SOFTWARE USER GUIDE

(Theory and general overview)

Computational Fluid x-Dynamics (CFxD)

(Version 2.11.05, July 12, 2026)

2026

Contents

Nomenclature

Symbol	Dimension	Definition
C_i , c_i		model coefficients
c_p	$J/(kg \times K)$	specific heat capacity by constant pressure
e_{ijk}		Levi-Civita tensor
g_i	m/s^2	gravity acceleration
Kn		Knudsen number
k	m^2/s^2	turbulent kinetic energy
l	m	characteristic length scale
M		Mach number
m	kg	mass
Nu		Nusselt number
P_k		turbulent production
Pr		Prandtl number
Pr_t		Prandtl turbulent number
p	Pa	static pressure
R	$J/(kg \times K)$	gas constant
Re		Reynolds number
Sc		Schmidt number
S_{ij}	$1/s$	mean strain-rate tensor (components)
T	K	temperature
t	s	time
u, u_i	m/s	velocity vector/components
$\overline{u_i' u_j'}$	m^2/s^2	Reynolds stress tensor (components)
$\overline{u_j' \varphi'}$		scalar flux (components)
V	m^3	volume
x_i	m	cartesian coordinates

Greek letters

Symbol	Dimension	Definition
$\alpha_i, \beta_i, \delta_i, \gamma_i$		model coefficients

$\dot{\gamma}$		strain rate
Γ_{φ}		Molecular diffusion coefficient of a general scalar quantity, φ
δ_{ij}		Cartesian components of unit tensor (Kronecker delta)
ε	m^2/s^2	dissipation rate of turbulent kinetic energy
λ	$J/(s \times K \times m)$	thermal conductivity
φ		general scalar quantity
κ		Karman constant
μ, μ_t	$kg/(m \times s)$	dynamic molecular/turbulent viscosity
ν, ν_t	m^2/s	kinematic molecular/ turbulent viscosity
ρ	kg/m^3	density
σ	m^2	surface
σ_{φ}		turbulent Schmidt (or Prandtl) number for variable
τ	s	turbulent time scale
τ_{ij}		Reynolds stress tensor (components)
ω		specific dissipation rate

Abbreviations

Abbreviation	Definition
CAD	Computer-aided design
CDS	Central Difference Scheme
CFD	Computational Fluid Dynamics
CV	Control Volume
DNS	Direct Numerical Simulation
EVM	Eddy-Viscosity Model
LES	Large Eddy Simulation
RANS	Reynolds Averaged Navier-Stokes
RSM	Reynolds Stress Model
RST	Reynolds Stress Tensor
SIMPLE	Semi Implicit Method for Pressure Linked Equations
SIP	Strongly Implicit Procedure
UDS	Upwind Difference Scheme
URANS	Unsteady RANS

Introduction

Computational Fluid x-Dynamics (CFxD) is a software application designed to model and simulate fluid flow. It includes the complete workflow for Computational Fluid Dynamics (CFD) analysis, from geometry creation to result visualization, in a streamlined and user-friendly interface. The typical simulation pipeline in CFxD includes:

Geometry → Mesh → Setup → Solution → Result.

CFxD allows users to create geometrical models from scratch or import existing geometry, generate computational meshes, define physical models and boundary conditions, solve the flow field using robust solvers, and visualize simulation results.

The software features a simplified and intuitive graphical interface, designed to provide a convenient tool for engineers without extensive experience in CFD. The current version supports **steady single-phase flow** in both **laminar and turbulent regimes**. It supports **incompressible** and **compressible** flow, as well as **heat transfer**, including **conjugate heat transfer**.

CFxD functions as a graphical shell that leverages free and open-source libraries, including:

- Open CASCADE Technology (OCCT) for CAD and geometry operations
- OpenFOAM for mesh generation and flow simulation
- VTK (Visualization Toolkit) for post-processing and result visualization.

The application has been tested on **Windows 10/11** platform. Hardware requirements depend on the complexity of the CAD geometry and mesh density. CFxD supports multi-core CPUs to accelerate computation.

Chapter I. Theoretical background

CFxD is based on the principles of **Computational Fluid Dynamics (CFD)**, which involves solving the governing equations of fluid motion using numerical methods. This section provides a brief overview of the physical models and numerical techniques used in CFxD.

