TECHNICAL REFERENCE
SOLIDWORKS FLOW SIMULATION 2021
EN

Table of Contents

Chapter 3
Boundary Conditions and Engineering Devices 85

Chapter 4
Numerical Solution Technique 95

Chapter 5
Mesh Settings 111

Chapter 6

Chapter 7

List of Figures

List of Tables

Chapter 1
Physical Capabilities of Flow Simulation

SOLIDWORKS Flow Simulation is a software based on SOLIDWORKS for computing fluid (gas or liquid) flows inside and outside SOLIDWORKS models, as well as heat transfer to (from, between, in) these models due to convection, radiation, and conduction with a proved computational fluid dynamics (CFD) technology.

With Flow Simulation it is possible to study a wide range of fluid flow and heat transfer phenomena that include the following:

- External and internal fluid flows;

- Steady-state and time-dependent fluid flows;

- Compressible gas and incompressible fluid flows;

- Subsonic, transonic, and supersonic gas flows;

- Free, forced, and mixed convection;

- Fluid flows with boundary layers, including wall roughness effects;

- Laminar and turbulent fluid flows;

- Multi-species fluids and multi-component solids;

- Fluid flows in models with moving/rotating surfaces and/or parts;

- Heat conduction in fluid, solid and porous media with/without conjugate heat transfer and/or contact heat resistance between solids and/or radiation heat transfer between opaque solids (some solids can be considered transparent for radiation), and/or volume (or surface) heat sources, e.g. due to Peltier effect, etc.

- Joule heating due to direct electric current in electrically conducting solids;[1]

- Various types of thermal conductivity in solid medium, i.e. isotropic, unidirectional, biaxial/axisymmetrical, and orthotropic;

- Fluid flows and heat transfer in porous media;

- Flows of non-Newtonian liquids;

- Flows of compressible liquids;

- Real gases;

---

1. This feature is available for the Electronics Cooling module users only.

- Cavitation in incompressible liquid flows;

- Equilibrium volume condensation of water from steam and its influence on fluid flow and heat transfer;

- Relative humidity in gases and mixtures of gases;

- Two-phase (fluid + particles) flows;

- Periodic boundary conditions.

Chapter 2
Governing Equations

This chapter provides the theoretical background for the basic governing equations for a wide range of incompressible and compressible, laminar and turbulent fluid flows and for transport phenomena such as heat transfer and chemical reactions.

The Navier-Stokes Equations for Laminar and Turbulent Fluid Flows

 Flow Simulation solves the Navier-Stokes equations, which are formulations of mass, momentum and energy conservation laws for fluid flows. The equations are supplemented by fluid state equations defining the nature of the fluid, and by empirical dependencies of fluid density, viscosity and thermal conductivity on temperature. Inelastic non-Newtonian fluids are considered by introducing a dependency of their dynamic viscosity on flow shear rate and temperature, and compressible liquids are considered by introducing a dependency of their density on pressure. A particular problem is finally specified by the definition of its geometry, boundary and initial conditions.

Flow Simulation is capable of predicting both *laminar* and *turbulent* flows. Laminar flows occur at low values of the Reynolds number, which is defined as the product of representative scales of velocity and length divided by the kinematic viscosity. When the Reynolds number exceeds a certain critical value, the flow becomes turbulent, i.e. flow parameters start to fluctuate randomly.

Most of the fluid flows encountered in engineering practice are turbulent, so Flow Simulation was mainly developed to simulate and study turbulent flows. To predict turbulent flows, the *Favre-averaged Navier-Stokes equations* are used, where time-averaged effects of the flow turbulence on the flow parameters are considered, whereas the other, i.e. large-scale, time-dependent phenomena are taken into account directly. Through this procedure, extra terms known as the Reynolds stresses appear in the equations for which additional information must be provided. To close this system of equations, Flow Simulation employs transport equations for the turbulent kinetic energy and its dissipation rate, the so-called *k-ε* model.

 Flow Simulation employs one system of equations to describe both laminar and turbulent flows. Moreover, transition from a laminar to turbulent state and/or vice versa is possible.

Flows in models with moving walls (without changing the model geometry) are computed by specifying the corresponding boundary conditions. Flows in models with rotating parts are computed in coordinate systems attached to the models rotating parts, i.e. rotating with them, so the models' stationary parts must be axisymmetric with respect to the rotation axis.

The conservation laws for mass, angular momentum and energy in the Cartesian coordinate system rotating with angular velocity Ω about an axis passing through the coordinate system's origin can be written in the conservation form as follows:

(2-1)

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho u_i\right) = S_M^p$$

$$(2\text{-}2)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_i u_j\right) + \frac{\partial p}{\partial x_i} = \frac{\partial}{\partial x_j}\left(\tau_{ij} + \tau_{ij}^R\right) + S_i + S_{Ii}^p, \quad i = 1,2,3;\ j = 1,2,3$$

$$(2\text{-}3)$$

$$\frac{\partial \rho H}{\partial t} + \frac{\partial \rho u_i H}{\partial x_i} = \frac{\partial}{\partial x_i}\left(u_j(\tau_{ij} + \tau_{ij}^R) + q_i\right) + \frac{\partial p}{\partial t} - \tau_{ij}^R \frac{\partial u_i}{\partial x_j} + \rho\varepsilon + S_i u_i + S_H^p + Q_H$$

$$H = h + \frac{u^2}{2} + \frac{5}{3}k - \frac{\Omega^2 r^2}{2} - \sum_m h_m^0 y_m$$

where u is the fluid velocity, ρ is the fluid density, S_i is a mass-distributed external force per unit mass due to a porous media resistance (S_i^{porous}), a gravity ($S_i^{gravity} = \rho g_i$, where g_i is the gravitational acceleration component along the i-th coordinate direction), and the coordinate system's rotation ($S_i^{rotation}$), i.e., $S_i = S_i^{porous} + S_i^{gravity} + S_i^{rotation}$, h is the thermal enthalpy; S_M^p, S_{Ii}^p, S_H^p are additional interfacial exchange terms due to Euler-Lagrange particle interaction; Q_H is a heat source or sink per unit volume, τ_{ij} is the viscous shear stress tensor, q_i is the diffusive heat flux, Ω is an angular velocity of the rotating coordinate system, r is the distance from a point to rotation axis in rotation reference frame, k is kinetic energy of turbulence, h_m^0 is an individual thermal enthalpy of the m-th mixture component, y_m is a concentration of the m-th mixture component and it is described by the species conservation equation 2-18 (see "Mass Transfer in Fluid Mixtures" on page 18). The subscripts are used to denote summation over the three coordinate directions.

For calculations with the **High Mach number flow** option enabled, the following energy equation is used:

$$(2\text{-}4)$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho u_i\left(E + \frac{p}{\rho}\right)}{\partial x_i} = \frac{\partial}{\partial x_i}\left(u_j(\tau_{ij} + \tau_{ij}^R) + q_i\right) - \tau_{ij}^R \frac{\partial u_i}{\partial x_j} + \rho\varepsilon + Q_H$$

$$E = e + \frac{u^2}{2}$$

where e is the internal energy.

For Newtonian fluids the viscous shear stress tensor is defined as:

$$(2\text{-}5)$$

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

Following Boussinesq assumption, the Reynolds-stress tensor has the following form:

$$(2\text{-}6)$$

$$\tau_{ij}^R = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) - \frac{2}{3} \rho k \delta_{ij}$$

Here δ_{ij} is the Kronecker delta function (it is equal to unity when $i = j$, and zero otherwise), μ is the dynamic viscosity coefficient, μ_t is the turbulent eddy viscosity coefficient and k is the turbulent kinetic energy. Note that mt and k are zero for laminar flows. In the frame of the k-ε turbulence model, μ_t is defined using two basic turbulence properties, namely, the turbulent kinetic energy k and the turbulent dissipation ε,

$$(2\text{-}7)$$

$$\mu_t = f_\mu \frac{C_\mu \rho k^2}{\varepsilon}$$

Here f_μ is a turbulent viscosity factor. It is defined by the expression:

$$(2\text{-}8)$$

$$f_\mu = \left[1 - \exp\left(-0.0165 R_y \right) \right]^2 \cdot \left(1 + \frac{20.5}{R_T} \right)$$

where

$$R_T = \frac{\rho k^2}{\mu \varepsilon}, \qquad R_y = \frac{\rho \sqrt{k} y}{\mu}$$

and y is the distance from the wall. This function allows us to take into account laminar-turbulent transition.

Two additional transport equations are used to describe the turbulent kinetic energy and dissipation,

$$(2\text{-}9)$$

$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_i} \left(\rho u_i k \right) = \frac{\partial}{\partial x_i} \left(\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right) + S_k$$

$$(2\text{-}10)$$

$$\frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial}{\partial x_i} \left(\rho u_i \varepsilon \right) = \frac{\partial}{\partial x_i} \left(\left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_i} \right) + S_\varepsilon$$

where the source terms S_k and S_ε are defined as:

$$(2-11)$$

$$S_k = \tau_{ij}^R \frac{\partial u_i}{\partial x_j} - \rho\varepsilon + \mu_t P_B$$

$$(2-12)$$

$$S_\varepsilon = C_{\varepsilon_1} \frac{\varepsilon}{k} \left(f_1 \tau_{ij}^R \frac{\partial u_i}{\partial x_j} + \mu_t C_B P_B \right) - C_{\varepsilon_2} f_2 \frac{\rho\varepsilon^2}{k}$$

Here P_B represents the turbulent generation due to buoyancy forces and can be written as:

$$(2-13)$$

$$P_B = -\frac{g_i}{\sigma_B} \frac{1}{\rho} \frac{\partial\rho}{\partial x_i}$$

where g_i is the component of gravitational acceleration in direction x_i, the constant $\sigma_B = 0.9$, and constant C_B is defined as: $C_B = 1$ when $P_B > 0$, and 0 otherwise;

$$(2-14)$$

$$f_1 = 1 + \left(\frac{0.05}{f_\mu} \right)^3, \quad f_2 = 1 - \exp\left(-R_T^2 \right)$$

The constants C_μ, $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_k, σ_ε are defined empirically. In Flow Simulation the following typical values are used:

$$(2-15)$$

$$C_\mu = 0.09, C_{\varepsilon 1} = 1.44, C_{\varepsilon 2} = 1.92, \sigma_\varepsilon = 1.3, \sigma_k = 1$$

Where Lewis number $Le = 1$ the diffusive heat flux is defined as:

$$(2-16)$$

$$q_i = \left(\frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\sigma_c} \right) \frac{\partial h}{\partial x_i}, \quad i = 1,2,3$$

Here the constant $\sigma_c = 0.9$, Pr is the Prandtl number, and h is the thermal enthalpy.

These equations describe both laminar and turbulent flows. Moreover, transitions from one case to another and back are possible. The parameters k and μ_t are zero for purely laminar flows.

Laminar/turbulent Boundary Layer Model

Laminar/turbulent Boundary Layer Model

A laminar/turbulent boundary layer model is used to describe flows in near-wall regions. The model is based on the so-called *Modified Wall Functions* approach. This model is employed to characterize laminar and turbulent flows near the walls, and to describe transitions from laminar to turbulent flow and vice versa. The modified wall function uses a Van Driest's profile instead of a logarithmic profile. If the size of the mesh cell near the wall is more than the boundary layer thickness the integral boundary layer technology is used. The model provides accurate velocity and temperature boundary conditions for the above mentioned conservation equations.

Mass Transfer in Fluid Mixtures

The mass transfer in fluid mixtures is governed by species conservation equations.

The equations that describe concentrations of mixture components can be written as:

(2-17)

$$\frac{\partial \rho y_m}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i y_m) = \frac{\partial}{\partial x_i}\left(\left(D_{mn} + D_{mn}^t\right)\frac{\partial y_m}{\partial x_i}\right) + S_m,$$

$$m = 1,2,..., M$$

Here D_{mn}, D_{mn}^t are the molecular and turbulent matrices of diffusion, S_m is the rate of production or consumption of the m-th component.

In case of Fick's diffusion law:

(2-18)

$$D_{mn} = D \cdot \delta_{mn}, \quad D_{mn}^t = \delta_{mn} \cdot \frac{\mu_t}{\sigma}$$

The following obvious algebraic relation between species concentrations takes place:

(2-19)

$$\sum_m y_m = 1$$

Constitutive Laws and Thermophysical Properties

The system of Navier-Stokes equations is supplemented by definitions of thermophysical properties and state equations for the fluids. Flow Simulation provides simulations of gas and liquid flows with density, viscosity, thermal conductivity, specific heats, and species diffusivities as functions of pressure, temperature and species concentrations in fluid mixtures, as well as equilibrium volume condensation of water from steam can be taken into account when simulating steam flows.

Generally, the state equation of a fluid has the following form:

(2-20)

$$\rho = f(P, T, \mathbf{y})$$

where $y = (y_1, ..., y_n)$ is the concentration vector of the fluid mixture components.

Excluding special cases (see below subsections concerning Real Gases, Water vapor condensation and relative humidity), **Gases** are considered ideal, i.e. having the state equation of the form:

(2-21)

$$\rho = \frac{P}{RT}$$

where R is the gas constant which is equal to the universal gas constant R_{univ} divided by the fluid molecular mass M, or, for the mixtures of ideal gases,

(2-22)

$$R = R_{univ} \sum_i \frac{y_i}{M_i}$$

where y_i, $i = 1, 2, ..., n$, are the mass fractions of mixture components, and M_i is the molecular mass of the i-th component.

Specific heat at constant pressure, as well as the thermophysical properties of the gases, i.e. viscosity and thermal conductivity, are specified as functions of temperature. In addition, proceeding from the state equation 2-21, each of such gases has constant specific heat ratio C_p/C_v.

Excluding special cases (see below subsections Compressible Liquids, Non-Newtonian Liquids), liquids are considered incompressible, i.e. the density of an individual liquid depends only on temperature:

(2-23)

$$\rho = f(T)$$

and the state equation for a mixture of liquids is defined as

(2-24)

$$\rho = \left(\sum_i \frac{y_i}{\rho_i} \right)^{-1}$$

In Flow Simulation the specific volume, i.e. reciprocal to the density, is used in defining the state of the liquid.

The specific heat and the thermophysical properties of the liquid (i.e. viscosity and thermal conductivity), are specified as functions of temperature.

The specific heat at constant pressure for a mixture of liquids is defined as:

(2-25)

$$C_p = \sum_i y_i C_{pi}$$

where C_{pi} is the specific heat of the i-th component.:

The dynamic viscosity for a mixture of liquids is defined as

(2-26)

$$\frac{\mu}{\rho} = \sum_i y_i^v \frac{\mu_i}{\rho_i}$$

where y_i^v is the volume fraction, and μ_i is the dynamic viscosity of the i-th component.

The thermal conductivity for a mixture of liquids is defined as:

(2-27)

$$k = \left(\sum_i y_i k_i^{-2} \right)^{-1/2}$$

where k_i is the thermal conductivity of the i-th component.

Real Gases

Flow Simulation is capable of simulating the behavior of real gases in a broad range of parameters, including both sub- and supercritical regions. If the considered gas may happen to get near the gas-liquid phase transition or above the critical point (i.e. becomes supercritical fluid), so that due to the increased intermolecular forces the ideal gas state equation can no longer describe the gas behavior properly (e.g. the Joule-Thomson effect), the real gas option should be switched on, so that the real gas state equations will be employed instead.

The state equation of ideal gas equation 2-21 (see "Constitutive Laws and Thermophysical Properties" on page 19) become inaccurate at high pressures or in close vicinity of the gas-liquid phase transition curve. Taking this into account, a real gas state equation together with the related equations for thermodynamical and thermophysical properties should be employed in such conditions. At present, this option may be used only for a single gas, probably mixed with ideal gases.

In case of user-defined real gas, Flow Simulation uses a custom modification of the Redlich-Kwong equation, that may be expressed in dimensionless form as follows:

(2-28)

$$P_r = T_r \cdot \left(\frac{1}{\Phi_r - b} - \frac{a \cdot F}{\Phi_r \cdot (\Phi_r + c)} \right)$$

where $P_r = P/P_c$, $T_r = T/T_c$, $\Phi_r = V_r \cdot Z_c$, $V_r = V/V_c$, $F = T_r$-1.5, P_c, T_c, and V_c are the user-specified critical parameters of the gas, i.e. pressure, temperature, and specific volume at the critical point, and Z_c is the gas compressibility factor that also defines the a, b, and c constants.

A particular case of the equation 2-28 with $Z_c = 1/3$ (which in turn means that $b = c$) is the original Redlich equation as given in Ref. 13.

Alternatively, one of the modifications (Ref. 13) taking into account the dependence of F on temperature and the Pitzer acentricity factor ω may be used:

- the Wilson modification:

(2-29)

$$F = 1 + (1.57 + 1.62\omega)(T_r^{-1} - 1)$$

- the Barnes-King modification:

(2-30)

$$F = 1 + (0.9 + 1.21\omega)(T_r^{-1.5} - 1)$$

- the Soave modification:

(2-31)

$$F = \frac{1}{T_r}\left[1 + (0.48 + 1.574\omega + 0.176\omega^2)(1 - T_r^{0.5})\right]^2$$

The specific heat of real gas at constant pressure (C_p) is determined as the combination of the user-specified temperature-dependent '*ideal gas*' specific heat (C_p^{ideal}) and the automatically calculated correction. The former is a polynomial with user-specified order and coefficients. The specific heat at constant volume (C_v) is calculated from C_p by means of the state equation.

Likewise, the thermophysical properties are defined as a sum of user-specified '*basic*' temperature dependency (which describes the corresponding property in extreme case of low pressure) and the pressure-dependent correction which is calculated automatically.

The basic dependency for dynamic viscosity μ of the gas is specified in a power-law form:

$$\mu = a \cdot T^n$$

The same property for liquid is specified either in a similar power-law form or in an exponential form:

$$\mu = 10^{a(1/T - 1/n)}$$

As for the correction, it is given according to the Jossi-Stiel-Thodos equation for non-polar gases or the Stiel-Thodos equations for polar gases (see Ref. 13), judging by the user-specified attribute of polarity. The basic dependencies for thermal conductivities k of the substance in gaseous and liquid states are specified by the user either in linear form:

$$k = a + n \cdot T$$

or in power-law form:

$$k = a \cdot T^n$$

and the correction is determined from the Stiel-Thodos equations (see Ref. 13).

All user-specified coefficients must be entered in SI unit system, except those for the exponential form of dynamic viscosity of the liquid, which should be taken exclusively from Ref. 13. These equations predict the user-defined real gases properties accurately in the specified range of parameters, including both sub- and supercritical regions (at pressure up to $1.1 \cdot P_c$).

In case of pre-defined real gas, the custom modification of the Redlich-Kwong equation of the same form as equation 2-28 is used, with the distinction that the coefficients a, b, and c are specified explicitly as dependencies on T_r in order to reproduce the gas-liquid phase transition curve at $P < P_c$ and the critical isochore at $P > P_c$ with higher precision:

(2-32)

$$P_r = T_r \cdot \left(\frac{1}{\Phi_r - b} - \frac{a}{\Phi_r \cdot (\Phi_r + c)} \right)$$

where $P_r = P/P_c$, $T_r = T/T_c$, $\Phi_r = V \cdot P_c/(R \cdot T_c)$ are the reduced pressure, temperature, and specific volume respectively, V is the molar volume, P_c and T_c are the critical pressure and temperature respectively, and R is the gas constant. This equation predicts the pre-defined real gases properties accurately in the specified range of parameters, including both sub- and supercritical regions (at pressure up to $2 \cdot P_c$).

When the calculated (p, T) point drops out of the region bounded by the temperature and pressure limits (zones 1 - 8 on Figure 2-1) or gets beyond the gas-liquid phase transition curve (zone 9 on Figure 2-1), the corresponding warnings are issued and properties of the real gas are extrapolated linearly.

**Figure 2-1. Pressure-temperature phase diagram
(blue zone = liquid, green zone = vapor, yellow zone = supercritical).**
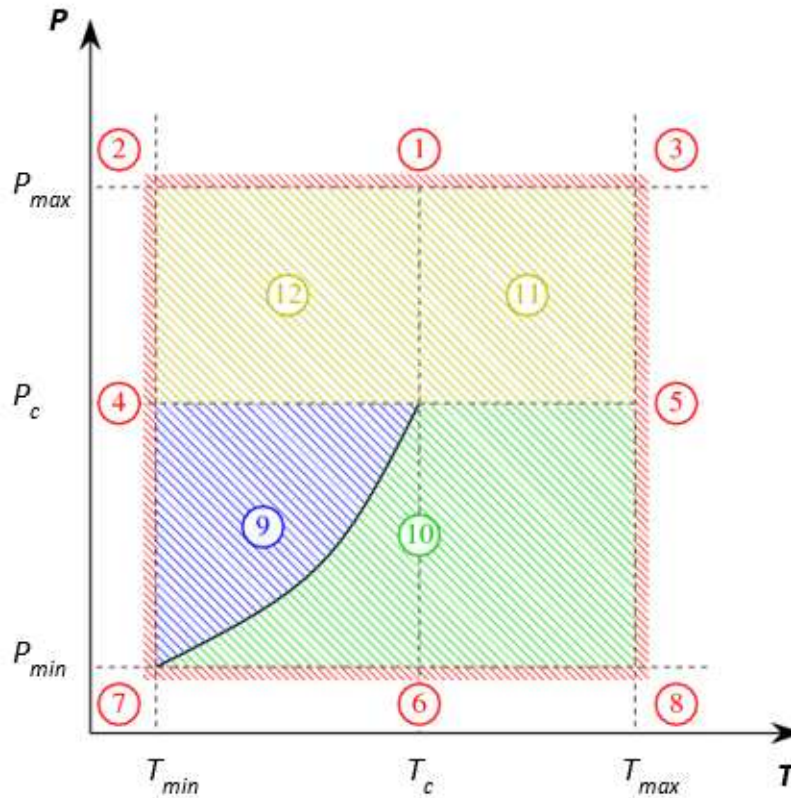
If a real gas mixes with ideal gases (at present, mixtures of multiple real gases are not considered), properties of this mixture are determined as an average weighted with mass or volume fractions:

$$(2\text{-}33)$$

$$v = \sum_{i=1}^{N} y_i v_i$$

where v is the mixture property (i.e., C_p, μ, or k), N is the total number of the mixture gases (one of which is real and others are ideal), y_i is the mass fraction (when calculating C_p) or the volume fraction (when calculating μ and k) of the i-th gas in the mixture.

The real gas model has the following limitations and assumptions:

- The precision of calculation of thermodynamic properties at nearly-critical temperatures and supercritical pressures may be lowered to some extent in comparison to other temperature ranges. Generally speaking, the calculations involving user-defined real gases at supercritical pressures are not recommended.

- The user-defined dependencies describing the specific heat and transport properties of the user-defined real gases should be applicable in the whole $T_{min}...T_{max}$ range (or, speaking about liquid, in the whole temperature range where the liquid exists).

- T_{min} for user-defined real gas should be set at least 5...10 K higher than the triple point of the substance in question.

Compressible Liquids

Compressible liquids whose density depends on pressure and temperature can be considered within the following approximations:

- obeying the logarithmic law:

(2-34)

$$\rho = \rho_0 \left/ \left(1 - C \cdot \ln \frac{B+P}{B+P_0} \right) \right.$$

where ρ_0 is the liquid's density under the reference pressure P_0, C, B are coefficients, here ρ_0, C, and B can depend on temperature, P is the calculated pressure;

- obeying the power law:

(2-35)

$$\rho = \rho_0 \cdot \left(\frac{P+B}{P_0+B} \right)^{1/n}$$

where, in addition to the above-mentioned designations, n is a power index which can be temperature dependent.

Also if the liquid dynamic viscosity depends on pressure, it can be considered within the following approximation:

(2-36)

$$\mu = \mu_0 e^{a(P-P')}$$

where, in addition to the above-mentioned designations, μ_0 is the liquid's dynamic viscosity under the reference pressure P_0, a is a coefficient which can be temperature dependent and $P' = 10^5$ Pa is a constant.

Non-Newtonian Liquids

Flow Simulation is capable of computing laminar flows of inelastic non-Newtonian liquids.

In this case the viscous shear stress tensor is defined, instead of equation 2-5 (see "The Navier-Stokes Equations for Laminar and Turbulent Fluid Flows" on page 14), as:

(2-37)

$$\tau_{ij} = \mu(\dot{\gamma}) \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

where shear rate,

(2-38)

$$\dot{\gamma} = \sqrt{d_{ij}^2 - d_{ii} \cdot d_{jj}} \quad , \quad d_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}$$

and for specifying a viscosity function $\mu(\dot{\gamma})$ the following five models of inelastic non-Newtonian viscous liquids are available in Flow Simulation:

- **Herschel-Bulkley model**:

(2-39)

$$\mu(\dot{\gamma}) = K \cdot (\dot{\gamma})^{n-1} + \frac{\tau_o}{\dot{\gamma}}$$

where K is the liquid's consistency coefficient, n is the liquid's power law index, and τ_0 is the liquid's yield stress. This model includes the following special cases:

- $n = 1$, $\tau_0 = 0$ describes Newtonian liquids, in this case K is the liquid's dynamic viscosity;

- $n = 1$, $\tau_0 > 0$ describes the *Bingham model* of non-Newtonian liquids, featured by a non-zero threshold yield stress (τ_0) below of which the liquid behaves as a solid, so to achieve a flow this threshold shear stress must be exceeded. (In Flow Simulation this threshold is modeled by automatically equating K, named plastic viscosity in this case, to a substantially high value at $\tau < \tau_0$);

- $0 < n < 1$, $\tau_0 = 0$ describes the power law model of '*shear-thinning*' non-Newtonian liquids (see also below);

- $n > 1$, $\tau_0 = 0$ describes the power law model of '*shear-thickening*' non-Newtonian liquids (see also below);

- **Power-law model**:

(2-40)

$$\mu(\dot{\gamma}) = K \cdot (\dot{\gamma})^{n-1}$$

in contrast to the Herschel-Bulkley model's special case, the μ values are restricted: $\mu_{min} \leq \mu \leq \mu_{max}$;

- **Carreau model**:

(2-41)

$$\mu = \mu_\infty + (\mu_o - \mu_\infty) \cdot \left[1 + (K_t \cdot \dot{\gamma})^2 \right]^{(n-1)/2}$$

where:

- μ_∞ is the liquid's dynamic viscosity at infinite shear rate, i.e., the minimum dynamic viscosity;

- μ_0 is the liquid's dynamic viscosity at zero shear rate, i.e., the maximum dynamic viscosity;

- K_t is the time constant;

- n is the power law index.

This model is a smooth version of the power law model with the μ restrictions.

In all these three models described above, all parameters with the exception of the dimensionless power law index can be temperature-dependent.

- **Cross-WLF model** (Cross-William-Landel-Ferry) is another modification of the power-law model for shear-thinning liquids. It takes into account the effects of the temperature T:

(2-42)

$$\mu(T, \dot{\gamma}, p) = \frac{\mu_0(T, p)}{1 + \left[\dfrac{\mu_0(T, p) \cdot \dot{\gamma}}{\tau_*} \right]^{(1-n)}}$$

where:

- $\mu_0(T)$ is the zero-shear viscosity or the '*Newtonian limit*' in which the viscosity approaches a constant at very low shear rates:

$$\mu_0(T) = D_1 \cdot \exp\left(-\frac{A_1 \cdot (T - T_*)}{A_2 + (T - T_*)} \right)$$

- $T_* = D_2$ is the glass-transition temperature;

- n is the power-law index in the high shear rate regime;

- τ_* is the critical stress level at the transition to *shear thinning*;

○ A_1, A_2, D_1 and D_2 are the data-fitted constants.

- **Second Order model**:

(2-43)

$$\ln \mu(\dot\gamma, T) = C_1 + C_2 \ln \dot\gamma + C_3 \ln^2 \dot\gamma + C_4 T + C_5 T \ln \dot\gamma + C_6 T^2$$

where C_i, $i = 1 \ldots 6$ are the user-specified coefficients.

The six coefficients C_i can be found by taking the two input viscosity sets $(\dot\gamma_i, \mu_i)$ at T_1 and T_2 respectively, and operating a simultaneous least-squares multi-regression minimization. Thus, this equation represents the master curve for the viscosity representation inside a narrow range of processing temperatures between the lower limit T_1 and the upper limit T_2.

Outside of this range, or if the range becomes too wide, its validity falls progressively down.

The minimum shear rate, below which the viscosity is considered constant, can be determined automatically as the point where $\mu(\dot\gamma)$ reaches its maximum value. The minimum shear rate can also be manually specified by user. The maximum shear rate, after which the viscosity is considered constant, can be specified by user also.

- The **Viscosity table model** defines the liquid's viscosity $\mu(\dot\gamma, T)$ by linear interpolation or polynomial approximation of the user-specified tabular dependencies of the viscosity μ on the shear rate $\dot\gamma$ at the various temperatures T.

Coefficients of the 2nd and 3rd order polynomials are automatically determined with the least squares method.

The 3rd order polynomials are preferable for shear-thickening liquids which viscosity rises intensively when the shear rate approaches zero.

In addition, the user can specify the minimum and maximum shear rates, outside of which the viscosity μ is constant. The minimum shear rate for 2nd order polynomials can be determined automatically as the point where $\mu(\dot\gamma)$ reaches its maximum value.

Water vapor condensation and relative humidity

If the gas whose flow is computed includes steam, Flow Simulation can predict an equilibrium volume condensation of water from this steam (without any surface condensation) taking into account the corresponding changes of the steam temperature, density, enthalpy, specific heat, and sonic velocity.

In accordance with the equilibrium approach, local mass fraction of the condensed water in the local total mass of the steam and the condensed water is determined from the local temperature

of the fluid, pressure, and, if a multi-component fluid is considered, the local mass fraction of the steam. Since this model implies an equilibrium conditions, the condensation has no history, i.e. it is a local fluid property only.

In addition, it is assumed that:

- the volume of the condensed water is neglected, i.e. considered zero, so this prediction works properly only if the volume fraction of the condensed water does not exceed 5%,

- The temperature and pressure in the phase transition area should be within the following ranges: $200.15 \text{ K} < T < 583.15 \text{ K}$ and $10^3 \text{ Pa} < P < 10^7 \text{ Pa}$.

Flow Simulation also allows you to consider the relative **Humidity** of the gas or mixture of gases. This allows you to analyze engineering problems where the condensation of water vapor contained in the air (or other gas), or, more generally speaking, where any differences in physical properties of wet and dry air play an important role.

Cavitation

A liquid subjected to a low pressure below some threshold ruptures and forms vaporous cavities. More specifically, when the local pressure at a point in the liquid falls below the saturation pressure at the local temperature, the liquid undergoes phase transition and form cavities filled with the vapor with an addition of gas that has been dissolved in the liquid. This phenomenon is called cavitation.

If the characteristic time-scale of the vapor formation process is much less than the characteristic time-scale of the liquid flow, the cavitation process occurs in conditions close to thermodynamic equilibrium. Therefore, the equilibrium approach can be used to simulate the liquid-vapor phase transition.

The following models of cavitation are available in Flow Simulation:

- **Equilibrium cavitation model** (for pre-defined water only):

  This model is based on a homogeneous equilibrium model of cavitation, which provides a simple technique for analyzing two-phase flows, and is available for pre-defined water only. It has the capability to account the thermal effects (for example, localized boiling of water due to intense heating).

- **Isothermal cavitation model** (for user-defined liquids only):

  This model is based on the approach considering isothermal, two-phase flows.

Equilibrium cavitation model (for pre-defined water only)

The homogeneous equilibrium approach is employed. It is applicable for a variety of important industrial processes.

The fluid is assumed to be a homogeneous gas-liquid mixture with the gaseous phase consisting of the vapor and non-condensable (dissolved) gas. The vapor mass fraction is defined at the local equilibrium thermodynamic conditions. The dissolved gas mass fraction is a constant, which can be modified by user.

The velocities and temperatures of the gaseous (including vapor and non-condensable gas) and liquid phases are assumed to be the same.

The following assumptions and limitations are made in the cavitation models:

- In the **Equilibrium Cavitation Model**, cavitation is currently available only for the pre-defined water (when defining the project fluids you should select **Water** from the list of **Pre-Defined** liquids).

- Cavitation process is assumed to be equilibrium.

- The mixture is assumed to be a homogeneous two-phase fluid., the velocities and temperatures of the gaseous (including vapor and non-condensable gas) and liquid phases are assumed to be the same.

- The model does not describe the detailed structure of the cavitation area, i.e. parameters of individual vapor bubbles and migration of bubbles.

- The temperature and pressure in the phase transition area should be within the following ranges: 280.15 K $< T <$ 583.15 K, 10^3 Pa $< P < 10^7$ Pa.

- The volume fraction of vapor is limited by 0.95. The parameters of the flow at the initial and boundary conditions must satisfy this requirement.

- The **Cavitation** option is not applicable if you calculate a fluid flow in the model without flow openings (inlet and outlet).

- The fluid region where cavitation occurs should be well resolved by the computational mesh.

- If the calculation has finished or has been stopped and the **Cavitation** option has been enabled or disabled, the calculation cannot be resumed or continued and must be restarted from the beginning.

