial x_j}\phi = 0 \quad (2.16)$$

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \quad (2.17)$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i + \gamma \kappa \delta^I(\phi) n_i \quad (2.18)$$

$$\rho C_p \left( \frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left( \lambda \frac{\partial T}{\partial x_j} \right) + \frac{Dp}{Dt} + \Phi + \dot{Q}''' \quad (2.19)$$

where  $\gamma$  is the surface tension coefficient,  $\kappa$  is the interface curvature,  $n$  is the normal vector to the interface,  $\delta^I$  is a smoothed Dirac delta function centered at the interface,  $T$  is the temperature,  $C_p$  is the heat capacity of the fluids,  $\lambda$  is the heat conductivity, and  $\dot{Q}$  is the volumetric heat source. Material properties are updated using  $\chi$ :

$$\rho(\chi, t) = \sum \chi^k \rho^k \quad (2.20)$$

$$\mu(\chi, t) = \sum \chi^k \mu^k \quad (2.21)$$

$$C_p(\chi, t) = \frac{\sum \chi^k \rho^k C_p^k}{\sum \chi^k \rho^k} \quad (2.22)$$

$$\lambda(\chi, t) = \sum \chi^k \lambda^k \quad (2.23)$$

## 2.4 Equations of state

### 2.4.1 Ideal Gas equation of state

The simplest way to link density, pressure and temperature is to use the ideal gas equation of state. This is applicable only for gases. It reads

$$\rho = \frac{\mathcal{M}P}{\mathcal{R}T} \quad (2.24)$$

where  $\mathcal{M}$  is the molecular weight and  $\mathcal{R}$  the universal gas constant. The different parameters must respect the equality

$$\gamma = \frac{C_p}{C_p - \frac{\mathcal{R}}{\mathcal{M}}} \quad (2.25)$$

where  $\mathcal{R} = 8.314472 \text{ J/K mol}$  is the universal gas constant. The speed of sound is then written

$$c = \sqrt{\frac{\gamma \mathcal{R} T}{\mathcal{M}}} \quad (2.26)$$

### 2.4.2 Stiffened Gas Equation of State

The Stiffened Gas Equation is an equation of state used for water under very high pressures. This equation can be applied for certain other liquids, but specific heat capacity is likely to be miscalculated. It reads

$$\rho = \frac{\gamma (p + p_\infty)}{C_p (\gamma - 1) T} \quad (2.27)$$

The fluid will behave as an ideal gas at pressure  $p = p_\infty$ .

The different parameters must respect the equality

$$\gamma = \frac{C_p}{C_p - \frac{\mathcal{R}}{\mathcal{M}}} \quad (2.28)$$

where  $\mathcal{R} = 8.314472 \text{ J/K mol}$  is the universal gas constant. The speed of sound can be written

$$c = \sqrt{C_p (\gamma - 1) T} \quad (2.29)$$

### 2.4.3 Tait Equation of State for Water

The Tait equation is an equation of state derived for water and sea-water [Cooke & Chen \(1992\)](#). Density of water as a function of pressure can then be expressed as

$$\rho = \rho_0 \left( \frac{p + B}{p_0 + B} \right)^{\frac{1}{N}} \quad (2.30)$$

where  $\rho_0 = 1000.0 \text{ kg/m}^3$ ,  $p_0 = 0.1 \text{ MPa}$ ,  $B = 295.9 \text{ MPa}$ , and  $N = 7.415$ . For this equation of state, speed of sound can be written

$$c = \sqrt{\frac{N}{\rho_0} \frac{(p_0 + B)^{\frac{1}{N}}}{(p + B)^{\frac{1-N}{N}}}} \quad (2.31)$$

#### 2.4.4 Tait Equation of State for Ammonia

The Tait equation of state is also available in [TransAT](#) for ammonia. The density is expressed as:

$$\rho = \frac{\rho_0}{1 - C \cdot \log \left( \frac{p + B}{p_0 + B_0} \right)} \quad (2.32)$$

where:

$$\rho_0 = 683.44 \text{ kg/m}^3,$$

$$p_0 = 257.79 \text{ kPa},$$

$$C(T) = 1.45 (0.1180 - 10^{-13.0628+4.4657\log_{10}(T)}),$$

$$B(T) = \max(1, 8241.8 - 38.241T + 0.043507T^2) \text{ atm},$$

$$B_0 = B(T_0),$$

$$T_0 = 238.7 \text{ K}$$

#### 2.4.5 Incompressible

[TransAT](#) also offers the possibility to use an incompressible equation of state. In that case the density and other properties do not vary with pressure and the constant values specified by the user will be used.

#### 2.4.6 Peng-Robinson Equation of State for Water (Liquid)

Another possibility when working with water is to use the Peng-Robinson equation of state. In general the Peng-Robinson equation of state writes:

$$p = \frac{RT}{V_m - b} - \frac{a\alpha}{V_m^2 + 2bV_m - b^2} \quad (2.33)$$

Started from this equation a cubic equation for the molar specific volume is obtained and solved. The density is then obtained using following relation:

$$\rho = \frac{M_w}{V_m - V_t} \quad (2.34)$$

with  $V_t = 2.58775 \cdot 10^{-6} \text{ m}^3/\text{mol}$  the volume translation and  $M_w = 18 \text{ g/mol}$  the molecular weight

#### 2.4.7 Peng-Robinson Equation of State for Water (Vapour)

The Peng-Robinson equation of state is also available for water vapour. The implementation is similar to the liquid state implementation.

### 2.4.8 Sutherland's law

The viscosity  $\mu$  and the thermal conductivity  $\lambda$  can be evaluated using Sutherland's law. They are then defined by the equations:

$$\mu = \mu_0 \left( \frac{T}{T_{\mu,0}} \right)^{3/2} \frac{T_{\mu,0} + S_\mu}{T + S_\mu} \quad (2.35)$$

$$\lambda = \lambda_0 \left( \frac{T}{T_{\lambda,0}} \right)^{3/2} \frac{T_{\lambda,0} + S_\lambda}{T + S_\lambda} \quad (2.36)$$

where  $T$  is the local temperature. These expressions can also be used for incompressible flows.

## 2.5 Pressure Solver Formulation

### 2.5.1 Mass conservation

The phasic continuity equation is written below in the volume fraction and mass fraction forms as,

$$\frac{\partial}{\partial t} (\alpha_k \rho_k) + \frac{\partial}{\partial x_j} (\alpha_k \rho_k u_j^m) = 0 \quad (2.37)$$

$$\frac{\partial}{\partial t} (\rho^m Y_k) + \frac{\partial}{\partial x_j} (\rho^m Y_k u_j^m) = 0 \quad (2.38)$$

where superscript  $m$  stands for mixture. It can be verified for both Eqs. (2.37) and (2.38) that summation over the phase index  $k$  gives the mixture continuity equation,

$$\frac{\partial \rho^m}{\partial t} + \frac{\partial}{\partial x_j} (\rho^m u_j^m) = 0 \quad (2.39)$$

Equation (2.38) can also be written in a material derivative form as follows, which will be useful in the derivation of the pressure equation,

$$\rho^m \left[ \frac{\partial Y_k}{\partial t} + u_j^m \frac{\partial Y_k}{\partial x_j} \right] + Y_k \left[ \frac{\partial \rho^m}{\partial t} + \frac{\partial \rho^m u^m}{\partial x_j} \right] = 0 \quad (2.40)$$

and plugging Eq. 2.39 gives

$$\rho^m \left[ \frac{\partial Y_k}{\partial t} + u_j^m \frac{\partial Y_k}{\partial x_j} \right] = 0 \quad (2.41)$$

We introduce now the notation of material derivative advected by the mixture velocity as,

$$\frac{D^m \phi}{Dt} = \frac{\partial \phi}{\partial t} + u_j^m \frac{\partial \phi}{\partial x_j} \quad (2.42)$$

where  $\phi$  is any advected quantity and the superscript  $m$  denotes that the advection velocity is the mixture velocity. Using this notation we can write the material derivative of  $Y_k$  as,

$$\rho^m \frac{D^m Y_k}{Dt} = 0 \quad (2.43)$$

### 2.5.2 Pressure equation

An equation for the evolution of the pressure of a multiphase system is derived here. It is based on a single pressure, single temperature assumption where the pressure  $p$  and temperature  $T$  are the mixture pressure and temperature, respectively. The starting point to derive a pressure equation for compressible multiphase flow is the mixture continuity equation (Eq. (2.39)), but written in a non-conservative form as,

$$\frac{\partial \rho^m}{\partial t} + u_j^m \frac{\partial \rho^m}{\partial x_j} + \rho^m \frac{\partial u_j^m}{\partial x_j} = 0 \quad (2.44)$$

This can be rewritten in the material derivative notation as,

$$\frac{D^m \rho^m}{Dt} = -\rho^m \frac{\partial u_j^m}{\partial x_j} \quad (2.45)$$

Taking  $\rho^m = \rho^m(p, T, Y_k)$  we can expand the above equation as a linearization based on the independent unknowns as,

$$\left. \frac{\partial \rho^m}{\partial p} \right|_{T, Y_k} \frac{D^m p}{Dt} + \left. \frac{\partial \rho^m}{\partial T} \right|_{p, Y_k} \frac{D^m T}{Dt} + \sum_k \left. \frac{\partial \rho^m}{\partial Y_k} \right|_{p, T} \frac{D^m Y_k}{Dt} = -\rho^m \frac{\partial u_j^m}{\partial x_j} \quad (2.46)$$

To evaluate the constrained derivatives, we start with the definition of density with respect to phasic densities and mass fractions.

$$\frac{1}{\rho^m} = \sum_k \frac{Y_k}{\rho_k} \quad (2.47)$$

where  $\rho_k = \rho_k(p, T)$ . Taking the derivative of  $\rho^m$  with respect to  $p$  at constant  $(T, Y_k)$  gives,

$$\left. \frac{\partial \rho^m}{\partial p} \right|_{T, Y_k} = (\rho^m)^2 \sum_k \frac{Y_k}{\rho_k^2} \left. \frac{\partial \rho_k}{\partial p} \right|_T = \rho^m \sum_k \frac{\alpha_k}{\rho_k} \left. \frac{\partial \rho_k}{\partial p} \right|_T \quad (2.48)$$

Similarly the derivative at constant  $(p, Y_k)$  can be written as,

$$\left. \frac{\partial \rho^m}{\partial T} \right|_{p, Y_k} = (\rho^m)^2 \sum_k \frac{Y_k}{\rho_k^2} \left. \frac{\partial \rho_k}{\partial T} \right|_p = \rho^m \sum_k \frac{\alpha_k}{\rho_k} \left. \frac{\partial \rho_k}{\partial T} \right|_p \quad (2.49)$$

The derivative of density with respect to  $Y_k$  at constant  $(p, T)$  is obtained as,

$$\frac{\partial \rho^m}{\partial Y_k} \Big|_{p,T} = -\frac{(\rho^m)^2}{\rho_k} \quad (2.50)$$

Additionally we write the mixture temperature equation 2.9 using material derivatives

$$\rho^m C_p^m \frac{D^m T}{Dt} = \frac{D^m p}{Dt} + \frac{\partial}{\partial x_j} \left( \lambda \frac{\partial T}{\partial x_j} \right) + \Phi + Q''' \quad (2.51)$$

where  $\Phi$  is the viscous dissipation of kinetic energy. The material derivative of  $T$  advected by the mixture velocity can then be written as,

$$\frac{D^m T}{Dt} = \frac{1}{\rho^m C_p^m} \left( \frac{D^m p}{Dt} + \Xi \right) \quad (2.52)$$

where  $\Xi$  represents the entropic part of heat transfer (the last three terms in the RHS of Eq. (2.51)).

Plugging Eqs. (2.48)-(2.50), Eq. (2.52) and Eq. (2.43) into Eq. (2.46), we obtain

$$\left[ \sum_k \rho^m \frac{\alpha_k}{\rho_k} \left( \frac{\partial \rho_k}{\partial p} \Big|_T + \frac{1}{\rho^m C_p^m} \frac{\partial \rho_k}{\partial T} \Big|_p \right) \right] \frac{D^m p}{Dt} + \left[ \sum_k \rho^m \frac{\alpha_k}{\rho_k} \frac{\partial \rho_k}{\partial T} \Big|_p \right] \frac{\Xi}{\rho^m C_p^m} = -\rho^m \frac{\partial u_j^m}{\partial x_j} \quad (2.53)$$

where the identity  $\rho^m Y_k = \rho_k \alpha_k$  is used. This is the final form of the pressure equation for n-phase compressible flow.

### 2.5.3 Limit of incompressible multiphase flow

In the limit of incompressible flow, the LHS of Eq. (2.53) can be set to zero, giving,

$$\frac{\partial u_j^m}{\partial x_j} = 0 \quad (2.54)$$

### 2.5.4 Limit of compressible single phase flow

For the limit of compressible single-phase flow, we set  $\rho_k = \rho$ ,  $\alpha_k = 1$  in Eq. (2.53) to obtain,

$$\left( \frac{\partial \rho}{\partial p} \Big|_T + \frac{1}{\rho C_p} \frac{\partial \rho}{\partial T} \Big|_p \right) \frac{D^m p}{Dt} + \frac{\partial \rho}{\partial T} \Big|_p \frac{\Xi}{\rho C_p} = -\rho \frac{\partial u_j^m}{\partial x_j} \quad (2.55)$$

Note that for an ideal gas  $p = \rho RT$ , it can be shown that,

$$\frac{\partial \rho}{\partial p} \Big|_T + \frac{1}{\rho C_p} \frac{\partial \rho}{\partial T} \Big|_p = \frac{\partial \rho}{\partial p} \Big|_s \quad (2.56)$$

where  $s$  denotes entropy. Substituting the partial derivatives, we get

$$\left. \frac{\partial \rho}{\partial p} \right|_s = \frac{1}{RT} + \frac{1}{\rho C_p} \left( \frac{-p}{RT^2} \right) = \frac{1}{RT} \left( 1 - \frac{R}{C_p} \right) = \frac{1}{\gamma RT} \quad (2.57)$$

where we have used the relationships  $C_p - C_v = R$  and  $C_p/C_v = \gamma$ .