1.1 Governing equations

Mass conservation equation

The **integral form** of the continuity equation (mass conservation) over a fixed control volume V with enclosing surface S is

$$\underbrace{\frac{\partial}{\partial t}\int_V \rho dV}_{I} + \underbrace{\int_\sigma \rho\vec{u}\,\vec{n}d\sigma}_{II} = 0\,, \tag{1}$$

where ρ is fluid density, $\vec{u}$ is velocity vector and $\vec{n}$ is the outward-pointing unit normal vector on the surface S. This equation states that the rate of mass change inside the volume (Term I) is equal to the net mass flux across the surface (Term II).

Applying **Gauss divergence theorem**, the surface integral is converted to a volume integral:

$$\frac{\partial}{\partial t}\int_V \rho dV + \int_V div(\rho\vec{u})dV = 0\,. \tag{2}$$

The corresponding **differential form** of the continuity equation is

$$\frac{\partial\rho}{\partial t} + \frac{\partial\rho u_i}{\partial x_i} = 0\,. \tag{3}$$

For **incompressible flow** ($\rho = const$) the continuity equation reduces to

$$\frac{\partial u_i}{\partial x_i} = 0\,. \tag{4}$$

Momentum conservation equation

The **integral form** of momentum conservation equation is

$$\frac{\partial}{\partial t}\int_V \rho\vec{u}dV + \int_S \rho\vec{u}(\vec{u}\cdot\vec{n})dS = \int_S \vec{\tau}\cdot\vec{n}dS + \int_V \rho\vec{g}dV, \tag{5}$$

where $\vec{g}$ is the gravity acceleration vector and τ is the stress tensor, which includes both pressure and viscous contributions.

Applying Gauss divergence theorem to the surface integrals and assuming the control volume becomes infinitely small, the differential form of the momentum conservation equation becomes:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i. \tag{6}$$

For Newtonian fluids, the stress tensor τ_{ij} is modeled using the Stokes hypothesis [9] as

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_\kappa}\delta_{ij}, \tag{7}$$

where is μ dynamic viscosity and δ_{ij} is Kronecker delta.

Taking into account (7), the equation for the conservation law of momentum can be written in the following form:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_\kappa}\delta_{ij}\right] + \rho g_i\ . \tag{8}$$

This equation is known as Navier-Stokes equation. For incompressible flows with constant viscosity, Navier-Stokes equation simplifies to

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\mu}{\rho}\frac{\partial^2 u_i}{\partial x_i \partial x_j} + g_i. \tag{9}$$

For flow in a rotating reference frame, the Coriolis $\left(-2\vec{\Omega} \times \vec{u}\right)$ and the centrifugal acceleration $\left(-\vec{\Omega} \times \vec{\Omega} \times \vec{x}\right)$ must be added:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p^*}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + f_i^*, \tag{10}$$

where the quantity pressure p^* is defined as

$$p^* = p - \frac{1}{2}\rho[(x_i x_i)(\Omega_k \Omega_k) - (x_k \Omega_k)^2]. \tag{11}$$

The body-force term including the Coriolis contribution is

$$f_i^* = f_i - 2\rho e_{ijk}\Omega_j u_k, \tag{12}$$

where e_{ijk} is the Levi-Civita tensor.

Energy, scalar, and species conservation equation

The **integral form** of scalar transport equation is

$$\frac{\partial}{\partial t}\int_V \rho\phi dV + \int_S \rho\phi\vec{u}\cdot\vec{n}dS = \int_S D_\phi \cdot \vec{n}\, dS + \int_V S_\phi dV. \tag{13}$$

The diffusive flux D_ϕ can be modeled using Fourier's law, which relates the flux to the gradient of a scalar. Applying Gauss divergence theorem and and assuming the control volume becomes infinite small, the **differential form** of the scalar transport equation is:

$$\frac{\partial\phi}{\partial t} + \frac{\partial\phi u_i}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\Gamma_\phi \frac{\partial\phi}{\partial x_i}\right) + S_\phi. \tag{14}$$

The diffusion coefficient Γ_ϕ is generally expressed as a ratio of kinematic viscosity ν to Prandtl number Pr :

$$\Gamma_\phi = \frac{\nu}{Pr}. \tag{15}$$

Prandtl number is often used in the heat transfer calculations. It characterizes the ratio of momentum and thermal diffusivities in fluid. The typical values for diatomic gases are $Pr \geq 0.72$ and for triatomic and polyatomic are $0.75 \leq Pr \leq 1$.