The density of the gas-liquid mixture is calculated as:

(2-44)

$$\rho = \frac{1}{v} \ , \quad v = y_g \frac{R_{univ}T}{P\mu_g} + (1 - y_g - y_v)v_l(T,P) + y_v \frac{R_{univ}Tz_v(T,P)}{P\mu_v}$$

where v is the specific volume of the gas-liquid mixture, v_l is the specific volume of liquid, $z_v(T,P)$ is the vapor compressibility ratio, R_{univ} is the universal gas constant, P is the local static pressure, T is the local temperature, y_v is the mass fraction of vapor, μ_v is the molar mass of vapor, y_g is the mass fraction of the non-condensable gas; μ_g is the molar mass of the non-condensable gas.

The properties of the dissolved non-condensable gas are set to be equal to those of air. By default, the mass fraction of non-condensable gas is set to 10^{-4}. This is a typical model value appropriated in most cases but it can be modified by the user in the range of $10^{-3}\ldots10^{-5}$.

The mass fraction of vapor y_v is computed numerically from the following non-linear equation for the full enthalpy gas-liquid mixture:

(2-45)

$$H = y_g h_g(T,P) + (1 - y_g - y_v)h_l(T,P) + y_v h_v(T,P) + \frac{I_c v^2}{2} + \frac{5}{2}k$$

where temperature of the mixture T is a function of pressure P and y_v. Here h_g, h_l, h_v are the enthalpies of non-condensable gas, liquid and vapor, respectively, k is the turbulent energy, I_c is the squared impulse defined as:

(2-46)

$$I_C = (\rho u_x)^2 + (\rho u_y)^2 + (\rho u_z)^2$$

Isothermal cavitation model (for user-defined liquids only)

This model provides a capability to analyze two-phase flows of industrial liquids which thermophysical properties are not described in details.

The basis of the model is formed from the assumption that the cavitation process occurs in conditions close to thermodynamic equilibrium. Moreover, it can be assumed that, in most cases, this process is isothermal, since the medium temperature changes due to phase transition are insignificant.

Therefore, analyses assume the existence of a barotropic relation for the two-phase homogeneous mixture. The barotropic state law assumes that the fluid pressure is a function of fluid density only. The multi-phase fluid density is calculated from the barotropic law as presented at the diagram below.

Figure 2-2. Density-pressure phase diagram: 1) ideal gas, 2) two-phase area, 3) compressible liquid).

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

The density of the gas-liquid mixture is calculated as:

(2-47)
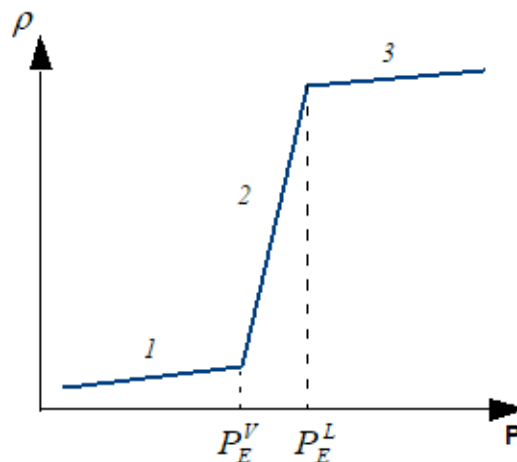
$$\rho = \frac{P - P_E(T_0)}{R_{univ}T_0}\frac{\mu_g}{y_g}\ , \qquad P_E^V \leq P \leq P_E^L$$

where R_{univ} is the universal gas constant, P is the local static pressure, T_0 is the local temperature, P_E is the saturation pressure of liquid at T_0, P_E^L is the local static pressure at which the vapor appears, P_E^V is the local static pressure at which the liquid turns into the vapor completely, y_g is the mass fraction of the non-condensable gas; μ_g is the molar mass of the non-condensable gas.

When the pressure is below P_E^V, a liquid disappears from the mixture and the fluid is treated as an ideal gas with the molar mass μ_0. In this case the density of the gas mixture is calculated as:

(2-48)

$$\rho = \frac{P}{R_{univ}T_0}\left(\frac{y_g}{\mu_g} + \frac{1 - y_g}{\mu_0}\right)^{-1}, \quad P \leq P_E^V$$

When the pressure is above the saturation pressure P_E^L, the fluid density equals the compressible liquid density:

(2-49)

$$\rho = \rho_E^L + \frac{P - P_E^L}{a^2}\ , \qquad P \geq P_E^L$$

where ρ_E^L is the liquid density at the saturation pressure P_E^L, a is the sonic velocity.

In the **Isothermal Cavitation Model**, the mass fraction of the dissolved (non-condensable) gas (y_g) is a variable value. The four gases can be used as a dissolved gas: Air, Carbon dioxide, Helium and Methane, in the **Equilibrium Cavitation Model** Air is used as a dissolved gas. By default, the mass fraction of the dissolved gas is set to 10^{-4}. This is a typical value under normal conditions and appropriate in most cases but it can be modified by the user in the range of $10^{-2}...10^{-6}$.

The following additional assumptions and limitations are made in this model:

- The process temperature is constant and the thermal effects are not considered.

- In the **Isothermal Cavitation Model**, cavitation is currently available only for the user-defined liquids.

This model requires a minimal number of setting fluid parameters, such as density, molar mass, saturation pressure, and dynamic viscosity at the local temperature T_0. The required data are available in the literature for the most industrial liquids, such as gasolines, diesel fuel, mineral and synthetic oils.

Conjugate Heat Transfer

Flow Simulation allows to predict simultaneous heat transfer in solid and fluid media with energy exchange between them.

Heat transfer in fluids is described by the energy conservation equation 2-3 (see "The Navier-Stokes Equations for Laminar and Turbulent Fluid Flows" on page 14), where the heat flux is defined by equation 2-16. The phenomenon of anisotropic heat conductivity in solid media is described by the following equation:

(2-50)

$$\frac{\partial \rho e}{\partial t} = \frac{\partial}{\partial x_i}\left(\lambda_i \frac{\partial T}{\partial x_i}\right) + Q_H$$

where e is the specific internal energy, $e = C \cdot T$, C is specific heat, Q_H is specific heat release (or absorption) per unit volume, and λ_i are the eigenvalues of the thermal conductivity tensor. It is supposed that the heat conductivity tensor is diagonal in the considered coordinate system. For isotropic medium $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$.

If a solid consists of several solids attached to each other, then the thermal contact resistances between them (on their contact surfaces), specified in the **Engineering database** in the form of contact conductance (as $m^2 \cdot K/W$), can be taken into account when calculating the heat conduction in solids. As a result, a solid temperature step appears on the contact surfaces. In the same manner, i.e. as a thermal contact resistance, a very thin layer of another material between solids or on a solid in contact with fluid can be taken into account when calculating the heat conduction in solids, but it is specified by the material of this layer (its thermal conductivity taken from the **Engineering database**) and thickness.

The surface heat source (sink) due to Peltier effect may also be considered (see "Thermoelectric Coolers" on page 92).

The energy exchange between the fluid and solid media is calculated via the heat flux in the direction normal to the solid/fluid interface taking into account the solid surface temperature and the fluid boundary layer characteristics, if necessary.

If a solid medium is porous with a fluid flowing through it, then a conjugate heat transfer problem in this porous-solid/fluid medium can be solved also in the manner described below (see "Flows in Porous Media" on page 66).

Joule Heating by Electric Current in Solids

Flow Simulation is able to calculate steady-state (and quasi time-dependent) direct electric current in electroconductive solids.

_____ **Restriction** _____

This feature is available for the Electronics Cooling module users only.

_____

In presence of the electric current, the corresponding specific Joule heat Q_J, W/m^3, is released and included in Q_H of the heat transfer equation 2-50 (see "Conjugate Heat Transfer" on page 35). In the case of isotropic material Q_J is:

(2-51)

$$Q_J = r \cdot i^2$$

where r is the solids' electrical resistivity, $\Omega \cdot$m, (it can be temperature-dependent) and i is the electric current density, A/m^2.

The electric current density vector:

(2-52)

$$\mathbf{i} = -\left(\frac{1}{r_{11}} \frac{\partial \varphi}{\partial x_1}, \frac{1}{r_{22}} \frac{\partial \varphi}{\partial x_2}, \frac{1}{r_{33}} \frac{\partial \varphi}{\partial x_3} \right)$$

is determined via the electric potential φ, V. To obtain the electric potential φ, Flow Simulation utilizes the steady-state Laplace equation:

(2-53)

$$\frac{\partial}{\partial x_i} \left(\frac{1}{r_{ii}} \frac{\partial \varphi}{\partial x_i} \right) = 0$$

Here r_{ii} is the temperature-dependent electrical resistivity in the i-th coordinate direction.

Transient electric problems with boundary conditions depending on time are considered as quasi-steady-state. In this case the steady-state problem for potential is solved at each time step, not taking into account transient electrical processes in solids.

The Laplace equation is solved numerically in the computational subdomain (it may be a part of the overall computational domain) of electroconductive materials. This computational subdomain automatically excludes dielectric solids and fluid areas inside. The total electric current in normal direction over a surface I_n, A, or electric potential φ, V, may be specified by user as boundary conditions for the problem. These conditions may be imposed on surfaces

between fluid/electroconductive solid, electroconductive solid/electroconductive solid, dielectric solid/electroconductive solid, and outer solid surfaces. If no electrical boundary conditions are specified by user, the $I_n = 0$ boundary condition is automatically specified by default on bounding surfaces.

A surface between electroconductive solids in the computational subdomain is either considered zero-resistance (default) or the electric contact resistance is specified on it. The resistance value is either given explicitly or calculated from the given material and its thickness.

A contact resistance specified on a surface implies that the current passing through it produces the corresponding Joule heating, which yields the following surface heat source Q_{JS}, W/m^2:

$$(2\text{-}54)$$

$$Q_{JS} = i_n \cdot \Delta\varphi$$

where i_n is the electric current density normal to the surface and $\Delta\varphi$ is the electric potential drop at this surface.

The anisotropic electrical resistivity of electroconductive solids can be anisotropic, i.e. specified by its components in the coordinate system's directions r_i, $i = 1, 2, 3$. The isotropic/anisotropic type of material is specified for electrical resistivity and thermal conductivity simultaneously, i.e. so that their main axes coincide.

Radiation Heat Transfer Between Solids

In addition to heat conduction in solids, Flow Simulation is capable of calculating radiation heat transfer between solids. If necessary, a heat radiation from the computational domain far-field boundaries or the model openings can also be defined and considered either as thermal, i.e. by specifying the boundaries emissivity and temperature, or/and as a solar radiation defined by the specified location (on the surface of the Earth) and time (including date) or by a constant or time-dependent direction and intensity. Moreover, it is assumed that the project fluids neither emit nor absorb heat radiation (i.e., they are transparent to the heat radiation).

In Flow Simulation the following radiation models are employed, each of them has its own limitations and benefits:

- **Discrete Transfer**. Its main idea can be described as followed: the radiation leaving the surface element in a certain range of solid angles can be approximated by a single ray. The radiation heat is transferred along a series of rays emanating from the radiative surfaces only. Rays are then traced as they traverse through the fluid and transparent solid bodies until it hits another radiative surface. This approach, usually called '*ray tracing*', allows '*exchange factors*' to be calculated as the fractions of the total radiation energy emitted from one of the radiative surfaces that is intercepted by other radiative surfaces (this quantity is a discrete analog of view factors). '*Exchange factors*' between radiative surface mesh elements are calculated at initial stage of solver, it allows to form matrix of coefficients for a system of linear equations which is solved on each iteration.

- **Discrete Ordinates**. This model solves the *radiative transfer equation* (RTE) for a finite number of discrete solid angles, each associated with a vector direction. This method allows the solution of radiation in absorptive (semi-transparent) media and to model spectral dependencies. The accuracy of the solution depends on the number of discrete directions used.[1]

These models have the following capabilities:

Table 2-1. Capabilities of radiation models.

Capability	Discrete Transfer	Discrete Ordinates
Reflection	supported with limitations[1]	supported with limitations[2]
Refraction	supported	supported with limitations[2]
Spectral dependencies (wavelength dependency)	not supported	supported

---

1. Available for the HVAC module users only.

Table 2-1. Capabilities of radiation models. (cont.)

Capability	Discrete Transfer	Discrete Ordinates
Absorption in semi-transparent solids	not supported	supported
Geometric optics (focusing on lenses, sharp shadows)	supported[3]	supported with limitations[2]
Smooth distribution of the radiative heat fluxes	supported with limitations[4]	supported
Parameters on which the calculation time depends	Quadratic dependence on **View factor resolution level**	Quadratic dependence on **Discretization level**; Linear dependence on **Number of bands**

[1] **Discrete Transfer** is capable for simulating only diffusive reflection or 100% specular reflection if defined by the **Radiative Surface** of **Symmetry** type.

[2] Discretization of space with the ordinates is applicable for majority of tasks though the method is known as one providing moderate solution for cases where geometric optics is strong and essential, for example the solar radiation focusing on lenses.

[3] Geometric optics (focusing on lenses, sharp shadows) is accurately simulated with a directional radiation source (i.e. **Solar radiation source**) in case the **Forward Solar ray tracing** is used. By default the **Backward Solar ray tracing** is used. For the difference between **Forward** and **Backward** ray tracing see "Exchange Factor Calculation" on page 46.

[4] Increasing number of rays makes distribution of fluxes less noisy often at the expense of longer computational time.

When deciding which radiation model to use, consider the following:

- The **Discrete Ordinates** model is recommended for most cases with low-temperature variations and/or with semi-transparent media. At the same time in most cases with low-temperatures, it is not necessary to consider spectral dependencies (i.e. **Number of bands** can be set to 0). Also the **Discretization level** can be set to 2 or 3 to get an acceptable accuracy. If the geometric-optical effects of the shadows are significant, it is recommended to increase the **Discretization level** to improve the geometrical characteristics of the model.

  If that is not enough, the using of **Discrete Transfer** model is recommended.

- For cases with high-temperature gradients (i.e. with high power heat sources such as incandescent lamps, furnaces or other heat sources of temperatures grater then 1000K) and geometric-optical effects of the shadows, focusing and etc. it is recommended to use the **Discrete Transfer** model.

But this model does not allow to simulate the absorption and/or spectral dependencies.

Environment and Solar Radiation

Environmental and solar radiation can be applied to external and internal problems. In fact, the environment radiation is the non-directional energy flux generated by the walls of an imaginary huge "room" that surrounds the body. This flux has predefined radiation parameters.

In contrast to the environment radiation, the solar radiation is modeled by the directional energy flux. Therefore, the solar radiation is defined via its power flow (intensity) and its directional vector. In addition to the solar radiation from the computational domain boundaries, a solar radiation source emitting directional radiation can be specified.

The Sun directional vector can be calculated using the following formula:

$$\vec{S} = \left[\vec{E} \cdot \sin \phi_s + \vec{N} \cdot \cos \phi_s \right] \cdot \cos \theta_s + \vec{Z} \cdot \sin \theta_s \qquad \text{(2-55)}$$

where $\vec{Z}$ is the zenith direction, $\vec{N}$ is the north direction, $\vec{E} = \vec{N} \times \vec{Z}$ is the east direction;

θ_s is the solar elevation angle. That is, the angle between the direction of the geometric center of the sun's apparent disk and the (idealized) horizon;

ϕ_s is the solar azimuth angle. That is, the angle from due north in a clockwise direction.

The solar elevation angle θ_s can be calculated, to a good approximation, using the following formula:

(2-56)

$$\sin \theta_s = \cos h \cos \delta \cos \Phi + \sin \delta \sin \Phi$$

where Φ is the local latitude, δ is the current Sun declination,

$h = \dfrac{t}{86400} 2\pi - \pi$ is the hour angle of the present time t [s] (for example, at solar noon 12:00AM, $t = 43200$ s and $h = 0$).

The solar azimuth angle ϕ_s can be calculated, to a good approximation, using the following formulas:

(2-57)

$$\sin \phi_s = -\frac{\sin h \cos \delta}{\cos \theta_s}$$

$$\cos \phi_s = \frac{\sin \delta \cos \Phi - \cos h \cos \delta \sin \Phi}{\cos \theta_s}$$

The current Sun declination δ is approximately given by the following expression:

(2-58)

$$\delta = -23.44° \cdot \cos\left[\frac{360°}{365} \cdot (N + 10)\right]$$

where N is the day of the year beginning with $N = 0$ at midnight *Coordinated Universal Time* as January 1 begins (i.e. the days part of the ordinal date -1). The number 10, in $(N + 10)$, is the approximate number of days after the December solstice to January 1. The number *23.44* is the Earth's axial tilt.

The solar radiation intensity can be calculated using the following formula:

(2-59)

$$q_s(\theta_s) = \frac{Q_{s0}}{1 + C_m m_1} \cdot (1 - n)$$

where $C_m = 0.092$ and $m_1 = 1.029/(\sin\theta_s + 0.029)$ are empirical values;

n is the cloudiness, i.e. the presence of clouds in the sky and ranges from 0 (no clouds in the sky) to 1 (all the sky is covered by clouds);

Q_{s0} = 1360 W/m^2 is the solar constant (i.e. the energy reaching the Earth from the Sun at the top of the atmosphere).

Discrete Transfer

The **Discrete Transfer** model is a method that has its origin in the flux model but also exhibits features of the Hottel zone and Monte Carlo techniques.

The following are general assumptions regarding the **Discrete Transfer** model:

- The heat radiation from the solid surfaces, both the emitted and reflected, is assumed diffuse (except for the symmetry radiative surface type), i.e. obeying the Lambert law, according to which the radiation intensity per unit projected source area per unit solid angle is the same in all directions.

- The radiative solid surfaces which are not specified as a blackbody or whitebody are assumed an ideal graybody, i.e. having a continuous emissive power spectrum similar to that of blackbody, so their monochromatic emissivity is independent of the emission wavelength. For certain materials with certain surface conditions, the graybody emissivity can depend on the surface temperature.

- The solar radiation is absorbed and reflected by surfaces independently from thermal radiation from all other heat radiation sources.

- The propagating heat radiation passes through a solid specified as radiation transparent without any absorption. A solid can be specified as transparent to the solar radiation only, or transparent to the thermal radiation from all sources except the solar radiation, or transparent to both types of radiation, thermal and solar.

The Discrete Transfer Equations

In a general case, the surfaces participating in the heat radiation (hereinafter radiative surfaces) can emit, absorb, and reflect a heat radiation (both the solar and thermal).

The heat radiation leaving a radiative surface or radiation source is defined as:

- for thermal radiation:

$$(2\text{-}60)$$

$$Q_T^{out} = \varepsilon \cdot \sigma \cdot T^4 \cdot A + (1-\varepsilon) \cdot Q_T^{in}$$

where: ε is the surface emissivity, σ is the Stefan-Boltzmann constant, T is the temperature of the surface ($\varepsilon \cdot \sigma \cdot T^4$ is the heat flux radiated by this surface in accordance with the Stefan-Boltzmann law), A is the radiative surface area, Q_T^{in} is the incident thermal radiation arriving at this surface;

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

- for solar radiation:

(2-61)

$$Q_S^{out} = (1 - \varepsilon) \cdot \left(Q_S^{in} + Q_S^{source} \right)$$

where: Q_S^{in} is the incident solar radiation arriving at this surface, Q_S^{source} is the radiation arriving at this surface from solar sources.

The net radiation Q^{net} being the difference between the heat radiation Q^{out} leaving this surface and the incident heat radiation Q^{in} arriving at it:

(2-62)

$$Q^{net} = Q^{out} - Q^{in} = \left(Q_T^{out} + Q_S^{out} \right) - \left(Q_T^{in} + Q_S^{in} \right)$$

are calculated for each of the surfaces participating in the radiation heat transfer.

Clustering and Ray Tracing

In order to reduce memory requirements, the problem of determining the leaving and net heat radiation fluxes is solved using a discrete ray Monte-Carlo approach consisting of the following main elements:

- To reduce the number of radiation rays and, therefore, the required calculation time and resources, the computational mesh cells containing faces approximating the radiative surfaces are joined in clusters by a special procedure that takes into account the face area and angle between normal and face in each partial cell. The cells intersected by boundaries between radiative surfaces of different emissivity are considered as belonging to one of these surfaces and cannot be combined in one cluster. This procedure is executed after constructing the computational mesh before the calculation and after each solution-adaptive mesh refinement, if any.

- For each of the clusters, the hemisphere governed by the ray's origin and the normal to the face at this origin is evenly divided into several nearly equal solid angles generated by several zenith (latitudinal) angles Θ (at least 3 within the 0...90° range, including the zero zenith angle of the normal to the face) and several azimuth (longitudinal) angles Φ (at least 12 within the 0...360° range).

A radiation ray is emitted in each of the solid angles in a direction that is defined randomly within this solid angle. Each ray is traced through the fluid and transparent solid bodies until it intercepts the computational domain's boundary or a cluster belonging to another radiative surface, thus defining a '*target*' cluster. Since the radiation heat is transferred along these rays only, their number and arrangement govern the accuracy of calculating the radiation heat coming from one radiative surface to another (naturally, the net heat radiated by a radiative surface does not depend on number of these rays).

Figure 2-3. Definition of rays emitted from cluster: *P* - polar ray, *Θ* - zenith (latitudinal) angle, *Φ* - azimuth (longitudinal) angle.
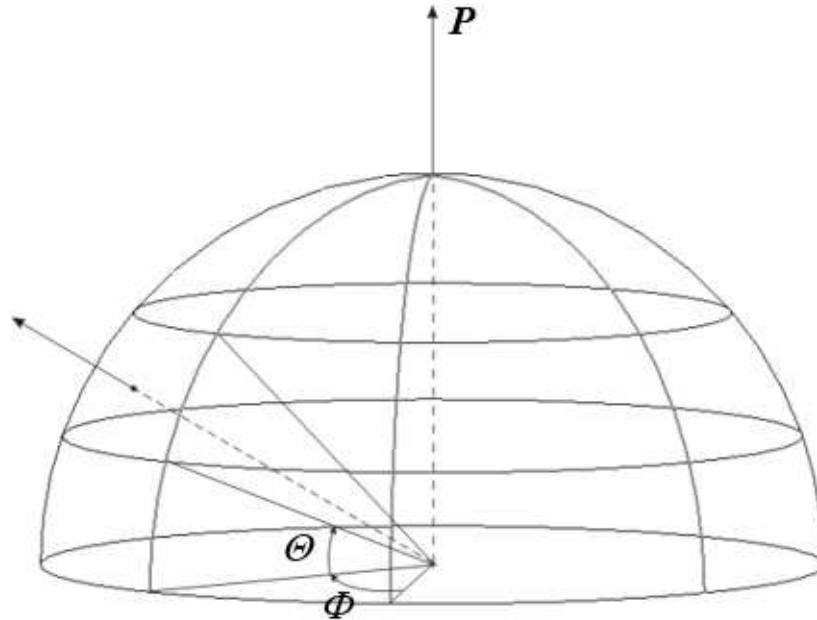
The total number of rays emitted from a cluster is calculates as:

$$N = (m-1) \cdot n + 1 \tag{2-63}$$

where m is the number of different latitude values for the rays (including the polar ray), n is the number of different longitude values ($n = 2$ for 2D case).

The value of m is defined directly by the **View factor resolution level** which can be changed by the user via the **Calculation Control Options** dialog box. The value of n depends on m as follows:

$$n = m \cdot 4 \tag{2-64}$$

The higher the **View factor resolution level**, the better the accuracy of the radiation heat transfer calculation, but the calculation time and required computer resources increase significantly when high values of **View factor resolution level** are specified.

To increase the accuracy of heat radiation calculation, the number of radiation rays emitted from each cluster can be increased automatically during the calculation, depending on the surface temperature and emissivity, to equalize the radiation heat emitting through the solid angles.

- When a radiation ray intercepts a cluster of other radiative surfaces, the radiation heat carried by this ray is evenly distributed over the area of this cluster. The same procedure is performed if several radiation rays hit the same cluster. To smooth a possible non-uniformity of the incident radiation heat distribution on a radiative surface, a fraction of

the radiation heat arriving with rays at a cluster can be transferred to the neighboring clusters also. In addition, small fluctuations are smoothed by the heat conduction in solid regions.

Exchange Factor Calculation

The *'exchange factor'* F_{ij} between two clusters is the fraction of the total radiation energy emitted from one of the clusters that is intercepted by other clusters. Hence, given a point j at the center of a cluster on the radiative surface, a radiation ray is fired from j for each one of the specified directions. Let i be the impingement point, starting from the origin j, the ray is traced to the i.

Then, the incident radiation at point i, Q_i^{in}, is calculated by adding up the contributions due to all the rays that reach point i:

- the incident thermal radiation is defined as:

(2-65)

$$Q_{Ti}^{in} = \sum_j F_{ij} Q_{Tj}^{out}$$

- the incident solar radiation is defined as:

(2-66)

$$Q_{Si}^{in} = \sum_j F_{ij} Q_{Sj}^{out}$$

The radiation leaving from the i-th cluster, Q_i^{out}, is then computed as a sum of the reflected portion of Q_i^{in} and the emissive power of the surface:

- for thermal radiation:

(2-67)

$$Q_{Ti}^{out} = (1 - \varepsilon_i) Q_{Ti}^{in} + \varepsilon_i \sigma T_i^4 A_i$$

- for solar radiation:

(2-68)

$$Q_{Si}^{out} = (1 - \varepsilon_{Si}) \cdot (Q_{Si}^{in} + Q_{Si}^{source})$$

The radiation arriving at i-th cluster from solar sources, Q_{Si}^{source}, is calculated as:

- for the backward solar ray tracing method:

(2-69)

$$Q_{Si}^{source} = \sum_k I_{Sk} \cdot F_{Ski} \cdot \delta_{ki}$$

- for the forward solar ray tracing method:

(2-70)

$$Q_{Si}^{source} = Q_S^{ray} \cdot N_i$$

'*Backward*' means that rays are emitted from the radiative surfaces. Each solar radiation source produces one ray that follows the directional vector backward. After it reaches the outer boundary or the surface having appropriate radiation boundary condition, the exchange factor can be estimated as:

(2-71)

$$F_{Ski} = \left(\vec{n}_k, \vec{n}_i \right) \cdot A_i$$

The I_{Sk} is the solar radiation intensity of the k-th solar source. The value δ_{ki} equals 1 if the i-th cluster is visible for the k-th solar source, and 0 if not.

'*Forward*' means that the rays are emitted from the radiation source. The radiation emitted by all solar radiation sources can be distributed evenly over all rays traced from radiation sources. The radiation transferred by each ray, Q_S^{ray}, can be estimated by summing up the radiation emitted by all solar sources, Q_{Sk}^{source}, and then divided by the number of emitted rays, N_{rays}:

(2-72)

$$Q_S^{ray} = \frac{\sum_k Q_{Sk}^{source}}{N_{rays}}$$

The value N_i is the number of rays intercepted the i-th cluster. The forward method is recommended in case of refraction and reflection radiative surfaces.

The solar ray tracing method and the number of rays traced from radiation sources can be changed by the user via the **Calculation Control Options** dialog box.

Please note that for the sake of simplicity the set of equations described here defines the radiation heat transfer between clusters only and does not take into account the outer boundaries radiation and diffusive radiation sources. In the full set of equations these sources are also considered.

Discrete Ordinates

The **Discrete Ordinates** model is a way to approximately solve the *radiative transfer equation* (RTE) by discretizing both the XYZ-domain and the angular variables that specify the direction of radiation.

___ **Restriction** ___

This feature is available for the HVAC module users only.

The following are general assumptions regarding the **Discrete Ordinates** model:

- The whole 4π directional domain at any location within the computational domain is discretized into the specified number of equal solid angles.

- Radiation absorptive (semi-transparent) solids absorb and emit heat radiation in accordance with the specified solid material absorption coefficient. Scattering is not considered.

- Surfaces of opaque solids absorb the incident heat radiation in accordance with their specified emissivity coefficients, the rest incident radiation is reflected specularly or diffusively, or both specularly and diffusively, in accordance with the specified specularity coefficient.

- Radiation absorptive solids reflect radiation specularly, the radiation is refracted in accordance with the specified refraction indices of the solid and adjacent medium (another radiation absorptive solid, or a transparent solid or fluid, which refraction index is always considered as equal to 1). The refraction index value cannot exceed 4.

The Discrete Ordinates Method

In the **Discrete Ordinates** method the *radiative transfer equation* (RTE) is solved for a set of discrete directions s representing the directional domain of 4π at any position within the computational domain defined by the position vector r.

The directional domain is broken down into the specified number of equal solid angles or directions. The total number of directions is defined as:

(2-73)

$$N_{ord} = 8 \cdot \frac{RL \cdot (RL + 1)}{2}$$

where *RL* is the **Discretization level** specified by the user. Within each direction the radiation intensity is considered constant.

In the absence of scattering the *radiative transfer equation* (RTE) can be written as:

(2-74)

$$\frac{dI(\vec{s},\vec{r})}{ds} = \alpha \cdot \left[n^2 \cdot I_b(\vec{r}) - I(\vec{s},\vec{r}) \right]$$

where *I* is the radiation intensity per solid angle, $I_b = \frac{\sigma T^4}{\pi}$ is the blackbody radiation intensity, α is the medium absorption coefficient, *n* is the refraction index.

At the surfaces of opaque solids the incident heat radiation is absorbed depending on the specified emissivity coefficient, the rest of the incident radiation is reflected. The opaque surfaces can also emit heat radiation diffusively in accordance with the surface temperature and the specified emissivity coefficient.

Since all fluids are considered as transparent to heat radiation, the heat radiation propagates through them, as well as through transparent solids, without any interaction with them. However, as the heat radiation is traced through the computational domain by the discrete ordinates method, the '*false scattering*' effect caused by the discretization inaccuracies can appear - it is similar to the '*numerical diffusion*' effect in fluid flow calculations. Creating a finer computational mesh allows to reduce this effect.

When the radiation propagates from a small source to a significant distance, the '*ray effect*' can be encountered as a result of breaking down the directional domain into several directions. As radiation in these directions is traced from the source, it begin to demonstrate a ray-like behavior because more and more cells fall within the same direction with the distance and the radiation intensity distribution within the direction becomes non-uniform. This effect must be considered when decreasing the cell size to avoid the '*false scattering*' effect, so the discretization level must be increased if the finer mesh is used.

Absorption and Transmission

The absorptive (semi-transparent) solids absorb heat radiation in accordance with the specified absorption coefficient.

___ **Restriction** _____

This feature is available for the HVAC module users only.

___ **Note** _____

This option is available for the **Discrete Ordinates** model only.

In the absence of emitting the *radiative transfer equation* (RTE) 2-74 (see "Discrete Ordinates" on page 48) can be written along axis *x* as:

$$\frac{dI(x)}{dx} = -\alpha \cdot I(x)$$

The radiation intensity continuously decreases with distance *x* that the light traverses:

$$I(x) = I_0 e^{-\alpha x}$$

where I_0 is the initial intensity of radiation and α, the absorption coefficient, is the characteristic of particular material.

At the interface between two transparent or semi-transparent media with different refractive indices the incident radiation changes its direction in accordance with the Snell's law:

$$\frac{\sin \theta_2}{\sin \theta_1} = \frac{n_1}{n_2}$$

where n_1 and n_2 are the refractive indices of the first and second medium (*n* is always equal to 1 for all fluids and for a fully transparent solid material, if its radiation properties are not specified in the **Engineering Database**) and θ_1 and θ_2 are the incident and refraction angles correspondingly.

Figure 2-4. Radiation reflection and transmission at the interface of absorptive media.
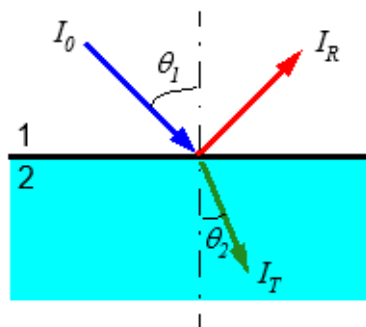
The radiation reflection at the interface between two transparent or semi-transparent media with different refractive indices follows the Fresnel's relation for unpolarized light:

(2-78)

$$\rho = \frac{1}{2}\left[\frac{\tan^2(\theta_1 - \theta_2)}{\tan^2(\theta_1 + \theta_2)} + \frac{\sin^2(\theta_1 - \theta_2)}{\sin^2(\theta_1 + \theta_2)}\right]$$

where ρ is the reflectivity which define the fraction of reflected radiation.

The fraction of radiation transmitted through the interface is defined as:

(2-79)

$$\tau = 1 - \rho$$

If the light is normal to the interface, then:

(2-80)

$$\rho = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2, \qquad \tau = \frac{4n_1 n_2}{(n_1 + n_2)^2}$$

The phenomena of absorption, reflection, and transmission may be applied to the passage of light through a semi-transparent solid plate. In case of the incident light is normal to the solid surface, the transmitted intensity can be estimated as shown in Figure 2-5.