### 2.5.5 Pressure Solver Formulation for reactive flows

A conservative formulation is used for the pressure solver in the case of reactive, compressible flows. We start with Eq. (1.13) and multiply it through by  $\rho_P^{k+1}$  to obtain:

$$\rho_P^{k+1} (u_{P,j}^{k+1} - u_{P,j}^{k*}) = -\rho_P^{k+1} A'_{P,ij} \frac{\partial p'_P}{\partial x_i} \Omega_P \quad (2.58)$$

Taking the divergence of the above equation we obtain:

$$\frac{\partial}{\partial x_j} (\rho_P^{k+1} u_{P,j}^{k+1}) - \frac{\partial}{\partial x_j} (\rho_P^{k+1} u_{P,j}^{k*}) = -\frac{\partial}{\partial x_j} \left( \rho_P^{k+1} A'_{P,ij} \frac{\partial p'_P}{\partial x_i} \Omega_P \right) \quad (2.59)$$

Splitting  $\rho_P^{k+1} = \rho_P^{k*} + \rho'_P$  and setting,

$$\frac{\partial}{\partial x_j} (\rho_P^{k+1} u_{P,j}^{k+1}) = -\frac{\partial \rho_P^{k+1}}{\partial t} \quad (2.60)$$

we obtain,

$$-\frac{\partial \rho_P^{k*}}{\partial t} - \frac{\partial \rho'_P}{\partial t} - \frac{\partial}{\partial x_j} (\rho_P^{k*} u_{P,j}^{k*}) - \frac{\partial}{\partial x_j} (\rho'_P u_{P,j}^{k*}) = -\frac{\partial}{\partial x_j} \left( (\rho_P^{k*} + \rho'_P) A'_{P,ij} \frac{\partial p'_P}{\partial x_i} \Omega_P \right) \quad (2.61)$$

Neglecting the  $\rho' p'$  non-linear terms in the RHS and substituting  $\rho'_P = p'_P/c_P^2$ , where  $c_P$  is the sound speed, we obtain,

$$-\frac{\partial \rho_P^{k*}}{\partial t} - \frac{\partial}{\partial x_j} (\rho_P^{k*} u_{P,j}^{k*}) = \frac{\partial}{\partial t} \left( \frac{p'_P}{c_P^2} \right) + \frac{\partial}{\partial x_j} \left( \frac{u_{P,j}^{k*} p'_P}{c_P^2} \right) - \frac{\partial}{\partial x_j} \left( \rho_P^{k*} A'_{P,ij} \frac{\partial p'_P}{\partial x_i} \Omega_P \right) \quad (2.62)$$

## 2.6 Numerical Schemes

The solver for compressible flows is based on the work of Karki & Patankar (1989) and Michelassi *et al.* (1994). The starting point is the incompressible solver presented in Chapter Numerical Methods, developed by Zhu (1992). The mass conservation equation for steady-state conditions reads

$$\sum_n \left( \rho_n^* + \rho'_n \right) \left( c_n^* + c'_n \right) \cdot n_n = 0 \quad (2.63)$$

where  $\rho$  is the density and  $c_n \cdot n_n$  is the volume flux through the face  $n$ . Superscript \* denotes guessed properties while ' denotes corrections to ensure continuity. This equation can be rewritten as follows

$$\underbrace{\sum_n \rho_n^* c_n^* \cdot n_n}_T_0 + \underbrace{\sum_n \rho_n^* c'_n \cdot n_n}_T_1 + \underbrace{\sum_n \rho'_n c_n^* \cdot n_n}_T_2 + \underbrace{\sum_n \rho'_n c'_n \cdot n_n}_T_3 = 0 \quad (2.64)$$

Where  $T_0$  and  $T_1$  are the incompressible contribution,  $T_2$  is the compressible contribution and  $T_3$  is neglected.  $\rho'$  is linked to  $p'$  using the equation of state relation

$$\rho' = \left( \frac{\partial \rho}{\partial p} \right)_\Phi p' \quad (2.65)$$

where  $\Phi$  is the temperature or entropy, whether we use the hypothesis of isothermal or entropic transformation.

Equation 3.13 is then rewritten for compressible flows

$$A_p^C p'_p - \sum_n A_n^C p'_n = \sum_n \rho_n^* c_n^* \cdot n_n \quad (2.66)$$

where

$$A_p^C = \sum_n A_n^C - S_p^C \quad (2.67)$$

$$A_n^C = A_n^I + A_n^{corr} \quad (2.68)$$

where  $A_n^I$  are the incompressible coefficients given in section 3.2, and  $A_n^{corr}$  and  $S_p^C$  are correction terms due to compressibility effects. They depend on the way of evaluating  $\rho'$  at the cell face.

### 2.6.1 Interpolation schemes for compressible density

In this section, we note  $P$  the current cell,  $W$  and  $E$  the neighbouring cells and  $w$  and  $e$  the west and east faces of cell  $p$ .

#### Upwind scheme

For high velocities, an upwind scheme can be used. We can thus write

$$\rho'_w = f_w^+ \rho'_W + (1 - f_w^+) \rho'_P \quad (2.69)$$

$$\rho'_e = f_e^+ \rho'_P + (1 - f_e^+) \rho'_E \quad (2.70)$$

with

$$f_n^+ = \frac{1 + sgn(c_n \cdot n_n)}{2} \quad (2.71)$$

$$sgn(\phi) = \begin{cases} 1 & if \phi > 0 \\ -1 & if \phi < 0 \end{cases} \quad (2.72)$$

With this scheme, we obtain

$$A_w^{corr} = \max(c_w \cdot n_w, 0) \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_W \quad (2.73)$$

$$A_e^{corr} = -\min(c_e \cdot n_e, 0) \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_E \quad (2.74)$$

$$S_P^C = \sum_n A_n^{corr} + \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_P \left( \sum_{n=W,S,B} \min(c_n \cdot n_n, 0) - \sum_{n=E,N,T} \max(c_n \cdot n_n, 0) \right) \quad (2.75)$$

### Linear scheme

This scheme can be used for compressible flows at low velocities. We have

$$\rho'_w = f_w \rho'_W + (1 - f_w) \rho'_P \quad (2.76)$$

$$\rho'_e = f_e \rho'_P + (1 - f_e) \rho'_E \quad (2.77)$$

with  $f_w$  and  $f_e$  the linear interpolation factors for the  $W - P$  and  $P - E$  intervals, respectively. With this scheme, we obtain

$$A_w^{corr} = f_w (c_w \cdot n_w, ) \left( f_w \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_W + (1 - f_w) \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_P \right) \quad (2.78)$$

$$A_e^{corr} = -(1 - f_e) (c_e \cdot n_e, ) \left( f_e \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_P + (1 - f_e) \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_E \right) \quad (2.79)$$

$$S_P^C = \sum_n A_n^{corr} + (1 - f_w) (c_w \cdot n_w, ) \left( f_w \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_W + (1 - f_w) \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_P \right) \quad (2.80)$$

$$-(f_e) (c_e \cdot n_e, ) \left( f_e \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_P + (1 - f_e) \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_E \right) \quad (2.81)$$

### 2.6.2 Unsteady simulation

When performing an unsteady simulation, an extra-term must be added to the compressible contribution  $S_P^C$ . The latter becomes:

$$S_{P,unsteady}^C = S_{P,steady}^C - \frac{\Omega}{\Delta t} \left[ \left( \frac{\partial \rho}{\partial p} \right)_{\Phi} \right]_P \quad (2.82)$$

where  $\Omega$  is the cell volume and  $\Delta t$  the time step.

# Chapter 3

## Numerical Methods

### 3.1 Governing Equations for Steady Flows

The partial differential equations governing steady incompressible flows in nonorthogonal coordinates may be written in the following general form (Patankar, 1980):

$$\frac{\partial}{\partial x_i} (C_i \phi - D_{i\phi}) = S_\phi, \quad i = 1, 2, 3 \quad (3.1)$$

where the coefficients  $C_i$  and  $D_{i\phi}$  are related to convection and diffusion, respectively and  $S_\phi$  the source terms for any primary unknown  $\phi$ . Eq. 3.1 uses the velocity components  $V_i$ , which are along the coordinates  $y_i$  instead of along the grid-aligned directions  $x_i$ .

Eq. 3.1 has two important features which are worthy of mention: (i) it is in conservation form which implies all terms arising from the divergence operator are under differential operators, and (ii) it does not contain second-derivatives of coordinates (curvature terms) which are very sensitive to grid smoothness.

### 3.2 Discretisation of Governing Equations

#### 3.2.1 Flux balance equations

Integrating Eq. 3.1 over a typical control volume centered at P (Figure 3.1) leads to a flux balance equation

$$I_e - I_w + I_n - I_s + I_t - I_b = \int_{\Delta V} S_\phi dV \quad (3.2)$$

where  $I_f$  represents the total flux of  $\phi$  across the cell-face  $f$  ( $= e, w, n, s, t$  or  $b$ ). Each of the surface fluxes  $I_f$  contains a convective contribution  $I_f^C$  and a diffusive contribution  $I_f^D$ , that is

$$I_f = I_f^C + I_f^D \quad (3.3)$$

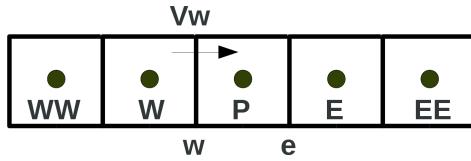


Figure 3.1: Nodes required by convection schemes in  $x_1$ -direction

Eq. 3.2 involves no approximation and represents a finite-volume analogue of the differential equation (3.1).

### 3.2.2 Approximation of convection terms

The convective contribution in Eq. 3.3 can be written as:

$$I_f^C = C_f \phi_f \quad (3.4)$$

where  $C_f$  is the mass flux across the cell-face  $f$ , and can be calculated, for w-, s- and b-faces, as:

$$\begin{aligned} C_w &= (\rho V_j)_w \\ C_s &= (\rho V_j)_s \\ C_b &= (\rho V_j)_b \end{aligned} \quad (3.5)$$

The determination of  $\phi_f$  is a key element for both accuracy and stability of numerical solutions. The more accurate schemes tend to be less stable, and vice versa. Four schemes are incorporated in the program, leaving a choice for the user. They are the standard hybrid differencing (Spalding, 1972) which is basically of first-order accuracy for convection-dominant flows, third-order unbounded QUICK (Leonard, 1979), and second-order bounded HLPA (Zhu, 1991). Consider the w-face of the control volume shown in Figure 3.1 and assume, without loss of generality, that  $V_w \geq 0$  ( $V_w$  is the normal velocity at the w-face).

The **HLPA** scheme can be written as

$$\phi_w = \phi_W + \gamma_w (\phi_P - \phi_W) \frac{\phi_W - \phi_{WW}}{\phi_P - \phi_{WW}} \quad (3.6)$$

where

$$\gamma_w = \begin{cases} 1 & \text{if } |\hat{\phi}_W - 0.5| < 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

All the schemes but Hybrid are implemented in the program in a deferred way proposed by Khosla & Rubin (1974)

$$\phi_f^{l+1} = \phi_f^{u,l+1} + \lambda(\phi_f^{h,l} - \phi_f^{u,l}) \quad (3.8)$$

where  $u$  and  $h$  indicate the upwind and higher-order schemes, and  $l$  represents the iteration level; the parameter  $\lambda$  blends the two schemes with limiting values  $\lambda = 0$  for the upwind and  $\lambda = 1$  for the higher-order scheme. The differenced correction leads to an always diagonally dominant coefficient matrix, thus lending the necessary stability to the numerical process while restoring higher-order solution at convergence.

All the schemes but Hybrid require two upstream nodes for each cell-face, which will involve a value outside the solution domain for a near-boundary control volume. Therefore in the program the Hybrid scheme is used for all the control volumes adjacent to boundaries.

### 3.2.3 Approximation of diffusion terms

The diffusive flux  $I_f^D$  in Eq. 3.3 can be divided into two parts

$$I_f^D = I_f^{DN} + I_f^{DC} \quad (3.9)$$

The first part  $I_f^{DN}$  contains only one term which has the first derivative of  $\phi$  in the direction "normal" to the cell-face  $f$ . It can be written, for the w-face for example, as:

$$I_w^{DN} = D_w(\phi_P - \phi_W); \text{ with } D_w = [\Gamma_\phi / \Delta V]_w \quad (3.10)$$

The second part  $I_f^{DC}$  contains all the other terms. Only the normal derivative diffusion flux,  $I_f^{DN}$ , is coupled with the convective flux,  $I_f^C$ , to calculate the main coefficients of the difference equations, while the cross-derivative diffusion flux,  $I_f^{DC}$ , is treated explicitly as a pseudo-source term to avoid the possibility of producing negative coefficients in an implicit treatment.

### 3.2.4 Approximation of source terms

The source terms  $S_\phi$  are linearised as

$$S_\phi = S_\phi^U + S_\phi^P \phi_P \quad (3.11)$$

The coefficient  $S_\phi^P$  is defined so that it is always not more than zero for all the conservation equations, and  $S_\phi^U$  always assumes non-negative values for both k- and  $\epsilon$ -equations. This enhances the stability of the numerical process and prevents the calculated values of k and  $\epsilon$  from becoming negative in low turbulence regions. The volume integral of the source term can therefore be approximated as

$$\int_{\Delta V} S_\phi dV \simeq S_\phi^U \Delta V + S_\phi^P \Delta V \phi_P \quad (3.12)$$

### 3.2.5 Final form of discretised equations

After replacing all the terms in Eq. 3.2 by their discretised analogies, the following difference equation results in:

$$A_P \phi_P = \sum_{nb} A_{nb} \phi_{nb} + S_U, \quad nb = W, E, S, N, B, T \quad (3.13)$$

where

$$A_P = \sum_{nb} A_{nb} - S_P \quad (3.14)$$

The main coefficients  $A_{nb}$  that relate the principal unknown  $\phi_P$  to its neighbours  $\phi_{nb}$  contain the combined contribution from convection and normal diffusion. The physical source term (Eq. 3.12) and the cross-derivative diffusion flux  $I^{DC}$  (Eq. 3.9) are included in the coefficients  $S_U$  and  $S_P$ .

In order to stabilise the iterative solution process, it is often necessary to under-relax the current solution by

$$\phi_P := \phi_P^o + \alpha_\phi (\phi_P - \phi_P^o) \quad (3.15)$$

where  $\alpha_\phi (\in [0, 1])$  is an under-relaxation factor and the superscript "o" refers to the old value at the previous iterative level. Introducing Eq. 3.15 into Eq. 3.13 leads to an under-relaxed equation which is of the same form as Eq. 3.13 except that the coefficients  $S_U$  and  $A_P$  are replaced by

$$S_U := S_U + \frac{1 - \alpha_\phi}{\alpha_\phi} A_P \phi_P^o \quad (3.16)$$

$$A_P := A_P / \alpha_\phi \quad (3.17)$$

## 3.3 Boundary Conditions

The most frequently encountered boundary conditions, i.e., inflow, outflow, symmetry and rigid wall conditions, are implemented in the program. The pressures at all the boundary nodes are evaluated by linear extrapolation from values at interior nodes. The program also provides a provision allowing the user to specify boundary conditions other than these four types.

### 3.3.1 Inflow planes

At the inflow planes, the value of each variable  $\phi$  is prescribed. Furthermore, the program defines the following 'n' factors to normalise the residual:

$$F_1 = \sum_{ii} \dot{m}_{ii}, \quad F_n = \sum_{ii} \dot{m}_{ii} \phi_{n,ii}, \quad n = 2, 3, \dots, n \quad (3.18)$$

with

$$\phi = \phi_{in} \quad (3.19)$$

where the summation index ii runs over all the inlet computational nodes,  $\dot{m}$  is the mass flux and the subscript "in" refers to the inlet values.  $F_n$  ( $n=1, 2, \dots$  or n) is set to 1 if the corresponding value calculated by Eq. 3.18 is 0.

### 3.3.2 Outflow planes

At the outflow planes, the stream-wise gradients of all variables are set to zero, implying a fully developed flow condition. In addition, the velocities  $V_j$  ( $j = 1, 2, 3$ ) are corrected in the following way to ensure the overall mass conservation at the outflow planes:

$$V_{j,ii} := V_{j,ii} F_a \quad (3.20)$$

$$F_a = F_1 / \sum_{ii} \dot{m}_{out,ii} \quad (3.21)$$

where ii runs over all the outlet computational nodes and  $\dot{m}_{out}$  is the outflow mass flux.

For pressure outflows with a pressure loss coefficient  $\zeta$ , the pressure at the boundary is defined as

$$p_{BC} = p_\infty + \frac{1}{2} \zeta \rho \bar{u}^2 \quad (3.22)$$

where  $p_\infty$  is the set pressure value,  $\zeta$  is the value set for the pressure loss coefficient,  $\rho$  is the fluid density and  $\bar{u}$  is the bulk outflow velocity, i.e. the average outflow velocity accounting for all the cells that are part of the same outflow.

### 3.3.3 Symmetry planes

At the symmetry planes, the normal velocity component and the normal gradients of other variables are set to zero. The convective and normal diffusive fluxes  $I_f^C$  (Eq. 3.4) and  $I_f^{DN}$  (Eq. 3.9) are also explicitly set to zero, where "f" refers to the boundary surface.