Substituting $\phi = c_p T$, the scalar transport equation becomes the conservation of energy in differential form

$$\rho\frac{\partial c_p T}{\partial t} + \frac{\partial u_i c_p T}{\partial x_j} = -\frac{\partial q_i}{\partial x_i} + \frac{\partial \tau_{ij} u_i}{\partial x_j} + S\,, \tag{16}$$

where $q_i = -k\frac{\partial T}{\partial x_i}$ defined by Fourier's law.

Gas mixture can be modeled with a transport equation for the species mass fractions. The mass fraction of species i is defined as

$$Y_i = \frac{m_i}{m}, \tag{17}$$

where m_i is the mass of species and m is the total mas of the gas mixture. The sum of all species mass fractions is equal to one:

$$\sum_{i=1}^{N} Y_i = 1. \tag{18}$$

For a non-reacting gas mixture, the conservation equation for species i can be written in differential form as

$$\frac{\partial\rho Y_i}{\partial t} + \frac{\partial\rho u_j Y_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\rho D_{i,e}\frac{\partial Y_i}{\partial x_j}\right), \tag{19}$$

where Y_i is the mass fraction of species i, $D_{i,e}$ is the effective mass diffusivity, and ρ is the mixture density.

Equation of state

To **close the system of governing equations**, an additional relationship is required to connect thermodynamic variables such as **pressure**, **temperature**, and **density**. This is provided by an **equation of state** (EOS), which relates to fluid properties. A commonly used EOS for gases is the **ideal gas law**:

$$p = \rho R T, \tag{20}$$

where p is pressure, ρ is density, R is the specific gas constant and T is temperature. The ideal gas law assumes that gas molecules do not interact with each other and occupy no volume. It is valid for many engineering applications.

1.2 Flow regimes

CFxD supports the simulation of both laminar and turbulent flows. The term *laminar* originates from the Latin word “*lamina”*, meaning a plate or a layer. In laminar flow, fluid moves in parallel layers with minimal mixing between them, typically at **low velocities**. An example of laminar flow through a channel is shown in Figure 1, where smoke added to the air helps visualize the smooth flow paths.

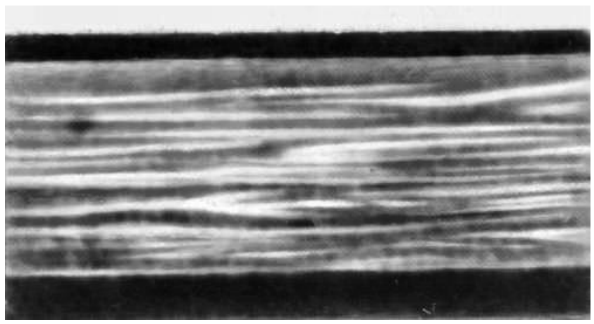


Figure 1. Laminar flow in a channel.

As the flow speed increases, the fluid layers begin to interact and mix. This intermediate state is known as **transitional flow**. Eventually, the flow may become **fully turbulent**, characterized by chaotic, irregular motion and strong mixing (Figure 2).

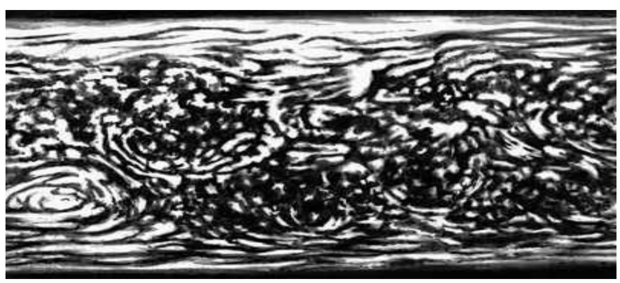


Figure 2. Turbulent flow in channel.