Figure 2-5. Radiation transmission through a transparent medium.
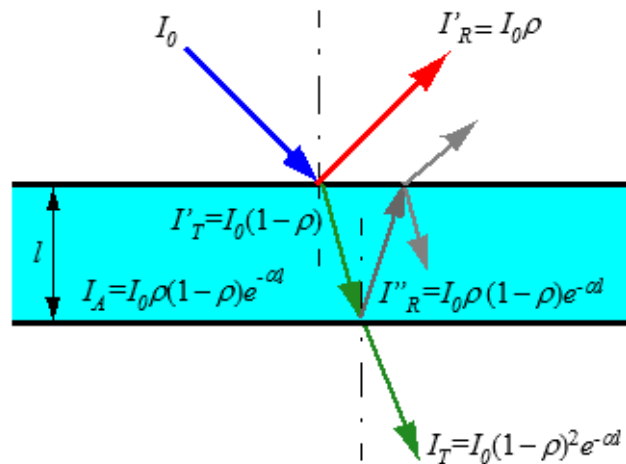
For an incident beam of intensity I_0 that impinges on the front surface of a semi-transparent solid plate of thickness l and absorption coefficient α the total reflected intensity I_R from the back face can be estimated as:

(2-81)

$$I_R = I_0 \rho \left[1 + \frac{(1-\rho)^2 e^{-2\alpha l}}{1-\rho^2 e^{-2\alpha l}} \right]$$

The total transmitted intensity I_T at the back face can be estimated as:

(2-82)

$$I_T = I_0 \frac{(1-\rho)^2 e^{-\alpha l}}{1-\rho^2 e^{-2\alpha l}} \approx I_0 (1-\rho)^2 e^{-\alpha l}$$

The total absorbed intensity I_A in the solid plate can be estimated as:

(2-83)

$$I_A = I_0 - I_R - I_T = I_0(1-\rho) \left[1 - \frac{(1-\rho)e^{-\alpha l}}{1-\rho e^{-\alpha l}} \right]$$

So the absorption coefficient α can be expressed as:

(2-84)

$$\alpha = -\frac{1}{l} \ln \left(\frac{I_A - I_0(1-\rho)}{I_A \rho - I_0(1-\rho)} \right)$$

Thus, the fraction of incident light that is transmitted through a semi-transparent material depends on the losses that are incurred by absorption and reflection.

So in accordance with equation 2-82, the absorption coefficient α can be approximately expressed as:

(2-85)

$$\alpha \approx -\frac{1}{l} \ln \left(\frac{\frac{I_T}{I_0}}{(1-\rho)^2} \right)$$

Reflection

At the surfaces of opaque solids the incident heat radiation is absorbed depending on the specified emissivity coefficient, the rest of the incident radiation is reflected by one of the following ways or their combination:

- diffusive (Lambertian) reflection,

- diffusive and specular (ideal) reflection which available for the **Discrete Ordinates** model only.

The reflection can be expressed as a transfer function that records for a given incoming direction, the amount of light that is reflected in a certain outgoing direction. The function is called *BRDF* (*bi-directional reflectance distribution function*). This is function of four angles, two incident (θ_i, φ_i) and two reflected (θ_r, φ_r).

Figure 2-6. Radiation reflection at the surface (blue - incident light, red - reflected light).
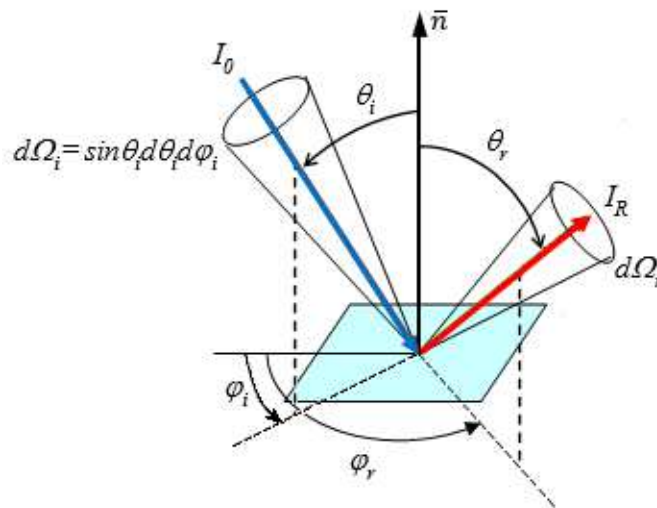
The BRDF function $\rho_{bd}(\theta_i, \varphi_i, \theta_r, \varphi_r)$ is defined in terms of incident and reflected radiance by the following integral:

$$(2\text{-}86)$$

$$L_r(\theta_r, \varphi_r) = \int\limits_0^{2\pi}\int\limits_0^{\pi/2} L_i(\theta_i, \varphi_i)\rho_{bd}(\theta_i, \varphi_i, \theta_r, \varphi_r)\cos\theta_i \sin\theta_i d\theta_i d\varphi_i$$

where:

φ is the azimuthal angle measured about the surface normal,

$L_r(\theta_r, \varphi_r)$ is the reflected radiance [W/sr/m^2],

$L_i(\theta_i, \varphi_i)$ is the incident radiance,

$\rho_{bd}(\theta_i, \varphi_i, \theta_r, \varphi_r)$ is the BRDF [sr^{-1}].

Depending on the surface reflective properties the BRDF can be estimated as follows:

- **diffusive (Lambertian) surface**: light is equally likely to be reflected in any output direction, independent of the incoming direction.

Figure 2-7. Diffuse reflection (blue - incident light, red - reflected light).
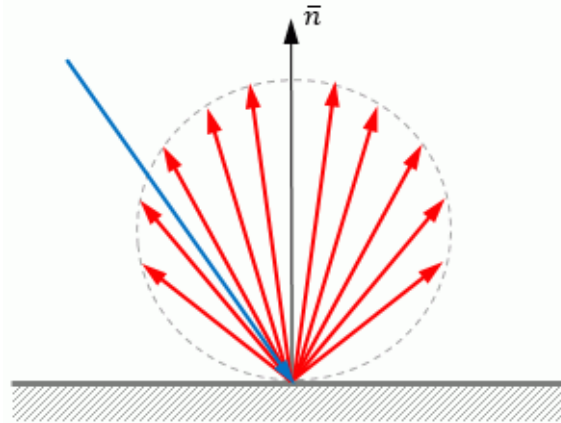
The BRDF of Lambertian surface is constant:

$$(2\text{-}87)$$

$$\rho_{bd}\left(\theta_i, \varphi_i, \theta_r, \varphi_r\right) = \frac{\rho_d}{\pi}$$

where ρ_d is the diffuse reflectance.

- **ideal specular surface**: the surface that reflects all the light coming from the direction (θ_i, φ_i) into the direction $(\theta_i, \varphi_i + \pi)$.

Figure 2-8. Ideal specular (mirror) reflection (blue - incident light, red - reflected light).
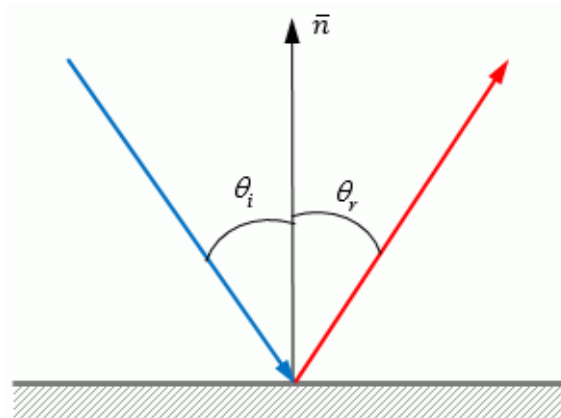
So the BRDF is zero everywhere except where: $\theta_r = \theta_i$ and $\varphi_r = \varphi_i + \pi$. Thus the BRDF is a delta function at direction of ideal mirror reflection:

(2-88)

$$\rho_{bd}(\theta_i, \varphi_i, \theta_r, \varphi_r) = \frac{\rho_s \delta(\theta_i - \theta_r)\delta(\varphi_i + \pi - \varphi_r)}{\cos\theta_i \sin\theta_i}$$

where ρ_s is the specular reflectance.

In case of diffusive and specular (ideal) reflection the BRDF is defined as follows:

(2-89)

$$\rho_{bd}(\theta_i, \varphi_i, \theta_r, \varphi_r) = \frac{\rho_d}{\pi} + \frac{\rho_s \delta(\theta_i - \theta_r)\delta(\varphi_i + \pi - \varphi_r)}{\cos\theta_i \sin\theta_i}$$

The surface specularity (ρ_s) defines the fraction of ideal specular reflected radiation. And the diffusively reflected fraction is determined as ($\rho_d = 1 - \rho_s$).

Radiation Spectrum

Spectrum can be specified for radiation from the computational domain boundaries and for radiation sources set on surfaces of opaque solids (representing openings).

___ **Restriction** _____

This feature is available for the HVAC module users only.

___ **Note** _____

This option is available for the **Discrete Ordinates** model only.

In case of using the **Discrete Ordinates** model, a multiband (**Discrete Bands**) approach is used to model spectral dependencies in radiative heat transfer.

The radiation spectrum is considered as consisting of several bands, which edges are specified by the user. Properties of radiation sources, surfaces and materials are considered constant within each band. The wavelength-dependent properties of solid materials are averaged over the specified spectrum bands, so it is recommended to specify the band edges at the wavelengths, at which the material properties change substantially.

If the radiation spectrum is considered, the *radiation transfer equation* (RTE) 2-74 (see "Discrete Ordinates" on page 48) takes the following form:

$$\frac{dI_{\lambda_i}(\vec{s},\vec{r})}{ds} = k_{\lambda_i} \cdot \left[n^2 \cdot I_{b\lambda_i}(\vec{r}) - I_{\lambda_i}(\vec{s},\vec{r}) \right]$$

where I_{λ_i} is the heat radiation intensity in the *i*-th spectrum band, $I_{b\lambda_i}$ is the intensity of the blackbody radiation in the *i*-th spectrum band, k_{λ_i} is the specified the medium absorption coefficient in the *i*-th spectrum band.

Radiative Surface and Radiation Source Types

Radiative surfaces allow to consider the selected surfaces in contact with the fluid as a radiative surface for a conjugate heat transfer problem. And radiation sources allow to consider heating of the model components and solid bodies due to the radiation coming into the internal space of the model through openings.

Radiative Surfaces

A radiative surface can be specified on the surface of either an opaque or a semi-transparent and a transparent solid.

For the **Wall**, **Wall to ambient** and **Wall to environment wall** boundary condition, the program gets T from the current results set.

Table 2-2. Radiative Surface Parameters.

Radiative surface type	Specified values	Dependent values
Wall	$T, \varepsilon, \varepsilon_{solar}, \rho_s$	$\rho = 1 - \varepsilon$, $\alpha = \varepsilon$
Wall to ambient		$\rho_{solar} = 1 - \varepsilon_{solar}$
Wall to environment wall		$\rho_d = 1 - \rho_s$
Symmetry (ideal reflection)	No parameters	
Absorbent wall (no ray starting)	No parameters	
Non-radiating wall (no rays)	No parameters	

The rays are emitted only from surfaces and boundaries on which the **Wall** or radiation source conditions are applied.

Wall to ambient and **Wall to environment wall** reproduces the most elementary phenomenon among the radiation effects. The walls with this condition are treated as black body and do not interact with any other surfaces. They can only exhaust energy into the space that surrounds the computational domain. Heat flux from such surface could be calculated as:

$$(2\text{-}91)$$

$$Q = \sigma \cdot \left(T^4 - T_{out}^4 \right)$$

where T_{out} is the temperature of the environmental radiation. In case of the **Wall to environment wall** boundary condition, a particular temperature of the environmental radiation can be specified at the wall boundary.

When a ray reaches the surface of a radiation source or the **Wall to ambient** (or **Wall to environment wall**) surface, it disappears. All energy carried by this ray also dies away.

_____ **Note** _____

The **Wall to ambient** and **Wall to environment wall** surface type of radiative surfaces is not consistent with the **Discrete ordinates** method, and is substituted in the calculation with the **Wall**.[2]

_____

The **Symmetry** boundary condition forces the walls to which it is applied to reflect rays as an ideal mirror.

Surfaces with the specified **Absorbent wall** boundary condition are taken into account during the calculation but they can act as absorptive surfaces only. This wall type takes all heat from the radiation that reaches it and does not emit any heat.

Non-radiating boundary condition removes specific surfaces from the radiation heat transfer analysis, so they do not affect the results.

_____ **Note** _____

The **Absorbent wall** and **Non-radiating** surface types of radiative surfaces are not consistent with the **Discrete ordinates** method, and are substituted in the calculation with **Whitebody wall**, which emissivity is considered equal to 0 (i.e. to that of whitebody), so that the surface fully reflects all the incident radiation (in accordance with the Lambert's law) and does not emit any heat by itself.[3]

_____

Radiation Sources

A radiation source can only be specified on the surface of an opaque solid. Note that when calculating net radiation rate on such surface, radiation source is not accounted. All the incident radiation disappears without absorption or reflection on the radiation source surface, unless there is a radiative surface condition defined on the same surface.

Table 2-3. Radiation Source Parameters.

Radiation source type	Specified values
Diffusive radiation source	Power (Q) or Intensity (I) or Temperature (T) and/or spectrum[*]

_____

2. These capabilities and features are available for the HVAC module users only.
3. These capabilities and features are available for the HVAC module users only.

Table 2-3. Radiation Source Parameters. (cont.)

Radiation source type	Specified values
Directional radiation source	Power (Q) or Intensity (I) or Temperature (T) and/or spectrum[*]

[*] The spectrum definition is available for the HVAC module users only.

The directional radiation source makes the wall to emit radiation like the outer solar radiation. It is specified by the direction vector and power or intensity or temperature. The solar radiation at the computational domain boundaries can be specified not only by the direction vector and intensity, but also by the location (on the surface of the Earth) and time.

Simultaneous Use of Radiative Surface and Radiation Source Conditions

The user can specify a radiative surface and radiation source on the same surface of an opaque solid.

In this case the total heat radiation leaving the surface is determined as:

$$(2\text{-}92)$$

$$q = (1 - \varepsilon)q_i + \varepsilon \cdot \sigma \cdot T^4 + q_s$$

where: ε is the emissivity of the radiative surface, q_i is the incident heat radiation arriving on the surface, T is the surface temperature and q_s is the heat radiation emitted by the radiation source. Please note that the radiative surface emissivity coefficient and temperature do not influence the heat radiation emitted by the radiation source defined on the same surface – the radiation source properties are specified independently.

Viewing Results

The main result of the radiation heat transfer calculation is the solids' surface or internal temperatures. But these temperatures are also affected by heat conduction in solids and solid/fluid heat transfer.

To see the results of radiation heat transfer calculation only, the user can view the **Incident** (q_{inc}), **Transmitted** (q_{trs}) and **Leaving** (q_{eff}) **radiant fluxes** and distribution of **Net radiant flux** (q_{net}) over the selected radiative surfaces at **Surface Plots**. User can also see the maximum, minimum, and average values of these parameters as well as the **Incident**, **Transmitted** and **Leaving radiation rate** and **Net radiation rate** as an integral over the selected surfaces in Surface Parameters.

All these parameters can be viewed separately for the **Solar radiation** and radiation from directional radiation sources (**Directional radiation**) and the radiation from all other heat radiation sources (**Thermal radiation**).

When the absorptive (semi-transparent) solids are considered, the additional parameters such as **Absorption volume radiant flux** (q_{abs}), **Net volume radiant flow** and **Net volume radiant flux** become available both for the solar and thermal radiation, as well as for the total radiation and heat flux. However, the **Leaving** and **Net** radiant fluxes and radiation rates are not available at the surfaces of the absorptive solids.

The relations between the radiant fluxes can be described as follows:

$$q_{inc} = q_{trs} + q_{ref} \tag{2-93}$$

$$q_{eff} = q_{em} + q_{ref} \tag{2-94}$$

$$q_{net} = q_{em} - q_{abs} = q_{eff} - q_{inc} \tag{2-95}$$

where q_{em} is the radiant flux emitted by the body surface; q_{ref} is the radiant flux reflected by the body surface.

Figure 2-9. Definition of radiant fluxes.
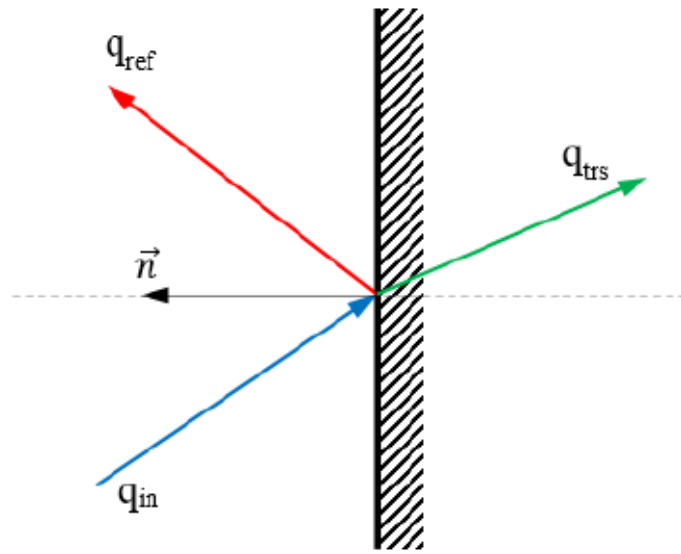
The availability of radiant fluxes dependent on the solid body transparency for radiation is shown below.

Table 2-4. Radiant fluxes dependent on solid bodies transparency.

Radiant fluxes	Semi-Transparent	Opaque
Net	not supported	supported
Leaving	not supported	supported
Absorption (volume)	supported	not supported
Incident	supported	supported
Transmitted	supported	supported

If the spectral dependencies are specified in terms of discrete bands, the distribution of radiant heat fluxes (**Net** and **Leaving**) over the selected radiative surfaces can be shown in each band.

Rotation

Flow Simulation allows to simulate a rotating equipment by modeling the flow in the coordinate system rotating with the rotating equipment.

> **Note**
>
> The **Rotation** option is incompatible with the **High Mach number flow** and **Free Surface** options.

The following models of rotation are available in Flow Simulation:

Global rotating. With this option the model and the global coordinate system are considered rotating with specified angular velocity. Global rotating reference frame is applicable when all non-rotating model components are axisymmetrical with respect to the selected rotation axis.

Local region(s) (Averaging) option allows to simulate fluid flows in regions surrounding rotating solids, which are not bodies of revolution by specifying multiple local rotating reference frame within the model in cases where the rotor-stator interaction is weak. The flow parameters within the rotating regions are calculated using a steady-state approach and averaged on the rotating regions' boundaries.

Local region(s) (Sliding) option allows to specify multiple local rotating coordinate systems within the model in cases where the rotor-stator interaction is strong. The *Sliding* approach assumes that the flow field is unsteady and it is available for **Time-dependent** analysis only. This assumption allows to obtain more accurate simulation than by using the *Averaging* approach, however, that because this approach requires an unsteady numerical solution, it is computationally more demanding than the *Averaging* approach.

Global Rotating Reference Frame

The **Global rotating** is assumed by default that all model walls rotate at the speed of the rotating reference frame.

The rotation of the coordinate system is taken into account via the following mass-distributed force:

(2-96)

$$S_i^{rotation} = -2\, e_{ijk}\Omega_j \rho u_k + \rho\, \Omega^2 r_i$$

where e_{ijk} is the Levy-Civita symbols (function), Ω is the angular velocity of the rotation, r is the vector coming to the point under consideration from the nearest point lying on the rotation axis.

Local rotating region(s) (Averaging)

This option is employed for calculating time-dependent (transient) or steady-state flows in regions surrounding rotating solids, which are not bodies of revolution (e.g. impellers, mixers, propellers, etc), when a single global rotating reference cannot be employed. For example, local rotating regions can be used in analysis of the fluid flow in the model including several components rotating over different axes and/or at different speeds or if the computational domain has a non-axisymmetrical (with respect to the rotating component) outer solid/fluid interface.

In accordance with the employed approach, each rotating solid component is surrounded by an axisymmetrical (with respect to the component's rotation axis) **Rotating region**, which has its own coordinate system rotating together with the component. If the model includes several rotating solid components having different rotation axes, the rotating regions surrounding these components must not intersect with each other. The fluid flow equations in the stationary (non-rotating) regions of the **Computational Domain** or **Fluid Subdomain** are solved in the inertial (non-rotating) Cartesian Global Coordinate System. The influence of the rotation on the flow is taken into account in each of the rotating coordinate systems.

To connect solutions obtained within the rotating regions and in the non-rotating part of the computational domain, special internal boundary conditions are set automatically at the fluid boundaries of the rotating regions. To correctly set these boundary conditions and the rotating coordinate system, a rotating region must always be a body of revolution. The rotating region's boundaries are sliced into rings of equal width as shown on the Figure 2-10. Then the values of flow parameters, transferred as boundary conditions from the adjacent fluid regions, are calculated in the local rotating coordinate system and then averaged circumferentially over each of these rings.

**Figure 2-10. The flow parameters averaging on the stator-rotor interface:
1) Computational domain or fluid subdomain; 2) Local rotating region and its
boundaries sliced into rings.**
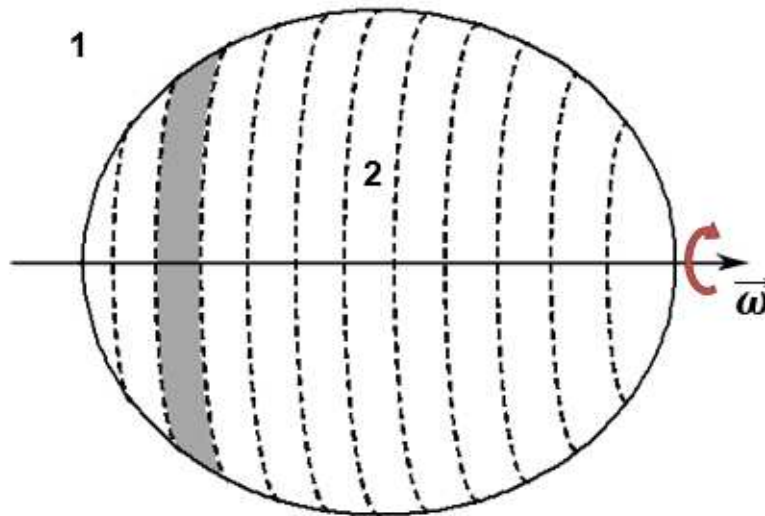
To solve the problem, an iterative procedure of adjusting the flow solutions in the rotating regions and in the adjacent non-rotating regions, therefore in the entire computational domain, is performed with relaxations.

___Note___

Please note that even in the case of time-dependent (transient) analysis the flow parameters within the rotating regions are calculated using a steady-state approach and averaged on the rotating regions' boundaries as described above.

Local rotating region(s) (Sliding)

This option is employed for calculating time-dependent (transient) flows in cases where the rotor-stator interaction is strong, when a time-accurate solution for rotor-stator interaction (rather than a time-averaged solution) is desired.

The **Sliding** approach assumes that the flow field is unsteady. This assumption allows to obtain more accurate simulation than by using the **Averaging** approach, however, that because this approach requires an unsteady numerical solution, it is computationally more demanding than the '*Averaging*' approach.

In the Sliding technique rotor and stator cells zones are connected with each other through '*sliding interface*'. During the calculation, zones linked through '*sliding interface*' remain in contact with each other (i.e., the rotor cells '*slide*' relative to stator cells along the interface boundary in discrete steps as shown in Figure 2-11).

Figure 2-11. Rotor-stator grid interface: a) initial position of the rotor and stator cells; b) rotor cells slides with respect to the stator cells ($\varphi = \omega t$).
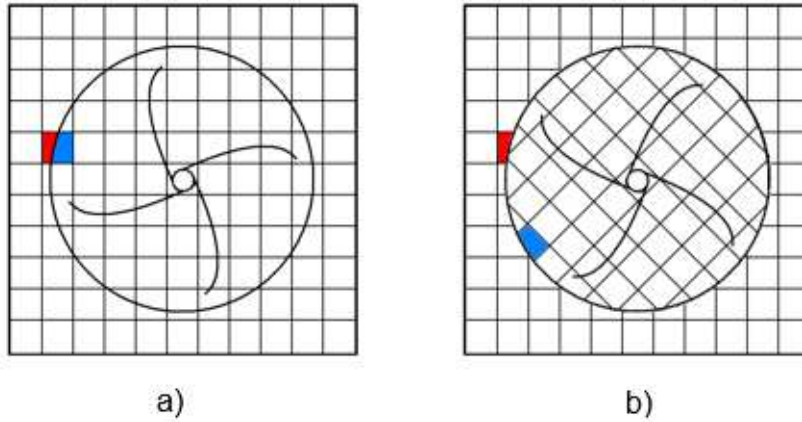
The *Sliding* approach has the following limitations and/or assumptions:

- The solid body motion is not taken into account in heat transfer analysis.

Flows in Porous Media

There are two approaches implemented in Flow Simulation which allow to simulate porous media. The **Porous Medium** is intended to treat some of the model's components as porous media with the fluid flow through them, simulating them as fluid cavities with a distributed resistance to fluid flow. The **Perforated Plate** is an integral model of a thin plate (modeled as infinitely thin) with multiple holes, installed at inlet or outlet model opening or fan.

General Approach

Porous media are treated in Flow Simulation as distributed resistances to fluid flow, so they can not occupy the whole fluid region or fill the dead-end holes.

In addition, if the **Heat conduction in solids** option is switched on, the heat transfer between the porous solid matrix and the fluid flowing through it is also considered. Therefore, the porous matrix act on the fluid flowing through it via the S_i, $S_i u_i$, and (if heat conduction in solids is considered) Q_H terms in equations 2-2 and 2-3 (see "The Navier-Stokes Equations for Laminar and Turbulent Fluid Flows" on page 14), whose components related to porosity are defined as:

$$S_i^{porous} = -k\,\delta_{ij}\,\rho u_j \tag{2-97}$$

$$Q_H^{porosity} = \gamma \cdot \left(T_P - T\right) \tag{2-98}$$

where k is the resistance vector of the porous medium (see below), γ is the user-defined volumetric porous matrix/fluid heat transfer coefficient which can dependent on the flow velocity, T_p is the temperature of the porous matrix, T is temperature of the fluid flowing through the matrix, and the other designations are given in "The Navier-Stokes Equations for Laminar and Turbulent Fluid Flows" on page 14. In addition, the fluid density in equations 2-1 – 2-3 is multiplied by the porosity n of the porous medium, which is the volume fraction of the interconnected pores with respect to the total medium volume.

In the employed porous medium model turbulence disappears within a porous medium and the flow becomes laminar.

If the heat conduction in porous matrix is considered, then, in addition to solving equations 2-1 – 2-3 describing fluid flow in porous medium, the equation 2-50 (see "Conjugate Heat Transfer" on page 35) describing the heat conduction in solids is also considered within the porous medium. In this equation the source Q_H due to heat transfer between the porous matrix and the fluid is defined in the same manner as in the equation 2-96 but with the opposite sign. The values of γ and C for the porous matrix may differ from those of the corresponding bulk

solid material and hence must be specified independently. Density of the solid material is multiplied by the solid volume fraction in the porous matrix, i.e. by $(1 - n)$.

Thermal conductivity of the porous matrix can be specified as anisotropic in the same manner as for the solid material.

The conjugate heat transfer problem in a porous medium is solved under the following restrictions:

- heat conduction in a porous medium not filled with a fluid is not considered,

- porous media are considered transparent for radiation heat transfer,

- heat sources in the porous matrix can be specified in the forms of heat generation rate or volumetric heat generation rate only; heat sources in a form of constant or time-dependent temperature can not be specified.

To perform a calculation in Flow Simulation, you have to specify the following porous medium properties: the effective porosity of the porous medium, defined as the volume fraction of the interconnected pores with respect to the total medium volume. Later on, the **Permeability type** of the porous medium must be chosen among the following:

- isotropic (i.e., the medium permeability is independent of direction),

- unidirectional (i.e., the medium is permeable in one direction only),

- axisymmetrical (i.e., the medium permeability is fully governed by its axial and transversal components with respect to a specified direction),

- orthotropic (i.e., the general case, when the medium permeability varies with direction and is fully governed by its three components determined along three principal directions).

Then you have to specify some constants needed to determine the porous medium resistance to fluid flow, i.e., vector k defined as $k = -\text{grad}(P)/(\rho \cdot V)$, where P, ρ, and V are fluid pressure, density, and velocity, respectively. It is calculated according to one of the following formulas:

- $k = \Delta P \cdot S/(\dot{m} \cdot L)$, where ΔP is the pressure difference between the opposite sides of a sample parallelepiped porous body, $\dot{m}$ is the mass flow rate through the body, S and L are the body cross-sectional area and length in the selected direction, respectively. You can specify ΔP as a function of $\dot{m}$, whereas S and L are constants. Instead of mass flow rate you can specify volume flow rate, v. In this case Flow Simulation calculates $\dot{m} = v \cdot \rho$. All these values do not specify the porous body for the calculation, but its resistance k only.

- $k = \Delta P/(V \cdot L \cdot \rho)$, where ΔP is the pressure difference between the opposite sides of a sample parallelepiped porous body, V and ρ are the fluid velocity and density,

respectively, L is the body length in the selected direction. You can specify ΔP as a function of V, whereas L is constant, since V and ρ are calculated.

- $k = (A \cdot V + B)/\rho$, where V is the fluid velocity, A and B are constants, ρ is the fluid density. Here, only A and B are specified, since V and ρ are calculated.

- $k = 32\mu/(\varepsilon \cdot \rho \cdot D^2)$, where μ and ρ are the fluid dynamic viscosity and density, D is the reference pore size determined experimentally, ε is the porous medium's porosity. Here, only D and ε are specified, since μ and ρ are calculated.

- $k = \mu/(2\rho \cdot D^2) \cdot f(Re)$, differing from the previous formula by the $f(Re)$ factor, yielding a more general formula. Here, in addition to D and ε, $f(Re)$ as a formula dependency is specified: $f(Re) = Re \cdot \varphi(Re) \cdot 1/\varepsilon$, where $\varphi(Re)$ is the flow resistance coefficient of a narrow channel.

The specified reference pore size is also used to calculate the turbulence dissipation after the porous medium. In case of the two first formula, the specified pore size is used to calculate the turbulence dissipation after the porous medium only. By default, it is set to 0.00001 m.

To define a certain porous body, specify both the body position in the model and, if the porous medium has a unidirectional or axisymmetrical permeability, the reference directions in the porous body.

Perforated Plates in Boundary Conditions

A perforated plate is a particular case of porous media treated in Flow Simulation as distributed hydraulic resistances.

Flow Simulation allows to specify a perforated plate as a special feature imposed on a pressure-opening- or fan-type flow boundary condition (see "Internal Flow Boundary Conditions" on page 85) via the **Perforated Plates** option. With the use of this feature, the user specifies a perforated plate via its porosity e (defined as the fraction of the plate area covered by holes) and the holes' shape and size (holes may be either round with specified diameter, or rectangular with specified width and height, or regular polygons with specified side length and number of vertices). Then it can be assigned to any of the above mentioned flow boundary conditions as a distributed hydraulic resistance yielding the additional pressure drop at this boundary

(2-99)

$$\Delta p = -\zeta \cdot \frac{1}{2} \rho u^2$$

where ρ is the fluid density, u is the fluid velocity inside the plate's holes, and ξ is the perforated plate's hydraulic resistance, calculated according to Ref. 8 from the plate porosity ε and the Reynolds number of the flow in the holes:

(2-100)

$$Re = \frac{\rho \cdot u \cdot D_h}{\mu}$$

where $D_h = 4F/\Pi$ is the hole hydraulic diameter defined via the area of a single hole F and its perimeter Π, and μ is the fluid's dynamic viscosity. Please note that the $\xi(Re,\varepsilon)$ dependence taken from Ref 8 and employed in Flow Simulation is valid for non-swirled, normal-to-plate flows only.

Two-phase (fluid + particles) Flows

Flow Simulation has capability to calculate two-phase fluid-particles flows.

Lagrangian Approach for the Post-Processor Simulation

Flow Simulation allows to calculate two-phase flows as a motion of particles in a steady-state flow field where the influence of the particles on the fluid flow (including its temperature) is negligible.

Generally, in this case the particles mass flow rate should be lower than about 30% of the fluid mass flow rate. To simulate dilute two-phase flows, where flows of gases or liquids contaminated with particles, the Lagrangian approach is used.

The particles of a specified (liquid or solid) material and constant mass are assumed to be spherical. The particles trajectories are determined by numerical integration of the equation:

$$(2\text{-}101)$$

$$m_p \frac{d\vec{U}_p}{dt} = \frac{1}{8}\pi\mu d\, Re\, C_D \left(\vec{U} - \vec{U}_p \right) - \frac{1}{6}\pi d^3 \nabla P + m_p \vec{g}$$

where m_p is the particle mass, $\vec{U}_p$ is the particle velocity vector, d is the particle diameter, C_D is the particle resistance coefficient, $\vec{g}$ is the gravitational acceleration vector, and Re is the Reynolds number based on the particle diameter, the relative velocity, and freestream density and viscosity. The particle resistance coefficient is calculated with Henderson's formula (Ref. 6), derived for continuum and rarefied, subsonic and supersonic, laminar, transient, and turbulent flows over the particles, and taking into account the temperature difference between the fluid and the particle. The particles' rotation, their interaction with each other, Brown motion and additional forces are not taken into account.