### 3.3.4 Rigid wall

At the rigid walls, the no-slip condition is used for laminar flows, i.e., the velocity components of the flow are set to those of the wall. The standard wall-function approach (Launder & Spalding, 1974) is used for turbulent flows. In this, the resultant wall shear stress  $\vec{\tau}_w$  is related to the flow velocity vector  $\vec{V}$  by

$$\vec{\tau}_w = -\lambda_w \vec{V}_P \quad (3.23)$$

where

$$\lambda_w = \begin{cases} \mu/y_P & \text{if } y_P^+ < 11.6 \\ \rho C_\mu^{1/4} k_P^{1/2} \kappa / \ln(Ey_P^+) & \text{otherwise} \end{cases} \quad (3.24)$$

$$y_P^+ = \rho C_\mu^{1/4} k_P^{1/2} y_P / \mu, \quad \kappa = 0.41, \quad E = 9.793 \quad (3.25)$$

the subscript  $P$  refers to the first control volume centre from the wall, and  $y_P$  is the normal distance from the wall. Further, the diffusive flux of  $k$  is set to zero at the wall, and the near-wall values of the production rate  $G$  and the dissipation rate  $\epsilon$  are determined from:

$$G_P = \frac{\tau_w^2}{\kappa \mu y_P^+} \quad (3.26)$$

$$\epsilon_P = \frac{C_\mu^{3/4} k_P^{3/2}}{\kappa y_P} \quad (3.27)$$

### 3.3.5 Total pressure boundary condition

For inflow conditions one can define the total pressure  $P_0$  or total temperature  $T_0$ . The mass flow rate is not known a priori. The static pressure at the inlet  $p_i$  is extrapolated from inside as long as subsonic conditions prevail at the inlet. In case of a supersonic flow,  $p_i$  must be defined by the user.

#### Ideal gas

For an ideal gas, isentropic expansion ( $pv^\gamma = \text{constant}$ ) from  $P_0$  to  $p_i$  is assumed and knowing this relationship, the Mach number can be solved for.

$$\frac{P_0}{p_i} = \left(1 + \frac{(\gamma - 1)}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} \quad (3.28)$$

Velocity can then be obtained by substituting  $\rho = \rho_0(p/p_0)^{1/\gamma}$

$$\frac{u^2}{2} = \left(\frac{\gamma}{\gamma-1}\right) \left(\frac{1}{\rho}\right) \left(p_0 \left(\frac{p}{p_0}\right)^{\frac{1}{\gamma}} - p\right) \quad (3.29)$$

#### Incompressible or weakly compressible fluids

For a liquid phase, either with compressible and incompressible formulation, the density is a very weak function of pressure, or in other words, density remains almost constant even for large variations in pressure. The relationship between  $P_0$  and  $p_i$  is given by the standard form of the Bernoulli equation,

$$\frac{P_0}{\rho_0} = \frac{p_i}{\rho_i} + \frac{1}{2} u^2 \quad (3.30)$$

### Mixture of compressible fluids

The  $P_0$  to  $p_i$  relationships for the two equations of state depends on the constant entropy expansion process, however they are both based on the general form of Bernoulli's equation.

The general form of the Bernoulli equation can be derived as follows from the conservation of mass and conservation of total energy equations for steady one-dimensional compressible flow where entropy producing terms such as heat conduction, heat sources, viscous dissipation etc. have been set to zero.

$$\frac{d}{dx}(\rho u) = 0 \quad (3.31)$$

and

$$\frac{d}{dx} \left( (\rho e + p + \frac{1}{2} \rho u^2) u \right) = \frac{d}{dx} \left( (e + \frac{p}{\rho} + \frac{1}{2} u^2) \rho u \right) = 0 \quad (3.32)$$

Using the product rule and using the conservation of mass equation, we obtain,

$$\frac{d}{dx} \left( e + \frac{p}{\rho} + \frac{1}{2} u^2 \right) = 0 \quad (3.33)$$

Since we are interested in the process from state 1 to 2 we can write the above equation as,

$$de + d \left( \frac{p}{\rho} \right) + d \left( \frac{u^2}{2} \right) = 0 \quad (3.34)$$

**Temperature:** To derive the relationship between the temperatures at the two states, we rewrite the equation in terms of enthalpy  $h = e + p/\rho$  and integrate from state 1 to 2,

$$\begin{aligned} \int_1^2 dh + \int_1^2 d \left( \frac{u^2}{2} \right) &= 0 \\ h_2 - h_1 + \frac{u_2^2}{2} - \frac{u_1^2}{2} &= 0 \end{aligned} \quad (3.35)$$

For constant  $C_p$  and assuming state 2 to be stagnation condition ( $u_2 = 0$ ), we obtain

$$T_0 = T_1 + \frac{u_1^2}{2C_p} \quad (3.36)$$

If  $C_p$  is not constant we can write the enthalpy difference as,

$$h_2 - h_1 = \int_{T_0}^{T_2} C_p dT - \int_{T_0}^{T_1} C_p dT = \int_{T_1}^{T_2} C_p dT = \bar{C}_p [T_1 : T_2] (T_2 - T_1) \quad (3.37)$$

Using Eq. (3.37) in Eq. (3.35) and as setting as before state 2 to be the stagnation state ( $u_2 = 0$ ) we get,

$$T_0 = T_1 + \frac{u_1^2}{2\bar{C}_p} \quad (3.38)$$

**Pressure:** To derive the relationship between the pressures at the two states, one has to incorporate the relationship between pressure and density for an isentropic expansion. We start with the thermodynamic relationship,

$$TdS = de + pdv \quad (3.39)$$

For an isentropic process ( $dS = 0$ ), we obtain,

$$de = -pdv = -p d\left(\frac{1}{\rho}\right) = \frac{p}{\rho^2} d\rho \quad (3.40)$$

Using the above relationship and substituting into Eq. (3.34) and further simplifying we obtain,

$$\frac{dp}{\rho} + d\left(\frac{u^2}{2}\right) = 0 \quad (3.41)$$

In the above equation if  $\rho = \text{constant}$ , integrating from state 1 to 2 gives the standard Bernoulli equation (Eq. (3.30)) for incompressible flow and assuming  $p \propto \rho^\gamma$  gives Eq. (3.28).

For a homogeneous multiphase mixture one can use Eq. (3.41) to calculate the inflow velocity as follows. We write the mixture density as,

$$\frac{1}{\rho_m} = \sum_k \frac{Y_k}{\rho_k} \quad (3.42)$$

where  $k$  is the phase index,  $Y_k$  is the mass fraction of phase  $k$ , and  $\rho_k$  is the density of phase  $k$ . We assume that the mixture has a single pressure, temperature and velocity. The mass fractions of each phase is assumed to be specified. Integrating Eq. (3.41) between states  $s_1$  and  $s_2$  for a two phase mixture, we obtain,

$$\int_{s_1}^{s_2} \frac{Y_1}{\rho_1} dp + \int_{s_1}^{s_2} \frac{Y_2}{\rho_2} dp = \frac{u_{s_1}^2}{2} - \frac{u_{s_2}^2}{2} \quad (3.43)$$

Assuming the phase 1 to be an ideal gas and phase 2 to be a constant density liquid, and further noting that the mass fractions do not change during the expansion, we obtain,

$$\frac{Y_2}{\rho_2}(p_{s_2} - p_{s_1}) + Y_1 \left( \frac{\gamma}{\gamma - 1} \right) \left( \frac{p_{s_2}}{\rho_{1,s_2}} - \frac{p_{s_1}}{\rho_{1,s_1}} \right) = \frac{u_{s_1}^2}{2} - \frac{u_{s_2}^2}{2} \quad (3.44)$$

Setting state 2 to be stagnation condition ( $u_{s_2} = 0$ ) and state 1 the inflow conditions ( $u_1 = u_{inflow}$ ), we obtain,

$$\frac{u_{inflow}^2}{2} = \frac{Y_2}{\rho_2}(p_{s_2} - p_{s_1}) + Y_1 \left( \frac{\gamma}{\gamma - 1} \right) \left( \frac{p_{s_2}}{\rho_{1,s_2}} - \frac{p_{s_1}}{\rho_{1,s_1}} \right) \quad (3.45)$$

### Linearization for pressure coefficient and flux update

To find the pressure coefficient arising out of specifying the inflow velocity using the isentropic relation for the ideal gas, we take the derivative of Eq. (3.29) with respect to pressure to obtain,

$$u_i \frac{\partial u_i}{\partial p} = - \left( \frac{1}{\rho_0} \right) \left( \frac{p_0}{p} \right)^{\frac{1}{\gamma}} \quad (3.46)$$

Multiplying both sides by the square of the inflow cell area  $A_i A_i$  and defining the flux as  $\tilde{u} = u_i A_i$ , we get,

$$\frac{\partial \tilde{u}}{\partial p} = - \left( \frac{A_i A_i}{\rho_0 \tilde{u}} \right) \left( \frac{p_0}{p} \right)^{\frac{1}{\gamma}} \quad (3.47)$$

or rewriting in terms of  $\rho$  we get,

$$\frac{\partial \tilde{u}}{\partial p} = - \frac{A_i A_i}{\rho \tilde{u}} \quad (3.48)$$

## 3.4 Solution Procedure

### 3.4.1 Convergence criterion

The residuals are defined for the continuity equation as:

$$R_{L2} = \left[ \sum_{ii} (C_e - C_w + C_n - C_s + C_t - C_b)_{ii}^2 \right]^{1/2} / F_1 \quad (3.49)$$

$$R_{L\infty} = \max_{ii} (|C_e - C_w + C_n - C_s + C_t - C_b|_{ii}) / F_1 \quad (3.50)$$

For all the other equations the residuals are defined as:

$$R_m(L_2) = \left[ \sum_{ii} \left( \sum_{nb} A_{nb} \phi_{nb} + S_U - A_P \phi_P \right)_{ii}^2 \right]^{1/2} / F_m \quad (nb = E, W, N, S, T, B) \quad (3.51)$$

$$R_m(L_\infty) = \max_{ii} \left( \left| \sum_{nb} A_{nb} \phi_{nb} + S_U - A_P \phi_P \right|_{ii} \right) / F_m \quad (nb = E, W, N, S, T, B) \quad (3.52)$$

where ii runs over all the computational nodes and  $F_m$  ( $m=1$  to 7) are the normalisation factors defined in Eq. 3.18. The calculation is declared converged when,

$$R_{max} = \text{MAX}(R_1, \dots, R_7) \leq \text{EPS} \quad (3.53)$$

where EPS is the convergence criterion prescribed by the user. **TransAT** makes use of the  $L_\infty$  norm, which is a stricter convergence condition than the  $L_2$  norm.

### 3.4.2 Calculation sequence

The sequence in which the calculation is carried out is as follows:

- a. Initialise all field values by guess.
- b. Solve the  $V_1$ ,  $V_2$  and  $V_3$ -momentum equations using the guessed pressure field.
- c. Solve the pressure-correction equation to obtain the pressure-correction at the cell-centers; correct the convective fluxes at the cell-faces, the velocities and pressure at the cell-centers.
- d. Solve the  $k$ - and  $\epsilon$ -equations and update  $\mu_t$ , if the flow is turbulent.
- e. Solve the scalar S-equation, if required.
- f. Return to step b with updated field values.

Sequence of steps b to f is repeated until the convergence criterion (Eq. 3.53) is satisfied.

A complete sequence of steps b to f is defined as an **outer** loop, and a solution of the system of Eqs. 3.13 for each variable over the entire solution domain as an **inner** loop. The system of Eqs. 3.13 is solved by using the strongly implicit procedure of [Stone \(1968\)](#). For all variables but the pressure-correction, one inner loop is normally enough to reduce the corresponding residue to a reasonable low level, especially in case of highly convective flows. However, sufficient inner loops are required for the pressure-correction  $p'$ -equation, mainly due to the fact that the  $p'$ -equation is of diffusion type in which the coefficient matrix is symmetric but without any convective fluxes, thus making convergence relatively slow. The inner loops of solving the  $p'$ -equation are stopped when either of the following two criteria is satisfied ([van Doormaal & Raithby, 1984](#)):

$$R^m \leq \lambda R^i \quad (3.54)$$

$$m > \text{NSWC}_1 \quad (3.55)$$

where  $R^i$  and  $R^m$  are the residuals calculated in the same way as in Eq. 3.52 at the initial stage and after the  $m$ -th loop,  $\lambda = 0.1$  and  $\text{NSWC}_1=20$  are used in the program.

# Chapter 4

# Reactive Flow Modelling

## 4.1 Conservation equations

### 4.1.1 Species equation

**General formulation:** Mass fraction

The species equation for every species  $k$  can be written as (Poinsot & Veynante, 2001)

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial}{\partial x_i} (\rho(u_i^m + V_{k,i})Y_k) = \dot{\omega}_k^m \quad (4.1)$$

Where  $V_{k,i}^m$  is the diffusion velocity and the source term  $\dot{\omega}_k^m$  has units of  $[\text{kg}/\text{s m}^3]$ . The flow velocity  $u_i^m$  is defined in mass weighted fashion as

$$u_i^m = \sum_{k=1}^N Y_k u_i^k \quad (4.2)$$

where  $u_i^k$  is the velocity of species  $k$ . The sum of the mass weighted diffusion velocities is then zero by definition:

$$\sum_{k=1}^N V_{k,i}^m Y_k = 0 \quad (4.3)$$

The diffusion velocities are determined by the multicomponent diffusion equation which can be derived for a low density ideal gas (Williams, 1958; R.J. Kee, 2003):

$$\begin{aligned} \nabla X_k &= \sum_{j=1}^N \frac{X_k X_j}{D_{kj}} (\mathbf{v}^j - \mathbf{v}^k + (Y_k - X_k) \frac{\nabla p}{p}) \\ &\quad + \frac{\rho}{p} \sum_{j=1}^N Y_k Y_j (\mathbf{f}_k - \mathbf{f}_j) + \sum_{j=1}^N \left[ \frac{X_k X_j}{\rho D_{kj}} \left( \frac{D_{T,j}}{Y_j} - \frac{D_{T,k}}{Y_k} \right) \right] \frac{\nabla T}{T} \end{aligned} \quad (4.4)$$

where  $D_T$  is the thermal diffusion coefficient,  $D_{kj}$  is the binary diffusion coefficient of k in j and  $\mathbf{f}_k$  is the volume force. The diffusion velocity can also be approximated with Fick's law where the binary diffusion coefficients is replaced by a diffusion coefficient of species k in the mixture. Furthermore pressure and temperature gradients are neglected. It is important to realise that Fick's law is an approximation for the multicomponent diffusion equation and the constraint for diffusion velocities. Eq.(4.3) is no more fulfilled in general (Poinsot & Veynante, 2001).

$$V_{i,k}^m Y_k = -D_k \frac{\partial Y_k}{\partial x_i} \quad (4.5)$$

The species equation is then:

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho u_i^m Y_k}{\partial x_i} = \frac{\partial}{\partial x_i} \rho D_k \frac{\partial Y_k}{\partial x_i} + \dot{\omega}_k^m \quad (4.6)$$

### General formulation: Mole fraction

The species equation in terms of mole fractions reads:

$$\frac{\partial c X_k}{\partial t} + \frac{\partial}{\partial x_i} (c X_k (u_i^c + V_{k,i}^c)) = \dot{\omega}_k^c \quad (4.7)$$

Where c is the total concentration,  $X_k$  the mole fraction of species k and  $\dot{\omega}_k$  has units of [mol/s m<sup>3</sup>]. The concentration weighted velocity is then:

$$u_i^c = \sum_{k=1}^N X_k u_i^k \quad (4.8)$$

For the diffusion velocities the resulting constraint is:

$$\sum_{k=1}^N V_{i,k}^c X_k = 0 \quad (4.9)$$

In general  $V_{i,k}^c = V_{i,k}^m$  is only true if the molar weights of all species are equal. Again the diffusion velocities can be approximated with Ficks law where the same assumptions as in the mass fraction formulation are made:

$$V_{i,k}^c X_k = -D_k \frac{\partial X_k}{\partial x_i} \quad (4.10)$$

This finally results in:

$$\frac{\partial c X_k}{\partial t} + \frac{\partial c u_i^c X_k}{\partial x_i} = \frac{\partial}{\partial x_i} c D_k \frac{\partial X_k}{\partial x_i} + \dot{\omega}_k^c \quad (4.11)$$

As this equation was derived under the assumption of ideal gas, constant temperature and constant pressure, it can only be applied in the case of constant concentration.