The flow regime can be characterized by Reynolds number, defined as

$$Re = \frac{l_0 \cdot u_0}{\nu}, \tag{21}$$

where l_0 is a characteristic length, u_0 is a characteristic (bulk) velocity and ν is a kinematic viscosity of fluid. The *Reynolds number* expresses the ratio between inertial and viscous (or molecular) forces. If this ratio is small, the viscous (or molecular) forces are comparable to the inertial forces, and the flow keeps its regular structure (laminar flow). If the ratio becomes larger, the viscous forces do not suffice to compensate the inertial forces. The flow turns unstable, and small initial perturbations destroy the regular flow structure, causing turbulence (turbulent flow). For internal flow, such flow in a pipe, the transition to turbulence typically occurs around $Re > 2300$.

1.3 Compressibility

CFxD supports both **incompressible** ($\rho = const$) and **compressible** ($\rho \neq const$) flow simulations. These terms refer to whether the **fluid density** varies within the fluid:

- A flow is considered **incompressible** when the fluid density remains approximately the same
    - For example, **water** is typically modeled as incompressible because its density changes very little under pressure.
    - **Air** can also be treated as incompressible **at low velocities**, where density variation is negligible.
- A flow is **compressible** when changes in pressure or temperature cause **significant variations in fluid density**. This is typical at **high velocities**, such as in gas dynamics or supersonic flows.

The degree of compressibility is often characterized by **Mach number**

$$M = \frac{u}{a}, \tag{22}$$

where u is flow velocity and a is speed of sound in the medium. At low speeds $M < 0.3$, the flow is generally considered incompressible. At high speeds, $M \geq 0.3$, the compressibility effects begin to play a significant role. At $M > 1$, the flow becomes supersonic and special phenomena such as shock waves may occur. An example of compressibility is shown in Figure 3, where a sphere moving at high speed creates a **shock wave** — a sharp pressure discontinuity caused by the fluid being compressed faster than disturbances can propagate.

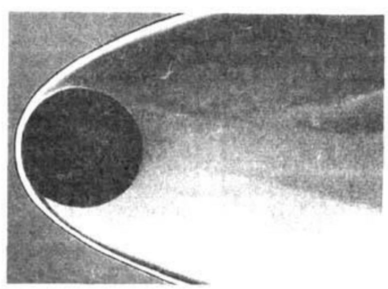


Figure 3. A shock wave in front of a sphere moving at a high speed.

Similar shock waves may occur in internal flows, such as nozzles or channels, when the flow velocity exceeds the local speed of sound. In such cases, mesh refinement near the shock is often necessary to accurately resolve the steep gradients in pressure, temperature, and velocity**.**

1.4 Unsteadiness

CFxD supports both **steady** and **unsteady** flow simulations. All flows are **unsteady** — they change with time. However, if the flow reaches a statistically stable pattern or the time-dependent effects are negligible, it can be approximated as **steady**, meaning the flow variables do not vary with time, t:

- **Steady flow**: $\frac{\partial}{\partial t} = 0$
- **Unsteady flow**: $\frac{\partial}{\partial t} \neq 0$

Choosing between steady and unsteady modeling depends on the physical nature of the problem and the level of accuracy required. For example:

- Flow through a long straight pipe at constant inlet conditions can often be treated as steady.
- Vortex shedding behind a cylinder or pressure fluctuations in a combustion chamber require unsteady modeling.

1.5 Turbulence models

CFxD supports two commonly used turbulence models:

- **k–ε model**
- **Shear Stress Transport (SST) model**

Both are based on the **Reynolds-Averaged Navier–Stokes (RANS)** approach, which simplifies turbulent flows by averaging out high-frequency fluctuations in time. This allows efficient simulation of turbulent flows without resolving all the small-scale structures, as would be required in **Direct Numerical Simulation (DNS)** or **Large Eddy Simulation (LES)**.