For subsonic flow ($M \le 1$):

(2-102)

$$C_D^{sub} = 24 \left[Re + S \left(4.33 + \frac{3.65 - 1.53 \frac{T_p}{T}}{1 + 0.353 \frac{T_p}{T}} \exp\left(-0.247 \frac{Re}{S} \right) \right) \right]^{-1} +$$

$$+ \exp\left(-\frac{0.5M}{\sqrt{Re}} \right) \left[\frac{4.5 + 0.38(0.03Re + 0.48\sqrt{Re})}{1 + 0.03Re + 0.48\sqrt{Re}} + 0.1M^2 + 0.2M^8 \right] +$$

$$+ \left[1 - \exp\left(-\frac{M}{Re} \right) \right] 0.6S$$

For supersonic flow ($M \ge 1.75$):

(2-103)

$$C_D^{super} = \frac{0.9 + \frac{0.34}{M_\infty^2} + 1.86 \left(\frac{M_\infty}{Re_\infty} \right)^{0.5} \left[2 + \frac{2}{S_\infty^2} + \frac{1.058}{S_\infty} \left(\frac{T_p}{T} \right)^{0.5} - \frac{1}{S_\infty^4} \right]}{1 + 1.86 \left(\frac{M_\infty}{Re_\infty} \right)^{0.5}}$$

For the flow regimes with Mach between 1 and 1.75, a linear interpolation is to be used:

(2-104)

$$C_D = C_D^{sub}(1, Re) + \frac{4}{3}(M_\infty - 1)\left[C_D^{super}(1.75, Re_\infty) - C_D^{sub}(1, Re) \right]$$

where $C_D^{sub}(1, Re)$ is the drag coefficient calculated using the correlation for subsonic flow with $M = 1$, $C_D^{super}(1.75, Re_\infty)$ is the drag coefficient calculated using the correlation for supersonic flow with $M = 1.75$, M is the Mach number based on the relative velocity between the particle and the fluid flow, $S = M\sqrt{\gamma/2}$ is the molecular speed ratio, γ is the specific heat ratio, T is the fluid temperature in freestream, T_p is the particle temperature, and subscript '∞' refers to freestream conditions.

Thermal energy equation for a particle is given by:

(2-105)

$$m_p C_p \frac{dT_p}{dt} = \pi d \cdot k \cdot Nu(T - T_p)$$

where C_p is the specific heat of particle, T_p is the particle temperature, k is the thermal conductivity of fluid, and Nu is the Nusselt number. The particle/fluid heat transfer coefficient is calculated with the formula proposed in Ref. 1:

(2-106)

$$Nu = \frac{2 + 0.459 Re^{0.55} Pr^{0.33}}{1 + 3.42 \dfrac{M}{RePr} \left(2 + 0.459 Re^{0.55} Pr^{0.33}\right)}$$

If necessary, the gravity is taken into account. Since the particle mass is assumed constant, the particles cooled or heated by the surrounding fluid change their size.

The interaction of particles with the model surfaces is taken into account by specifying either full absorption of the particles (that is typical for liquid droplets impinging on surfaces at low or moderate velocities) or ideal or non-ideal reflection (that is typical for solid particles). The ideal reflection denotes that in the impinging plane defined by the particle velocity vector and the surface normal at the impingement point, the particle velocity component tangent to surface is conserved, whereas the particle velocity component normal to surface changes its sign. A non-ideal reflection is specified by the two particle velocity restitution (reflection) coefficients, e_n and e_τ, determining values of these particle velocity components after reflection, $V_{2,n}$ and $V_{2,\tau}$, as their ratio to the ones before the impingement, $V_{1,n}$ and $V_{1,\tau}$:

(2-107)

$$e_n = \frac{V_{2,n}}{V_{1,n}}, \qquad e_\tau = \frac{V_{2,\tau}}{V_{1,\tau}}$$

As a result of particles impingement on a solid surface, the total erosion mass rate ($R_{\Sigma e}$) and the total accretion mass rate ($R_{\Sigma a}$) are determined as follows:

(2-108)

$$R_{\Sigma e} = \sum_{i=1}^{N} K_i \cdot V_{pi}^b \cdot f_{1i}(\alpha_{pi}) \cdot f_{2i}(d_{pi}) d\dot{m}_{pi}$$

(2-109)

$$R_{\Sigma a} = \sum_{i=1}^{N} M_{pi}$$

where:

N is the number of fractions of particles specified by user as injections in Flow Simulation (the user may specify several fractions of particles, also called injections, so that the particle properties at inlet, i.e. temperature, velocity, diameter, mass flow rate, and material, are constant within one fraction),

i is the fraction number,

M_{pi} is the mass impinging on the model walls in unit time for the i-th particle fraction,

K_i is the impingement erosion coefficient specified by user for the i-th particle fraction,

V_{pi} is the impingement velocity for the i-th particle fraction,

b is the user-specified velocity exponent ($b = 2$ is recommended),

$f_{1i}(\alpha_{pi})$ is the user-specified dimensionless function of particle impingement angle α_{pi},

$f_{2i}(d_{pi})$ is the user-specified dimensionless function of particle diameter d_{pi}.

Free Surface

Flow Simulation allows to model the flow of two immiscible fluids with a free surface. Liquids are considered as immiscible if they are completely insoluble in each other. A free surface is an interface between immiscible fluids, for example, a liquid and a gas (any pair of fluids belonging to gases, liquids or non-Newtonian liquids are allowed excluding gas-gas contact).

However, any phase transitions (including humidity, condensation, cavitation), rotation problems, surface tension and boundary layer on an interface between immiscible fluids are not allowed in this version.

Free surfaces are modeled with the *Volume of Fluid* (VOF) technique by solving a single set of momentum equations and tracking the volume fraction of each of the fluids throughout the domain.

The VOF technique is based on the concept of a fluid volume fraction α_q ($q = 0...N_q - 1$), which must have a value between 0 and 1. In a two-phase system, for example, in mesh cells (control volumes) of liquid $\alpha_0 = 0$ and $\alpha_1 = 1$, while in cells of gas $\alpha_0 = 1$ and $\alpha_1 = 0$. The location of a free surface is where α_q changes from 0 to 1. In each control volume, the volume fractions of all phases sum to unity:

(2-110)

$$\sum_{q=0}^{N_q-1} \alpha_q = 1$$

where N_q is the number of immiscible fluids (component phases).

 Flow Simulation allows to model a two-phase system with $N_q = 2$ only. Pairs of liquid-gas or liquid-liquid are available.

The density in each cell is either purely representative of one of the phases or a mixture of the phases, depending upon the volume fraction values:

$$\rho = \sum_{q=0}^{N_q-1} \alpha_q \rho_q$$

(2-111)

The absence of immiscible fluids mixing means that there is no convective transfer of mass, momentum, energy and other parameters through the free surface. Also there is no diffusion transfer of the mixture components. However there are diffusion fluxes for momentum, energy, and turbulent parameters. It is assumed that velocities, shear stresses, static pressures, temperatures and heat fluxes are equal at the interface between immiscible fluids.

The terms $grad(\rho_q)$ are assumed to be negligible within each fluid. For a gas phase changes in density ρ_q are mainly related to temperature variations, not pressure:

$$\frac{\max(\rho_q)}{\min(\rho_q)} < 3, \quad q = 0..N_q - 1$$

(2-112)

It is assumed that a gas phase flow is low-compressible, that is Mach number $M < 0.3$.

At the free surface the ratio between immiscible fluids densities (or viscosities) may be large, e.g., for water to air the ratio between densities is about 10^3 and for Non-Newtonian liquid to air the ratio between viscosities may be up to 10^{10}.

The volume fraction equation is written as follows:

$$\frac{\partial \alpha_q}{\partial t} + \frac{\alpha_q}{\rho_q} \frac{\partial \rho_q}{\partial t} + \sum_i \frac{\partial \alpha_q u_i}{\partial x_i} = 0, \quad q = 0..N_q - 1$$

(2-113)

Then as a consequence of the volume fraction equations, the conservation law for mass has the following form:

$$\sum_{q=0}^{N_q-1} \left(\frac{\alpha_q}{\rho_q} \frac{\partial \rho_q}{\partial t} \right) + \sum_i \frac{\partial u_i}{\partial x_i} = 0$$

(2-114)

The momentum equation can be written as:

$$(2\text{-}115)$$

$$\rho\frac{\partial u_i}{\partial t} + \rho u_i \sum_{q=0}^{N_q-1}\left(\frac{\alpha_q}{\rho_q}\frac{\partial \rho_q}{\partial t}\right) + \rho\sum_j\frac{\partial}{\partial x_j}\left(u_i u_j\right) + \frac{\partial p}{\partial x_i} =$$

$$= \sum_j\frac{\partial}{\partial x_j}\left(\tau_{ij} + \tau_{ij}^R\right) + S_i, \qquad i = 0,1,2$$

The energy equation can be written as:

$$(2\text{-}116)$$

$$\rho\frac{\partial H}{\partial t} + \rho H \sum_{q=0}^{N_q-1}\left(\frac{\alpha_q}{\rho_q}\frac{\partial \rho_q}{\partial t}\right) + \rho\sum_i\frac{\partial u_i H}{\partial x_i} =$$

$$= \sum_{ij}\frac{\partial}{\partial x_i}\left(u_j(\tau_{ij} + \tau_{ij}^R) + q_i\right) + \frac{\partial p}{\partial t} - \sum_{ij}\tau_{ij}^R\frac{\partial u_i}{\partial x_j} + \rho\varepsilon + \sum_i S_i u_i + Q_H$$

Each immiscible fluid is one substance, not a mixture.

The state equation of immiscible fluids has the following form:

$$(2\text{-}117)$$

$$h = \sum_{q=0}^{N_q-1}\frac{h_q \alpha_q \rho_q}{\rho}$$

The other fluid properties (viscosity, heat conductivity and heat capacity) are defined in similar way:

$$(2\text{-}118)$$

$$\mu = \sum_{q=0}^{N_q-1}\alpha_q \mu_q$$

$$k = \sum_{q=0}^{N_q-1}\alpha_q k_q$$

$$C_p = \sum_{q=0}^{N_q-1}\frac{\alpha_q \rho_q}{\rho} C_{p,q}$$

The free surface position can be restored at any time for visualization. The restored free surface position is not used in the numerical algorithm.

HVAC

Heating, ventilation, and air conditioning (HVAC) is the technology of indoor and vehicular environmental comfort. Its goal is to provide thermal comfort and acceptable indoor air quality. HVAC is an important part of residential structures such as single family homes, apartment buildings, hotels and senior living facilities, medium to large industrial and office buildings such as skyscrapers and hospitals, vehicles such as cars, trains, airplanes, ships and submarines, and in marine environments, where safe and healthy building conditions are regulated with respect to temperature and humidity, using fresh air from outdoors.

Tracer Study

If a gaseous (or liquid) substance (e.g. a contaminant) diffuses in a gaseous (or liquid) fluid (if this fluid flows and carries this substance, this fluid is usually named the carrier fluid) and this substance's mass fraction y in the carrier fluid is too small, i.e. $y \ll 1$, so it can not influence the carrier fluid flow's properties (velocity, pressure, temperature), then distribution of this substance over the computational domain due to carrying it by the fluid flow and its diffusion in this fluid can be calculated with the **Tracer Study** option.

_____ **Restriction** _____

This feature is available for the HVAC module users only.

_____

According to this option, this substance's diffusion is calculated in the previously calculated steady-state or unsteady carrier fluid flow by solving the following Tracer Study equation taking the substance's concentration non-uniformity and the carrier fluid flow's pressure gradient (for gaseous fluids only) into account:

(2-119)

$$\frac{\partial \rho y}{\partial t} + \frac{\partial}{\partial x_i}\left[\rho y u_i - \frac{\rho R T}{pm}\left(\frac{\mu}{Pr \cdot Le} + \frac{\mu_t}{Pr_t \cdot Le_t}\right)\frac{\partial y}{\partial x_i}\right] =$$

$$= \frac{\partial}{\partial x_i}\frac{m_1 m_2}{m^2}\left[\frac{\rho y v_1 - y}{p}\left(\frac{\mu}{Pr \cdot Le} + \frac{\mu_t}{Pr_t \cdot Le_t}\right)\frac{\partial p}{\partial x_i}\right]$$

where:

ρ is the carrier fluid and the substance's mixture density (since $y \ll 1$, ρ can be considered as the carrier fluid's density),

t is time,

x_i is the i-th component of the used coordinate system,

u_i is the i-th component of the carrier fluid's velocity (the substance has the same velocity),

p is the carrier fluid's static pressure,

R is the universal gas constant,

m is the carrier fluid and the substance's mixture molar mass,

m_1 is the substance's molar mass,

m_2 is the carrier fluid's molar mass,

v_1 is the substance's specific volume,

μ is the carrier fluid's laminar viscosity,

μ_t is the carrier fluid's turbulent viscosity,

Pr, Pr_t are the carrier fluid's laminar and turbulent Prandtl numbers,

Le, Le_t are the carrier fluid's laminar and turbulent Lewis numbers.

The Tracer Study equation is solved after finishing the carrier fluid flow calculation, i.e. in the postprocessor, either as steady-state or as unsteady (time-dependent, with its own time).

The substance's physical properties required to solve the Tracer Study equation are specified in the **Engineering Database** as the substance's molar mass (m_1), binary (this substance in the carrier fluid) diffusion coefficient $D = \mu/(Pr{\cdot}Le)$ ($D_t = \mu_t/(Pr_t{\cdot}Le_t) = D$ is assumed) or Lewis number Le ($Le_t = Le$, $Pr_t = Pr$, $\mu_t = \mu$ are assumed), and saturation pressure curve vs. temperature (optional).

The Tracer Study equation is solved in the computational domain (or its subdomain) with the user-specified boundary conditions, initial conditions, and volume sources of the substance carried by the fluid flow. The substance's boundary conditions are superimposed on the carrier fluid flow boundary conditions and consist of the substance's surface sources and wall condition specified on the user-selected surfaces. The substance's surface sources are specified by the substance's mass fraction, or mass flow rate, or specific (i.e. per unit surface area) mass flow rate, or evaporation on the user-selected walls and/or by the substance's mass fraction on the user-selected inlet flow openings. At that the substance's evaporation can be specified (by enabling the **Liquid Surface** option) only for those substances whose saturation pressure curve has been specified in the **Engineering Database**. The substance's wall condition is the

substance's condensation calculated on the user-selected model walls (only for those substances whose saturation pressure curve has been specified in the **Engineering Database**).

The substance's volume sources are specified by the substance's mass fraction, or mass generation rate (total over the source volume), or volumetric (i.e. per unit volume) mass generation rate in the user-selected fluid volumes.

To solve the Tracer Study equation the user must also specify the substance's initial conditions in the form of the substance's mass fraction (it may be zero) distribution over the computational domain (or its subdomain).

As a result of the Tracer Study calculation you can see the substance's mass fraction, mass flow rates (through the user-selected surfaces or in the user-selected volumes), **CRE** and **LAQI** (see page 83 and page 82) distributions over the computational domain (or its subdomain).

The **Tracer Study** can be performed simultaneously for several substances in the same fluid flow carrying them.

Local Mean Age

Local mean age (**LMA**) is the average time τ for fluid to travel from the selected inlet opening to the point with taking both the velocity and the diffusion into account.

It is determined by solving the following equation:

(2-120)

$$\sum_{i=1}^{3} \frac{\partial}{\partial x_i}\left(\rho\tau u_i - \left(\frac{\mu}{\sigma} + \frac{\mu_t}{\sigma_t}\right)\frac{\partial\tau}{\partial x_i}\right) = \rho$$

where x_i is the i-th coordinate, ρ is the density, u_i is the i-th velocity component, μ is the dynamic viscosity coefficient, μ_t is the turbulent eddy viscosity coefficient, σ and σ_t are the laminar and turbulent Schmidt numbers. The equation is solved under the $\tau = 0$ boundary condition on the inlet opening.

Dimensionless LMA (τ^*) is the LMA divided by the V/Q ratio, where V is the computational domain fluid volume, Q is the volume flow rate of the air entering this fluid volume.

LACI (Local Air Change Index) is equal to $1/(\tau^*)$, i.e. reciprocal to the **Dimensionless LMA**.

By default, the calculation of these parameters is disabled. You can enable the calculation of **LMA**, **Dimensionless LMA** and **LACI** in the **Calculation Control Options**.

Comfort Parameters

 Flow Simulation has the capability to predict the general thermal sensation, degree of discomfort (thermal dissatisfaction) of people exposed to moderate thermal environments and estimate air quality by calculating comfort criteria.

───── **Restriction** ─────────────────────────────

This feature is available for the HVAC module users only.

See Ref. 19, Ref. 9, and Ref. 10. These criteria are used when designing occupied spaces and their HVAC systems and are intended to determine whether environmental conditions are acceptable in terms of general thermal comfort and air quality or represent discomfort. The calculation of the comfort criteria assumes that the analyzed fluid is Air.

Mean Radiant Temperature (MRT) is the uniform surface temperature of an imaginary black enclosure in which an occupant would exchange the same amount of radiant heat as in the actual non-uniform space.

The mean radiant temperature (T_r) is defined as follows:

(2-121)

$$T_r^4 = \frac{1}{4\sigma}\int I_{dif}(\Omega)d\Omega + \frac{1}{4\sigma}\sum I_{sun}$$

where I_{dif} is the intensity of the diffuse (thermal) radiation ($W/m^2/rad$), I_{sun} is the intensity of the solar radiation (W/m^2), σ is the Stefan-Boltzmann constant.

To calculate the **Mean Radiant Temperature**, it is assumed that the emissivity of all the surfaces within the computational domain equals to unity.

Operative Temperature is the uniform temperature of an imaginary black enclosure, in which an occupant would exchange the same amount of heat by radiation plus convection as in the actual non-uniform environment.

The operative temperature (T_c) is defined as follows (Ref. 9):

(2-122)

$$T_c = \frac{T_r + T\sqrt{10V}}{1 + \sqrt{10V}}$$

where T_r is the mean radiant temperature (°C), T is the fluid temperature (°C), V is the fluid velocity (m/s).

Predicted Mean Vote (PMV) is an index that predicts the mean value of the votes of a large group of persons on the 7-point thermal sensation scale, based on the heat balance of the human body. Thermal balance is obtained when the internal heat production in the body is equal to the loss of heat to the environment. In a moderate environment, the human thermoregulatory system will automatically attempt to modify skin temperature and sweat secretion to maintain heat balance (Ref. 10).

cold	cool	slightly cool	neutral	slightly warm	warm	hot
-3	-2	-1	0	+1	+2	+3

The *PMV* is defined as follows:

(2-123)

$$PMV = (0.303e^{-0.036M} + 0.028) \cdot$$
$$\cdot \left\{ \begin{array}{l} (M-W) - 3.05 \cdot 10^{-3}[5733 - 6.99(M-W) - p_a] - \\ -0.42[(M-W) - 58.15] - 1.7 \cdot 10^{-5}M(5867 - p_a) - 0.0014M(34 - T_a) \\ -3.96 \cdot 10^{-8}f_{cl}[(T_{cl} + 273)^4 - (T_r + 273)^4] - f_{cl}h_c(T_{cl} - T_a) \end{array} \right\}$$

where:

(2-124)

$$T_{cl} = 35.7 - 0.028(M-W)$$
$$- I_{cl}\left\{ 3.96 \cdot 10^{-8} f_{cl}\left[(T_{cl} + 273)^4 - (T_r + 273)^4\right] + f_{cl}h_c(T_{cl} - T_a) \right\}$$

(2-125)

$$h_c = \max\left\{ 2.38(T_{cl} - T_a)^{0.25}, 12.1\sqrt{V} \right\}$$

$$f_{cl} = \begin{cases} 1.00 + 1.29I_{cl}, \text{ for } I_{cl} \le 0.078 \dfrac{m^2}{kW} \\ 1.05 + 0.645I_{cl}, \text{ for } I_{cl} > 0.078 \dfrac{m^2}{kW} \end{cases}$$

where:

M is the metabolic rate (W/m^2 of the body area). It is the rate of transformation of chemical energy into heat and mechanical work by metabolic activities within an organism (by default it is set to 70 W/m^2);

W is the external work (W/m^2 of the body area). It accounts for the effective mechanical power (by default it is set to 0 W/m^2);

I_{cl} is the clothing thermal resistance (m²K/W). It is the resistance to sensible heat transfer provided by a clothing ensemble. The definition of clothing insulation relates to heat transfer from the whole body and, thus, also includes the uncovered parts of the body, such as head and hands. The typical values of thermal resistance for a certain clothing ensemble can be found in Ref. 10 (by default it is set to 0.11 m²K/W);

f_{cl} is the ratio of clothed surface area to nude surface area;

T_a is the air temperature (°C);

T_r is the mean radiant temperature (°C);

V is the relative air velocity (m/s);

p_a is the water vapor partial pressure (Pa) calculated in accordance with the saturation curve, the air temperature T_a and the **Relative Humidity** (see "Water vapor condensation and relative humidity" on page 28);

h_c is the convective heat transfer coefficient (W/m²/K);

T_{cl} is the clothing surface temperature (°C).

Predicted Percent Dissatisfied (PPD) is an index that provides information on thermal discomfort or thermal dissatisfaction by predicting the percentage of people likely to feel too warm or too cool in a given environment (Ref. 10). It can be obtained from the *PMV* using the following equation:

$$(2\text{-}126)$$

$$PPD = 100 - 95 \exp\left(-0.03353 PMV^4 - 0.2179 PMV^2\right)$$

Draught Rate is the percentage of people predicted to be bothered by draught. The draught rate (*DR*) is defined as follows:

$$(2\text{-}127)$$

$$DR = \left(34 - T_{a,l}\right)\left(\bar{v}_{a,l} - 0.05\right)^{0.62}\left(0.37 \cdot \bar{v}_{a,l} \cdot Tu + 3.14\right)$$
$$\text{for } \bar{v}_{a,l} < 0.05\,\text{m/s:} \quad \bar{v}_{a,l} = 0.05\,m/s$$
$$\text{for } T_{a,l} > 34°C: \quad T_{a,l} = 34°C$$
$$\text{for } DR > 100\%: \quad DR = 100\%$$

where:

$T_{a,l}$ is the local air temperature (°C), 20 °C to 26 °C;

$v_{a,l}$ is the local mean air velocity (m/s), < 0.5 m/s;

Tu is the local turbulence intensity (%), 10 % to 60 %. If the **Laminar Only** flow type is selected, Tu equals 40 %.

This model applies to people at light, mainly sedentary activity with a thermal sensation for the whole body close to neutral and for prediction of draught at the neck. At the level of arms and feet, the model could overestimate the predicted draught rate. The sensation of draught is lower at activities higher than sedentary (> 1.2 met) and for people feeling warmer than neutral.

Draft Temperature is the difference in temperature between any point in the occupied zone and the control condition. '*Draft*' is defined as any localized feeling of coolness or warmth of any portion of the body due to both air movement and air temperature, with humidity and radiation considered constant (Ref. 19).

The draft temperature (T_d) is defined as follows:

(2-128)

$$T_d = T - T_m - 7.6553 \, (V - 0.1524)$$

where:

T is the local fluid temperature (°C);

T_m is the average fluid temperature within the control space (°C);

V is the local fluid velocity (m/s).

Air Diffusion Performance Index (ADPI) is the percentage of the space in which the air speed is less than 0.35 m/s and the Draft Temperature falls between -1.7 ºC and 1.1 ºC (Ref. 19).

If the **Draft Temperature** or **ADPI** are calculated in **Volume Parameters**, then the '*control space*' will correspond to the specified volume region. In all other cases the '*control space*' corresponds to the whole computational domain.

Contaminant Removal Effectiveness (CRE) is an index that provides information on the effectiveness of a ventilation system in removing contaminated air from the whole space. For a perfect mixing system $CRE = 1$. Values above 1 are good, values below 1 are poor.

This parameter is only available if more than one fluid is present in the control space. Its value is defined as follows:

(2-129)

$$CRE = \frac{C_e}{<C>}$$

where:

C_e is the bulk average mass fraction of the contaminant calculated over all faces through which the fluids outflow from the computational domain;

$<C>$ is the bulk average mass fraction of the contaminant calculated over the whole computational domain.

Local Air Quality Index (LAQI) is an index that provides information on the effectiveness of a ventilation system in removing contaminated air from a point. For a perfect mixing system $LAQI = 1$. For other systems, the higher the value at a point, the better the capability of the ventilation system in removing contaminated air from that point. This parameter is only available if more than one fluid is present in the control space. Its value is defined as follows:

(2-130)

$$LAQI = \frac{C_e}{C}$$

where:

C_e is the bulk average mass fraction of the contaminant calculated over all faces through which the fluids outflow from the computational domain;

C is the mass fraction of the contaminant at a point.

The **Flow Angle** shows the flow deviation from the design flow direction. Consider one of the axis of selected coordinate system as the design flow direction, the results can then be viewed as the deviation from the design. Typically, flow angles of less than 15° might be considered as good. The flow angle components (α_i) are defined as follows:

(2-131)

$$\alpha_x = \arccos(V_x/V)$$
$$\alpha_y = \arccos(V_y/V)$$
$$\alpha_z = \arccos(V_z/V)$$

where V_x, V_y, V_z are the X, Y, and Z components of the fluid velocity and V is the absolute value of the fluid velocity vector.

This chapter describes flow boundaries conditions implemented in Flow Simulation and a wide range of the engineering device models specifying conditions at the model surfaces and in solid components.

Internal Flow Boundary Conditions

For internal flows, i.e., flows inside models, Flow Simulation offers the following two options of specifying the flow boundary conditions: manually at the model inlets and outlets (i.e. model openings), or to specify them by transferring the results obtained in another Flow Simulation calculation in the same coordinate system (if necessary, the calculation can be performed with another model, the only requirement is the flow regions at the boundaries must coincide).

With the first option, all the model openings are classified into '*pressure*' openings, '*flow*' openings, and '*fans*', depending on the flow boundary conditions which you intend to specify on them.

A '*pressure*' opening boundary condition, which can be static pressure, or total pressure, or environment pressure is imposed in the general case when the flow direction and/or magnitude at the model opening are not known a priori, so they are to be calculated as part of the solution. Which of these parameters is specified depends on which one of them is known. In most cases the static pressure is not known, whereas if the opening connects the computational domain to an external space with known pressure, the total pressure at the opening is known. The Environment pressure condition is interpreted by Flow Simulation as a total pressure for incoming flows and as a static pressure for outgoing flows. If, during calculation, a vortex crosses an opening with the Environment pressure condition specified at it, this pressure

considered as the total pressure at the part of opening through which the flow enters the model and as the static pressure at the part of opening through which the flow leaves the model.

Note that when inlet flow occurs at the '*pressure*' opening, the temperature, fluid mixture composition and turbulence parameters have to be specified also.

A '*flow*' opening boundary condition is imposed when dynamic flow properties (i.e., the flow direction and mass/volume flow rate or velocity/ Mach number) are known at the opening. If the flow enters the model, then the inlet temperature, fluid mixture composition and turbulence parameters must be specified also. The pressure at the opening will be determined as part of the solution. For supersonic flows the inlet pressure must be specified also.

A '*fan*' condition simulates a fan installed at a model opening. In this case the dependency of volume flow rate on pressure drop over the fan is prescribed at the opening. These dependencies are commonly provided in the technical documentation for the fans being simulated.

With the second option, you specify the boundary conditions by transferring the results obtained in another Flow Simulation calculation in the same coordinate system. If necessary, the calculation can be performed with another model, the only requirement is the flow regions at the boundaries must coincide. At that, you select the created boundary conditions' type: either as for external flows (so-called '*ambient*' conditions, see "External Flow Boundary Conditions" on page 86), or as for '*pressure*' or '*flow*' openings, see above. If a conjugate heat transfer problem is solved, the temperature at the part of the boundary lying in a solid body is transferred from the other calculation.

Naturally, the flow boundary conditions specified for an internal flow problem with the first and/or second options must be physically consistent with each other, so it is expedient to specify at least one '*pressure*'-type boundary condition and at least one '*flow*'-type boundary condition, if not only '*ambient*' boundary conditions are specified.

If one or several non-intersecting axisymmetric rotating regions (local rotating reference frames) are specified, the flow parameters are transferred from the adjacent fluid regions and circumferentially averaged over rotating regions' boundaries as boundary conditions.

External Flow Boundary Conditions

For external problems such as flow over an aircraft or building, the parameters of the external incoming flow (so-called '*ambient*' conditions) must be defined. Namely the velocity, pressure, temperature, fluid mixture composition and turbulence parameters must be specified. Evidently, during the calculation they can be partly violated at the flow boundary lying downstream of the model.

Wall Boundary Conditions

In Flow Simulation the '*default*' velocity boundary condition at solid walls corresponds to the well-known '*no-slip*' condition. The solid walls are also considered to be '*impermeable*'. In addition, the wall surface's translation and/or rotation (without changing the model's geometry) can be specified. If a calculation is performed in a rotating coordinate system, then some of the wall surfaces can be specified as stationary, i.e. a backward rotation in this coordinate system (without changing the model geometry).

Flow Simulation also provides the **Ideal Walls** condition that corresponds to the well-known '*slip*' condition. For example, **Ideal Walls** can be used to model planes of flow symmetry.

If the flow of non-Newtonian liquids is considered, then the following slip condition at all solid walls can be specified for each non-Newtonian liquid in the project separately: if the shear stress τ exceeds the yield stress value $\tau_{0,slip}$, then a slip velocity v_{slip} is determined from:

(3-1)

$$v_{slip} = C_1 \left(\tau - \tau_{0,slip} \right)^{C_2}$$

where C_1 and C_2, as well as $\tau_{0,slip}$, are specified by user.

If conjugate heat transfer in fluid and solid media is not considered, one of the following boundary conditions can be imposed at solid walls: either the wall temperature:

(3-2)

$$T = T_w$$

or the heat flux,

(3-3)

$$q = q_w$$

being positive for heat flows from fluid to solid, equal to zero for adiabatic (heat-insulated) walls, and negative for heat flows from solid to fluid.

When considering conjugate heat transfer in fluid and solid media, the heat exchange between fluid and solid is calculated by Flow Simulation, so heat wall boundary conditions are not specified at the walls.

Periodic Boundary Conditions

The '*periodicity*' condition may be applied if the model consist of identical geometrical features arranged in periodic linear order. Periodic boundary conditions are specified at the pair of computational domain boundaries for the selected direction in which a geometrical feature or a group of features repeats regularly over distance. Periodic boundary conditions allows to reduce

the analysis time by calculating the fluid flow only for a small group of identical geometrical features or even just for one feature, but taking into account influence of other identical features in the pattern. Please note that the number of basic mesh cells along the direction in which the '*periodicity*' condition is applied must be no less than five.

Heat Pipes

'*Heat pipe*' is a device (of arbitrary form: a tube, a plate, etc.) transferring heat from its hotter surface to its colder surface due to evaporating a liquid (water, etc.) and condensing its vapor in this device's inner hollow. This liquid evaporates near this hollow's hotter surface and its vapor condensates near this hollow's colder surface. The condensed liquid then returns to the hotter end due to gravity or by a wick. If such a device operates under its design conditions, the heat transfer rate provided by this device exceeds substantially any value achievable by a solid body of similar dimensions.

_____ **Restriction** _____

This feature is available for the Electronics Cooling module users only.

In Flow Simulation a **Heat Pipe** is modeled simplistically as an extremely heat-conducting body. To model real efficiency of a heat pipe, a non-zero thermal resistance can be assigned to it.

Thermal Joints

A '*thermal joint*' model implemented in Flow Simulation can be used to simplify the heat transfer simulation from one (hot) surface S_1 to the other (cold) surface S_2 with the actual thermal joint geometry excluded from the analysis.

Here, to simulate heat transfer between these surfaces, the value of contact resistance θ (or its reciprocal, heat transfer coefficient α) is used. The resulting heat transfer rate from the hot surface to the cold surface is calculated as $Q = \alpha(T_1 - T_2)$, where T_1 and T_2 are the temperatures of the surfaces S_1 and S_2.

_____ **Note** _____

Notice that once you select the faces to specify thermal joint between them, these faces become thermally insulated in respect to the surrounding medium and only participate in the heat exchange between each other.

Two-resistor Components

A two-resistor model implemented in Flow Simulation can be used for simplified solving of heat transfer problems in an electronic device with small electronic packages (chips, etc.).

_____ **Restriction** _____

This feature is available for the Electronics Cooling module users only.

_____

A small package is considered as a flat solid plate, which is mounted on the printed circuit board (see Figure 3-1). The compact model consists of three nodes: _Junction_, _Case_ and _Board_. These are connected together by two thermal resistors which are the user-specified values of the junction-to-board θ_{JB} (from the junction to the board on which it is mounted) and junction-to-case θ_{JC} (from the junction to the top surface of the package) thermal resistances (in K/W in SI).

$$(3-4)$$

$$\theta_{JB} = (T_J - T_B)/P_H$$

$$(3-5)$$

$$\theta_{JC} = (T_J - T_C)/P_H$$

where T_J is the junction temperature, T_B is the board temperature and T_C is the case temperature, measured at the package top surface, P_H is the heat generation rate which produced the change in junction temperature and applied at the _Junction_.

The _Board_ is considered to be in direct thermal contact with the local environment immediately below the footprint of the package; normally the printed circuit board (_PCB_).