### General formulation: Concentration

One can write the transport equation directly for concentration of species k ( $c_k$ )

$$\frac{\partial c_k}{\partial t} + \frac{\partial}{\partial x_i}(c_k(u_i^c + V_{k,i}^c)) = \dot{\omega}_k^c \quad (4.12)$$

The concentration weighted velocity is then

$$u_i^c = \sum_{k=1}^N c_k u_i^k \quad (4.13)$$

with the constraint:

$$\sum_{k=1}^N V_{i,k}^c c_k = 0 \quad (4.14)$$

The diffusion velocity is then written according to Ficks law as:

$$V_{i,k}^c c_k = -D_k \frac{\partial c_k}{\partial x_i} \quad (4.15)$$

Which results in the concentration equation:

$$\frac{\partial c_k}{\partial t} + \frac{\partial u_i^c c_k}{\partial x_i} = \frac{\partial}{\partial x_i} D_k \frac{\partial c_k}{\partial x_i} + \dot{\omega}_k^c \quad (4.16)$$

For constant concentration, Eq.(4.16) and Eq.(4.11) are the same but the latter can be used to derive an transport equation for a passive scalar.

### Passive scalars: Mass fraction

In a passive scalar formulation the flow velocity is not determined by mass or concentration averaged velocities. The velocity field  $u^b$  is assumed to be known and independent of the species. Therefore the diffusion velocities are simply added to this base flow field. In this case  $D_k$  is the diffusion coefficient of species k in the base flow. For passive mass fractions the following equation is solved:

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho u_i^b Y_k}{\partial x_i} = \frac{\partial}{\partial x_i} \rho D_k \frac{\partial Y_k}{\partial x_i} + \dot{\omega}_k^m \quad (4.17)$$

One component can be constrained to ensure mass conservation.

### Passive scalars: Concentration

As mentioned above the mole fraction formulation of the species equation can not be used for a general passive scalar as the total concentration can vary in space and time. However by replacing the concentration averaged velocity by the base velocity  $u_i^b$  in Eq. (4.16) one obtains a transport equation at the passive concentration (Fox, 2003).

$$\frac{\partial c_k}{\partial t} + \frac{\partial u_i^b c_k}{\partial x_i} = \frac{\partial}{\partial x_i} D_k \frac{\partial c_k}{\partial x_i} + \dot{\omega}_k^c \quad (4.18)$$

### Passive scalars: Mixture fraction and progress variable

In order to apply the mixture fraction approach the passive scalar must obey an equation of the form (Fox, 2003)

$$\frac{\partial \Phi}{\partial t} + u_i^b \frac{\partial \Phi}{\partial x_i} = D_k \frac{\partial^2 \Phi}{\partial x_i \partial x_i} + \dot{\omega}_\Phi \quad (4.19)$$

Assuming constant D and using the continuity equation this can be rewritten in a form suitable for TransAT

$$\frac{\partial \Phi}{\partial t} + \frac{\partial \Phi u_i^b}{\partial x_i} = \frac{\partial}{\partial x_i} D \frac{\partial \Phi}{\partial x_i} + \dot{\omega}_\Phi + \Phi \frac{\partial u_i^b}{\partial x_i} \quad (4.20)$$

This equation is solved for the mixture fraction and progress variable where in the case of the mixture fraction the source term will vanish.

### Chemical source term

In a laminar flow the chemical source term is determined by Arrhenius rate constants:

$$\dot{\omega}_k^m = W_k \sum_{l=1}^r \nu_{il} \dot{\omega}_l \quad (4.21)$$

where  $r$  is the number of reactions,  $W_k$  is the molecular weights of component  $k$ ,  $\nu_{kl} = \nu''_{kl} - \nu'_{kl}$  is the difference between forward and backward stoichiometric coefficients and  $\dot{\omega}_l$  is given by

$$\dot{\omega}_l = k_{fl} \prod_{j=1}^n \left( \frac{\rho Y_j}{W_j} \right)^{\nu'_{jl}} - k_{bl} \prod_{l=1}^n \left( \frac{\rho Y_j}{W_j} \right)^{\nu''_{jl}} \quad (4.22)$$

and  $k_{fl}, k_{bl}$  are the rate coefficients of the forward and backward reaction, respectively. It results that the sum  $\sum_{i=1}^n \dot{\omega}_k^m = 0$ . The rate coefficients are usually represented in the Arrhenius equation

$$k = A e^{-\frac{E_a}{RT}} \quad (4.23)$$

where  $A$  is the pre-exponential factor,  $T$  is the temperature,  $E_a$  is the activation energy and  $R$  is the gas constant.

### Definition of mixture fraction and progress variable based on concentration

Besides the specific derivation for a combustion system, mixture fraction and progress variable for a system can be derived in a general way. One tries to make a coordinate transformation from the physical space (in terms of concentration or mass fractions) to a space where as few transported scalars as possible have a chemical source term. Consider a system with three species: reactant A, reactant B, and the product P which take part in the reaction:



All three species fulfill a transport equation of the form

$$\frac{\partial \Phi}{\partial t} + \frac{\partial \Phi u_i^b}{\partial x_i} = \frac{\partial}{\partial x_i} D \frac{\partial \Phi}{\partial x_i} + \dot{\omega}_\Phi^c + \Phi \frac{\partial u_i^b}{\partial x_i} \quad (4.25)$$

where  $\Phi$  can be replaced by the concentration of the species ( $c_A, c_B$  or  $c_P$ ). The inlets of the system contain either species A with concentration  $c_{A0}$  or species B with  $c_{B0}$ . A conserved scalar, (mixture fraction Z) can then be defined as:

$$Z = \frac{\nu_A c_B - \nu_B c_A + \nu_B c_{A0}}{\nu_A c_{B0} + \nu_B c_{A0}} \quad (4.26)$$

To obtain a transport equation for the mixture fraction,  $\Phi$  in Eq.(4.25) can be replaced by Z where  $\dot{\omega}_Z = 0$ . Without a reaction the pure mixing solution is then  $c_A = c_{A0}(1 - Z)$  and  $c_B = c_{B0}Z$  (Poinsot & Veynante, 2001).

As there is only one reaction in this system the source term of all species is directly linked to the reaction rate of this reaction. R is defined as the molar reaction rate [mol/s m<sup>3</sup>] with which species the unity stoichiometric coefficient would be produced or destroyed. If the reaction rates are determined with an Arrhenius formulation we have  $R = k c_A c_B$  in the case of a one step reaction of two reactants. One can then define the vector  $\Upsilon$  as  $\dot{\omega}_i^c = \Upsilon_i R$ . For the one step reaction this yields:

$$\Upsilon = \begin{pmatrix} -\nu_A \\ -\nu_B \\ \nu_P \end{pmatrix} \quad (4.27)$$

The progress variable ( $\xi$ ) describes the extend of reaction. It is defined such that it is zero in the unreacted state and one when the reaction has finished. The source term of the progress variable is defined to be  $R/\gamma$  where  $\gamma$  is an appropriate scaling factor to obtain the correct scaling ( $0 < \xi < 1$ ). The concentration vector  $\mathbf{c}$  can be written as a linear combination of mixture fraction and progress variable (Fox, 2003):

$$\mathbf{c} = \mathbf{c}_0 + \mathbf{M}_Z Z + \mathbf{M}_\xi \xi \quad (4.28)$$

$\mathbf{c}_0$  and  $\mathbf{M}_Z$  can be determined from the pure mixing solution as

$$\mathbf{c}_0 = \begin{pmatrix} c_{A0} \\ 0 \end{pmatrix} \quad (4.29)$$

and

$$\mathbf{M}_Z = \begin{pmatrix} -c_{A0} \\ c_{B0} \end{pmatrix} \quad (4.30)$$

If we write the the transport equation (4.25) for  $\mathbf{c}$  the source term which remains is  $\mathbf{M}_\xi/\gamma R$  as  $Z$  and  $\mathbf{c}_0$  are conserved scalars. As the source term for the equations was defined as  $\Upsilon R$ ,  $\mathbf{M}_\xi$  is equal to  $\gamma \Upsilon$ . For Eq.(4.24) the appropriate scaling factor is (Fox, 2003)

$$\gamma = \frac{c_{A0}c_{B0}}{\nu_A c_{B0} + \nu_A c_{B0}} \quad (4.31)$$

This finally yields the concentration of all three species expressed with mixture fraction and progress variable:

$$c_A = c_{A0}(1 - Z - (1 - Z_{st})\xi) \quad (4.32)$$

$$c_B = c_{B0}(Z - Z_{st}\xi) \quad (4.33)$$

$$c_P = \nu_P c_{B0} Z_{st} \xi \quad (4.34)$$

where  $Z_{st}$  is the stoichiometric mixture fraction defined as

$$Z_{st} = \frac{\nu_B c_{A0}}{\nu_A c_{B0} + \nu_B c_{A0}} \quad (4.35)$$

The source term for the progress variable in Arrhenius formulation is then

$$\dot{\omega}_\xi = \frac{R}{\gamma} \quad (4.36)$$

$$= \frac{k c_A c_B}{c_{B0} Z_{st}} \quad (4.37)$$

$$\dot{\omega}_\xi = k Z_{st} c_{B0} \left( \frac{1 - Z}{1 - Z_{st}} - \xi \right) \left( \frac{Z}{Z_{st}} - \xi \right) \quad (4.38)$$

### Definition of mixture fraction and progress variable based on mass fraction

If we define the mixture fraction for mass fractions the mixture fraction definition is slightly different

$$Z = \frac{\nu_A M_A Y_B - \nu_B M_B Y_A + \nu_B M_B Y_{A0}}{\nu_A M_A Y_{B0} + \nu_B M_B Y_{A0}} \quad (4.39)$$

As the mixture definition is now in terms of masses the mass stoichiometric coefficients  $\nu_i M_i$  have to be used where  $M_i$  is the molar weight of species i. Again we have the species A,B and P which take part in the reaction defined in Eq.(4.24). The inlets of the system contain either species A with mass fraction  $Y_{A0}$  or species B with  $Y_{B0}$ . Again a linear combination of mixture fraction and progress variable can be defined for the mass fractions:

$$\mathbf{Y} = \mathbf{Y}_0 + \mathbf{M}_Z Z + \mathbf{M}_\xi \xi \quad (4.40)$$

From the pure mixing solution we can obtain:

$$\mathbf{Y}_0 = \begin{pmatrix} Y_{A0} \\ 0 \end{pmatrix} \quad (4.41)$$

and

$$\mathbf{M}_Z = \begin{pmatrix} -Y_{A0} \\ Y_{B0} \end{pmatrix} \quad (4.42)$$

Using the same definition as above reaction vector is then

$$\boldsymbol{\Upsilon} = \begin{pmatrix} -\frac{M_A \nu_A}{\rho} \\ -\frac{M_B \nu_B}{\rho} \\ \frac{M_P \nu_P}{\rho} \end{pmatrix} \quad (4.43)$$

As the product mass fraction has already the desired scaling ( $0 < Y_P < 1$ ) it can directly be used as reaction progress variable. The scaling factor  $\gamma$  can then be obtained from:

$$\dot{\omega}_\xi = \dot{\omega}_P = R/\gamma \quad (4.44)$$

Therefore

$$1/\gamma = M_P \nu_P / \rho \quad (4.45)$$

and

$$\mathbf{M}_\xi = \begin{pmatrix} -\frac{M_A \nu_A}{M_P \nu_P} \\ -\frac{M_B \nu_B}{M_P \nu_P} \\ 1 \end{pmatrix} \quad (4.46)$$

From this we get:

$$Y_A = Y_{A0}(1 - Z) - \frac{M_A \nu_A}{M_P \nu_P} \xi \quad (4.47)$$

$$Y_B = Y_{B0}Z - \frac{M_B \nu_B}{M_P \nu_P} \xi \quad (4.48)$$

$$Y_P = \xi \quad (4.49)$$

The source term for the progress variable in an Arrhenius form reaction rate expression is then

$$\dot{\omega}_\xi = \frac{R}{\gamma} \quad (4.50)$$

$$= \frac{\nu_P M_P}{\rho} k c_A c_B \quad (4.51)$$

$$= \frac{\nu_P M_P k}{\rho} \frac{\rho Y_A}{M_A} \frac{\rho Y_B}{M_B} \quad (4.52)$$

$$\dot{\omega}_\xi = \frac{\nu_P M_P \rho k}{M_A M_B} \left( Y_{A0}(1 - Z) - \frac{M_A \nu_A}{M_P \nu_P} \xi \right) \left( Y_{B0}Z - \frac{M_B \nu_B}{M_P \nu_P} \xi \right) \quad (4.53)$$

For the eddy dissipation the source term is always divided by one of the three mass stoichiometric coefficients. Therefore the mass stoichiometric coefficients can be normalised by the the mass stoichiometric coefficient of component A:

$$r = \frac{\nu_B M_B}{\nu_A M_A} \quad (4.54)$$

$$s = \frac{\nu_P M_P}{\nu_A M_A} \quad (4.55)$$

According to the eddy dissipation theory, the mass reaction rate of A is

$$\dot{\omega}_A^m = -A \rho \frac{\epsilon}{k} \min \left( Y_A, \frac{Y_B}{r}, \frac{BY_P}{s} \right) \quad (4.56)$$

From this we conclude that

$$R = -\frac{\dot{\omega}_A^m}{\nu_A M_A} \quad (4.57)$$

Therefore the source term for the progress variable is:

$$\dot{\omega}_\xi = \frac{R}{\gamma} \quad (4.58)$$

$$= \frac{\nu_P M_P}{\nu_A M_A} \frac{\dot{\omega}_A^m}{\rho} \quad (4.59)$$

$$\dot{\omega}_\xi = s A \frac{\epsilon}{k} \min \left( Y_A, \frac{Y_B}{r}, \frac{BY_P}{s} \right) \quad (4.60)$$

#### 4.1.2 Temperature equation

A balance equation can be also written for the temperature

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \mathbf{v} \cdot \nabla T = \frac{\partial p}{\partial t} + \nabla \cdot (\lambda \nabla T) - \sum_{i=1}^n \rho c_{p,i} D \nabla Y_i \cdot \nabla T + \omega_T \quad (4.61)$$

where  $\lambda$  is the thermal conductivity and  $c_p$  is the heat capacity at constant pressure. An equation can be written for temperature source term

$$\omega_T = \frac{1}{c_p} \sum_{k=1}^r Q_k w_k \quad (4.62)$$

where  $Q_k = -\sum_{i=1}^n \nu_{ik} W_i h_i$  is the heat of combustion of reaction k.

The vector  $(Y_1, Y_2, \dots, Y_n, T)$  can be called the “reactive scalars” vector  $\psi$ .

If different diffusivity must be considered, the convection velocity of the species must be corrected by adding a correction velocity  $V_i^c = \sum_{k=1}^N D_k \frac{\partial Y_k}{\partial x_i}$  to ensure global mass conservation.