Reynolds-Averaged Equations

In the RANS framework, each variable (e.g., velocity, pressure, temperature) is decomposed into:

$$u_i = \overline{u}_i + u_i', \tag{23}$$

where $\overline{u}_i$ is the time-averaged (mean) velocity and u_i' is the fluctuating component. This decomposition results in the **averaged equations** for incompressible flow:

- **Continuity**:

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0, \tag{24}$$

- **Momentum**:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} - \frac{\partial}{\partial x_j}\left(\nu \frac{\partial \bar{u}_i}{\partial x_i} + \frac{\partial \overline{u_i' u_j'}}{\partial x_j}\right) + g_i. \tag{25}$$

The term $\overline{u_i' u_j'}$ is the **Reynolds stress tensor**, which introduces new unknowns. These need to be modeled using turbulence models — a challenge known as the **closure problem**. Most of turbulence models used in the engineering applications are based on the concept of eddy viscosity [9]. Reynolds stress tensor can be represented as a product of eddy viscosity ν_t and mean strain-rate tensor $\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)$:

$$-\overline{u_i' u_j'} = \nu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3} k \delta_i, \tag{26}$$

where $k = \frac{1}{2}\overline{u_i' u_i'}$ is a turbulent kinetic. The given equation does not represent a model of turbulence but only characterizes the structure of such a model. So, the basic problem is to find the expression of the turbulent viscosity ν_t. It can be done by solving two differential equations as shown below.

k–ε Turbulence Model

The **standard k–ε model** introduces two additional transport equations:

- **k**: turbulent kinetic energy
- **ε**: turbulent dissipation rate.

The model equations are:

Turbulent kinetic energy:

$$\rho \frac{\partial k}{\partial t} + \rho \overline{u}_j \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial \overline{u}_i}{\partial x_j} - \rho \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]. \tag{27}$$

Dissipation rate:

$$\rho \frac{\partial \varepsilon}{\partial t} + \rho \overline{u}_j \frac{\partial \varepsilon}{\partial x_j} = c_{\varepsilon 1} \frac{\varepsilon}{k} \tau_{ij} \frac{\partial \overline{u}_i}{\partial x_j} - c_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]. \tag{28}$$

The **eddy viscosity** is computed as:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon}. \tag{29}$$

The model's constants are

$$c_{\varepsilon 1} = 1.44; c_{\varepsilon 2} = 1.92; C_\mu = 0.09; \sigma_k = 1.0; \sigma_\varepsilon = 1.3.$$

The k–ε model is widely used due to its **robustness** and **simplicity**, but it tends to underperform in flows with **strong separation**, **swirl**, or **adverse pressure gradients**.

SST (Shear Stress Transport) Model

The **SST model** blends the advantages of the:

- **k–ω model** near walls, and
- **k–ε model** in the free stream

This hybrid approach improves accuracy in **boundary layer separation** and **near-wall treatment**. The SST model uses the following transport equations:

- Turbulent kinetic energy (k)
- Specific dissipation rate (ω).

The basic *two-layer mode*l of Menter is presented below:

$$\rho \frac{\partial k}{\partial t} + \rho \overline{u}_j \frac{\partial k}{\partial x_j} = \widetilde{P_k} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\right], \tag{30}$$

$$\rho\frac{\partial\omega}{\partial t}+\rho\overline{u}_j\frac{\partial\omega}{\partial x_j}=\alpha\rho S-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_t)\frac{\partial\omega}{\partial x_j}\right]+$$
$$+2(1-F_1)\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}, \tag{31}$$

The turbulent viscosity μ_t and Reynolds stresses τ_{ij} are defined similar as in $k-\omega$ *model*. Denoting the generalized parameter φ_1 with a set of constants of the original model $k-\omega$ with index 1 and accordingly φ_2 with a similar set of constants of a transformed $k-\varepsilon$ *model*, the blending function can be written as

$$\varphi=F_1\varphi_1+(1-F_1)\varphi_2. \tag{32}$$

The blending function is determined as follows:

$$F_1=tanh(arg_1^4), \tag{33}$$
$$arg_1=min\left[max\left(\frac{\sqrt{k}}{\beta^*\omega y};\frac{500v}{y^2\omega}\right);\frac{4\rho\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right], \tag{34}$$
$$CD_{k\omega}=max\left\{2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j},10^{-10}\right\}. \tag{35}$$

The turbulent eddy viscosity is defined as

$$\nu_t=\frac{a_1k}{max(a_1\omega,SF_2)}, \tag{36}$$

where S is the strain rate, F_2 is second blending function defined similar to (30).