The _Case_ is considered to be in direct thermal contact with the local environment immediately above the top of the package (normally air, or a thermal interface material used in conjunction with a heat sink).

The _Junction_ plate is modeled as a high conductivity body with heat-insulating side walls, so the only surfaces through which heat can be exchanged with the environment are the top and bottom surfaces.

Figure 3-1. Flow Simulation representation of the two-resistor model: 1) High Conductivity Block (Junction), 2) Top Plate (Case, θ_{JC}), 3) Bottom Plate (Board, θ_{JB}), 4) Adiabatic walls, 5) PCB.
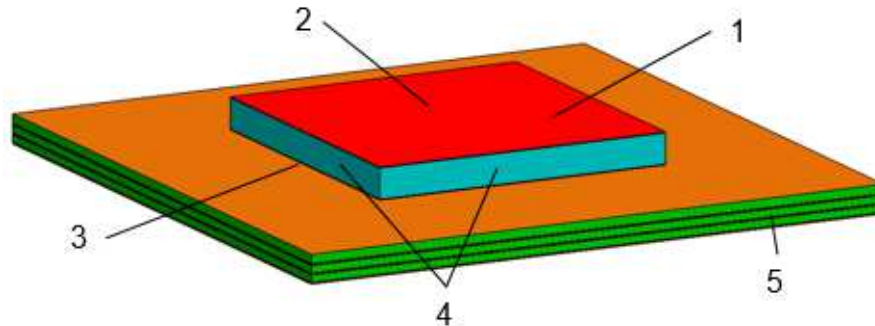
Printed Circuit Boards

Printed Circuit Board (PCB) is concerned as a special case of solid material having anisotropic thermal conductivity. The integral characteristics of a PCB, i.e. its effective density, specific heat, and components of thermal conductivity, are calculated on the basis of the PCB structure.

___ **Restriction** _____

This feature is available for the Electronics Cooling module users only.

_____

To define a PCB characteristics, you have to specify density, specific heat, and thermal conductivity for both dielectric (denoted by '*D*' index) and conductor (denoted by '*C*' index) materials of the PCB, and describe its internal structure in one of the provided types:

- **Conductor Volume Fraction** type requires the specification of the volume fraction of conductor material in the PCB *A*. The in-plane (planar) conductivity K_{in} and the through-plane (normal) conductivity K_{thro} are calculated as follows:

(3-6)

$$K_{in} = \frac{A}{100} \cdot K_C + \left(1 - \frac{A}{100}\right) \cdot K_D \;, \quad \frac{1}{K_{thro}} = \frac{\frac{A}{100}}{K_C} + \frac{1 - \frac{A}{100}}{K_D}$$

The effective density ρ and the effective specific heat C of the board are calculated as follows:

(3-7)

$$\rho = \frac{\rho_C \cdot V_C + \rho_D \cdot V_D}{V} \;, \quad C = \frac{C_C \cdot \rho_C \cdot V_C + C_D \cdot \rho_D \cdot V_D}{\rho \cdot V}$$

where V is the total volume of the PCB, $V_C = V \cdot \dfrac{A}{100}$ is the volume of the conductor material in the PCB, $V_D = V \cdot \left(1 - \dfrac{A}{100}\right)$ is the volume of the dielectric material in the PCB;

- **Board Mass** type allows to calculate the fraction of conductor material A in PCB by using the PCB total mass M and PCB total volume V:

$$A = \frac{\rho - \rho_C}{\rho_C - \rho_D} \cdot 100 \tag{3-8}$$

The effective density ρ is calculated as follows:

$$\rho = \frac{M}{V} \tag{3-9}$$

The in-plane (planar) conductivity K_{in} and the through-plane (normal) conductivity K_{thro} are calculated in the same manner;

- **Layer Definition** type implies that you must specify the PCB total thickness t and the number of conducting layers n_C. Also for each conducting layer you must specify the volume fraction of conductor material in the layer A_i and the layer thickness t_{Ci}. The in-plane (planar) conductivity K_{in} and the through-plane (normal) conductivity K_{thro} are calculated as follows:

$$K_{in} = \frac{\displaystyle\sum_{i=1}^{n_C} t_{Ci}\left[\frac{A_i}{100} \cdot K_C + \left(1 - \frac{A_i}{100}\right) \cdot K_D\right] + \left(t - \sum_{i=1}^{n_C} t_{Ci}\right) \cdot K_D}{t} \tag{3-10}$$

$$K_{thro} = \frac{t}{\displaystyle\sum_{i=1}^{n_C} t_{Ci} \cdot \left(\frac{\dfrac{A_i}{100}}{K_C} + \frac{\left(1 - \dfrac{A_i}{100}\right)}{K_D}\right) + \frac{\left(t - \displaystyle\sum_{i=1}^{n_C} t_{Ci}\right)}{K_D}} \tag{3-11}$$

The effective density ρ and the effective specific heat C of the board are calculated as follows:

(3-12)

$$\rho = \frac{\sum_{i=1}^{n_C} t_{Ci}\left[\frac{A_i}{100}\cdot \rho_C + \left(1-\frac{A_i}{100}\right)\cdot \rho_D\right] + \left(t - \sum_{i=1}^{n_C} t_{Ci}\right)\cdot \rho_D}{t}$$

(3-13)

$$C = \frac{\sum_{i=1}^{n_C} t_{Ci}\left[\frac{A_i}{100}\cdot \rho_C \cdot C_C + \left(1-\frac{A_i}{100}\right)\cdot \rho_D \cdot C_D\right] + \left(t - \sum_{i=1}^{n_C} t_{Ci}\right)\cdot \rho_D \cdot C_D}{\rho \cdot t}$$

The fraction of conductor material A in PCB is calculated as follows:

(3-14)

$$A = \frac{\sum_{i=1}^{n_C} t_{Ci}\cdot \frac{A_i}{100}}{t}\cdot 100$$

Thermoelectric Coolers

Thermoelectric cooler (TEC) is a flat sandwich consisting of two plates covering a circuit of p-n semiconductor junctions inside.

When a *direct electric current* (DC) i runs through this circuit, in accordance with the Peltier effect the $a\cdot i\cdot T_c$ heat, where a is the Seebeck coefficient, T_c is the TEC's '*cold*' surface temperature, is pumped from the TEC's '*cold*' surface to its '*hot*' surface (the '*cold*' and '*hot*' sides are determined from the DC direction). This heat pumping is naturally accompanied by the Joule (ohmic) heat release at both the TEC surfaces and the heat transfer from the hotter side to the colder (reverse to the Peltier effect). The ohmic heat release is defined as $R\cdot i^2/2$, where R is the TEC's electric resistance, while the heat transfer is defined as $k\cdot \Delta T$, where k is the TEC's thermal conductivity, $\Delta T = T_h - T_c$, T_h is the TEC's '*hot*' surface temperature. The net heat transferred from the TEC's '*cold*' surface to its '*hot*' surface, Q_c, is equal to:

(3-15)

$$Q_c = a\cdot i\cdot T_c - R\cdot i^2/2 - k\cdot \Delta T$$

Correspondingly, the net heat released at the TEC's '*hot*' surface, Q_h, is equal to:

(3-16)

$$Q_h = a\cdot i\cdot T_h + R\cdot i^2/2 - k\cdot \Delta T$$

In Flow Simulation a TEC is specified by selecting a flat plate (box) in the model, assigning its '*hot*' face, and applying one of the TECs already defined by user in the **Engineering Database**.

Figure 3-2. Flow Simulation representation of the Thermoelectric Cooler model: 1) object being cooled, 2) TEC, 3) heat sink, 4) cold side, 5) hot side.
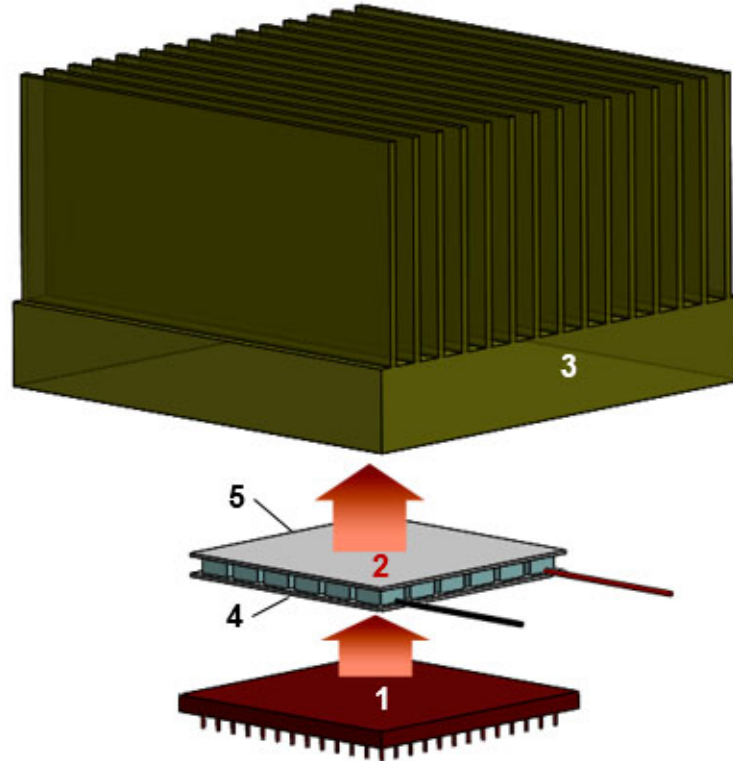
The following characteristics of TEC are specified in the **Engineering Database**:

- **Maximum pumped heat** is the maximum heat $Q_{c,max}$ transferred at this i_{max} from the '*cold*' face to the '*hot*' face at temperature difference between these faces $\Delta T = 0$;

- **Maximum temperature drop** is the maximum temperature difference ΔT_{max} between the '*hot*' and '*cold*' faces attained at the heat transferred between these faces $Q_c = 0$;

- **Maximum current strength** is the maximum DC current, i_{max};

- **Maximum voltage** is the voltage V_{max} corresponding to i_{max}.

All of these characteristics are specified for two T_h values, in accordance with the information usually provided by the TEC suppliers. Proceeding from these characteristics, the $a(T)$, $R(T)$, and $k(T)$ linear functions are determined. The functional boundary conditions are specified automatically on the TEC's '*cold*' and '*hot*' surfaces, which must be free from other boundary conditions.

The temperature solution inside the TEC and on its surfaces is obtained using a special procedure differing from the standard Flow Simulation calculation procedure for heat conduction in solids.

The TEC's '*hot*' face must be in contact with other solids, i.e it must not be in contact with any fluid. In addition, it is required that the obtained TEC solution, i.e. T_h and ΔT, lie within the TEC's operating range specified by its manufacturer.

Chapter 4
Numerical Solution Technique

The numerical solution technique employed in Flow Simulation is robust and reliable, so it does not require any user knowledge about the computational mesh and the numerical methods employed. But sometimes, if the model and/or the problem being solved are too complicated, so that the Flow Simulation standard numerical solution technique requires extremely high computer resources (memory and/or CPU time) which are not available, it is expedient to employ Flow Simulation options which allow the adjustment of the automatically specified values of parameters governing the numerical solution technique. To employ these options properly and successfully, take into account the information presented below about Flow Simulation' numerical solution technique.

Briefly, Flow Simulation solves the governing equations with a discrete numerical technique based on the *finite volume* (FV) method. Cartesian rectangular coordinate system is used. To obtain space discretization, the axis-oriented rectangular grid is used far from a geometry boundary. Thus, the *control volumes* (i.e. *mesh cells*) are rectangular parallelepipeds. Near the geometry boundary Cartesian cut cells approach is used. According to this approach, the near-boundary mesh is obtained from the original background Cartesian mesh by cutting original parallelepiped cells that intersect the geometry (see Figure 4-1). Consequently, the near-boundary cells are polyhedrons with both axis-oriented and arbitrary oriented plane faces in this case. Thus, Flow Simulation combines advantages of approaches based on regular grids and ones with highly accurate representation of geometry boundaries.

Also the local refinement of mesh is used in Flow Simulation to take into account geometry and solution peculiarities. It is usually exploited at the solid/fluid interface, in regions of high gradients, and so on.

All physical parameters are referred to control volume mass centers. Following the FV approach, the direct discretization of the integral form of the conservation laws is used. It guarantees that the basic quantities mass, momentum and energy remain conserved in the discrete representation.

The spatial derivatives are approximated with implicit difference operators of second-order accuracy. The time derivatives are approximated with an implicit first-order Euler scheme. The numerical viscosity generated by the discretization error of the scheme is small enough and allows obtaining adequately accurate results in the most practical cases.

Computational Mesh

Flow Simulation computational approach is based on locally refined rectangular mesh near geometry boundaries.

The mesh cells are rectangular parallelepipeds with faces orthogonal to the specified axes of the Cartesian coordinate system. However, near the boundary mesh cells are more complex. The near-boundary cells are portions of the original parallelepiped cells that cut by the geometry boundary. The curved geometry surface is approximated by set of polygons which vertexes are surface's intersection points with the cells' edges. These flat polygons cut the original parallelepiped cells. Thus, the resulting near-boundary cells are polyhedrons with both axis-oriented and arbitrary oriented plane faces in this case (see Figure 4-1). The original parallelepiped cells containing boundary are split into several control volumes that are referred to only one fluid or solid medium. In the simplest case there are only two control volumes in the parallelepiped, one is solid and another is fluid.

Figure 4-1. Computational mesh near the solid/fluid interface: black circles on a white background – centers of fluid control volumes, black circles on a gray background – centers of solid control volumes, red dashed line – original curved geometry boundary.
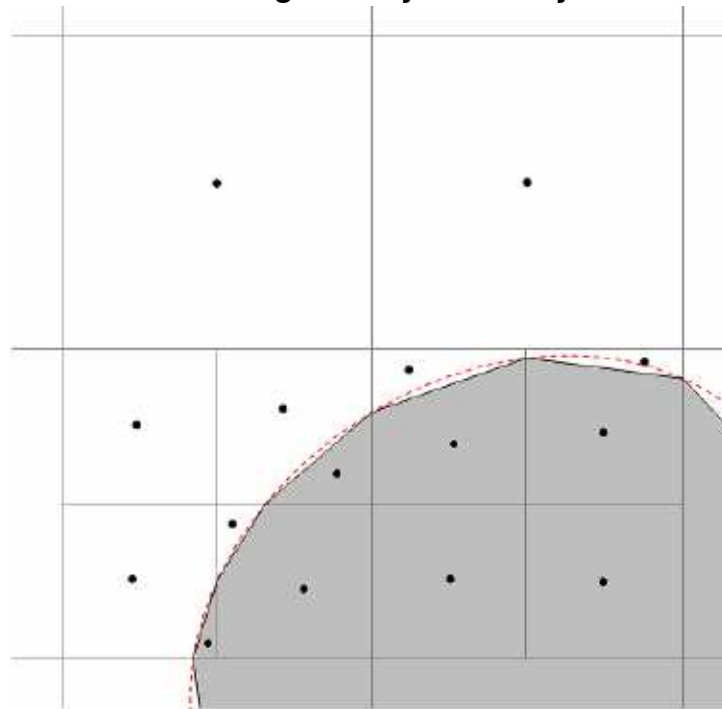
The rectangular **Computational Domain** is automatically constructed (may be changed manually), so it encloses the solid body and has the boundary planes orthogonal to the specified axes of the Cartesian coordinate system. Then, the computational mesh is constructed in the following several stages.

First of all, a **Basic Mesh** is constructed. For that, the computational domain is divided into slices by the basic mesh planes, which are evidently orthogonal to the axes of the Cartesian coordinate system. The user can specify the number and spacing of these planes along each of the axes. The so-called **Control Planes** whose position is specified by user can be among these planes also. The basic mesh is determined solely by the computational domain and does not depend on the solid/fluid interfaces.

Then, the basic mesh cells intersecting with the solid/fluid interface are split uniformly into smaller cells in order to capture the solid/fluid interface with mesh cells of the specified size (with respect to the basic mesh cells). The following procedure is employed: each of the basic mesh cells intersecting with the solid/fluid interface is split uniformly into 8 child cells; each of the child cells intersecting with the interface is in turn split into 8 cells of next level, and so on, until the specified cell size is attained.

At the next stage of meshing, the mesh obtained at the solid/fluid interface with the previous procedure is refined (i.e. the cells are split further or probably merged) in accordance with the solid/fluid interface curvature. The criterion to be satisfied is established as follows: the maximum angle between the normals to the surface inside one cell should not exceeds certain threshold, otherwise the cell is split into 8 cells.

Finally, the mesh obtained with these procedures is refined in the computational domain to satisfy the so-called narrow channel criterion: for each cell lying at the solid/fluid interface, the number of the mesh cells lying in the fluid region along the line normal to the solid/fluid interface and starting from the center of this cell must not be less than the criterion value. Otherwise each of the mesh cells on this line is split into 8 child cells.

As a result of all these meshing procedures, a locally refined rectangular computational mesh is obtained. The final set of mesh cells that includes parallelepiped as well as more complex polyhedrons is used for approximation of the governing equations.

Since all the above-mentioned meshing procedures are performed before the calculation, the obtained mesh is unable to resolve all the solution features well. To overcome this disadvantage, the computational mesh can be refined further at the specified moments during the calculation in accordance with the solution spatial gradients (both in fluid and in solid, see Online Help for details). As a result, in the low-gradient regions the cells are merged, whereas in the high-gradient regions the cells are split. The moments of the computational mesh refinement during the calculation are prescribed either automatically or manually.

Note

Note that in some cases the fine mesh may be unnecessary to give results of the required accuracy.

Two-Scales Wall Functions Model

The *Two-Scales Wall Functions* (2SWF) model in Flow Simulation consist of two approaches to coupling the boundary layer calculation with the main flow properties:

- To describe boundary layers on a fine mesh (the number of cells across a boundary layer is 6 or greater) the '*thick-boundary-layer*' approach is used. In this approach the calculation of parameters of laminar boundary layer is doing via Navier-Stokes equations and for the turbulent boundary layer is performed by modification of well known wall function approach. However instead of the classical approach where the logarithmic velocity profile is used in EFD technology the full profile proposed by Van Driest (Ref. 18) is used. All other assumptions are similar ones to the classical wall functions approach.

- The '*thin-boundary-layer*' approach is used to describe flow on a coarse mesh (the number of cells across a boundary layer is 4 or less). In this approach the Prandtl boundary layer equations already integrated along the normal to the wall from 0 (at the wall) to the dynamic boundary layer thickness (δ) are solved along a fluid streamline covering the walls. If the boundary layer is laminar these equations are solved with a method of successive approximations based on the Shvetz trial functions technology (Ref. 3). If the boundary layer is turbulent or transitional between laminar and turbulent, a generalization of this method to such boundary layers employing the Van Driest hypothesis about the mixing length in turbulent boundary layers is used (Ref. 18).

In intermediate cases, a compilation of the two above approaches is used, ensuring a smooth transition between the two models as the mesh is refined, or as the boundary layer thickens along a surface.

In addition to 2SWF model mentioned above, the '*thin-channel*' approach is used to describe flow through narrow slots (including flow in pipes, plane and circular Couette flows) on a coarse mesh (the number of cells across a slot is 7 or less). According to this approach, the shear stress and heat flux near the wall are calculated by using the approximations based on the experimental data.

By default, an appropriate boundary-layer approach is selected automatically according to the computational mesh.

___ **Note** ___

In the most cases all of these approaches provide the good accuracy, even on a coarse mesh. However, in some cases when the appropriate boundary-layer approach is selected automatically and the computational mesh is rather fine, the solution accuracy may fall off. The reasons for the accuracy decrease are that the mesh is excessively fine to apply the *thin-boundary-layer*, but it is insufficiently fine to resolve boundary layers and apply the *thick-boundary-layer*. The further refinement of the computational mesh improves the accuracy gradually.

Spatial Approximations

The *cell-centered finite volume* (FV) method is used to obtain conservative approximations of the governing equations on locally refined grid that consists of parallelepipeds and more complex polyhedrons near boundary. Following the FV method, the governing equations are integrated over a control volume which is a mesh cell, and then are approximated. All basic variables are referred to mass centers of control volumes. These cell-centered values are used for approximations.

The integral conservation laws may be represented in the form of the cell volume and surface integral equation:

(4-1)

$$\frac{\partial}{\partial t}\int U dv + \oint \boldsymbol{F} ds = \int \boldsymbol{Q} dv$$

are replaced by the discrete form:

(4-2)

$$\frac{\partial}{\partial t}(U \cdot v) + \sum_{N_c} \boldsymbol{F} \cdot \boldsymbol{S} = \boldsymbol{Q} \cdot v$$

where N_c is the number of cell faces.

The fluxes $\boldsymbol{F}$ are approximated in accordance with a type of the related faces. The faces are classified and divided into two groups in accordance with their properties: if they are axis-oriented and common for two adjacent control volumes or they are arbitrary oriented boundary faces. Approximations are constructed in different way for these two cases.

Second order approximations are used for the faces that are common for two adjacent control volumes. For convective fluxes the upwind approach is used. Nonlinear approximations with limiters (Ref. 14) are used to provide monotonic discrete solutions. For diffusive terms the central approximation is used. The approximations are treated implicitly.

In Flow Simulation, especially consistent approximations for the convective terms, 'div' and **'grad'** operators are employed in order to derive a discrete problem that maintains the fundamental properties of the parent differential problem in addition to the usual properties of mass, momentum and energy conservation.

Spatial Approximations at the Solid/Fluid Interface

On arbitrary oriented boundary faces, the fluxes are approximated in accordance with boundary conditions and taking into account the curved geometry boundary. To calculate accurately

boundary fluxes on solid/fluid interface, sophisticated Flow Simulation boundary layer model is used.

Conjugate heat transfer in fluid and solid is calculated at once as one problem, without splitting into two explicitly related to each other problems.

Temporal Approximations

The time derivatives are approximated with an implicit first-order Euler scheme.

Time-implicit approximations of the continuity and convection/diffusion equations (for momentum, temperature, etc.) are used together with an *operator-splitting* technique (Ref. 4, Ref. 11, and Ref. 16). This technique is used to efficiently resolve the problem of pressure-velocity decoupling. Following the SIMPLE-like approach (Ref. 12), an elliptic type discrete pressure equation is derived by algebraic transformations of the originally derived discrete equations for mass and momentum, and taking into account the boundary conditions for velocity.

Form of the Numerical Algorithm

Let index '*n*' denotes the time-level, and '***' denotes intermediate values of the flow parameters.

The following numerical algorithm is employed to calculate flow parameters on time-level ($n + 1$) using known values on time-level (n):

(4-3)

$$\frac{U^* - U^n}{\Delta t} + A_h\left(U^n, p^n\right)U^* = S^n$$

(4-4)

$$L_h \delta p = \frac{\text{div}_h\left(\rho u^*\right)}{\Delta t} + \frac{1}{\Delta t}\frac{\rho^* - \rho^n}{\Delta t}$$

(4-5)

$$\rho^* = \rho(p^n + \delta p, T^*, y^*)$$

(4-6)

$$\rho u^{n+1} = \rho u^* - \Delta t \cdot \mathbf{grad}_h \delta p$$

(4-7)

$$p^{n+1} = p^n + \delta p$$

(4-8)

$$\rho T^{n+1} = \rho T^*, \ \rho \kappa^{n+1} = \rho \kappa^*, \ \rho \varepsilon^{n+1} = \rho \varepsilon^*, \ \rho y^{n+1} = \rho y^*$$

(4-9)

$$\rho^{n+1} = \rho\left(p^{n+1}, T^{n+1}, \mathbf{y}^{n+1}\right)$$

Here $\mathbf{U} = (\rho u, \rho T, \rho\kappa, \rho\varepsilon, \rho\mathbf{y})^T$ is the full set of basic variables excluding pressure p, $\mathbf{u} = (u_1, u_2, u_3)^T$ is the velocity vector, $\mathbf{y} = (y_1, y_2, ..., y_M)^T$ is the vector of component concentrations in fluid mixtures, and $\delta p = p^{n+1} - p^n$ is an auxiliary variable that is called a pressure correction. These parameters are discrete functions stored at cell centers. They are to be calculated using the discrete equations 4-3 – 4-9 that approximate the governing differential equations. In equations 4-3 – 4-9 A_h, div_h, $\mathbf{grad}_h$ and $L_h = \text{div}_h\mathbf{grad}_h$ are discrete operators that approximate the corresponding differential operators with second order accuracy.

The equation 4-3 corresponds to the first step of the algorithm when fully implicit discrete convection/diffusion equations are solved to obtain the intermediate values of momentum and the final values of turbulent parameters, temperature, and species concentrations.

The elliptic type equation 4-4 is used to calculate the pressure correction δp. This equation is defined in such a way that the final momentum field $\rho\mathbf{u}^{n+1}$ calculated from the equation 4-3 satisfies the discrete fully implicit continuity equation. Final values of the flow parameters are defined by equations 4-6 – 4-9.

Methods to Resolve Linear Algebraic Systems

The following methods are used to resolve the linear algebraic systems.

Iterative Methods for Non-Symmetric Problems

The following iterative methods are used to solve the non-symmetric problems.

To solve the asymmetric systems of linear equations that arise from approximations of momentum, temperature and species equations 4-3 (see "Form of the Numerical Algorithm" on page 101), a preconditioned generalized conjugate gradient method (Ref. 15) is used.

Incomplete LU factorization is used for preconditioning.

Iterative Methods for Symmetric Problems

The following iterative methods are used to solve the symmetric problems.

To solve symmetric algebraic problem for the pressure-correction equation 4-4 (see "Form of the Numerical Algorithm" on page 101), an original double-preconditioned iterative procedure is used. It is based on a specially developed multigrid method (Ref. 5).

Multigrid Method

The multigrid method is a convenient acceleration technique which can greatly decrease the solution time. Basic features of the multigrid algorithm are as follows. Based on the given mesh, a sequence of grids (grid levels) are constructed, with a decreasing number of nodes. On every such grid, the residual of the associated system of algebraic equations is restricted onto the coarser grid level, forming the right hand side of the system on that grid. When the solution on the coarse grid is computed, it is interpolated to the finer grid and used there as a correction to the result of the previous iteration. After that, several smoothing iterations are performed. This procedure is applied repeatedly on every grid level until the corresponding iteration meets the stopping criteria.

The coefficients of the linear algebraic systems associated with the grid are computed once and stored.

Nested Iterations

Employment of an *operator-splitting* technique is one of basic numerical approaches in Flow Simulation. This means that not the whole set of non-linear governing equations with non-linear boundary conditions are solved simultaneously. Instead of this, equation by equation is solved in some linearized form with linearized boundary conditions.

In order to provide more robust and accurate numerical method, 'nested iterations' are introduced on each time step. These iterations allow treating mutual influence of flow parameters and non-linear terms in equations and boundary conditions.

These iterations are incorporated within each time step and repeated until a solver convergence is obtained or until a **Maximum number of nested iterations** is reached or each conservation law convergence condition is satisfied.

Let the governing equations are treated by using the nested iterations on the time span $[t, t + \Delta t]$. Let k be a number of the nested iteration, the k-th iteration is finished and the $(k+1)$-th iteration is starting.

Criterion of mass equation convergence

Let ρ^0 be the density at time t and ρ^k is calculated approximation for the density at time $(t + \Delta t)$.

While performing the $(k+1)$-th nested iteration, the residual error (r^k_{mass}) for the mass equation is calculated and defined as follows:

$$(4\text{-}10)$$

$$r^k_{mass} = \left\| r^k_{mass,cv} \right\|_1 = \sum_{\Omega} \left| r^k_{mass,cv} \right|$$

where $\sum_{\Omega}$ means summing over all control volumes in the domain (sub-domain) Ω and the residual error at the control volume (cv) is:

$$(4\text{-}11)$$

$$r^k_{mass,cv} = \frac{\left(\rho^k - \rho^0 \right)}{\Delta t} V_{cv} + \sum_{f_{cv}} j^k_{f_{cv}}$$

where $\dot{j}_{mass,f_{cv}}$ is the mass flux at the control volume face (f_{cv}).

To make decision if $\dot{r}_{mass}^{k}$ is small enough, a reference residual value ($\dot{r}_{mass,ref}^{k}$) is defined as:

$$(4\text{-}12)$$

$$\dot{r}_{mass,ref}^{k} = \sum_{\Omega} \dot{r}_{mass,cv,ref}^{k} = \sum_{\Omega} \sum_{f_{cv}} \left| \dot{j}_{mass,f_{cv}}^{k} \right| = 2 \left\langle \dot{j}_{mass,cv} \right\rangle$$

So a sum of absolute values of inlet and outlet fluxes at control volume faces can be considered as doubled mass flux through the control volume (cv).

At the end of the (k +1)-th nested iteration, a stopping criteria are checked. For the mass equation it is:

$$(4\text{-}13)$$

$$\dot{r}_{mass}^{k} \le \varepsilon_{mass} \cdot \dot{r}_{mass,ref}^{k}$$

and the stopping criteria 4-13 can be treated as:

$$(4\text{-}14)$$

$$\frac{\left\langle \dot{r}_{mass,cv} \right\rangle^{k}}{2 \left\langle \dot{j}_{mass,cv} \right\rangle^{k}} \le \varepsilon_{mass}$$

Criterion of energy equation convergence

The residual error ($\dot{r}_{en}^{k}$) for the energy equation is defined as follows:

$$(4\text{-}15)$$

$$\dot{r}_{en}^{k} = \left\| \dot{r}_{en,cv}^{k} \right\|_{1} = \sum_{\Omega} \left| \dot{r}_{en,cv}^{k} \right|$$

where the residual error at the control volume (cv) is:

$$(4\text{-}16)$$

$$\dot{r}_{en,cv}^{k} = \rho^{0} \frac{\left(H^{k} - H^{0} \right)}{\Delta t} V_{cv} + L_{H,Conv+Diff}^{k} \left(H^{k} \right) - f_{H}^{k}$$

For the energy equation, the calculation of reference residual value is based on the use of approximation coefficients. The discrete operator:

(4-17)

$$L_{H,Conv+Diff}^{k}\left(H^{k}\right)=\sum_{j\in\omega_{cv}}c_{j}\cdot H_{j}^{k}$$

approximates convection and diffusion terms of the equation at the control volume (*cv*). Here ω_{cv} is a stencil of the operator *L* at the control volume (*cv*), c_0 is a coefficient for *H* at the control volume under consideration and $c_{j(j\neq0)}$ are coefficients for other control volumes from the stencil. Note, that the coefficients of the approximation are:

(4-18)

$$c_{j}\sim\left(\rho u_{n}+\frac{\lambda_{h}}{\dfrac{h_{cv}}{2}}\right)S_{f_{cv}}$$

where u_n is the velocity component normal to the face and h_{cv} is the characteristic size of the control volume.

The explicit member f_{H}^{k} contains approximations of other terms of the equation and the heat source at the control volume face equals $q\cdot V_{cv}$.

The reference residual value ($r_{en,ref}^{k}$) is introduces as:

(4-19)

$$r_{en,ref}^{k}=\sum_{\Omega}r_{en,cv,ref}^{k}=\sum_{\Omega}\left(\max\left(\left|c_0+\frac{\rho^0}{\Delta t}V_{cv}\right|,\max_{j\in\omega,j=0}\left|c_j\right|\right)\cdot\left|H^k\right|+\left|f_H^k+\frac{\rho^0 H^0}{\Delta t}V_{cv}\right|\right)$$

In common case of $\Delta t>>\dfrac{h_{cv}}{u_n}+\dfrac{h_{cv}^2}{\lambda_H/\rho}$, the first summand is the reference value of the energy flux through the control volume (*cv*). The second summand is the reference value of the energy flux in/out of the control volume due to the heat source $q\cdot V_{cv}$.

In case of non-large Δt: $r_{en,cv,ref}^{k}\xrightarrow[\Delta t\to0]{}\infty$.

Thus, $r_{en,cv,ref}^{k}$ is treated as the reference value of the energy flux ($j_{en,cv}$) through the control volume. Finally, the convergence condition for the energy equation is:

$$(4\text{-}20)$$

$$r_{en}^{k} \leq \varepsilon_{en} \cdot r_{en,ref}^{k}$$

and the stopping criteria 4-20 can be treated as:

$$(4\text{-}21)$$

$$\frac{\langle r_{en,cv} \rangle^{k}}{\langle j_{en,cv} \rangle^{k}} \leq \varepsilon_{en}$$

Criterion of momentum equation convergence

In a way that is quite similar to the energy equation, the residual error (r_{mom}^{k}) for the momentum equation is defined as follows:

$$(4\text{-}22)$$

$$r_{mom}^{k} = \left\| r_{mom,cv}^{k} \right\|_{1} = \sum_{\Omega} \left(\left| r_{mom_{0},cv}^{k} \right| + \left| r_{mom_{1},cv}^{k} \right| + \left| r_{mom_{2},cv}^{k} \right| \right)$$

where the residual error for i-th component of momentum at the control volume (cv) is:

$$(4\text{-}23)$$

$$r_{mom,cv}^{k} = \frac{\left(m_{i}^{k} - m_{i}^{0} \right)}{\Delta t} V_{b} + L_{m_{i},Conv+Diff} \left(m_{i}^{k} \right) - f_{m_{i}}^{k} , \quad i = 0,1,2$$

Here:

$$(4\text{-}24)$$

$$L_{m_{i},Conv+Diff}^{k} \left(m_{i}^{k} \right) = \sum_{j \in \omega_{cv}} c_{ij} \cdot m_{ij}^{k}$$

where:

$$(4\text{-}25)$$

$$c_{ij} \sim \left(u_n + \frac{\mu + \mu_t}{\dfrac{h_{cv}}{2}} \right) S_{f_{cv}}$$

where the member $f_{m_i}^k$ contains approximations of other terms including a pressure gradient and forces.