## 4.2 Turbulent reacting flow

### 4.2.1 RANS

When the species have different densities, the usual Reynolds average gives new unclosed terms  $\overline{\rho' u'_j}$  in the mass balance equation corresponding to the correlation between density and velocity fluctuations. To avoid this difficulty, mass-weighted averages (called Favre averages) are usually preferred,

$$\tilde{f} = \frac{\overline{\rho f}}{\overline{\rho}} \quad (4.63)$$

where  $f$  is a generic quantity. Splitting  $Y_i$  into a Favre mean  $\tilde{Y}_i$  and a fluctuation  $Y''_i$  Eq.(4.17) becomes

$$\bar{\rho} \frac{\partial \tilde{Y}_i}{\partial t} + \bar{\rho} \tilde{\mathbf{v}} \cdot \nabla \tilde{Y}_i = \nabla \cdot (\overline{\rho D_i \nabla Y_i}) - \nabla \cdot (\widetilde{\rho \mathbf{v}'' Y''_i}) + \bar{\rho} \tilde{S}_i \quad (4.64)$$

In this equation the r.h.s. terms must be modelled.

The species turbulent fluxes  $\widetilde{\mathbf{v}'' Y''_i}$  are generally closed using a classical gradient assumption

$$\widetilde{\mathbf{v}'' Y''_i} = -D_t \nabla \tilde{Y}_i \quad (4.65)$$

where  $D_t$  is a turbulent diffusivity which is modelled by analogy to the eddy viscosity as  $D_t = \frac{\nu_t}{Sc_t}$  where  $Sc$  is a turbulent Schmidt number. For a reactive scalar this closure is not always acceptable and can be improved upon (Peters, 2000).

The laminar diffusive fluxes are generally neglected or retained by adding a laminar diffusivity to the turbulent viscosity in Eq.(4.65).

$$\bar{\rho} \tilde{\mathbf{v}} \cdot \nabla \tilde{Y}_i = \bar{\rho} \bar{D}_i \nabla \tilde{Y}_i \quad (4.66)$$

where  $\bar{D}_i$  is a “mean” species molecular diffusion coefficient.

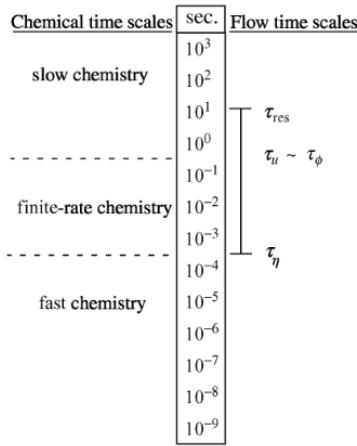


Figure 4.1: Comparison example between flow and chemical time scales. The flow time scales range from the Kolmogorov time scale  $\tau_\nu$ , through the turbulence time scale  $\tau_u$ , up to the mean residence time  $\tau_{res}$ . The micromixing time  $\tau_m$  will usually lie between  $\tau_\nu$  and  $\tau_u$ . Figure taken by Fox (2003)

#### 4.2.2 LES

The same equation obtained for RANS approach can be obtained with LES, substituting the filter operation to the Favre average and the same terms needs a closure that is achieved using the sub-grid scale viscosity and additional sub-grid scale model for source terms.

### 4.3 Non-Premixed Reactive Flows

In combustion this case is also called *Diffusion Flames* since the reactants are initially separated and the reaction takes place in a narrow region or surface.

In non-premixed flows the reactions are strictly connected with the mixing and by comparing these two phenomena we can distinguish different reaction regimes. With reference to Figure 4.1, we define “slow” chemistry for the reactions with time scales larger than the micromixing time scale, “fast” chemistry when the chemical time scales are all smaller than the Kolmogorov time scale and “finite-rate” chemistry in the middle. In the following we will consider first the two limiting cases, “frozen” (very-slow) chemistry and “infinitely-fast” chemistry.

For *non-premixed* reacting flows, it is often possible to define a *mixture-fraction vector*  $\mathbf{Z}$  that can be employed to develop a simpler set of equations with chemical source term. In Fox (2003) a general method for finding the mixture-fraction vector (when it exists) for a given set of initial and inlet conditions. When a mixture-fraction vector exists, the system is fully described for

infinitely-fast chemistry. For finite-rate chemistry a *reaction-progress vector*  $\xi$  must be taken into account. This vector is null for all initial and inlet conditions.

The importance of the mixture fraction is that it is a conserved quantity, a scalar quantity that is neither created nor destroyed so the equations for  $Z_i$  reduces to

$$\rho \frac{\partial Z_i}{\partial t} + \rho \mathbf{v} \cdot \nabla Z = -\nabla \cdot (-\rho D \nabla Z Y_i) \quad (4.67)$$

Many computational models for combustion are developed with a single mixture fraction which can be used only for two-feed systems. If more than two feeds enter into a combustion chamber, the concept of single mixture fraction can no longer be used. In fact, even if it can formally be extended to multiple mixture fraction variables, it becomes less attractive for modelling because the resulting flamelet equations are defined in a multi-dimensional space.

The usual definition of mixture fraction for non-premixed combustion is best derived for an homogeneous system in the absence of diffusion. In this case the reaction equation for complete combustion of hydrocarbon fuel  $C_m H_n$  can be written as

$$\frac{dY_O}{\nu'_O W_O} = \frac{dY_F}{\nu'_F W_F} \quad (4.68)$$

where  $F$  stands for fuel and  $O$  for oxidiser and  $\nu'$  are the stoichiometric coefficients. For a homogeneous system this equation may be integrated to obtain

$$\nu Y_F - Y_O = \nu Y_{F,u} - Y_{O,u} \quad (4.69)$$

where  $\nu = \frac{\nu'_O W_O}{\nu'_F W_F}$  is the stoichiometric oxygen-to-fuel mass ratio and the subscript  $u$  denotes the initial conditions in the unburnt mixture. If the diffusivities of fuel and oxidiser are equal, 4.69 is valid also for spatially inhomogeneous systems such as a diffusion flame. So the quantity  $\nu Y_F - Y_O$  is a conserved variable and the mixture fraction can be expressed as

$$Z = \frac{\beta - \beta_{O,u}}{\beta_{F,u} - \beta_{O,u}} \quad (4.70)$$

where  $\beta$  is a conserved variable. In our case it becomes

$$Z = \frac{\nu Y_F - Y_O + Y_{O,2}}{\nu Y_{F,1} + Y_{O,2}} \quad (4.71)$$

where subscript 1 denotes the the fuel stream and 2 the oxidiser stream.

The stoichiometric mixture fraction  $Z_{st}$  can be defined as the value of  $Z$  for a stoichiometric mixture. In this case the quantity  $\nu Y_F - Y_O$  vanishes and we can obtain

$$Z_{st} = \frac{Y_{O,2}}{\nu Y_{F,1}} \quad (4.72)$$

The mixture fraction can be related to the commonly used equivalence ratio  $r$ , which is defined as the fuel-to-air ratio in the unburnt mixture normalised by that of the stoichiometric mixture

$$r = \frac{\nu Y_{F,u}}{Y_{O,u}} \quad (4.73)$$

The mixture fraction can be also expressed starting from a quantity related to chemical elements. In fact, while the mass of chemical species may change due to chemical reactions, the mass of elements is conserved. (Poinsot & Veynante, 2001; Peters, 2000).

### Combustion regime indicators

- Turbulence mixing time scale  $\tau_m$
- Chemical time scale  $\tau_c$
- Turbulence dissipation time scale (Kolmogorov time scale)  $\tau_\nu$
- Lewis number  $Le = \frac{\alpha}{D}$  where  $\alpha$  is the thermal diffusivity and  $D$  is the mass diffusivity.
- Damkohler number  $Da = \frac{\tau_u}{\tau_c}$  is the ratio between the integral time scale and the chemical time scale
- Karlovitz number  $Ka = \frac{\tau_c}{\tau_\nu}$  is the ratio between chemical time scale and the Kolmogorov time. From this it follows that  $Re = Da^2Ka^2$ .

#### 4.3.1 Slow Chemistry

##### Arrhenius approach

In our discussion, the most important closure will be the source term  $\tilde{S}_i$ . The simpler model is to neglect the fluctuations and write the Arrhenius equation with the mean quantity  $\bar{Y}_i$ . This approximation is called the “Arrhenius approach” and the closure relation is simply

$$\overline{\omega_i(\mathbf{Y})} = \omega_i(\bar{\mathbf{y}}) \quad (4.74)$$

This model is relevant only for laminar flow or when chemical time scale are larger than turbulent time scale ( $\tau_\phi > \tau_u$ , low Damkohler number limit)

#### 4.3.2 Infinitely Fast Chemistry

In this representation one or more chemical reactions are specified but they are assumed to be infinitely fast. In modelling of combustion processes, this approach is known in terms of the “mixed is burned” assumption. The reactions are completely controlled by the mixture fraction

since reactants are converted directly to final products with a one-step description. So it is enough to solve the balance equation for the mixture fraction variable and then calculate mass fractions and temperature. This approach is valid only for low Mach number flow with constant thermodynamic pressure, same species heat capacities, Lewis number equal to unity and molecular diffusivities all equal to  $D$ .

### Laminar flames

The equation for the mixture fraction is:

$$\rho \frac{\partial Z}{\partial t} + \rho \mathbf{v} \cdot \nabla Z = \nabla \cdot (\rho D \nabla Z) \quad (4.75)$$

Once the solution is known in the entire flow field, the flame surface is defined as the surface of stoichiometric mixture, which is obtained by setting  $Z(\mathbf{x}, t) = Z_{st}$ . In the vicinity of that surface the reactive-diffusive structure can be described by the so-called “flamelet equations”

$$\rho \frac{\partial Y_i}{\partial t} = \frac{\rho}{Le_i} \frac{\chi}{2} \frac{\partial^2 Y_i}{\partial Z^2} + \omega_i \quad (4.76)$$

where  $\chi$  is the scalar instantaneous dissipation rate defined as  $\chi = 2D|\nabla Z|^2$ .

### Turbulent flames

The complexity added by turbulence comes from the averaging (or filtering) procedure. To determine the average mass fractions and temperature, the mean value of the mixture fraction is not sufficient: higher moments are needed. The two possible approaches are

- solve for the mixture fraction (mean and eventually higher moments) and then use a model or libraries to get species mass fractions and temperature.
- solve the balance equation of the mass fractions with an appropriate model for the source term

### Eddy Dissipation Concept

The eddy dissipation concept, devised by Magnussen and Mjertager (1997), directly extends the Eddy Break-Up model to non-premixed combustion. The main idea is to replace the chemical time scale of an assumed one-step reaction by the turbulent time scale  $\frac{k}{\epsilon}$ . Thereby the model eliminates the influence of chemical kinetics, representing the fast chemistry limit only. The fuel mean burning rate is estimated from fuel ( $Y_F$ ), oxidiser ( $Y_O$ ) and products ( $Y_P$ ) mean mass fractions as

$$\bar{\omega}_F = A \bar{\rho} \frac{\epsilon}{k} \min \left( \bar{Y}_F, \frac{\bar{Y}_O}{\nu}, \frac{B \bar{Y}_P}{1 + \nu} \right) \quad (4.77)$$

where A and B are modelling constants. For a highly exothermic reaction (combustion),  $\frac{BY_P}{1+\nu}$  is included because it is proportional to the heat release and therefore models the temperature dependence of the reaction. For isothermal reactions this term can be dropped.

The eddy dissipation concept is based on the assumption that scalar mixing lengthscales and turbulent lengthscales are the same, i.e. the concentration is uniform within a Kolmogorov length-scale. This is only valid for unity Schmidt number (observed in gas phase reactions) but not for  $Sc \gg 1$  which is the case in liquid fluids. In this case  $k$  and  $\epsilon$  have to be replaced by the scalar variance ( $\langle (\Phi')^2 \rangle$ ) and by the scalar dissipation rate ( $\epsilon_\phi$ ). In order to model  $(\frac{\epsilon_\phi}{\langle (\Phi')^2 \rangle})$  one can derive an algebraic equation depending on the Schmidt number. There also exist more complex models which involve the solution of transport equations for the variance in different parts of the Kolmogorov energy spectrum.

## Chapter 5

# Boussinesq Approximation

The Boussinesq approximation can be used to solve natural convection problems without having to solve the full compressible Navier-Stokes equations. This is achieved by selectively considering local variations in density, due to local variations of temperature, only for the gravity term in the Navier-Stokes equations. All remaining properties and terms remain unaffected.

The variation of density with temperature is linearized around a set reference density  $\rho_0$  and a corresponding reference Temperature  $T_0$ . The linear constant of proportionality is the thermal expansion coefficient, denoted by  $\beta_0$ .

Variations of density are then approximated through the linear relation

$$\tilde{\rho}(T) = \rho_0 \cdot (1 - \beta_0 (T - T_0)) \quad (5.1)$$

**Please note:** The linear approximation of density variations with temperature is generally only valid for a small temperature range. The expected temperature range should also be considered for the choice of reference properties and thermal expansion coefficient. For large temperature variations it is recommended to solve for compressible flows.



# Chapter 6

## Non-Newtonian Models

For Newtonian fluids, the viscosity is a constant that defines the ratio of stress-to-strain-rate is constant. This property is false for so-called Non-Newtonian fluids. This section describes the capabilities of TransAT for handling such Non-Newtonian fluids.

### Bingham model

The Bingham model may be used to describe plastic materials, which feature a yield stress. In this model, the viscosity  $\eta$  may be written

$$\eta = \begin{cases} \infty & \text{if } |\tau| \leq \tau_b \\ \eta_p + \frac{\tau_b}{\dot{\gamma}} & \text{if } |\tau| > \tau_b \end{cases} \quad (6.1)$$

where  $\dot{\gamma}$  is the shear rate,  $\tau$  is the shear stress.  $\tau_b$  and  $\eta_p$  are two model constants standing for the yield stress and the plastic viscosity, respectively.

### Casson model

The Casson model has been developed to simulate plastic materials. In this model, the viscosity is defined by

$$\eta^{1/2} = \frac{\tau_b^{1/2}}{\dot{\gamma}^{1/2}} + \eta_p^{1/2} \quad (6.2)$$

where  $\dot{\gamma}$  is the shear rate,  $\tau$  is the shear stress.  $\tau_b$  and  $\eta_p$  are two model constants standing for the yield stress and the plastic viscosity, respectively.

## Power-law model

The power-law model is a simple model for the simulation of pseudo-plastic or dilatant fluids. Viscosity in the power-law model reads

$$\eta = \eta_c \dot{\gamma}^n \quad (6.3)$$

where  $\dot{\gamma}$  is the shear rate, and  $\eta_c$  and the power law exponent  $n$  are constants of the model.

## Carreau model

The Carreau model may be used for polymeric liquids. it reads

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty) (1 + \lambda \dot{\gamma})^{\frac{n-1}{2}} \quad (6.4)$$

where  $\dot{\gamma}$  is the shear rate. The zero-shear viscosity  $\eta_0$ , the infinite shear viscosity  $\eta_\infty$ , the relaxation time of the fluid  $\lambda$  and the exponent  $n$  are model constants.

## Cross model

It reads

$$\eta = \frac{\eta_0}{(1 + (\lambda \dot{\gamma})^m)} \quad (6.5)$$

where  $\dot{\gamma}$  is the shear rate. The zero-shear viscosity  $\eta_0$ , the relaxation time of the fluid  $\lambda$  and the exponent  $n$  are model constants.

## Carreau-Yasuda model

The Carreau-Yasuda model may be used for the simulation of polymers. The viscosity is then written

$$\eta = \eta_\infty + (\eta_0 - \eta_\infty) (1 + (\lambda \dot{\gamma})^a)^{\frac{1-n}{a}} \quad (6.6)$$

where  $\dot{\gamma}$  is the shear rate. The zero-shear viscosity  $\eta_0$ , the infinite-shear viscosity  $\eta_\infty$ , the relaxation time  $\lambda$  and the exponents  $a$  and  $n$  are model constants.