The reference residual value ($r_{mom,ref}^k$) is defined as:

$$(4\text{-}26)$$

$$r_{mom,ref}^k = \sum_\Omega \left(\left| r_{mom_0,cv,ref}^k \right| + \left| r_{mom_1,cv,ref}^k \right| + \left| r_{mom_2,cv,ref}^k \right| \right)$$

where:

$$(4\text{-}27)$$

$$r_{mom_i,cv,ref}^k = \sum_\Omega \left(\max \left(\left| c_{i0} + \frac{V_b}{\Delta t} \right|, \max_{j \in \omega, j=0} \left| c_{ij} \right| \right) \cdot \left| m^k \right| + \left| f_{m_i}^k + \frac{m_i^0}{\Delta t} V_b \right| \right)$$

Thus, $r_{mom,cv,ref}^k$ is treated as the reference value of the momentum flux ($j_{mom,cv}$) through the control volume. Finally, the convergence condition for the momentum equation is:

$$(4\text{-}28)$$

$$r_{mom}^k \le \varepsilon_{mom} \cdot r_{mom,ref}^k$$

and the stopping criteria 4-28 can be treated as:

$$(4\text{-}29)$$

$$\frac{\left\langle r_{mom,cv} \right\rangle^k}{\left\langle j_{mom,cv} \right\rangle^k} \le \varepsilon_{mom}$$

Notes on the residuals behavior

In a particular case of complex flow it is not evident what the time scale is for unsteady processes. The residuals do not go down if the time step in calculations is comparable or greater than the time scale of unsteady processes.

In any case, any behavior of residuals does not guarantee accuracy of a discrete solution due to the fact that there are still approximation errors.

> **Note**
>
> Note, that residuals do not tell whether the solution is accurate. Low residuals do not automatically mean an accurate solution, and high residuals do not automatically mean a wrong solution. Only investigation of physical parameters convergence on a sequence of decreasing time steps tells about the actual accuracy of the discrete solution.

This chapter provides the fundamentals of working with the Flow Simulation computational mesh, describes the mesh generation procedure and explains the use of parameters governing both automatically and manually controlled meshes.

Introduction

First, let us introduce a set of definitions.

Flow Simulation considers the real model created in SOLIDWORKS and automatically generates a rectangular computational mesh in the **Computational Domain** distinguishing the fluid and solid domains.

The corresponding **Computational Domain** is generated in the form of a rectangular parallelepiped enclosing the model for both the 3D analysis and 2D analysis. Its boundaries are parallel to the Global Coordinate System planes. For **External flows**, the computational domain's boundary planes are automatically distanced from the model. For **Internal flows**, the computational domain's boundary planes automatically envelop either the entire model, if **Heat Conduction in Solids** is considered, or if **Heat Conduction in Solids** is not considered, the model's flow passage only.

In the mesh generation process, the computational domain is divided into uniform rectangular parallelepiped-shaped cells, which form a so-called **Basic Mesh**. Then, using information about the model geometry, the specified boundary conditions and goals Flow Simulation further constructs the mesh by means of various refinements, i.e. splitting of the **Basic Mesh** cells into smaller cells, thus better representing the model and fluid regions. The mesh, which the

calculation starts from, so-called **Initial Mesh**, is fully defined by the generated basic mesh and the refinement settings.

Each refinement has its criterion and level. The *refinement criterion* denotes which cells have to be split, and the *refinement level* denotes the smallest size, which the cells can be split to. Regardless of the refinement considered, the smallest cell size is always defined with respect to the basic mesh cell size so the constructed **Basic Mesh** is of great importance for the resulting computational mesh.

The main types of refinements are:

- **Cell Type Refinement**;

- **Channel Refinement**;

- **Advanced Refinements**:

    o **Small Solid Features Refinement**,

    o **Curvature Refinement**,

    o **Tolerance Refinement**.

In addition, the following type of refinements can be invoked locally:

- **Equidistant Refinement**;

- During the calculation, the **Initial Mesh** can be refined further using the *Solution-Adaptive Refinement* (see "Refinement of the Computational Mesh During Calculation" on page 144).

Though it depends on a refinement which criterion or level is available for user control, we will consider all of them (except for the *Solution-Adaptive Refinement*) to give you a comprehensive understanding of how the Flow Simulation meshing works.

Types of Cells

Any Flow Simulation calculation is performed in a rectangular parallelepiped-shaped computational domain which boundaries are orthogonal to the axes of the Cartesian Global Coordinate System. A computational mesh splits the computational domain with a set of planes orthogonal to the Cartesian Global Coordinate System's axes to form rectangular parallelepipeds called cells. The original parallelepiped cells containing boundary are split into several parts that are referred to only one fluid or solid medium.

The resulting computational mesh consists of cells of the following types:

- *Fluid cells* are the cells located entirely in the fluid.

- *Solid cells* are the cells located entirely in the solid.

- *Solid-fluid boundary cells* are the cells which are partly in the solid and partly in the fluid.

As an illustration let us look at the generated computational mesh (Figure 5-1).

Figure 5-1. The computational mesh cells of different types: 1) fluid cells, 2) solid-fluid boundary cells, 3) solid cells.
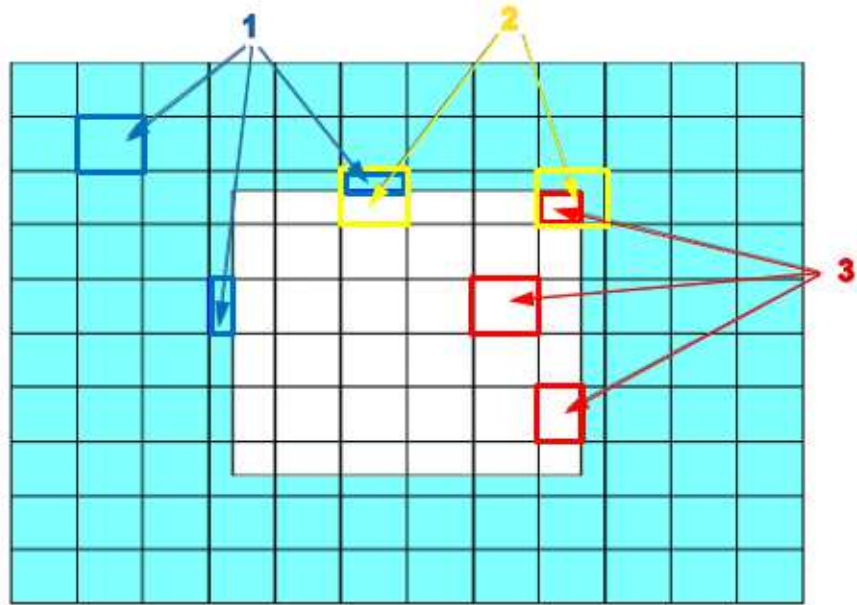
Mesh Construction Stages

The mesh is constructed in the following steps:

- Construction of the basic mesh taking into account the user-specified **Control Planes** and the respective values of cells number (or cells size) and cell size ratios.

- Refinement of all fluid and solid mesh cells up to the user-specified level.

- Resolving of the interface between substances to resolve the relatively small solid features and solid/solid interface, tolerance and curvature refinement of the mesh at a solid/fluid boundaries to resolve the interface curvature (e.g. small-radius surfaces of revolution, etc).

- Channels refinement, that is the refinement of the mesh in narrow channels taking into account the respective user-specified settings.

___ **Note** ___

The mesh at this stage is called the **Initial Mesh**. The initial mesh implies the complete basic mesh with the resolution of the solid/fluid (as well as solid/solid) interface by the small solid features refinements, the curvature and channels refinement also taking into account the local mesh settings.

After each of these stages is passed, the number of cells is increased to some extent.

___ **Tip** ___

If you switch on or off heat conduction in solids, or add/modify solid materials, you should rebuild the mesh.

In Flow Simulation you can control the following parameters and options which govern the computational mesh:

1. **Nx**, the number of basic mesh cells (zero level cells) along the X axis of the Global Coordinate System. $1 \leq N_x \leq 100\ 000$

2. **Ny**, the number of basic mesh cells (zero level cells) along the Y axis of the Global Coordinate System. $1 \leq N_y \leq 100\ 000$.

3. **Nz**, the number of basic mesh cells (zero level cells) along the Z axis of the Global Coordinate System. $1 \leq N_z \leq 100\ 000$.

4. **Control planes**. By adding and relocating them you can contract and/or stretch the basic mesh in the specified directions and regions. In case of the 3D analysis, six control planes coincident with the computational domain's boundaries are always present in any project.

5. **Channels refinement**: Characteristic number of cells across a channel (N_{ch}), Maximum Channels refinement level ($0 \leq L_{ch} \leq 9$), The minimum and maximum height of channels (H_{min} and H_{max}) to be refined.

6. **Small solid features refinement level** ($0 \leq L_{ssf} \leq 9$).

7. **Curvature refinement level** ($0 \leq L_{cur} \leq 9$).

8. **Curvature refinement criterion** ($0 \leq C_{cur} \leq \pi$).

9. **Tolerance refinement level** ($0 \leq L_{tol} \leq 9$).

10. **Tolerance refinement criterion** ($0 \leq C_{tol}$).

11. **Display refinement level**. The **Display Refinement Level** option is intended to inspect the probable refined initial mesh but not to change the specified refinement settings that will be applied. The specified refinement level is measured from the basic mesh.

These options are described in more details below.

Basic Mesh

The basic mesh is a mesh of zero level cells. It is constructed for the whole computational domain at the beginning of the meshing process and formed by dividing the computational domain into slices by parallel planes which are orthogonal to the Global Coordinate System's axes. The basic mesh can be defined either by the number of basic mesh cells or by the average size of the basic mesh cells along each coordinate direction.

In case of 2D calculation (i.e. if you select the **2D plane flow** option in the **Computational Domain** dialog box) only one basic mesh cell is generated automatically along the eliminated direction. By default Flow Simulation constructs each cell as close to cubic shape as possible.

Note

The number of basic mesh cells could be one or two less than the user-defined number (**Nx**, **Ny**, **Nz**). There is no limitation on a cell oblongness or aspect ratio, but you should carefully check the calculation results in all cases for the absence of too oblong or stretched cells.

Control Planes

The **Control Planes** option is a powerful tool for creating an optimal computational mesh, and the user should certainly become acquainted with this tool if he is interested in optimal meshes resulting in higher accuracy and decreasing the CPU time and required computer memory. Control planes allow you to contract the basic mesh locally to resolve a particular region of interest by a denser mesh and stretch the basic mesh to avoid excessively dense meshes in other regions.

All of the control planes are orthogonal to the Global Coordinate System's axes. The **Computational Domain** boundary planes (**X min**, **X max**, **Y min**, **Y max**, **Z min**, **Z max**) are among the **Control Planes** by default.

Contracting the Basic Mesh

Using control planes you may contract the basic mesh in the regions of interest.
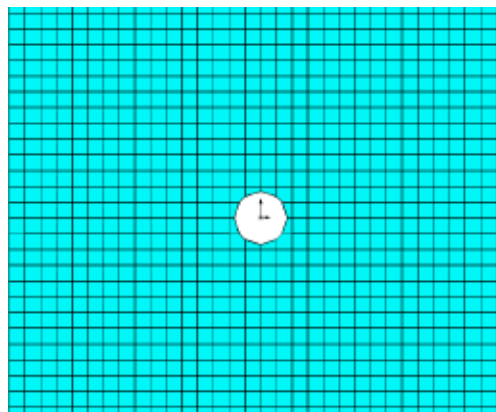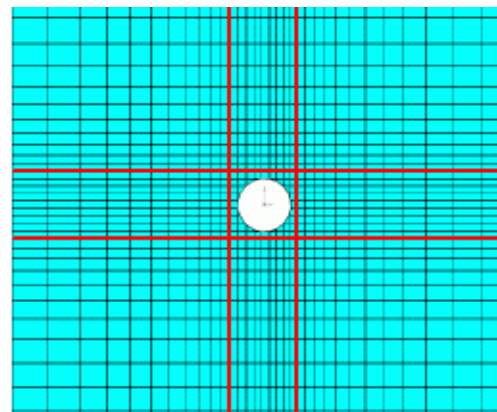
To do this, you need to set control planes surrounding the region and assign the proper **Ratio** values to the respective intervals. The cell sizes on the interval are changed gradually so that the proportion between the first and the last cells of the interval is close (but not necessarily equal) to the entered **Ratio** value. Negative values of the ratio correspond to the reverse order of cell size increase. Alternatively, you may explicitly set the **Number of cells** (N_{cell}) for each interval (I_i), in which case the **Ratio** value (R) becomes mandatory. For example, assume that there are two control planes *Plane 1* and *Plane 2* (see Figure 5-2) and the ratio on the interval between them is set to 2. Then the basic mesh cells adjacent to the *Plane 1* will be approximately two times longer than the basic mesh cells adjacent to the *Plane 2*.

Figure 5-2. Specifying custom control planes (P_d - default, P_c - custom):
I_1: N_{cell} = 12 (automatic), R = 2; I_2: N_{cell} = 12 (manual), R = 1;
I_3: N_{cell} = 3 (automatic), R = 1.5.
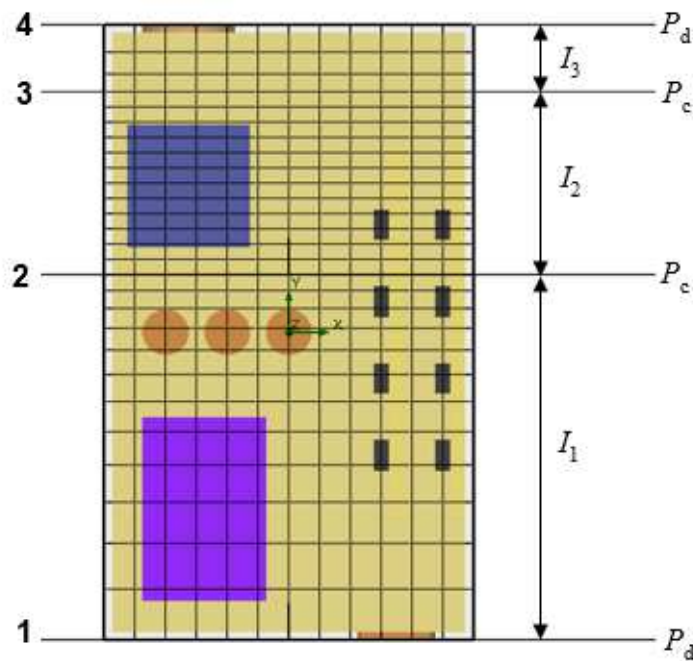
Use of control planes is especially recommended for external analyses, where the computational domain may be substantially larger than the model.

Table 5-1. Control planes examples.

a) Uniform mesh, default control planes only.	b) Two custom control planes in each direction.

In the Table 5-1 (b) two custom control planes are set around the body in each coordinate direction with the ratio set to 5 and –5, respectively.

Mesh Refinement

Refinement is a process of splitting a rectangular computational mesh cell into eight cells by three orthogonal planes that divide the cell's edges in halves. The non-split initial cells that compose the basic mesh are called basic cells or zero level cells. Cells obtained by the first splitting of the basic cells are called first level cells, the next splitting produces second level cells, and so on. The maximum level of splitting is nine. A ninth level cell is 8^9 times smaller in volume than the basic cell.

------ **Tip** ------
ⓘ During the solution-adaptive meshing the cells can be refined and merged. See "Refinement of the Computational Mesh During Calculation" on page 144.

The following *Cell Mating* rule is applied to the processes of refinement: the levels of two neighboring cells (i.e. cells having a common face) can only be the same or differ by one, so that, say, a fifth level cell can have only neighboring cells of fourth, fifth, or sixth level. This rule has the highest priority.

In the Figure 5-3 the fluid cells refined according to the *Cell Mating* rule are shown. The fourth-level red cells appearing after resolving the cog cause the neighboring cells to be split up to third level (yellow cells), that, in turn, causes the subsequent refinement producing second level cells (green cells) and first level cells (blue cells). The white zero level cell (basic mesh cell) remains unsplit since it borders on first level cells only, thus satisfying the rule.

Figure 5-3. Fluid cell refinement due to the Cell Mating rule.
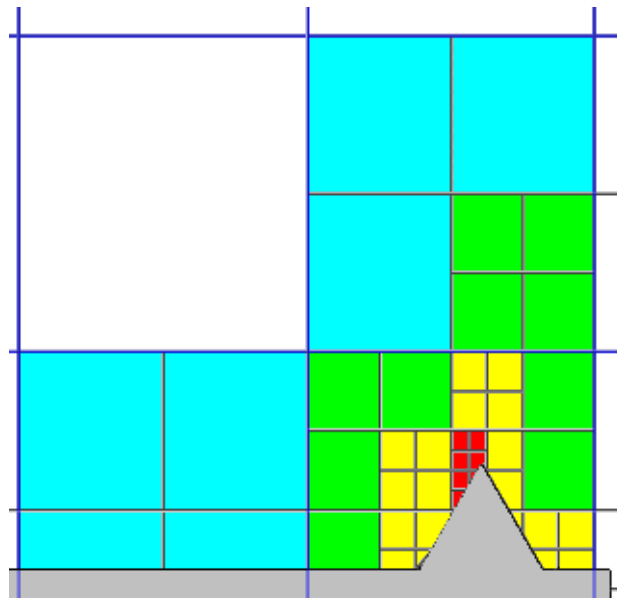
> **Note**
> The *Cell Mating* rule is strict and has higher priority than the other cell operations. The rule is also enforced for the cells that are entirely in a solid.

Refining Cells by Type

The refinement level of cells of a specific type (the combination of fluid cells, and solid cells) denotes the minimum level to which the corresponding cells must be split if it doesn't contradict the *Cell Mating* rule.

> **Note**
> The minimum level means the lower bound to which it is obligatory to split cells, though the cells can be split further if it is required to satisfy the other criteria such as **Small solid features refinement**, **Curvature refinement**, **Channels refinement** or **Tolerance refinement**.

Advanced Refinement at Interfaces Between Substances

Different interface types (solid/fluid or solid/solid) are checked on different refinement criteria, namely:

- *small solid features criterion* for solid/fluid, solid/solid, solid/porous and porous/fluid interfaces;

- *curvature refinement criterion* for solid/fluid, solid/porous and porous/fluid interfaces;

- *tolerance refinement criterion* for solid/fluid and solid/porous interfaces;

- *narrow channel refinement criterion* for solid/fluid and solid/porous interfaces.

Whereas the specified refinement levels are equally applied to any interface type.

Tip

The solid/solid interface is the interfaces between different solid materials. If two solids of the same material contact to each other, they are combined into one solid.

To resolve the interface between substances, the refined mesh is constructed in the following steps:

1. Analyze triangulation in each basic mesh cell at the interfaces between different substances (such as solid/fluid and solid/solid interfaces) in order to find the maximum angle between normals to the triangles which compose the interface within the cell. Depending on the maximum angle found, the decision whether to split the cell or not is made in accordance with the specified **Curvature Criterion** (C_{cur}), **Tolerance Criterion** (C_{tol}) and the **Small Solid Feature Refinement criterion** (C_{ssf}), also called the *120-degree criterion*.

2. Analyze the distances between the opposite walls of each flow passage in the direction normal to wall. Depending on the distances found, the decision whether to split the cell or not is made in accordance with the specified **Characteristic Number of Cells Across Channel** (N_{ch}), and **Minimum (Maximum) Height of Channel** (H_{min} and H_{max}).

3. The cells that marked to be split as described in items 1 – 2, are split if the specified **Small Solid Feature Refinement Level** (L_{ssf}), **Curvature Level** (L_{cur}), **Tolerance Level** (L_{tol}) and **Maximum Channel Refinement Level** (L_{ch}) are not reached.

Tip

If a cell belongs to a local initial mesh area, then the corresponding local refinement levels will be applied (see "Local Mesh Settings" on page 136).

---

4. If a basic mesh cell is split, the resulting child cells are analyzed as described in items 1 – 3, and split further, if necessary. The cell splitting will proceed until the interface

resolution satisfies the **Small Solid Feature Refinement criterion** (C_{ssf}), the specified C_{cur}, C_{tol}, N_{ch}, H_{min} and H_{max} or the corresponding level of splitting reaches its specified value.

> **Tip**
> ⓘ The specified levels of splitting denote the maximum admissible splitting, i.e. they show to which level a basic mesh cell can be split if it is required for resolving the solid/fluid interface within the cell.

5. The operations 1 to 4 are applied for the next basic mesh cell and so on, taking into account the following *Cell Mating* rule: two neighboring cells can be only cells which levels are similar or differ by one. This rule has the highest priority as it is necessary for simplifying numerical algorithm in solver.

Small Solid Features Refinement

The procedure of resolving small solid features refines only the cells where the solid/fluid and solid/solid interface curvature is too high: the maximum angle between the normals to a solid surface inside the cell is strictly greater than 120°, i.e. the solid surface has a protrusion within the cell. Such cells are split until the **Small Solid Feature Refinement Level** (L_{ssf}) of splitting mesh cells is achieved.

To make this **Small Solid Feature Refinement criterion** (C_{ssf}), also called the *120-degree criterion*, easier to understand, let us consider simple small solid features of planar faces only. The normal to triangles that form the planar face is normal to the planar face too. Therefore, instead of considering the normals to the triangles we can consider normals to faces, or better the angle between faces.

Figure 5-4. L_{ssf} = 1, L_{cur} = 0, L_{ch} = 0.
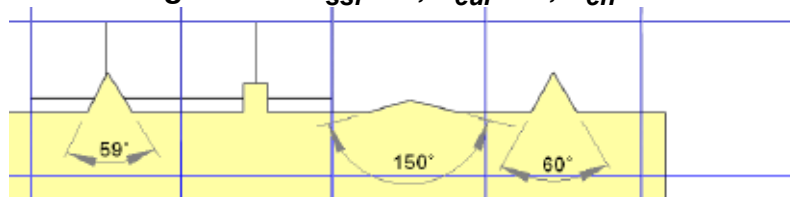
In Figure 5-4 the cells with the cogs of 150° and 60° were not split by the small solid features refinement because the maximum angles between the faces (i.e. between normals to the triangles enclosed by the cell) are 30° and 120°, respectively. If the angle between the normals becomes greater than 120° (121° for the 59°-cog) then the cell is split. The cell with the square

spike surely has to be split because the lateral faces of the spike have their normals at the angle of 180°, thus satisfying the *120-degree criterion*.

Note

Rectangular corners (like in the rightmost cell) do not satisfy the criterion and therefore will not be resolved by the small solid features refinement.

Remember that if the Channel refinement is enabled, the maximum level to which the small solid features refinement can split the cells is set as the maximum level from the specified **Small Solid Feature Refinement Level** (L_{ssf}) and **Maximum Channel Refinement Level** (L_{ch}). In other words, if the **Channel refinement** is enabled, the L_{ssf} has no effect if it is smaller than the L_{ch}..

Figure 5-5. $L_{ssf} = 0$, $L_{cur} = 0$, $L_{ch} = 1$.
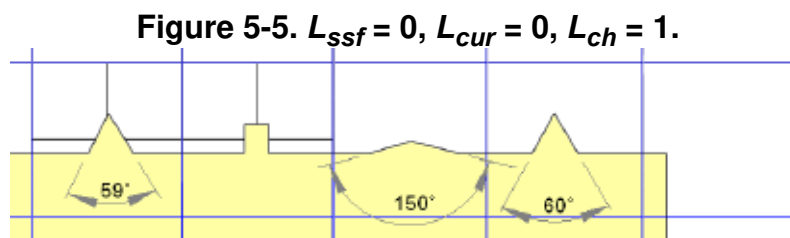
From Figure 5-5 it is clear that the cells are split by the *120-degree criterion* up to the first level, as defined by the narrow channel refinement level.

For the information about how the L_{ch} influences the narrow channel refinement see "Channel Refinement" on page 126.

Curvature Refinement

The **Curvature refinement level** (L_{cur}) is the maximum level to which the cells will be split during refinement of the computational mesh until the curvature of the solid/fluid interface within the cell becomes lower than the specified **Curvature refinement criterion** (C_{cur}).

The curvature refinement procedure has the following stages:

1. Each solid surface is triangulated: Flow Simulation gets triangles that make up the surfaces.

Note

The performance settings do not govern the triangulation performance.

---

2. A local (for each cell) interface curvature is determined as the maximum angle between the normals to the triangles within the cell.

3. If this angle exceeds the specified C_{cur}, and the curvature refinement level is not reached then the cell is split.
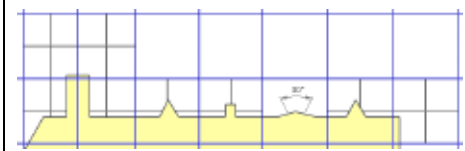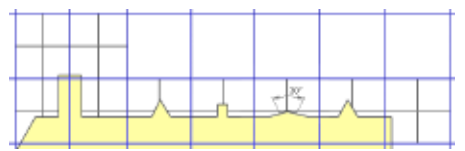
The curvature refinement works in the same manner as the small solid features refinement with the difference that the critical angle between the normals can be specified by the user (in radians) as **Curvature refinement criterion** (C_{cur}). Here, the smaller the criterion, the better resolution of the solid curvature. To give more precise and descriptive explanation, the following table presents several C_{cur} values together with the corresponding angles between normals and the angles between planar faces.

Table 5-2. Influence of the curvature criterion on the solid curvature resolution.

Curvature criterion, rad	0.3491	0.5062	0.5411	0.6982	1.0472	1.5708	2.0944	3.1416
Angle between normals (α)	20°	29°	31°	40°	60°	90°	120°	180°
Angle between faces (β)	160°	151°	149°	140°	120°	90°	60°	0°

The Table 5-2 states that if the C_{cur} is equal to 0.5062 rad, then all the cells where the angle between normals to the surface-forming triangles is more than 29° will be split.

Table 5-3. The dependency of the solid curvature resolution on the curvature criterion.

a) $L_{ssf} = 0$, $L_{ch} = 0$, $L_{cur} = 1$, $C_{cur} = 0.5411$ rad ($\alpha = 31°$, $\beta = 149°$)	b) $L_{ssf} = 0$, $L_{ch} = 0$, $L_{cur} = 1$, $C_{cur} = 0.5062$ rad ($\alpha = 29°$, $\beta = 151°$)

In Table 5-3 you can see that the curvature criterion set to 0.5062 rad splits the cells with the 151° cog.

Note
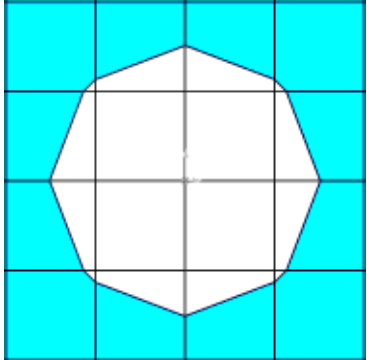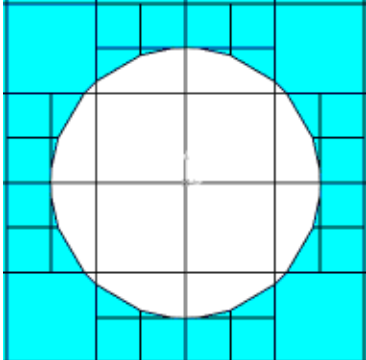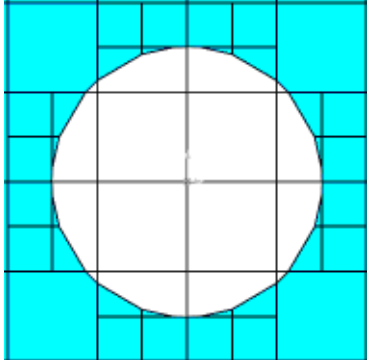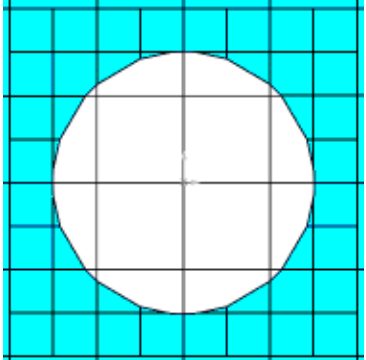
The curvature refinement works exactly as the small solid features refinement when the curvature criterion is equal to 2.0944 rad (120°).

However, the default curvature criterion values are small enough to resolve obtuse angles and curvature well. Increasing the curvature criterion is reasonable if you want to avoid superfluous refinement but it is recommended that you try different criteria to find the most appropriate one.

The curvature refinement is a powerful tool, so that the competent usage of it allows you to obtain proper and optimal computational mesh. Look at the illustrations in Table 5-4 to the curvature refinement by the example of a sphere.

You can see that the curvature criterion set to 0.317 rad and 0.1 rad splits the cells up to the first level only. In Table 5-4 (b) and (c) the cells with the cogs of 162° were split. In Table 5-4 (d) the cells with the cogs of 174° were split.

Table 5-4. The curvature criterion examples.

a) $L_{cur} = 0$; Total number of cells is 64.	b) $L_{cur} = 1$; $C_{cur} = 0.317$ rad ($\alpha = 18°$); Total number of cells is 120.
c) $L_{cur} = 2$; $C_{cur} = 0.317$ rad ($\alpha = 18°$); Total number of cells is 120.	d) $L_{cur} = 2$; $C_{cur} = 0.1$ rad ($\alpha = 6°$); Total number of cells is 148.

Small Solid Feature Refinement Level or Curvature Level

Why is it necessary to have two criteria? As you can see, the curvature refinement has higher priority than the small solid features refinement if the curvature criterion is smaller than 120°.

_____ **Note** _____

The Flow Simulation-specified values of the curvature criterion are always smaller than 120°. In other words, if you did not set the C_{cur} greater than 120° and if the L_{ssf} and L_{ch} are smaller than the L_{cur}, then the small solid feature refinement would be idle.

_____

Nevertheless, the advantage of the small solid features refinement is that being sensitive to relatively small geometry features it does not 'notice' the large-scale curvatures, thus avoiding

refinements in the entire computational domain but resolving only the areas of small features. At the same time, the curvature refinement can be used to resolve the large-scale curvatures. So both the refinements have their own coverage providing a flexible tool for creating an optimal mesh.

Tolerance Refinement

Tolerance refinement allows you to control how well (with what tolerance) mesh polygons approximate the real interface. The tolerance refinement may affect the same cells that were affected by the small solid features refinement and the curvature refinement. It resolves the interface's curvature more effectively than the small solid features refinement, and, in contrast to the curvature refinement, discerns small and large features of equal curvature, thus avoiding refinements in regions of less importance (see images below).

Figure 5-6. L_{cur} = 3; C_{cur} = 0.1 (α = 6°, β = 174°).
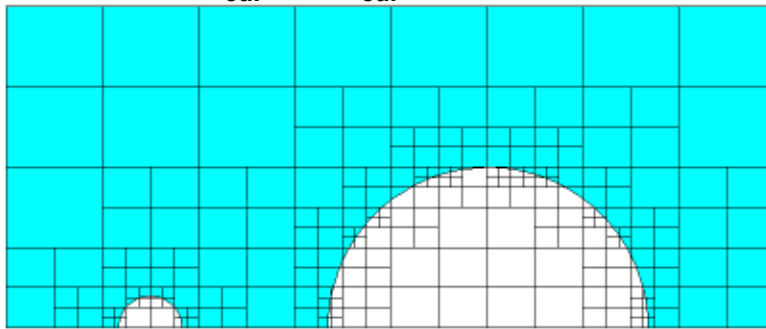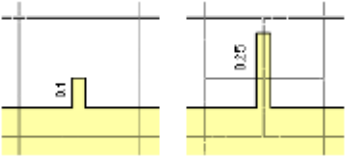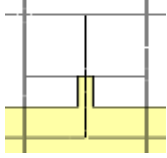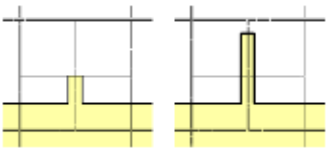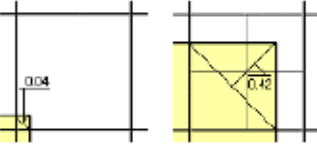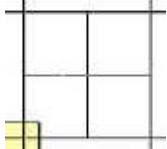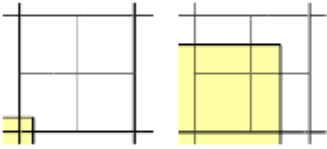
Figure 5-7. L_{tol} = 3; C_{tol} = 0.1 mm.
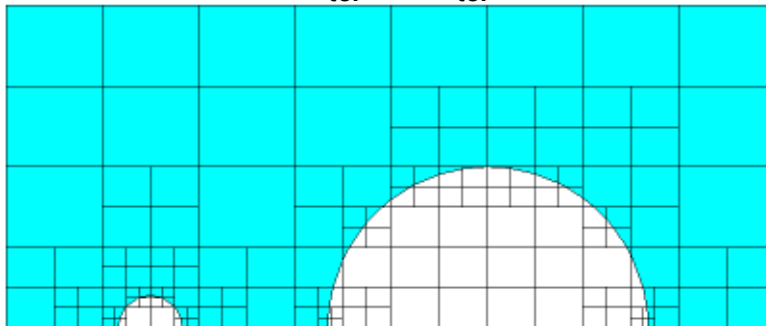
Any surface is approximated by a set of polygons which vertices are surface's intersection points with the cells' edges. This approach accurately represents flat faces though curvature surfaces are approximated with some deviations (e.g. a circle can be approximated by a polygon). The tolerance refinement criterion controls the precision of this approximation. A cell will be split if the distance (h) between the outermost interface's point within the cell and the polygon approximating this interface is larger than the specified criterion value.