## Thixotropic model

The Thixotropic model may be used for the simulation of crude oil. The viscosity is a function of a scalar quantity  $\lambda$

$$\eta = \eta_\infty + \alpha \lambda \quad (6.7)$$

where the infinite-shear viscosity  $\eta_\infty$  and  $\alpha$  are fluid properties and are considered to be model-constants. The scalar  $\lambda$  evolves according to the equation

$$\frac{D\lambda}{Dt} = -a\lambda\dot{\gamma} + (1 - \lambda)b\dot{\gamma} \quad (6.8)$$

where  $a$  is the break-down rate of  $\lambda$ ,  $b$  is the build-up rate and  $\frac{D\lambda}{Dt}$  is the substantial derivative of  $\lambda$ . The value of  $\lambda$  ranges from 0 to 1. At 0 shear-rates, the value of  $\lambda$  is 1 and gradually as shear increases  $\lambda$  decays to 0.

## Hershel-Bulkley model

The Hershel-Bulkley model may be used to describe In this model, the viscosity  $\eta$  may be written

$$\eta = \begin{cases} \infty & if \quad |\tau| \leq \tau_b \\ k(\dot{\gamma})^{n-1} + \frac{\tau_b}{\dot{\gamma}} & if \quad |\tau| > \tau_b \end{cases} \quad (6.9)$$

where  $\dot{\gamma}$  is the shear rate,  $\tau$  is the shear stress.  $\tau_b$  , $k$  and  $n$  are three model constants.  $\tau_b$  refers to the yield-stress of the fluid,  $n$  is an exponent and depending on whether it is greater than or less than 1 determines whether the fluid is shear-thickening or shear-thinning and  $k(\dot{\gamma})^{n-1}$  is the plastic viscosity.



# Chapter 7

## Viscoelasticity

For Newtonian fluids, the stress is simply the sum of the pressure gradient and the newtonian stress. A more general description is:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i \quad (7.1)$$

where  $\sigma$  is the total stress tensor, which can be decomposed:

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \quad (7.2)$$

where  $p$  is the pressure and  $\tau$  is the viscoelastic stress tensor.

For a newtonian or viscoplastic fluid (no elastic behaviour), the viscoelastic stress becomes the viscous stress:

$$\tau_{ij}^s = 2\mu_s S_{ij} - \frac{2}{3}\mu_s S_{kk}\delta_{ij} \quad (7.3)$$

For viscoelastic fluids the viscoelastic stress becomes:

$$\tau_{ij} = \tau_{ij}^s + \tau_{ij}^e \quad (7.4)$$

where  $\tau^e$  is the polymeric or viscoelastic part, determined via the stress governing-equation depending on the viscoelastic model used.

In general, the response of a viscoelastic material cannot be described with a single characteristic time scale. This is why the more accurate viscoelastic models are the so-called multimode models which can be described as:

$$\tau^e = \sum_m \tau_m^e \quad (7.5)$$

where each  $\tau_m^e$  (mode) correspond to a specific elastic relaxation time scale and obeys its own constitutive equation.

The momentum equation then reads:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}^s}{\partial x_j} + \sum_{m=1}^M \frac{\partial \tau_{ij,m}^e}{\partial x_j} \quad (7.6)$$

The role of the viscoelastic solver in **TransAT** is to solve for the stress equation and couple it with the general momentum equation written above. The stress equation depends on the model used. The following models are available:

## 7.1 Olroyd-B. model

The stress equation (for one mode) for the Olroyd-B model is:

$$\check{\tau}^e + \frac{\tau^e}{\lambda} = 2GS - \frac{2}{3}Gtr(S)I \quad (7.7)$$

The only model parameters that are required (for each individual mode) are the elastic time scale  $\lambda$  and the polymeric viscosity  $\mu_p$  (since  $G = \frac{\mu_p}{\lambda}$ ).

## 7.2 Exponential Phan-Thien-Tanner model (PTT)

The exponential PTT model writes:

$$\check{\tau}^e + \xi(S.\tau^e + \tau^e.S) + \exp\left(\frac{\epsilon tr(\tau^e)}{\mu_p}\right)\tau^e = 2GS - \frac{2}{3}Gtr(S)I \quad (7.8)$$

## 7.3 Linear PTT model

The linear PTT is the linearized version of the exponential PTT and writes:

$$\check{\tau}^e + \xi(S.\tau^e + \tau^e.S) + \frac{\tau^e}{\lambda} + \frac{\epsilon tr(\tau^e)}{\mu_p}\tau^e = 2GS - \frac{2}{3}Gtr(S)I \quad (7.9)$$

For both PTT models the required parameters are (for each mode):  $\epsilon$ ,  $\xi$ ,  $\lambda$  and  $\mu_P$ .

## 7.4 Temperature correction using Arrhenius law

Usually viscosity correction using the Arrhenius law writes:

$$\mu = \mu_0 e^{-\alpha(T-T_0)} \quad (7.10)$$

where  $\alpha$  and  $T_0$  are given parameters.

## 7.5 Multiphase viscoelasticity

The viscoelastic solver is currently not available for homogeneous and algebraic slip multiphase models, only with level-set. The properties of each phase are weighted using the level-set function and the procedure is the same than for single phase viscoelastic flows.

## 7.6 Logarithm of Conformation Tensor (LCT)

The solution of the viscoelastic constitutive equation is made difficult by the high Weissenberg number problem (HWNP) which occurs when the Weissenberg number reaches moderately high values. The origin of this numerical is not well explained but seems to correspond to loss of positivity of conformation tensor  $A$ .

$$A = \frac{1-\xi}{G} \tau + I$$

The formulation of the constitutive equation in terms of the matrix logarithm of the conformation tensor has the advantage of not exhibiting such unstable behavior. Transforming the stress equation to obtain the LCT equation results in an equivalent for the stress equation, that needs to be solved and coupled to the momentum equation.



## Chapter 8

# Radiation

The Rosseland Model (also known as diffusion approximation model) for radiation has been implemented in TransAT. This model can be quickly described as follow:

The Rosseland model assumes that the radiation can be taken into account as a temperature-dependent conductivity. This means that no additional equation is solved, but an effective thermal conductivity is calculated to account for radiative heat transfer.

The radiative heat transfer writes:

$$q_{rad} = \frac{16\sigma T^3}{3\beta} \nabla T$$

where  $\sigma$  is the Stefan-Boltzmann constant and  $\beta$  is the Rosseland extinction coefficient. The diffusion approximation states that the radiative contribution to the heat transfer can be treated similarly to the conductive part provided that a radiative conductivity can be calculated. This conductivity depends on temperature ( $T^3$  contribution) and on the extinction coefficient  $\beta$ . In general  $\beta$  itself is temperature-dependent and must be computed. The Rosseland approximation states that  $\beta$  can be expressed as a polynomial of temperature:

$$\beta = \sum_{i=0}^N a_i T^i$$

The  $a_i$  are phase-dependent coefficients called the Rosseland coefficients, given as user-input. If the extinction coefficient is to be temperature-independent it is equal to  $a_0$  with all others Rosseland coefficients set to 0.

This is often a rough approximation because the radiative properties of media are also spectrum-dependent. This means that the above approximation for  $\beta$  for a set of Rosseland coefficients is only valid in a certain range of wavelength. To remedy that the full spectrum can be cut into several sub-ranges (called radiation bands) for each of which a different set of Rosseland coefficients is provided.

The radiative conductivity can then be calculated as a sum of contributions over the radiation bands.

# Chapter 9

## Immersed Surfaces Technique (IST)

### 9.1 Introduction

Immersed boundary (IB) techniques were developed in the late nineties for the simulation of flow interacting with solid boundary (see for review [Mittal & Iaccarino \(2005\)](#)), under various formulations. The most widely recognised formulation for instance employs a mixture of Eulerian and Lagrangian variables, where the solid boundary is represented by discrete Lagrangian markers embedding in and exerting forces to the Eulerian fluid domain. The interactions between the Lagrangian markers and the fluid variables are linked by a simple discretised delta function. The IB methods are all based on the direct momentum forcing (penalty approach) on the Eulerian grids, with various forcing formulations on the Lagrangian marker. The forcing should be performed because it ensures the satisfaction of the no-slip boundary condition on the immersed boundary in the intermediate time step. This forcing procedure involves solving a banded linear system of equations whose unknowns consist of the boundary forces on the Lagrangian markers, thus, the order of the unknowns is one-dimensional lower than the fluid variables. Numerical experiments show that the stability limit can in general be altered by the proposed force formulation, in that the second-order accuracy of the adopted numerical scheme is degraded to first order, and in the best conditions to 1.5.

The Immersed Surfaces Technique (IST) has been developed as part of TransAT, although other similar approaches have been developed in parallel. It is implemented in the CMFD code TransAT. The underpinning idea is inspired from the Level Set interface tracking technique for two-phase flows, where free surfaces are described by a hyperbolic convection equation. In the IST, the solid is described as the second phase, with its own thermo-mechanical properties. The technique differs substantially from the IB method discussed above, in that the jump condition at the solid surface is implicitly accounted for, not via direct momentum forcing (using the penalty approach) on the Eulerian grids. It has the major advantage of being able to solve conjugate

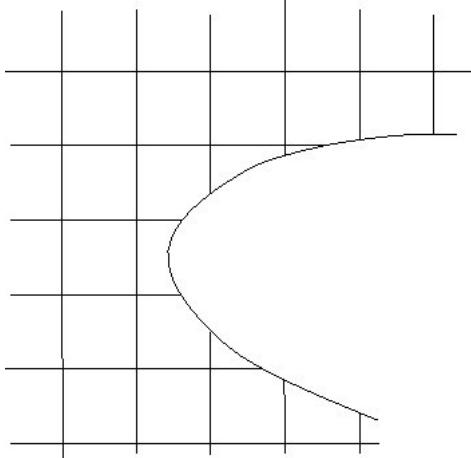


Figure 9.1: Representation of the solid surface within a cubical grid in the Immersed Surfaces Technique.  $\phi = 0$  at the solid boundaries.

heat transfer problems, in that conduction inside the body is directly linked to external fluid convection.

A simple example is reported in Figure 9.1, consisting of the flow past a conical shape or a round-shaped structure. The solid is first immersed into a cubical grid covered by a Cartesian mesh. The solid is defined by its external boundaries, which will be described by a sort of level set function denoted by  $\phi$ . The level set is a signed distance to the surface. It is zero at the surface, negative in the fluid and positive in the solid. The treatment of viscous shear at the solid surfaces is handled very much the same way as in all CFD codes, where wall indices are known from a boundary condition file. In the IST these wall cells are identified as those in which  $\phi = 0$ .

## 9.2 Equations and Implementation

### 9.2.1 Immersed Level Set and solid Heaviside functions

The immersed surface is represented on the fluid grid by a Level Set function ( $\phi_s$ ), where  $\phi_s = 0$  represents the fluid–solid interface.  $\phi_s$  is a signed distance function which is positive in the solid phase and negative in the fluid phase. The solid part of the domain is excluded during the solution of the fluid conservation equations through the use of a smooth Heaviside function  $H(\phi_s)$  which

has value 1 in the fluid phase and 0 in the solid phase.

$$H(\phi_s) = \frac{1}{2}(1 - \tanh(\frac{2\phi_s}{\delta_{sf}})) \quad (9.1)$$

where,  $\delta_{sf}$  is the finite solid–fluid interface thickness. This function is referred to as  $H^f$  since it indicates the fluid region.

We present the fluid conservation equations for the case where the solid is stationary and the solid velocity is set to zero ( $u_i^s = 0$ ). The standard Navier–Stokes equations are used for the fluid phase:

$$\frac{\partial}{\partial x_j}(H^f u_j^f) = 0 \quad (9.2)$$

$$\frac{\partial H^f \rho^f u_i^f}{\partial t} + \frac{\partial}{\partial x_j} \left( H^f \rho^f u_i^f u_j^f \right) = -H^f \frac{\partial p^f}{\partial x_i} + \frac{\partial}{\partial x_j} \left( 2\mu^f H^f S_{ij}^f \right) + H^f \rho^f g_i - 2\mu^f S_{ij}^f n_j \delta(\phi_s) \quad (9.3)$$

the last term in the RHS is a viscous shear at the wall, where  $n_j$  is the normal to the fluid–solid interface and  $\delta(\phi_s)$  is the Dirac delta function representing the location of the interface. The wall shear itself is modelled as (Beckermann *et al.*, 1999),

$$2\mu^f S_{ij}^f n_j = \mu^f \left( \frac{2u_i}{\delta_{sf}} \right) \quad (9.4)$$

When used in combination with RANS turbulence modelling and wall–functions, the wall shear is calculated using the logarithmic law of the wall.

### 9.2.2 Solid temperature equation

For an immersed surface TransAT can solve for the temperature distribution inside the solid. This separate equation for solid temperature is termed TSolid for short in the user manual and the graphical user interface.

For TSolid the steady state heat conduction equations are solved inside all solids selected

$$0 = \nabla \cdot (\lambda \nabla T) + \omega_T \quad (9.5)$$

where  $\lambda$  is the thermal conductivity and  $\omega_T$  is a heat source term.

Boundary conditions can be of the Dirichlet (constant temperature) or the Neumann (constant heat flux) type. Spatial variation in the boundary condition value is allowed.

**Please note:** Only steady state temperature distributions are solved inside the solid.

**Please note:** For non-zero heat flux through the immersed surface, an appropriate heat source term has to be applied inside the solid, in order to balance the heat equation.



# Chapter 10

## Smart Adaptive Run Parametrisation (SArP) algorithm

### 10.1 Theory

Evolutionary Genetic Algorithms (EGAs) are a family of computational models inspired by Darwin's evolution theory. These algorithms liken potential solutions to specific problems chromosome-like structures. Recombination operators are applied to these structures such that critical information is preserved. Despite being often viewed as function optimizers, genetic algorithms are applied to solve a wide range of problems.

A genetic algorithm implementation begins with a population of chromosomes (i.e. sets of parameters) which is expanded from generation to generation. The fitness of each chromosome is evaluated at each generation. The fitness of chromosomes is the determining factor for the generation of the following generations. Chromosomes of the next generation are produced by a reproduction operator.

From one generation to the other, the reproduction operator produces new chromosomes by means of mutation and crossover. The reproduction operator is implemented in such a way that it biased towards the fittest solutions of the population.

As the EGA progresses, the fittest solutions should prevail. The EGA is stopped if the convergence criteria of the EGA or the maximum number of generation are reached. The EGA can also be stopped earlier if the solution obtained from a chromosome meets the targets set for a satisfactory solution. In the case of the EGA implemented in [TransAT](#), satisfactory solutions can be obtained for steady simulation if residuals are below convergence threshold and for unsteady simulations if the physical time to be reached has been overcome.

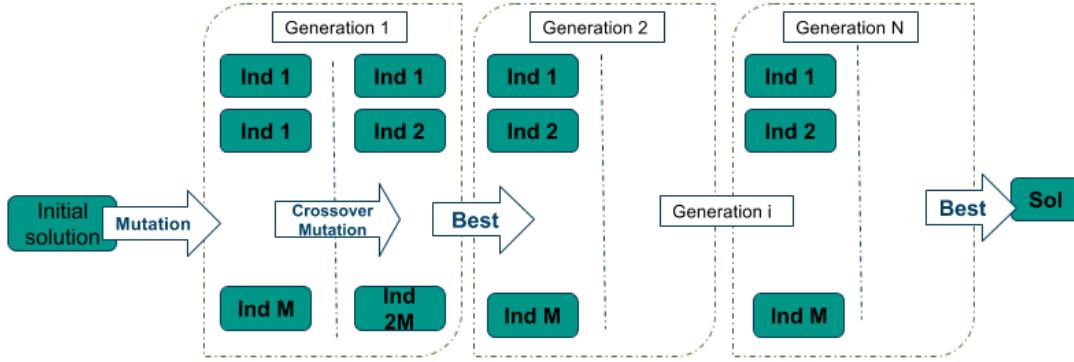


Figure 10.1: Overview of the genetic algorithm process

## 10.2 Genetic Algorithm

The EGA of the SArP module is a controller built around the numerical solver of [TransAT](#) with the aim of automatically selecting the numerical parameters to either optimise the convergence or stabilise a simulation. It selects the fittest solution in a finite population of solutions which evolves over different generations accordingly to a natural genetic process-like algorithm.