Table 5-5. Tolerance refinement vs. Small Solid Features and Curvature refinements (refinement level is set to 1).

Tolerance Refinement		Small Solid Feature Refinement	
C_{tol} = 0.1 mm	C_{tol} = 0.08 mm	(refinement occurs regardless of the feature's size)	
		Curvature Refinement	
C_{tol} = 0.1 mm Refines cells only if the solid part cut by the polygon is large enough (h > 0.1 mm)	C_{tol} = 0.03 mm	(refinement occurs regardless of the curvature only)	

Channel Refinement

After the primary mesh has been created, the channel refinement is put in action. The channel refinement is applied to each flow passage within the computational domain (or a region, in case that local mesh settings are specified) unless you specify for Flow Simulation to ignore passages of a specified height.

The *Narrow Channels* term is conventional and used for the definition of the flow passages of the model in the direction normal to the solid/fluid interface. Regardless of the real solid curvature, the mesh approximation is that the solid boundary is always represented by a set of flat elements, which nodes are the points where the model intersects with the cell edges. Thus, whatever the model geometry, there is always a flat element within a solid-fluid boundary cell and the normal to this element denotes the direction normal to the solid/fluid interface for this solid-fluid boundary cell.

The basic concept of narrow channel refinement is to resolve the narrow channels with a sufficient number of cells to provide a reasonable level of solution accuracy. It is especially important to have narrow channels resolved in analyses of low Reynolds numbers or analyses with long channels, i.e. in such analyses where the boundary layer thickness becomes comparable to the size of the solid-fluid boundary cells where the layer is developed.

There are two ways to specify the channels refinement:

- **Number of Cells** mode governs the procedure of mesh refining in the model's narrow channels by specifying the number of mesh cells across model's flow passages and restricting the refinement level;

- **Refinement Level** mode allows to define an uniform mesh across each model's flow passage by specifying the refinement level as a tabular dependency on the channel height.

The narrow channel settings available in the **Number of Cells** mode are the following:

- **Maximum Channel Refinement Level** – the maximum level of cells refinement in narrow channels with respect to the basic mesh cell.

- **Characteristic Number of Cells Across Channel** – the number of cells (including cells lying at the solid/fluid interface) that Flow Simulation will attempt to set across the model flow passages in the direction normal to the solid/fluid interface. If possible, the number of cells across narrow channels will be equal to the specified characteristic number, otherwise it will be as close to it as possible. The **Characteristic Number of Cells Across Channel** (N_{ch}) and the **Maximum Channel Refinement Level** (L_{ch}) both influence the mesh in narrow channels in the following manner: the basic mesh in narrow channels will be split to have N_{ch} number per channel, if the resulting cells satisfy the specified L_{ch}. In other words, whatever the specified N_{ch}, the smallest possible cell in a narrow channel is $8L$ times smaller in volume (or $2L$ times smaller in each linear dimension) than the basic mesh cell. This is necessary to avoid undesirable mesh splitting in very fine channels that may cause the number of cells to increase to an unreasonable value.

- **Minimum Height of Channel**, **Maximum Height of Channel** – the minimum and maximum bounds for the height outside of which a flow passage will not be considered as a narrow channel and thus will not be refined by the narrow channel resolution procedure.

For example, if you specify the minimum and maximum height of narrow channels, the cells will be split only in those fluid regions where the distance between the opposite walls of the flow passage in the direction normal to wall lies between the specified minimum and maximum heights.

The **Number of Cells** channel refinement mode operates as follows:
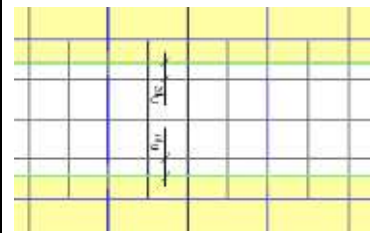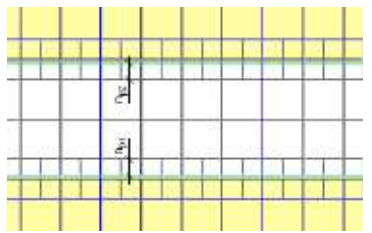
1. For each solid-fluid boundary cell Flow Simulation calculates the '*local*' narrow channel width as the distance between this solid-fluid boundary cell and the next solid-fluid boundary cell found on the line normal to the solid/fluid interface of this cell (i.e. normal to the flat surface element located in the cell).

> **Tip**
>
> ⓘ If the line normal to the solid/fluid interface crosses a local initial mesh area, then the corresponding local narrow channel refinement settings is applied to the cells in this direction.

2. If the distance value falls within the range defined by the **Minimum height of channel** (H_{min}) and **Maximum height of channel** (H_{max}) options, the number of cells per this interval is calculated including both cells lying at the solid/fluid interface and taking into account which portion of each cell is in fluid.

3. More precisely, the number of cells across the channel (i.e. on the interval between the two solid-fluid boundary cells) is calculated as $N = N_f + n_{p1} + n_{p2}$, where N_f is the number of fluid cells on the interval, and np1 and np2 are the fluid portions of the both solid-fluid boundary cells. This value is compared with the specified Characteristic number of cells across a channel (N_{ch}). If N is less than the specified N_{ch} then the cells on this interval are split. For example, on Table 5-6 (a) $N_f = 2$, $n_{p1} = n_{p2} = 0.4$, and $N = 2+0.4+0.4 = 2.8$ which is less than the criterion $N_{ch} = 3$. On Table 5-6 (b) the solid-fluid boundary cells are split, so that the fluid portions of the newly-formed solid-fluid boundary cells are $n_{p1} = n_{p2} = 0.9$, and the criterion is satisfied ($N > N_{ch}$).

Table 5-6. Examples of the Characteristic Number of Cells Across Channel.
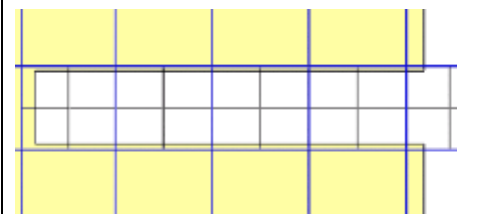
a) $L_{ch} = 2$; $N_{ch} = 3$; $N = 2.8 < N_{ch}$	b) $L_{ch} = 3$; $N_{ch} = 3$; $N = 3.8 > N_{ch}$

> **Note**
>
> 🗐 Like in the other refinements, the **Maximum channel refinement level** (L_{ch}) denotes the maximum level to which the cells can be split to satisfy the N_{ch} criterion. The L_{ch} has higher priority than the N_{ch}, so the refinement will proceed until the N_{ch} criterion is satisfied or all the cells reach the L_{ch}.

The channel refinement is symmetrical with respect to the midpoint of the interval and proceeds from the both ending solid-fluid boundary cells towards the midpoint. Since the actual number of cells across narrow channels can be greater by 1 than the specified characteristic number.

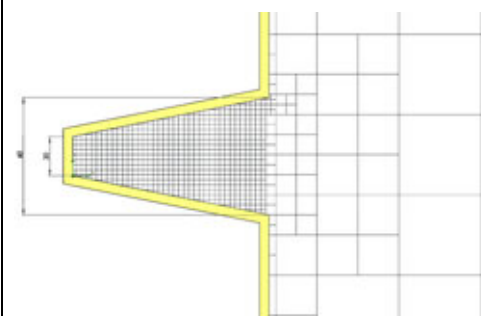**Table 5-7. Examples of the Maximum channel refinement level.**
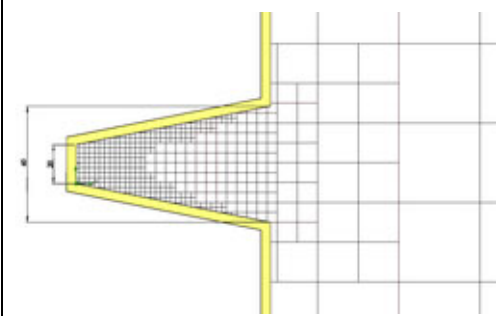
a) $N_{ch} = 5$; $L_{ch} = 1$	b) $N_{ch} = 5$; $L_{ch} = 3$

In Table 5-7 (a), the specified **Characteristic number of cells across a channel** (N_{ch}) is 5 but only two cells were generated since the **Maximum channel refinement level** (L_{ch}) of one allows only basic mesh cells and first-level cells to be generated.

In Table 5-7 (b), the specified **Maximum channel refinement level** (L_{ch}) is high enough to allow 5 cells to be placed across the channel, but there are 6 cells across the channel due to the symmetry requirement of the channel refinement.

If the channels height is variable along the channel length, pay attention to the N_{ch} criterion and make sure that the proper value of the N_{ch} criterion is satisfied along the entire channel.

Table 5-8. Examples of the Characteristic number of cells.
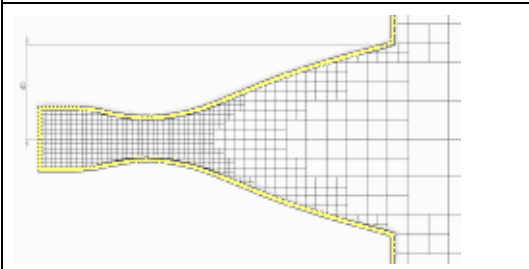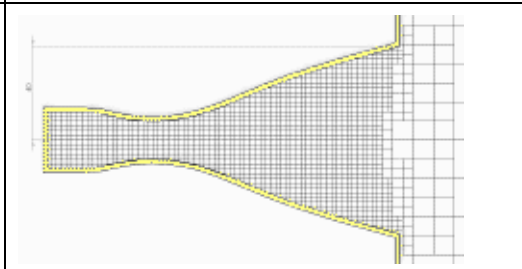
a) $H_{max} = 60$ mm; $L_{ch} = 4$; $N_{ch} = 10$	b) $H_{max} = 60$ mm; $L_{ch} = 4$; $N_{ch} = 25$

In Table 5-8 (a), the specified **Characteristic number of cells across a channel** (N_{ch}) is not enough to reach the specified **Maximum channel refinement level** (L_{ch}) across the entire channel. Only the cells located in the narrowest end of the channel reach the specified L_{ch}.

In Table 5-8 (b), the specified **Characteristic number of cells across a channel** (N_{ch}) is high enough to reach the specified L_{ch} across the entire channel.

Alternatively, to generate an uniform mesh across a channel, you can specify the refinement level as a tabular dependency on the channel height by using the **Refinement Level** mode.

Table 5-9. Examples of the mesh across a channel.

a) **Number of Cells** (H_{max} = 80 mm): L_{ch} = 3; N_{ch} = 10	b) **Refinement Level**: $H \leq 81$ mm: L_{ch} = 3

4. Next, for all the fluid cells within the entire computational domain the following *Fluid Cell Leveling* procedure is applied: if a fluid cell is located between two cells of higher level, it is split to be equalized with the level of neighboring smaller cells.

Although the settings that produce an optimal mesh depends on a particular task, here are some 'rule-of-thumb' recommendations for narrow channel settings:

1. Set the number of cells across a channel to a minimum of 5.

2. Use the minimum and maximum heights of narrow channels to concentrate on the regions of interest.

3. If possible, avoid setting high values for the narrow channels refinement level, since it may cause a significant increase in the number of cells where it is not necessary.

Figure 5-8. L_{ssf} = 3; Channels refinement is disabled.
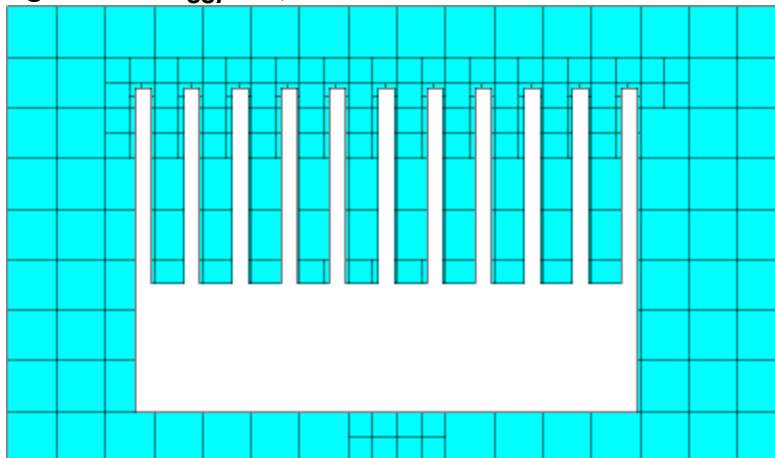
Figure 5-9. L_{ssf} = 3; Channels refinement is on: N_{ch} = 5, L_{ch} = 2.
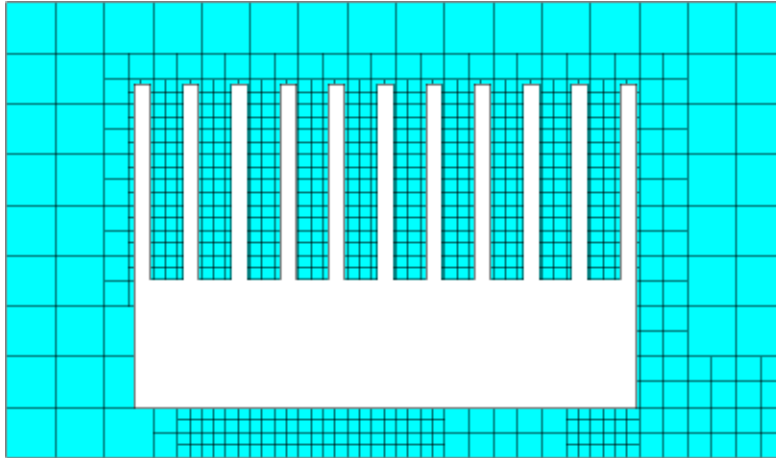
Figure 5-10. L_{ssf} = 3; Channels refinement is on: N_{ch} = 5, L_{ch} = 5.
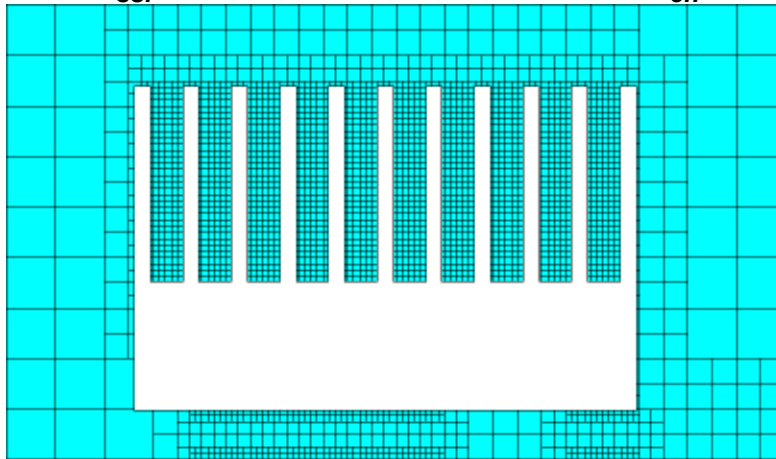
Thin walls resolution

In contrast to the narrow channels, thin walls can be resolved without the mesh refinement inside the wall, since the both sides of the thin wall may reside in the same cell. Therefore, the amount of cells needed to resolve a thin wall is generally lower than the number of cells needed to properly resolve a channel of the same width.

See Figure 5-11 - Figure 5-12 illustrating the thin walls resolution technology and its limitations.

Figure 5-11. One mesh cell can contain more than one fluid and/or solid volume; during calculation each volume has an individual set of parameters depending on its type (F_i - fluid or S_i - solid).
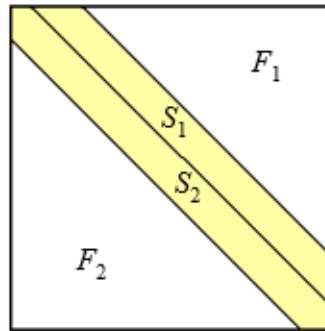
Figure 5-12. If the wall thickness is greater than the basic mesh cell's size across the wall or if the wall creates only one fluid volume in the cell, then the opposite sides of the wall will not lay within the same cell. Such walls are resolved with two or more cells across.
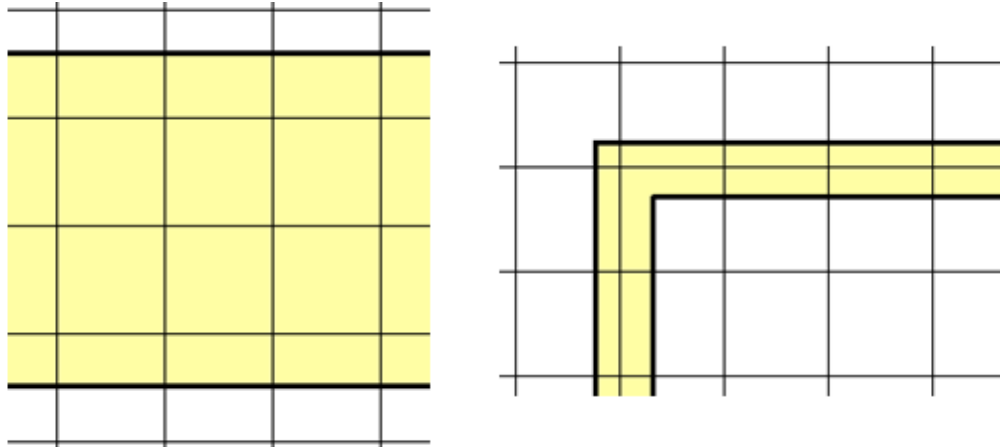
Automatic Mesh Setting

As noted above, Flow Simulation mesh generator uses a system of refinement criteria, each with their own values and refinement levels. However, in order to further simplify the process of defining the mesh a method to automatically define the refinement criteria, values and levels for a case *Automatic Parameters Definition* (APD) technology has been implemented.

The main idea of the APD technology is to define both the basic mesh and refinement settings from the physical definition of the model's boundary conditions, etc. together with the most general information about the geometry.

APD uses the following input parameters:

1. Problem type (Internal\External, Compressible\Incompressible, 3D\2D)

2. **Minimum gap size** (h_{gap}) and **Minimum wall thickness** (h_{wall})

3. **Level of initial mesh** ($1 \leq L_{ini} \leq 7$)

4. Computational model bounding box (B_{model})

5. Fluid region bounding box (B_{fluid})

6. Symmetry settings applied to the computational domain boundaries

7. Middle size of the computational model (H_{mean})

8. Wind direction for external flows (e_{wind}).

The **Level of initial mesh** (L_{ini}) is specified by user. The default values of **Minimum gap size** (h_{gap}) and **Minimum wall thickness** (h_{wall}) are based on the model geometry and defined automatically. However, they can be also set manually. The values of other parameters are set automatically by Flow Simulation.

Also you can enable the **Advanced channel refinement** option to improve the narrow channel refinement strategy.

APD output:

1. Basic mesh settings:

   a. Computational domain size (H_{CD})

   b. Control planes sets for *X, Y, Z* direction and cells ratios at these planes

   c. Number of cells for *X, Y, Z* direction (N_x, N_y, N_z).

2. Refinement settings:

   a. **Small solid feature refinement level** (L_{SSF})

    b.   **Tolerance refinement level** (L_{tol}) and **Tolerance criterion** (C_{tol})

    c.   **Maximum channel refinement level** (L_{ch}) and **Characteristic number of cells across channel** ($N_{ch} = N_{gap}$).

Minimum Gap Size and Minimum Wall Thickness

Before creating the initial mesh, Flow Simulation automatically determines the **Minimum gap size** and the **Minimum wall thickness** for the walls contacting the fluid with both sides. This is required for resolving the essential geometrical features of the model with the computational mesh. Flow Simulation creates the initial mesh so that the number of the mesh cells along the normal to the model surface must not be less than a certain number, if the distance along this normal from this surface to the opposite wall is not less than the **Minimum gap size** (h_{gap}). This insures that a flow passage or gap with the width larger than the specified minimum gap size will be resolved with a certain number of cells across it.

The **Minimum wall thickness** (h_{wall}) governs the resolution of the sharp edges such as tips of thin fins and it does not influence the meshing if it is equal to or greater than the **Minimum gap size** (h_{gap}). The default value of h_{wall} is calculated using information about the model size, the Computational Domain, volume sources, initial conditions, surface sources and surface goals.

The default value of h_{gap} is the smallest size of the surfaces with the boundary conditions specified, such as the model inlet and outlet openings in an internal analysis, as well as the surfaces and volumes with the heat sources, local initial conditions, surface and/or volume goals and some of the other conditions and features. If set manually, h_{gap} should be set to the smaller of the size of smallest flow passage to be resolved by the mesh, and the size of smallest geometric object in the flow that needs to be resolved. At that, it is necessary to remember that the number of the computational mesh cells generated to resolve the model geometrical features depends on the specified result resolution level.

Level of Initial Mesh

The **Level of initial mesh** governs the mean number of cells (N_{mean}) per the model's middle size (H_{mean}) and the number of cells (N_{gap}) per the smallest flow passage/channel height, which specified by the **Minimum gap size** value (h_{gap}).

First of all, the geometry resolution coefficient (K_{res}) is defined as the ratio of the model's middle size (H_{mean}) to the minimum gap size (h_{gap}). If it is smaller than 2.5, the number of cells per the model size (N_{mean}) and the number of cells per the minimum gap size (N_{gap}) are

specified by using the table below. Note, that the real number of cells per a gap is restricted by the maximum allowed channel refinement level (L^{*}_{ch}).

Table 5-10. APD mesh settings.

Level of initial mesh	N_{mean}			N_{gap}	L^{*}_{ch}
	Internal	**External**			
		Incompressible	**Compressible**		
1	5	2	3	2	0
2	7	3	5	3	1
3	10	5	7	5	2
4	14	7	10	7	2
5	20	10	14	10	2
6	28	14	20	14	4
7	40	20	28	20	4

See Table 5-11 illustrating how the APD technology works.

Table 5-11. APD technology example.

a) $L_{ini} = 4$; $h_{gap} = 18$ mm   b) $L_{ini} = 4$; $h_{gap} = 4$ mm
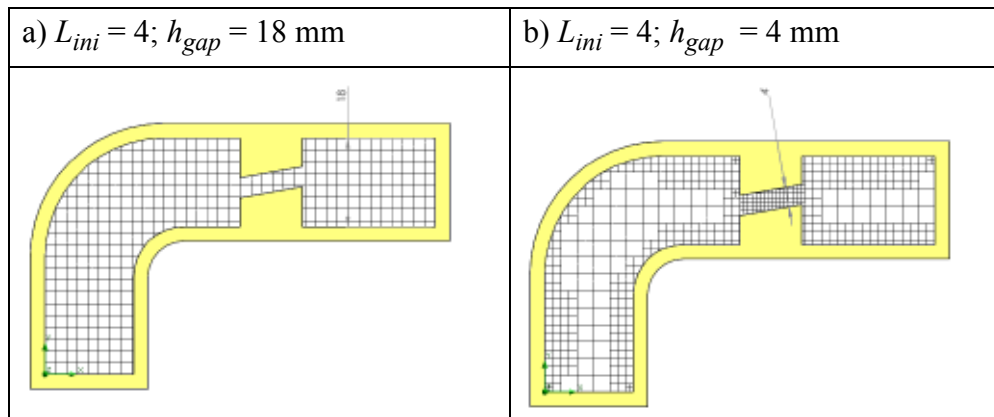
If K_{res} is greater than 2.5, the N_{mean} value is multiplied by the normalized coefficient K_{norm} depended on the K_{res}. Also in these cases the maximum allowed channel refinement level (L^{*}_{ch}) cannot provide a sufficient number of cells per a gap. So if the $K_{res} \gg 1$, it is recommended to enable the **Advanced channel refinement** option.

Advanced channel refinement

When checked, an improved narrow channel refinement strategy is effective which ensures that narrow channels/passages are resolved by a sufficient number of cells to predict the flow and heat transfer phenomena including boundary layers with high accuracy. The default **Maximum**

channel refinement level (L_{ch}) is set to the value which provides the number of cells across a gap, no less than N_{gap}. Otherwise, it depends on the **Level of initial mesh** (L_{ini}) and cannot be greater than four, resulting in the restriction of the minimum size of cells across the model's flow passage in the normal-to-solid/fluid-interface directions. The consequence may be a significant increase of the number of cells up to one or more orders of magnitude.

Local Mesh Settings

The local mesh settings option is one more tool to help create an optimal mesh. Use of local mesh settings is especially beneficial if you are interested in resolving a particular region within a complex model.

The local mesh settings influence only the initial mesh and do not affect the basic mesh in the local area, but are basic mesh sensitive in that all refinement levels are set with respect to the basic mesh cell.

The local mesh settings are applied to the cells intersected with the local mesh region which can be represented by a component, face, edge, vertex, or by simple geometric shape such as a cube, cylinder or sphere. By using **Equidistant Refinement** option, you can specify settings to refine the cells within the region bounded by surfaces that are equidistant (keep the same distance) from the selected objects. You can also apply local mesh settings to fluid regions and solid bodies. To apply the local mesh settings to a fluid region you need to specify this region as a solid part or subassembly and then disable this component in the **Component Control** dialog.

If a cell intersects with different local mesh setting regions, the refinement settings in this cell will be used to achieve the maximum refinement.

The local mesh settings have higher priority over the initial mesh settings. Therefore, the local mesh cells will be split to the specified local refinement levels regardless of the general L_{ssf}, L_{cur} and L_{ch} (specified in the **Global Mesh** dialog). This, however, may cause refinement of cells located outside of the local region due to imposing the *Cell Mating* rule.

Figure 5-13. The local mesh settings used: two narrow channels are refined to have 10 cells across them.
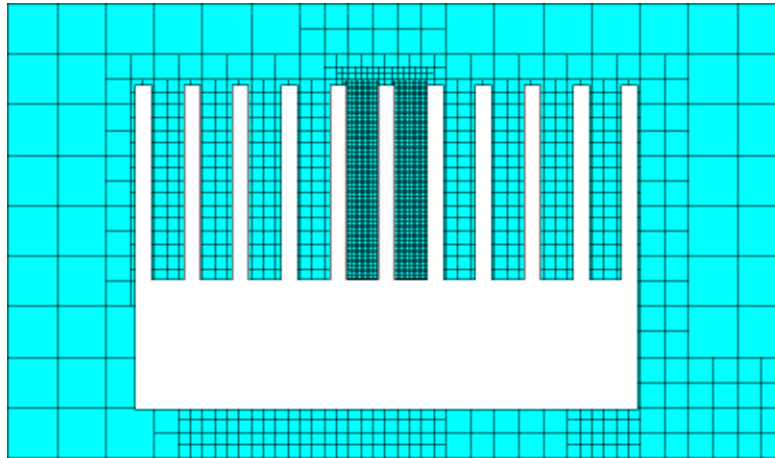
Recommendations for Creating the Computational Mesh

This topic provides step-by-step instructions how to create the **Computational Mesh**.

Procedure

1. First, define an appropriate problem type: Internal\External, Compressible\Incompressible, 3D\2D.

2. In case of **Internal flow** analysis, first of all, using information about the model geometry, estimate the ratios of overall model size to sizes of its smallest parts (or to heights of the narrowest flow passages).

   a. Note, that if you use some of engineering methods and techniques implemented in Flow Simulation (such as '*thin-channel*' and '*thin-boundary-layer*' approaches, see "Two-Scales Wall Functions Model" on page 99), in the most cases these approaches provide the good accuracy, even on a coarse mesh.

   b. If the estimated ratios are not very large, it is recommended to create the mesh using the default (**Automatic**) mesh settings. Start with the **Level of initial mesh** of 3 if both the model geometry and the flow field are relatively smooth. For more complex problems we recommend to perform the calculation at the result resolution level of 4 or 5. On this stage it is important to recognize the appropriate value of the **Minimum gap size** which will provide you with the suitable mesh. The default value of the **Minimum gap size** is calculated using information about the overall model dimensions, the **Computational Domain** size, and area of surfaces where conditions (boundary conditions, sources, etc.) and goals are specified.

   c. If the estimated ratio of overall model size to heights of the flow passages is rather large, closely analyze the obtained automatic mesh, paying attention to the total

numbers of cells, resolution of the regions of interest and narrow channels. If the automatic mesh does not satisfy you and changing of the **Minimum gap size** value do not give the desired effect you can proceed with the **Manual** mesh with the enabled **Channels** refinement.

d. If the **Channels** refinement does not allow to resolve the area of interest, try to use the **Local mesh** settings. In the most cases it makes sense to specify the **Refining Cells** by type. Note, that it is strictly recommended to construct a mesh as uniform as possible, especially in the high-gradient flow regions. So define the local mesh region so that its boundaries are not intersect the high-gradient flow regions.

e. To capture the relatively small solid features or to resolve the boundary between substances (fluid/solid, fluid/porous, porous/solid interfaces or boundary between different solids) you can use either the **Local mesh** settings or the **Advanced Refinement**. The **Local mesh** settings are preferable if there are few small features required to be resolved.

3. In case of **External flow** analysis, it makes sense to create the mesh using the default (**Automatic**) mesh settings.

a. Start with the **Level of initial mesh** of 3 if both the model geometry and the flow field are relatively smooth. For more complex problems we recommend, first of all, to perform the calculation at the result resolution level of 4 or 5.

b. As described above, note, that if you use some of Flow Simulation engineering techniques, in the most cases these approaches provide the good accuracy, even on a coarse mesh.

c. Closely analyze the obtained automatic mesh, paying attention to the total numbers of cells and resolution of the regions of interest. If the automatic mesh does not satisfy you, you can proceed with the **Manual** mesh with custom **Control Planes** and/or specify the **Local mesh** settings.

d. Use **Control Planes** to optimize the **Basic mesh** by setting custom control planes surrounding the region of interest and assigning the proper **Ratio** values to the respective intervals. Note, that it is strictly recommended to construct a mesh as uniform as possible, especially in the high-gradient flow regions.

e. If needed, try to use the **Local mesh** settings to resolve the area of interest. It makes sense to specify the **Refining Cells** by type or the **Equidistant Refinement** depending on the selected region.

4. In the high-gradient flow regions, it makes sense to use the *Solution-Adaptive Refinement*. At the beginning perform calculations with the disabled *Solution-Adaptive Refinement* to obtain the flow pattern on a coarse mesh, then perform calculations with enabled *Solution-Adaptive Refinement* to achieve the prescribed solution accuracy.

5. The problem of resolving a model with the **Computational Mesh** is always model-specific. In general, a denser mesh will provide better accuracy but you should tend to create an optimal mesh and to avoid redundant refinement.

6. When performing a calculation, try different mesh settings and analyze the obtained results carefully in order to understand whether it is necessary to refine the mesh or a coarser resolution is acceptable for the desired accuracy.

Solution-vs.-Mesh Convergence Investigation

One of the most overlooked issues in Computational Fluid Dynamics (CFD) that affects the accuracy of solution is the *mesh convergence*.

As it is mentioned in "Numerical Solution Technique" on page 95, Flow Simulation solves the governing equations numerically, in a discrete form on a computational mesh.

The accuracy of solution of the mathematical problem depends on how well the computational mesh resolves the regions of a non-linear behavior in the problem. To provide a good accuracy, the mesh has to be rather fine in these regions. Moreover, the usual way of estimating the accuracy of the solution consists in obtaining solutions on several different meshes, from coarse to fine. So, if beginning from some mesh in this set, the difference in the physical parameters of interest between the solutions obtained on the finer and coarser meshes becomes negligible from the standpoint of the solved problem (the solution flattens), then the accuracy of the solution of the mathematical problem is considered to be attained, since the so-called *solution mesh convergence* is attained.

The formal method of establishing the mesh convergence requires a curve of a critical result parameter (typically local or integral value) in a specific location, to be plotted against some measure of mesh density. Naturally, the Flow Simulation mesh convergence curve may looks non-monotonic. The reason for such behavior is that a number of engineering methods and techniques are implemented in Flow Simulation and different engineering techniques or their combinations are used automatically as the mesh gets finer or coarser (for example, see "Two-Scales Wall Functions Model" on page 99).

Figure 5-14. The critical result parameter value (*P*) vs.the total number of computational mesh cells (*N*$_{cell}$).
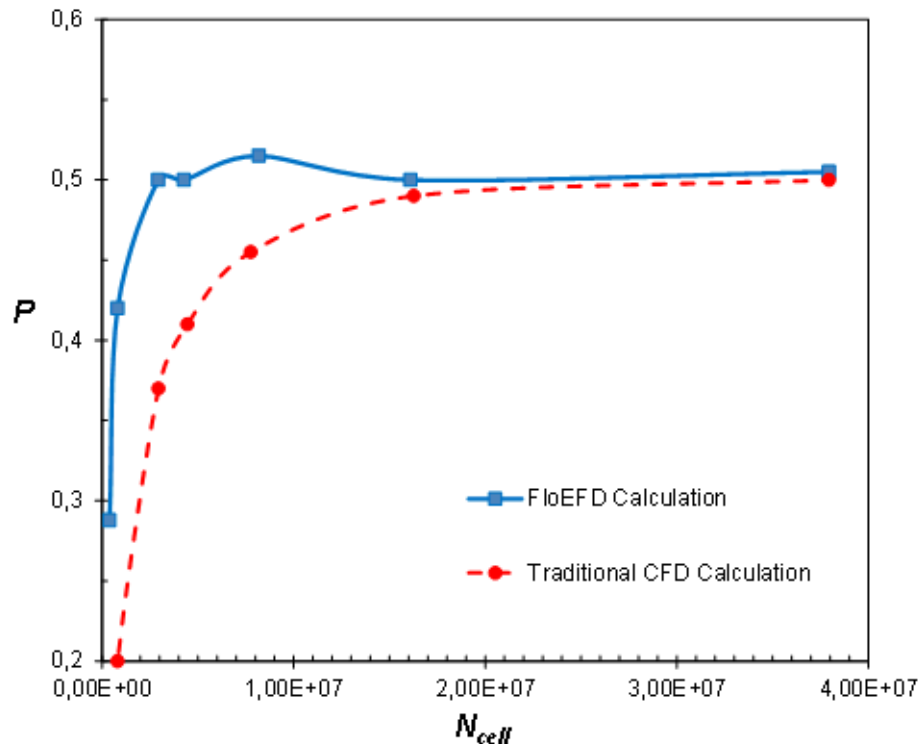
Therefore, the strategy of establishing mesh convergence with Flow Simulation consists, first of all, in performing several calculations on the same basic project (with the same model, inside the same computational domain, and with similar boundary and initial conditions) varying only the computational mesh and controlling if the same engineering techniques are enabled on the given mesh. Note moreover, that using engineering methods and techniques in Flow Simulation allows to obtain acceptable accuracy on coarser meshes as compared with traditional CFD codes.

This chapter provides the fundamentals of working with the **Calculation Control Options** dialog box and explains the use of parameters governing the calculation process.

The **Calculation Control Options** dialog box introduced into Flow Simulation allows you to control the following procedures:

- conditions of finishing the calculation,

- saving of the results during the calculation,

- refinement of the computational mesh during the calculation,

- freezing the flow calculation,

- time step for a time-dependent analysis,

- controlling the radiation model parameters on which the calculation time and accuracy depends.

This dialog box is accessible both before the calculation and during the calculation. In the last case the new-made settings are applied to the current calculation starting from the next iteration.

The main information on employing the options of **Finishing the calculation** and **Refining the computational mesh during calculation** is presented in this document.

Finishing the Calculation

Flow Simulation solves the time-dependent set of equations for all problems, including steady-state cases. For such cases it is necessary to recognize the moment when a steady-state solution is attained and therefore the calculation should be finished. A set of independent finishing conditions offered by Flow Simulation allow the user to select the most appropriate conditions and criteria on when to stop the calculation.

The following finishing conditions are offered by Flow Simulation:

- maximum number of refinements;

- maximum number of iterations;

- maximum physical time (for time-dependent problems only);

- maximum CPU time;

- maximum number of travels;

- convergence of the **Goals**.

Note

Travel is the number of iterations required for the propagation of a disturbance over the whole computational domain. Current number of iterations per one travel is presented in the Info box of the **Calculation monitor**.

In Flow Simulation you can select the finishing conditions that are most appropriate from your viewpoint to solve the problem under consideration, and specify their values. For the latter two conditions (i.e., for the maximum number of travels and the **Goals convergence** settings) Flow Simulation presents their default values (details are described below). You can also select the superposition mode for multiple finishing conditions in the **Finish Conditions** value cell: either to finish the calculation when all the selected finishing conditions are satisfied or when at least one of them is satisfied.

In any case, information on the finishing conditions due to which the calculation has finished is shown in the **Monitor**'s **Log** box.

The Goals convergence finishing condition is complex since it consists of satisfying all the specified Goals criteria. A specified Goal criterion includes a specified dispersion, which is the difference between the maximum and minimum values of the **Goal**, and a specified analysis interval over which this difference (i.e., the dispersion) is determined. The interval is taken from the last iteration rearwards and is the same for all specified **Goals**. The analysis interval is applied after an automatically specified initial calculation period (in travels), and, if refinement of the computational mesh during calculation is enabled, after an automatically or manually specified relaxation period (in travels or in iterations) since the last mesh refinement is reached. As soon as the Goal dispersion obtained in the calculation becomes lower than the specified dispersion, the **Goal** is considered converged. As soon as all Goals included in the Goals convergence finishing condition (by selecting them in the **On/Off** column) have converged, this condition is considered satisfied. The **Goals** not included into the Goals convergence finishing condition are used for information only, i.e., with no influence on the calculation finishing conditions.

Let us consider the Flow Simulation default values for the maximum number of travels and the Goals convergence settings in detail. These default (recommended by Flow Simulation) values depend on the **Result resolution level** either specified in the **Wizard** or changed by pressing the **Reset** button in the **Calculation Control Options** dialog box. For higher **Result resolution levels** the finishing conditions are tighter.

The default maximum number of travels depends on

- the type of the specified **Goal** (i.e., *dynamic* or *diffusive*);

  —————— **Note** ——————
  The *dynamic* goals are: Static Pressure, Dynamic Pressure, Total Pressure, Mass Flow Rate, Forces, Volume Flow Rate, and Velocity.
  The *diffusive* goals are: Temperature, Density, Mass in Volume, Heat flux, Heat transfer rate, Concentrations, Mass Flow Rate of species, and Volume Flow Rate of species.
  ————————————————————

- the specified **Result resolution level**;

- the problem's type (i.e., incompressible liquid or compressible gas, low or high Mach number gas flow, time-dependent or steady-state).

The default **Goals convergence** settings are the default analysis interval, which is shown in the **Finish** tab of the **Calculation Control Options** dialog box, and the default Goals criterion dispersion values, which are not shown in the **Calculation Control Options** dialog box, but, instead, are shown in the **Monitor**'s **Goal Table** or **Goal Plot** table (in the **Criteria** column), since they depend on the values of the Goal physical parameter calculated in the computational domain, and therefore are not known before the calculation and, moreover, can change during it. In contrast, the Goals criterion dispersion values specified manually do not change during the calculation.

As for the automatically specified initial calculation period (measured in travels), it depends on the problem type, the Goal type, and the specified **Result resolution level**.

Note, that the **Goal** features have the following characteristics depending on their type:

- the manually specified analysis interval for the **Goals convergence** finishing criteria must be substantially longer than the typical period of the flow field oscillation (if it occurs);

- the **Goals** determined on solid/fluid interfaces or model openings, as well as the post-processor **Surface Parameters**, yield the most accurate and correct numerical information on flow or solid parameters, especially integral ones;

- **Global Goals** yield the most reliable information on flow or solid parameters, although they may be too general;

- the CPU time depends slightly on the number of the specified **Goals**, but, in some cases, vary substantially in the case of presence of a **Surface Goal**;

- **Surface Goals** and **Volume Goals** provide exactly the same information that may be obtained via the **Surface Parameters** and **Volume Parameters** post-processor features, respectively.

Refinement of the Computational Mesh During Calculation

Solution-Adaptive Refinement (SAR) of the computational mesh during the calculation is a process of splitting the computational mesh cells in areas where the calculation error (specifically, the *local truncation error* (LTE)) may be sufficiently large and merging of the computational mesh cells in areas where the calculation error is definitely small.

In the solution-adaptive refinement the computational mesh cells are split until the specified **Refinement level** is satisfied. The **Refinement level** specifies how much times the initial mesh cells can be split to achieve the solution-adaptive refinement criteria, and thus governs the minimum computational mesh cell size. If specified **Refinement level** allows, the *re-meshing adaptation cycles* are performed with a sequentially increasing refinement level. If the **Refinement level** is already reached, the re-meshing occurs anyway, but at a constant refinement level.

If any local initial mesh is specified in the project, you can control the solution-adaptive refinement in these regions independently.

Figure 6-1. Refinement in the regions where the local initial meshes are specified: Global Domain level = 1, Local Mesh 1 level = 2, Local Mesh 2 level = Use Global, Local Mesh 3 = Disabled.
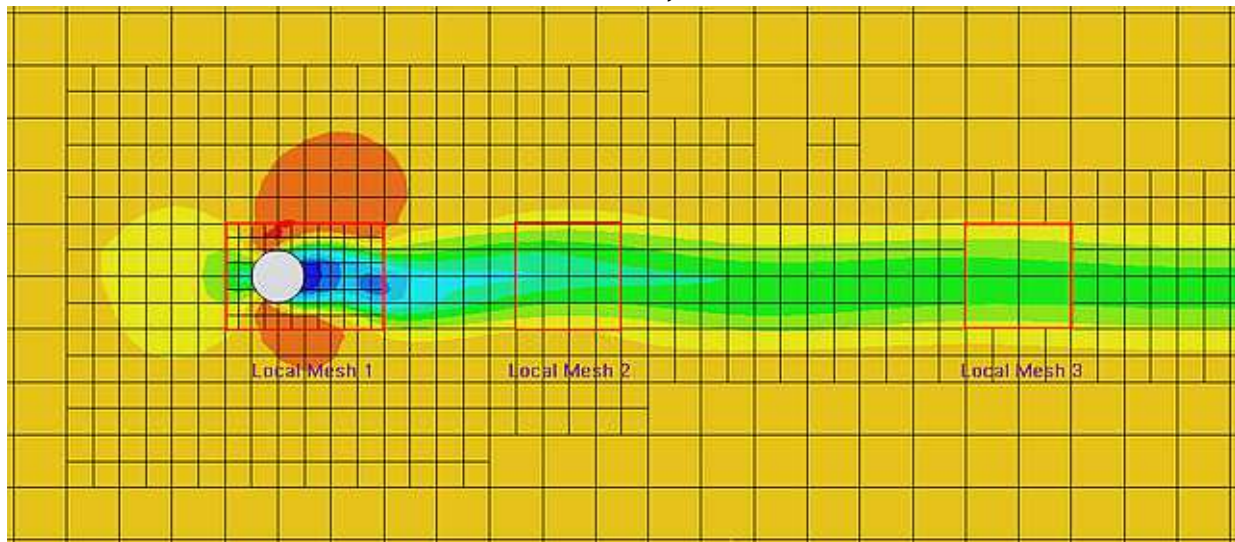
To locate regions of the computational domain that need mesh refinement, it is necessary to analyze the solution obtained with the mesh that existed at the last refinement cycle. *Indicator functions* are used to locate regions where the solution requires a mesh refinement to reduce the local truncation errors. The output of an indicator function is used to determine if the mesh cell must be split or merged or is adequately resolved.

The indicator function for the momentum conservation law is defined as follows:

(6-1)

$$C_m = \Gamma\left(\frac{n}{n_{min}}, foam\right) \cdot h \cdot S, , \quad \Gamma = 1 + \left(\frac{n}{n_{min}} - 1\right) \cdot foam$$

where $S = \sqrt{0.5 \cdot S_{ij} \cdot S_{ij}}$ is a convolution of the strain rate tensor, n is the total number of adjacent cells, n_{min} is the number of the coordinate directions (plus or minus) in which at least one cell is adjacent cell, $foam$ is the cell level gap displacement factor, which shifts the region where the mesh cells must be split to the area where the local truncation error is definitely small, h is the characteristic cell size defined as follows:

(6-2)

$$h = \frac{\sum\limits_{i \neq j}\sum\limits_{j} \left|\frac{\partial u_i}{\partial x_j}\right| h_j}{\sum\limits_{i \neq j}\sum\limits_{j} \left|\frac{\partial u_i}{\partial x_j}\right|}$$

where u is the fluid velocity.

The indicator functions for the mass, energy and species conservation laws are defined as follows:

(6-3)

$$C_\rho = \Gamma \cdot \left[\sum_i \left(\frac{1}{\rho}\frac{\partial \rho}{\partial x_i} h_i\right)^2\right]^{0.5}$$

$$C_e = \Gamma \cdot \left[\sum_i \left(\frac{1}{T}\frac{\partial T}{\partial x_i} h_i\right)^2\right]^{0.5}$$

$$C_y = \Gamma \cdot \left[\sum_i \left(\frac{\partial y}{\partial x_i} h_i\right)^2\right]^{0.5}$$

The mesh cell are split if the following condition is satisfied:

$$C_m > \varepsilon_{split}^m \text{ or } C_\rho > \varepsilon_{split}^\rho \text{ or } C_e > \varepsilon_{split}^e \text{ or } C_y > \varepsilon_{split}^y$$

and the mesh refinement level is less than the current maximum refinement level.

The child mesh cells are merged if all following condition is satisfied for each child cell:

$$C_m < \varepsilon_{merge}^m \text{ and } C_\rho < \varepsilon_{merge}^\rho \text{ and } C_e < \varepsilon_{merge}^e \text{ and } C_y < \varepsilon_{merge}^y$$

and if the refinement level difference between the resulting merged mesh cells and their neighbors will not exceed 1.

All limiting values (ε_{split} and ε_{merge}) are determined automatically at each refinement cycle by using mesh histograms.

The solution-adaptive refinement can dramatically increase the number of cells so that the available computer resources (physical RAM) will not be enough for running the calculation. To limit the total number of cells the **Approximate Maximum Cells** value can be specified. If the maximum cells number is exceeded, the number of cells will be limited for the last refinement cycle in such a way that the LTE, increasing at the refinement level gap surface, is minimized.

For a transient analysis the following strategies are available:

- **Periodic** refinement;

- **Tabular** refinement;

- **Manual Only** refinement;

- **Goal Convergence** refinement.

In the first two strategies the refinement moment is known beforehand. The solution gradients are analyzed over iterations belonging to the **Relaxation interval**, which is calculated from the current moment rearwards. As the result, only steady-state gradients are considered. The default length of the **Relaxation interval** can be adjusted manually. On the other hand, the analysis must not continue with the same relaxation interval defined from the start of the calculation, in order to avoid considering the initial highly unsteady period. Therefore, a period of at least two relaxation intervals is recommended before the first refinement. If the first assigned refinement is scheduled in a shorter term from the beginning, the period over which the gradients are analyzed is shortened accordingly, so that in an extreme case it can be as short as one current iteration. If you initiate a refinement manually within this period, the gradients are analyzed in one current iteration only. Naturally, such a short period gives not very reliable gradients and hence may result in an inadequate solution or excessive CPU time and memory requirements.

The Figure 6-2 illustrates this concept. Here, the letter '*r*' denotes the relaxation interval. This figure involves both the **Periodic** and **Tabular** refinements. Case 1 is the recommended normal approach. In the Case 2 the first refinement is too close to the starting point of the calculation, so the gradients are analyzed over the shorter interval (which could even be reduced to just one current iteration in an extreme case). Case 3 is a particular case when the refinement is initiated manually just before a previously assigned refinement. As the result, the manual refinement is well-defined, since the gradients have been analyzed over almost the entire relaxation interval,

but on the other hand, the previously assigned refinement is performed on the substantially shorter interval, and therefore its action can be incorrect. Thus, Case 3 demonstrates the possible error of performing manual and previously assigned refinements concurrently.

Figure 6-2. Refinement strategy: green line - collecting of the statistics is prohibited; blue line - monitoring interval where statistics are collected, red dot - refinement point.
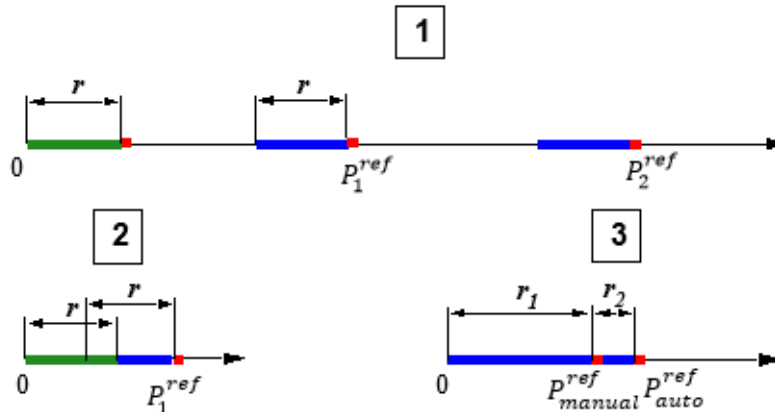
If you select a **Goal Convergence** refinement, you can select goals to refine the computational mesh at the moment when all specified goals have converged. Also you can specify the **Delay** to start refining a specified time after the goals convergence occurred.

The mesh refinement performed during the calculation is idling and the warning messages appear in the **Monitor** window in the following cases:

- **Refinement canceled: the limit of maximum number of cells was achieved**. This message warns that the **Approximate Maximum Cells** is reached.

- **Refinement canceled: refinement-unrefinement criteria is not satisfied**. This message indicates that there are no mesh cells where the conditions of splitting of merging are satisfied.

- **Refinement canceled: the limit of maximum refinement level was achieved in field gradients**. If there are mesh cells where the conditions of splitting or merging are satisfied, but the Refinement level has been already reached, since the field gradients have not been changed, this message appears. If the field gradients are changed and the mesh refinement is performed again, the mesh refinement will result in splitting or merging cells.

Flow Freezing

This topic provides fundamentals of working with the **Flow Freezing** option and explains the use of parameters governing the freezing procedure.

What is Flow Freezing?

Sometimes it is necessary to solve a problem that deals with different processes developing at substantially different rates. If the difference in rates is substantial (10 times or higher) then the CPU time required to solve the problem is governed almost exclusively by the slower process. To reduce the CPU time, a reasonable approach is to stop the calculation of the fastest process (which is fully developed by that time and does not change further) and use its results to continue the calculation of the slower processes. Such an approach is called '*freezing*'.

In the case of problems solved with Flow Simulation the processes of convective mass, momentum, and energy transport are the fastest processes to develop and to converge, whereas the processes of mass, momentum, and energy transfer by diffusion are the slowest ones. Accordingly, Flow Simulation offers the **Flow Freezing** option that allow you to freeze, or fix, the pressure and velocity field while continuing the calculation of temperature and composition. This option is especially useful in solving steady-state problems involving diffusion processes that are important from the user's viewpoint, e.g. species or heat propagation in dead zones of the flow. Time-dependent analyses with nearly steady-state velocity fields and diffusion processes developing with time are also examples of this class of problems. As a result, the CPU time for solving such problems can be substantially reduced by applying the **Flow Freezing** option.

Flow Simulation treats **Flow Freezing** for the **High Mach number flows** differently. All flow parameters are frozen, but the temperature of the solid is calculated using these fixed parameters at the outer of the boundary layer and user defined time step. Temperature change on the solid's surface and relevant variation of the heat flows are accounted in the boundary layer only. It is impossible (and makes no sense) to use **Flow Freezing** for calculation of concentration propagation in the **High Mach number flow**. If custom time step is not specified, the steady-state temperature of solid will be reached in one time step assumed to be infinite.

How It Works

To access the **Flow Freezing** option, open the **Calculation Control Options** dialog box and then the **Solving** tab.

This option has three modes: **Disabled** (by default), **Periodic**, and **Permanent**.

Flow Freezing in a Permanent Mode

As an example of applying the **Flow Freezing** option, let us consider a plane flow (2D) problem of heating the vortex core in a vessel.

See Figure 6-3.

Figure 6-3. Heating the vortex core in a vessel.
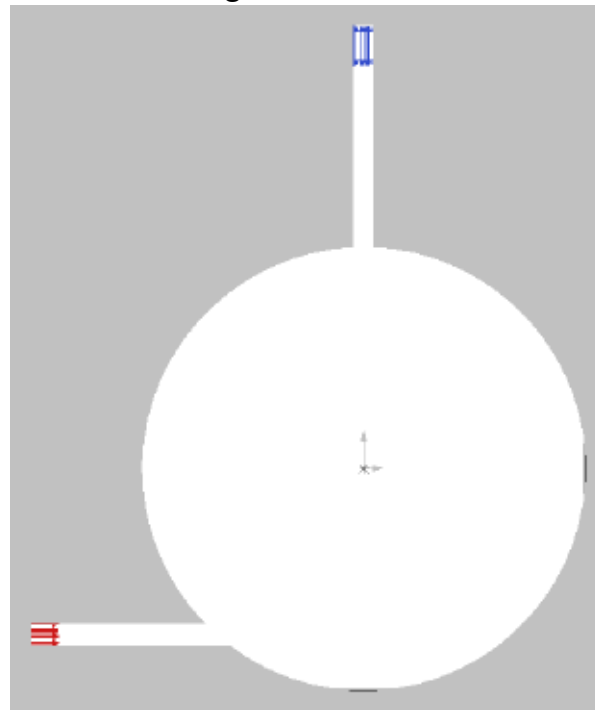
At the beginning the entire fluid region is filled with a cold (T_c = 300 K) liquid. A hot (T_h = 400 K) liquid enters the vessel through the lower channel (the upper channel is the exit).

As a result, a vortex with a cold core is developed in the vessel. The vortex core temperature is changed mainly due to heat diffusion. To measure it, a small body is placed at the vortex center and disabled in the **Component Control** dialog box, so that it is treated by Flow Simulation as a fluid region. Its minimum temperature (i.e., the minimum fluid temperature in this region) is the **Volume Goal** of the calculation.

First of all, let us consider **Flow Freezing** operating in the **Permanent** mode. The only user-specified parameter in **Permanent** mode is the starting moment of enabling the **Flow Freezing** option. Until this moment the calculation runs in a usual manner. After this moment the fluid velocity field becomes frozen, i.e., it is no longer calculated, but is taken from the last iteration performed just before the Flow Freezing **Start** moment. For the remainder of the run only the equations' terms concerning heat conduction and diffusion are calculated. As a result, the CPU time required per iteration is reduced.

The starting moment of the **Flow Freezing** option should be set not too early in order to let the flow field to fully develop. As a rule, an initial period of not less than 0.25 travels is required to satisfy this condition. In most problems the 0.5 travel initial period is sufficient, but there are problems that require a longer initial period.

Tip

ⓘ The Flow Freezing **Start** moment, as well as other parameters of the **Calculation Control Options** dialog box can be changed during a calculation.

Tip

ⓘ As soon as the **Flow Freezing** option is invoked, only the slowest processes are calculated. As a result, the convergence and finishing criteria can become non-optimal. Therefore, to avoid obtaining incorrect results when enabling the **Flow Freezing** option, it is recommended to increase the maximum number of travels specified at the **Finish** tab of the **Calculation Control Options** dialog box by 1.5…5 times compared to the number that was set automatically or required for the calculation performed without the **Flow Freezing** option.

When first solving the problem under consideration we set the maximum number of travels to 10. The calculation performed without applying the **Flow Freezing** option then required about 10 travels to reach the convergence of the project **Goal** (the steady-state minimum fluid temperature in the vortex core). However, the steady-state fluid velocity field was reached in about 0.5 travels, i.e., substantially earlier. So, by applying the **Flow Freezing** option in the Permanent mode (just after 0.5 travels) the same calculation requires substantially less time on the same computer to reach the convergence of the project **Goal**. If it is necessary to perform several calculations with the same fluid velocity field, but different temperatures and/or species concentrations, it is expedient to first calculate this fluid velocity field without applying the **Flow Freezing** option. Then, clone the Flow Simulation project into several projects (including copying the calculation results), make the required changes to these projects, and perform the remaining calculations for these projects using the calculated results as initial conditions and applying the **Flow Freezing** option in the **Permanent** mode with a zero **Start** period.

Tip

ⓘ If you forget to use the calculated results as initial conditions, then the saved fluid velocity field will be lost in the cloned project, so the project must be created again. To use the calculated results as initial conditions for the current project, select the **Transferred** type of **Parameter** definition for the initial conditions in the **General Settings** dialog box.

Flow Freezing in a Periodic Mode

In some problems the flow field depends on temperature (or species concentrations), so both the velocity and the temperature (concentrations) change simultaneously throughout the calculation. Nevertheless, since they change in a different manner, i.e., the velocity field changes faster than the temperature (concentrations) field, therefore approaching its steady state solution earlier, the **Flow Freezing** option can be used in a Periodic mode to reduce the CPU time required for solving such problems.

The Periodic mode of the **Flow Freezing** option consists of calculating the velocity field not in each of the iterations (time steps), but periodically for a number of iterations specified in **No freezing (iterations)** after a period of freezing specified in the **Freezing (iterations)** (see Figure 6-4) The temperatures and concentrations are calculated in each iteration. Examples include channel flows with specified mass flow rates and pressures, so the fluid density and, therefore, velocity depend on the fluid temperature, or flows involving free convection, where due to the buoyancy the hot fluid rises, so the velocity field depends on the fluid temperature.

Figure 6-4. The Periodic mode of the Flow Freezing (V - freezing, X - no freezing).
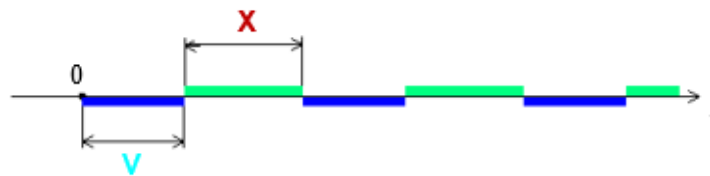
As an example, let us consider a 3D external problem of an air jet outflow from a body face into still air (see Figure 6-5, in which the jet outflow face is marked by a red line). Here, the wire frame is the computational domain. The other body seen in this figure is introduced and disabled in the **Component Control** dialog box (so it is a fluid region) in order to see the air temperature averaged over its face (the project **Goal**), depending on the air temperature specified at the jet outflow face.

Figure 6-5. Air jet outflow from a body face into a still air.
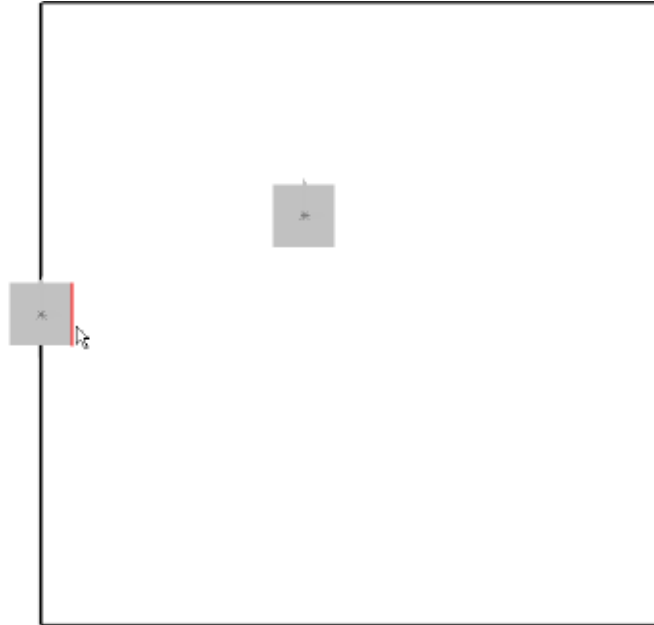
This problem is solved in several stages. At the first stage, the calculation is performed for the cold (T_c = 300 K, which is equal to the environment temperature) air jet. Then we clone the project including copying the results. Next, we set the outlet air temperature to T_h = 400 K, specify the **Periodic** mode of the **Flow Freezing** option by its **Start** moment of 0.25 travels (in order for the heat to have time to propagate along the jet to the measuring face) and under **Duration** specify 10 as both the **Freezing (iterations)** and **No freezing (iterations)** values.

Then perform the calculation on the same computational mesh with the **Take previous results** option in the **Run** box.

As a result, the calculation with flow freezing takes less CPU time than the similar calculation without the **Flow Freezing** option enabled.

Radiation Freezing

Radiation calculation might be CPU consuming and radiation process might develop faster than fluid flow. To reduce the CPU time, the radiation calculation can be stopped and you can use the previous iteration values of all radiation parameters to continue the calculation.

To access the **Radiation Freezing** option, open the **Calculation Control Options** dialog box, then the **Solving** tab. This option has three modes: **Disabled** (by default), **Periodic**, and **Auto**.

Radiation Freezing in a Periodic Mode

The **Periodic** freezing strategy allows to calculate the flow field and the radiation simultaneously throughout the calculation. Also this strategy allows to specify (in the iterations) the Periodicity, which means that the radiation calculates every specified iterations only, which allows to reduce the CPU time required for solving such problems.

Radiation Freezing in an Automatic Mode

The **Automatic** freezing strategy means that the radiation calculates every freezing step only and this strategy is similar to the **Periodic** mode.

However the **Automatic** mode allows to vary the freezing period automatically and it depends on the selected radiation model:

- If the **Discrete Transfer** model is selected, the freezing step can be automatically varied from 3 to 10 iterations.

- If the **Discrete Ordinates** model is selected, the freezing strategy is disabled.

Also in case of the **Time-dependent** (transient) or **Heat conduction in solids only** analysis, the freezing strategy is disabled too.

The list of references arranged alphabetically by author is given below.

1. Carlson, D. J., & Hoglund, R. F. (1964). *Particle drag and heat transfer in rocket nozzles*. AIAA Journal, 2(11), 1980-1984.

2. Faeth, G. M. (1983). *Evaporation and combustion of sprays*. Progress in Energy and Combustion Science, 9(1-2), 1-76.

3. Ginzburg, I. P. (1970). *Teoriya soprotivleniya i teploperedachi [Theory of Drag and Heat Transfer]*. Leningrad: Leningr. Gos. Univ [in Russian].

4. Glowinski, R., & Le Tallec, P. (1989). *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics (Vol. 9)*. SIAM.

5. Hackbusch, W. (1985). *Multi-grid Methods and Applications*. Berlin: Springer-Verlag.

6. Henderson, C. B. (1976). *Drag coefficients of spheres in continuum and rarefied flows*. AIAA Journal, 14(6), 707-708.

7. Hirsch, C. (1988). *Numerical computation of internal and external flows, Volume 1: Fundamentals of numerical discretization*. John Wiley and Sons, 9, 10.

8. Idelchik, I. E. (1986). *Handbook of hydraulic resistance*. Washington, DC, Hemisphere Publishing Corp., 662 p. Translation.

9. ISO 7726:1998, ED 2. *Ergonomics of the Thermal Environment. Instruments for Measuring Physical Quantities*. Geneva: International Organization for Standardization.

10. ISO 7730:2005, ED 3. *Ergonomics of the thermal environment – Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria*. Geneva: International Organization for Standardization.

11. Marchuk, G. I. (1982). *Methods of Numerical Mathematics*. Applications of Mathematics.

12. Patankar, S.V., (1980). *Numerical Heat Transfer and Fluid Flow*. New York: Hemisphere Publ. Corp.

13. Reid, R. C., Prausnitz, J. M., & Poling, B. E. (1987). *The properties of gases and liquids*. 4'th ed., New York: McGraw-Hill.

14. Roache, P. J. (1998). *Technical Reference of Computational Fluid Dynamics*. Albuquerque, New Mexico: Hermosa Publishers.

15. Saad, Y. (1996). *Iterative methods for sparse linear systems*. Boston: PWS Publishing Company.

16. Samarskii, A. A. (2001). *The theory of difference schemes (Vol. 240)*. CRC Press.

17. Schmehl, R., Klose, G., Maier, G., & Wittig, S. (1998). *Efficient numerical calculation of evaporating sprays in combustion chamber flows*. In RTO AVT Symposium on "Gas Turbine Engine Combustion, Emissions and Alternative Fuels", Lisbon, Portugal, 12-16 October 1998; RTO MP-14.. NATO.

18. Van Driest, E. R. (1956). On turbulent flow near a wall. Journal of the aeronautical sciences, 23(11), 1007-1011.

19. Wessel, D. J. (2001). *ASHRAE fundamentals handbook 2001 (SI edition)*. American Society of Heating, Refrigerating, and Air-Conditioning Engineers, 31.

20. White, W. B., Johnson, S. M., & Dantzig, G. B. (1958). *Chemical equilibrium in complex mixtures*. The Journal of Chemical Physics, 28(5), 751-755

Index

Reynolds-stress tensor, 16
Rotation
    global, 62
    local regions
        averaging, 62, 63
        sliding, 62, 64

— S —

Semi-transparent. See Absorption
SIMPLE, 101
Slip velocity, 87
Snell's law, 50
Spectrum, 55
State equation of ideal gas, 19

— T —

Thermal joint, 88
Thermoelectric cooler (TEC), 92
Thick-boundary-layer, 99
Thin-boundary-layer, 99
Thin-channel, 99
Total accretion mass rate, 72
Total erosion mass rate, 72
Tracer study, 76
Transmission, 51
Travel, 142
Two-resistor, 89
    board, 89
    case, 89
    junction, 89
Two-scales wall functions (2SWF), 99

— V —

View factor resolution level, 45
Volume of fluid (VOF), 73