The only information required to make the EGA progress over generations is the knowledge of the fitness of the individuals to sort them. The fitness of the individuals is obtained by evaluating the so-called fitness function. The fitness function is based on objectives (e.g. target residuals, physical time constraint, etc.).

The problem statement can be expressed in terms of fitness function as follows:

$$\underset{x \in \Omega}{\operatorname{argmax}} f(x) \quad (10.1)$$

where  $\Omega$  is the set of chromosomes (i.e. the set of numerical parameters) and  $f$  is the fitness function.

### 10.2.1 Structure

The flowcharts in Figure 10.2 and Figure 10.3 show the different stages. Figure 10.2 gives a general overview of the process while Figure 10.3 goes into more details about the selection process.

As shown in Figure 10.3, the initial population of the EGA is generated by altering the original set of parameters i.e. chromosome. New chromosomes are produced at every generation by means of mutating the fittest individual with the aim of enhancing the fitness. In a nutshell, a population made of  $N$  individuals, comprises one original chromosome and  $N-1$  mutations of that original chromosome. The mutation rate, probability to carry out a mutation on a chromosome, is set at a high value to guarantee a variety of chromosomes in the population. With high mutation rates, the search space is wider and it reduces the risk of encountering a local optimisation problem.

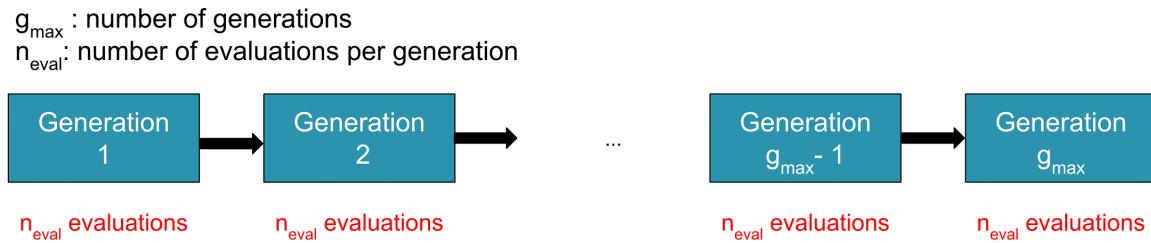


Figure 10.2: Genetic Algorithm Process

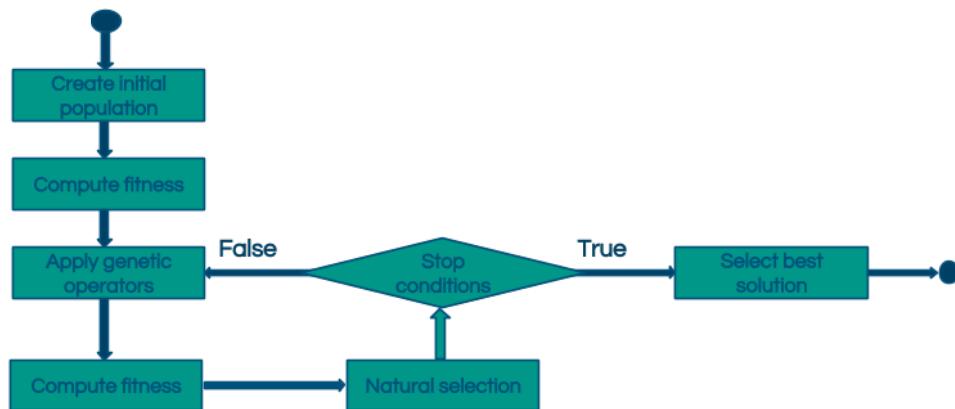


Figure 10.3: Genetic algorithm flowchart

Chromosomes are composed of genes. The genes are the individual parameters used to set numerical solvers. These parameters can be of one of the three types:

- Numerical type: value restricted to a range of numerical values
- Enumerate type: value restricted to elements of a list
- Boolean type: value is either true or false

The ranges of the different genes were set accordingly to the options available in [TransAT](#) and from experience.

### 10.2.2 Triggers and stopping criteria

#### Triggers

To avoid unnecessarily increasing the computation time, the EGA should only be used when a simulation diverges, crashes or does not converge fast enough. The activation conditions and the steps performed by the EGA vary depending on the type of simulations, stationary or transient.

The EGA has two modes which are depicted in the flowcharts of Figure ?? and Figure [10.5](#):

- optimisation: the set of solver parameters are optimised (convergence and/or time-marching process speed) by the EGA at the beginning of the simulation
- stability: the simulation normally runs and the EGA is only launched if certain trigger conditions are met during the simulation

In stability mode, the EGA is launched upon occurrence of one of the following conditions depending on the type of simulation:

- Stationary simulation (steady):
  1. Residuals converge slowly or diverge over the residual analysis window
  2. The solver of an equation encounters the following convergence issues:
    - maximum number of sweeps reached on  $n$  consecutive iterations with  $n$  a user-defined parameter
    - rate of unconverged iterations over the analysis window is greater than a user-defined threshold rate
  3. A jump in the residuals can be observed (discontinuity detection: ratio of residuals between current iteration and previous one becomes too high)

The convergence of the residuals is estimated by computing the slope of the linear function fitting the point cloud of residuals for each equation. The linear function is computed using the least squares method.

- Transient simulation (unsteady):
  1. Time-step duration becomes too small
  2. Time-step duration decreases over the analysis window
  3. The solver of an equation encounters the following convergence issues:
    - maximum number of sweeps reached on  $n$  consecutive iterations with  $n$  a user-defined parameter
    - rate of unconverged iterations over a time-step is greater than a user-defined threshold rate
  4. Target residuals are not reached for  $n$  consecutive time steps with  $n$  a user-defined parameter
  5. A jump in the residuals can be observed (discontinuity detection: ratio of residuals between current iteration and previous one becomes too high)

Similarly to residual convergence in steady cases, the time-step duration decrease is measured by fitting a linear function to the point cloud of time-step duration.

### Stopping criteria

The EGA stops when one of the stopping criteria is met. The stopping criteria vary accordingly to the nature of the simulation, steady or unsteady. For steady cases, the EGA stops if:

- the best solution found has not been improved upon for  $n$  consecutive generations with  $n$  a user-defined parameter
- the ratio between residuals at the end of the analysis window and the maximum residual at the beginning of the analysis window is less than a user-defined ratio. On top of this condition, the residuals of each equation must be overall decreasing or below the target residuals to exit the EGA
- the maximum number of generations,  $n_{max}$ , set by the user is reached without the EGA convergence criterion being met. In that case if at least one solution has been found during the process, the fittest solution found at the time is selected. If no fit solution is found, the simulation is stopped

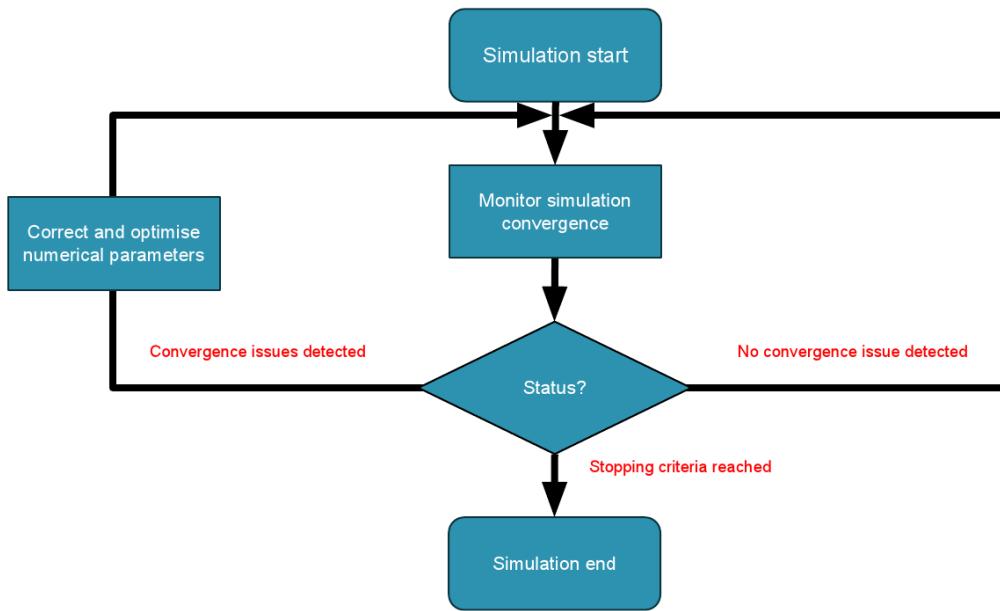


Figure 10.4: SArP Stability Mode Flowchart

A typical steady simulation fitness evaluation is shown in Figure 10.6. In that example the residuals exit condition is met by the variable represented with red disks and green crosses since their residuals drop below  $\gamma Res_{max}$  where  $\gamma$  is the user-defined residual drop factor. The exit condition is however not met by the variable represented with purple diamonds.

For unsteady cases, the EGA stops if:

- a solution for which the last time step has reached the target residual has been found;
- the best solution found has not been improved upon for  $n$  consecutive generations with  $n$  a user-defined parameter
- the user-defined maximum number of generations,  $n_{max}$ , is reached without the EGA convergence criterion being met. In that case if at least one solution has been found during the process, the fittest solution found at the time is selected. If no fit solution is found, the simulation is stopped.

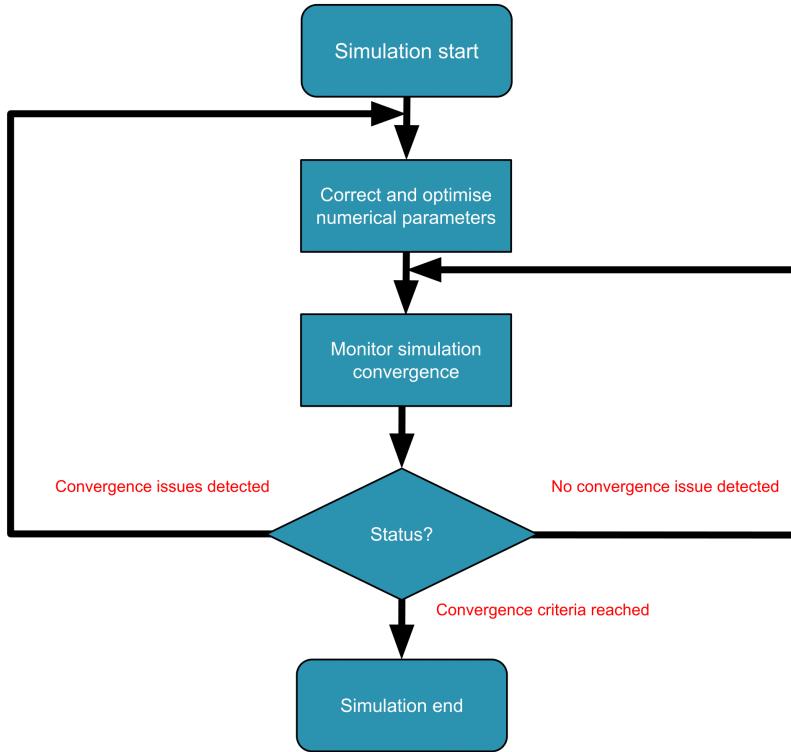


Figure 10.5: SArP Optimisation Mode Flowchart

### 10.2.3 Genetic Operations

Genetic operations are used to create new chromosomes based on an existing population. The genetic operations consist in applying chromosome crossover followed by mutation in each generation except the first one where the mutation operation is applied directly on the original set of parameters.

#### Mutation Operation

Mutation is only applied on a random number of new chromosomes. Mutation is then applied on a random set of genes, which size (ratio of number of genes in set to total number of genes in chromosome) is equal to the mutation rate. Thanks to mutation, the parameters space is more thoroughly explored which in turn prevents the algorithm from getting stuck in an area around a local minimum.

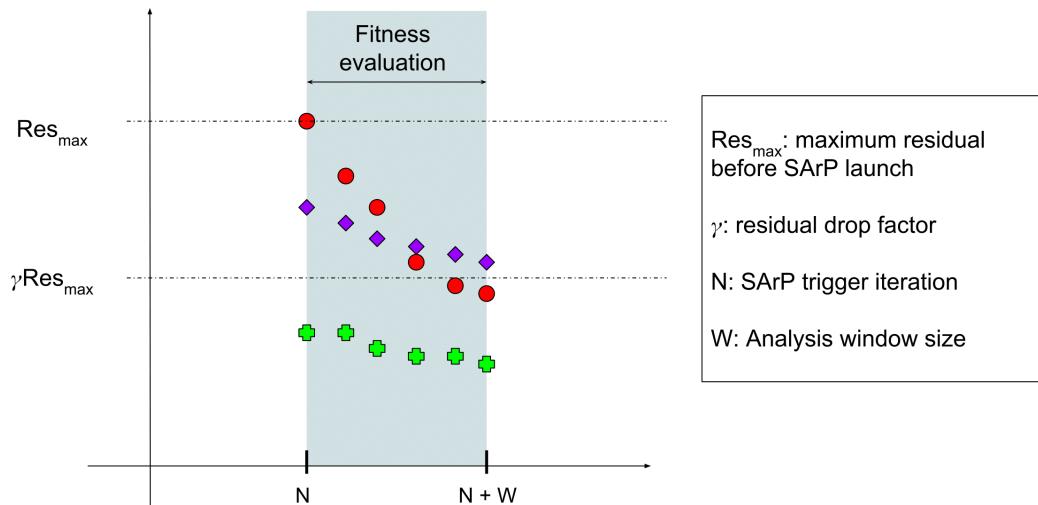


Figure 10.6: Steady Simulation Evaluation Example

There is a mutation operator for each type of parameter. They operate as follows:

- Numerical: get a random number from a Gaussian distribution centred on the current value and bounded by the minimum and maximum values allowed for the parameter
- Enumerate: select an item different than the current value in the list of possible choices
- Boolean: invert Boolean value

### Crossover Operation

The implemented crossover operation is based on a one-point crossover. According to the encoding method and the predefined physical rules, the overall consistency of the underlying solution and the validity of the individuals in the population must be maintained. To make sure that genes and combination of genes are within the possible ranges, an extra correction step is added in the breeding process of the EGA. These corrections allow avoiding crashes or unphysical solutions in [TransAT](#). In short, the corrections are akin to eliminating unfit chromosomes *a priori*. In cases where two individuals have the exact same chromosome, mutation is applied to the latter individual.

### 10.2.4 Selection Process

The basic part of the selection process is to stochastically select from one generation to create the basis of the next generation. The requirement is that the selection process should make the chances of survival of the fittest individuals greater than weaker ones. This replicates natural selection process in that fitter individuals will tend to have a better probability of survival and will go forward to form the mating pool for the next generation. Weaker individuals are not without a chance. In nature such individuals may have genetic coding that may prove to be useful for future generations.

#### Fitness function

In each generation, chromosomes are accordingly evaluated with [TransAT](#) to measure their fitness. Simulations are run for each chromosome over a window which size corresponds to a certain number of iterations or time-step depending on the simulation type. Residual values and convergence rates are analysed over the window to compute the fitness of the chromosome.

The fitness function indicates the fitness of a chromosome and serves as a metaheuristic to initiate the genetic algorithm. The multi-objective fitness function used to deduce the fitness of chromosomes is defined as a linear combination of simulation runtime efficiency and convergence quality. The distance to the target, quality of convergence and stability are the input variables to the fitness function.

Different indicators determining how good a simulation converges are isolated and weighted in the fitness function. In order to take the different weights of the indicators into account, the indicators are multiplied by weighted coefficients in the fitness function definition.

The stability and the quality of convergence at a point  $i$  (i.e. at a specific time-step or iteration depending on the simulation type), is measured with the least squares approximation method which is applied on the window  $[i - w, i]$ , with  $w$  the size of the window.

The fitness function reads:

$$\forall i \in [w, N] \quad f_i(d_i, c_i, e_i) = -\alpha_i |\log(d_i)| - \beta_i c_i - \gamma |\log(e_i)| \quad (10.2)$$

Where:

- $w$  : Size of the least square analysis windows
- $f$  : Fitness function
- $d$  : Distance of the current residual to the target

- $c$  : Convergence quality/rate (slope of the least square estimation)
- $e$  : Stability quality (mean error of least square estimation)
- $\alpha$ : Distance to the target coefficient
- $\beta$ : Convergence quality coefficient
- $\gamma$ : Stability quality coefficient
- $N$  : Number of iterations

For stationary simulations, the fitness function is exactly as defined in Eq. 10.2. For transient simulations, two extra terms are appended to the fitness function definition in Eq. 10.2. One is related to the convergence of the last time step and the other to the duration of time steps.

For steady cases, the following sequence of filtering is performed to select the fittest chromosome:

1. Select the chromosomes that reached the target residuals if any
2. Select the chromosomes that do not trigger SArP if the set created in 1 is empty
3. Select the chromosomes with the lowest runtime out of the set created in 2
4. Select the chromosome with the highest fitness at the last iteration out of the set created in 3

For unsteady test cases, the selection process is different than with steady test cases. The selection is done sequentially:

1. Select the chromosomes with converged time steps
2. Select the chromosomes that do not trigger SArP out of the set created in 1
3. Select the chromosomes with the highest time step to runtime ratio out of the set created in 2
4. Select the chromosome with the highest fitness at the last time step out of the set created in 3

## Selection Operators

To extract the fittest individuals, different selection strategies from the DEAP module ([Fortin et al. \(2012\)](#)) are implemented:

- **elitism**: the N (population size) fittest elements among the chromosomes of the current generation and the previous ones are kept;
- **tournament**: the best set of parameters is chosen after applying selective pressure to the different sets of parameters by evaluating the best set of parameters through a tournament between N (population size) individuals. The best set of parameters extracted using this method is the winner of the tournament i.e. the set of parameters with the highest score
- **roulette**: analogous to a roulette wheel where each set of parameters is associated with a slice of the wheel that is proportional in size to its fitness score. The size of the slices is set in such a way that the selection for the next generation is biased towards the current best chromosome.
- **nsga2**: non-dominated sorting genetic algorithm which can be categorised as a Multiple Objective Optimization (MOO) algorithm. It improves the adaptive fit of a population of candidate solutions to a Pareto front constrained by a set of objective functions ([Deb et al. \(2000\)](#)). It uses an evolutionary process with surrogates for evolutionary operators. The population is sorted into a hierarchy of sub-populations based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions
- **spea2**: Strength Pareto Evolutionary Algorithm is a Multiple Objective Optimization algorithm. It locates and maintains a front of non-dominated solutions, ideally a set of Pareto optimal solutions ([Zitzler et al. \(2001\)](#)). This is achieved by using an evolutionary process to explore the search space, and a selection process that uses a combination of the degree to which a candidate solution is dominated (strength) and an estimation of density of the Pareto front as an assigned fitness. An archive of the non-dominated set is maintained separate from the population of candidate solutions used in the evolutionary process, providing a form of elitism

The **elitism** strategy always enhances the fitness of the individuals and guarantees that the fittest solution is constantly selected for the next generation. In the selection process, the **elitism** strategy is applied alongside a selection operator set by the user among the ones above in each generation to add randomness to the algorithm. This hybrid solution combines the advantages of the two employed strategies: favouring the fittest solution and deep exploration of the search space.

### 10.2.5 Prediction and Fitness approximation

In theory, the fitness of a chromosome is obtained by evaluating the fitness function which means running a simulation. Bearing in mind the number of simulations required to run to evaluate the fitness function (one for each chromosome produced during the EGA process), running the EGA can be computationally costly.

To offset the potential excessive cost of the EGA, estimation models predicting the fitness based on past fitness evaluations implemented in the SArP module can be used. Instead of letting the EGA progress naturally – i.e. by evaluating the fitness with a [TransAT](#) run for every chromosome – the rules of evolution are redefined by integrating an associative memory that evaluates the fitness of a chromosome by predicting it based on past evaluations. The full flow chart of the EGA process with prediction enabled is depicted in Figure 10.7.

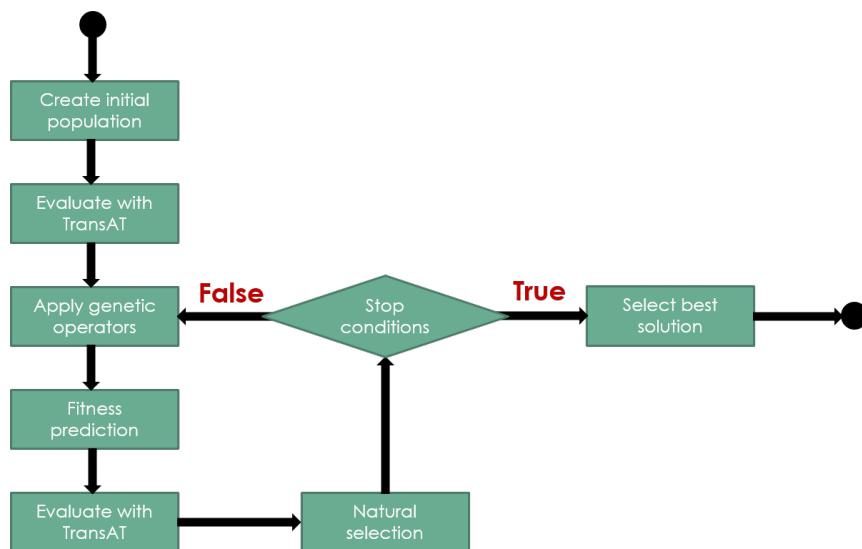


Figure 10.7: Genetic algorithm flowchart

The aim is to select more fitting individuals to be eventually evaluated with [TransAT](#). Such a bias in the evolution process gives rise to a self-adapting strategy and decreases the number of calls to the original fitness function which is expensive. Such approaches have been the focus of several academic studies which have shown that they can improve the computational performances.

The proposed solution in the SArP module is based on machine learning techniques such as non-linear regression models and decision trees.

## Strategy

The idea, as detailed in the previous diagram, is to launch the EGA in order to feed the machine learning algorithm with its output data. Indeed, the data used to feed the machine learning algorithms comes from a local database that is expanded by runs of the SArP module with fitness prediction enabled. When fitness prediction is enabled, fitness evaluations of chromosomes evaluated during the EGA are added to that local database.

If the database is large enough, it can be used to refine the machine-learning models. When prediction is enabled, the fitness of the chromosomes of a generation is first estimated using the prediction algorithms of machine-learning models. A set of half the population size consisting of the fittest elements is then extracted from the population to be evaluated using [TransAT](#). This way, the number of calls to the [TransAT](#) solver is significantly reduced.

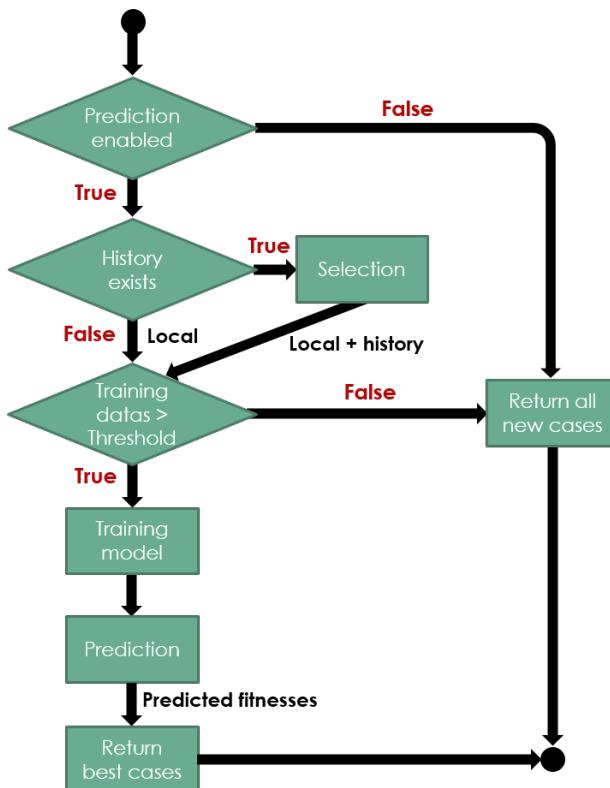


Figure 10.8: Predictive evaluation flowchart

As shown in Figure 10.8, the SArP module executes the following steps when prediction is enabled:

1. Verify whether data history exists for this particular simulation or if only the chromosomes created during the current EGA run should be relied on
2. Once the appropriate sample of data is located, proceed to selecting chromosomes that are similar to the current chromosome in terms of state and progress level. The selection is solely applied on data from chromosomes that are in the same phase, progress or state as the current chromosome for which the fitness is to be predicted
3. Check that the training samples size complies with the learning threshold. Learning thresholds were defined empirically from the study of the different models performances in varying the size of the training sample and analysing the results. As a result, minimum thresholds were fixed for each model in order to get satisfactory prediction results. If the training sample selected after a cross-validation process is not large enough to get satisfactory model performances, the fitness prediction model is not applied and the EGA proceeds to evaluate all the chromosomes
4. Train the machine learning algorithm on the selected data to generate the prediction model
5. The fitness of the chromosomes of the current generation are predicted based on the applied model
6. Return the chromosomes with the best predicted fitness to be passed on to the next generation

## Learning Methods

Two types of machine-learning models were implemented: classifiers and regressors.

The former type of model has a binary behaviour. Chromosomes can be either fit or unfit depending on their fitness. These chromosome classes are uniformly distributed within the data from the database. This classification method is an efficient way to eliminate cases that are quickly diverging. Although this model cannot order two solutions in the same class, it remains faster than regressors.

With regressors, regression is used to predict the actual value of the fitness value. It gives more accurate results than with classifiers. The estimation of the fitness function as a continuous function allows the differentiation and ordering of chromosomes. Non-linear regression models are used to handle the complexity of the problems at hand.

The list of implemented algorithms, based on the scikit-learn API ([Buitinck \*et al.\* \(2013\)](#)), in the SarP module is as follows:

- **Decision trees** classifier: a non-parametric supervised learning method used for classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features;
- **Bagging** regressor: an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregates their individual predictions (either by voting or by averaging) to form a final prediction. It improves the stability and accuracy of machine learning algorithms and reduces the variance and helps to avoid overfitting;
- **Extra trees** regressor: a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting;
- **Random forest** regressor: a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size.

### 10.2.6 Speeding up the EGA

The main strategy to speed up the EGA is to reduce the number of evaluations performed during the correction and optimisation process. There are three main options to do this:

- Option 1: adjust the genetic algorithm parameters by:
  - reducing the analysis window size
  - reducing the number of evaluations per generation
  - reducing the number of generations
- Option 2: adjust the genetic algorithm convergence criteria by
  - reducing the stagnating generation criterion
  - increasing the residual drop factor (only for steady cases)
- Option 3: enable fitness approximation

Note that all three options can be coupled as they are independent from one another.

Alternatively, the EGA can be stopped in the middle of the process if a satisfactory solution has been found. This can be done by clicking **.** in the **Execute** tab of **TransATUI** or creating the file **transat\_mb.rti** and setting its content to **stop\_sarp** (only one word in the file).



# Bibliography

- BECKERMANN, C., DIEPERS, H.-J., STEINBACH, I., KARMA, A. & TONG, X. 1999 Modeling melt convection in phase-field simulations of solidification. *J. Comp. Phys.* **154**, 468–496.
- BUITINCK, L., LOUPPE, G., BLONDEL, M., PEDREGOSA, F., MUELLER, A., GRISEL, O., NICULAE, V., PRETTENHOFER, P., GRAMFORT, A., GROBLER, J., LAYTON, R., VANDERPLAS, J., JOLY, A., HOLT, B. & VAROQUAUX, G. 2013 API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.
- COOKE, C. H. & CHEN, T. 1992 On shock capturing for pure water with general equation of state. *Comm. Appl. Num. Meth.* **8**, 219–233.
- DEB, K., AGRAWAL, S., PRATAP, A. & MEYARIVAN, T. 2000 *A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II*, pp. 849–858. Berlin, Heidelberg: Springer Berlin Heidelberg.
- VAN DOORMAAL, J. & RAITHBY, G. D. 1984 Enhancements of the simple method for predicting incompressible fluid flows. *Num. Heat Transfer* **7**, 147–163.
- FORTIN, F.-A., DE RAINVILLE, F.-M., GARDNER, M.-A., PARIZEAU, M. & GAGNÉ, C. 2012 DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research* **13**, 2171–2175.
- FOX, R. 2003 *Computational Models for Turbulent Reacting Flows*. Cambridge University Press.
- KARKI, K. & PATANKAR, S. 1989 Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations. *AIAA J.* **27**, 1167–1178.
- KHOSLA, P. K. & RUBIN, S. G. 1974 A diagonally dominant second-order accurate implicit scheme. *Computers & Fluids* **2**, 207–209.
- LAUNDER, B. & SPALDING, D. 1974 The numerical computation of turbulent flows. *Comp. Meth. Appl. Mech. Eng.* **3**, 269–289.

- LEONARD, B. P. 1979 A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comp. Meth. Appl. Mech. Eng.* **19**, 59–98.
- MICHELASSI, V., THEODORIDIS, G. & PAPANICOULAOU, E. 1994 Comparison of time-marching and pressure-correction solvers for compressible flows – part I: low-speed formulation. In *ASME 1995 International Mechanical Engineering Congress and Exposition*. San Francisco, CA, USA.
- MITTAL, R. & IACCARINO, G. 2005 Immersed boundary methods. *Ann. Review Fluid Mech.* **37**, 239–261.
- PATANKAR, S. V. 1980 *Numerical heat transfer and fluid flow*. New York, USA: Hemisphere.
- PETERS, N. 2000 *Turbulent Combustion*. Cambridge University Press.
- POINSOT, T. & VEYNANTE, D. 2001 *Theoretical and Numerical Combustion*. R. T. Edwards Inc.
- R.J. KEE, M.E. COLTRIN, P. G. 2003 *Chemically reacting flow*. John Wiley and Sons.
- SPALDING, D. B. 1972 A novel finite difference formulation for differential expressions involving both first and second derivatives. *Int. J. Num. Meth. Eng.* **4**, 551–559.
- STONE, H. L. 1968 Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM J. Num. Anal.* **5**, 530–558.
- WILLIAMS, F. A. 1958 Spray combustion and atomization. *Phys. Fluids* **1**, 541.
- ZHU, J. 1991 A low diffusive and oscillation-free convection scheme. *Comm. Appl. Num. Meth.* **7**, 225–232.
- ZHU, J. 1992 An introduction and guide to the computer program fast-3d. *Tech. Rep.* 691. Institut for Hydromechanics, University of Karlsruhe, Germany.
- ZITZLER, E., LAUMANNS, M. & THIELE, L. 2001 Spea2: Improving the strength pareto evolutionary algorithm. *Tech. Rep.*.