$$F_2=tanh(arg_2^2), \tag{37}$$
$$arg_2=max\left[2\sqrt{k}/0{,}09\omega y;\frac{500v}{y^2\omega}\right]. \tag{38}$$

The production limiter $\widetilde{P_k}$ is defined as

$$\widetilde{P_k}=\min(\tau_{ij}\frac{\partial\overline{u}_i}{\partial x_j},10\beta^*\rho k\omega). \tag{39}$$

The following constants are used:

1. Model constant of $k-\omega$ *model*

$$\beta^*=0.09;\alpha_1=\frac{5}{9};\beta_1=\frac{3}{40};\sigma_{k1}=0.85;\sigma_{\omega 1}=0.5;$$

2. Model constant of $k-\varepsilon$ *model*

$$\beta^*=0.09;\alpha_2=0.44;\beta_2=0.0828;\sigma_{k2}=1;\sigma_{\omega 2}=0.856;$$

Turbulent scalar/species transport

For turbulent flows, the effective diffusivity includes molecular and turbulent contributions in equation (19):

$$D_{i,e} = D_i + \frac{\nu_t}{Sc_t}, \tag{40}$$

where D_i is the molecular diffusivity, ν_t is the turbulent kinetic viscosity, and Sc_t is the turbulent Schmidt number. For a binary mixture, only one species equation is required because the second mass fraction is obtained from

$$Y_2 = 1 - Y_1\ , \tag{41}$$

This reduces the number of solved scalar equations and ensures conservation of the total mixture composition. For ideal-gas mixtures, the molecular weight mixture is calculated from the species mass fractions as

$$\frac{1}{M_{mix}} = \sum_{i=1}^{N} \frac{Y_i}{M_i}, \tag{42}$$

where M_{mix} is the molecular weight of species i. The mixture density is then calculated using the ideal-gas equation of state.

1.6 Heat transfer

Heat transfer in fluids occurs through the following mechanisms:

- **Convection** – transport of thermal energy by fluid motion,
- **Conduction** – molecular diffusion of heat through the fluids or solids,
- **Radiation** – energy transfer through electromagnetic waves.

CFxD supports **convection** and **conduction** in **single-phase** flows, including **conjugate heat transfer (CHT)**, where heat conduction in solid regions is coupled with heat transfer in adjacent fluids regions. **CFxD** also supports a **surface-to-surface** (**S2S**) radiation model for radiative heat exchange between wall surfaces.

CFxD computes temperature by solving an **energy (thermal) transport equation**. An additional turbulent heat-flux contribution is modeled through an effective thermal diffusivity. CFxD also supports **buoyancy** (natural convection): gravity introduces a density-driven body force in the momentum equation when temperature-dependent density effects are enabled. As result, CFxD can simulate **forced convection**, **conduction**, **buoyancy-driven** flow, **conjugate heat transfer** and **surface-to-surface** radiative heat exchange.

The general form of the energy (temperature) transport equation is

$$\frac{\partial \overline{\phi}}{\partial t} + \overline{u}_j \frac{\partial \overline{\phi}}{\partial x_j} = \frac{\partial}{\partial x_j}\left(D \frac{\partial \overline{\phi}}{\partial x_j} - \overline{u_j' \phi'}\right) + \bar{S}_\phi, \tag{43}$$

where $\overline{\phi}$ is the mean scalar (temperature), $\bar{S}_\phi$ represents a source term of the transported scalar and $-\overline{u_j' \phi'}$ is the turbulent scalar flux term. The turbulent scalar flux is commonly modelled using a gradient-diffusion approach [9]:

$$-\overline{u_j' \phi'} = \frac{\nu_t}{\sigma_\varphi} \frac{\partial \overline{\phi}}{\partial x_j}, \tag{44}$$

where σ_φ is the turbulent Schmidt/Prandtl number, depending on the scalar.

Surface-to-surface radiation

The **surface-to-surface (S2S)** radiation model is used to calculate radiative heat exchange between solid wall surfaces. It is suitable for cases where the participating medium is transparent, meaning radiation absorption, emission, and scattering inside the fluid volume are neglected. Therefore, radiation is exchanged only between surfaces.

In the S2S model, wall surfaces are treated as **gray diffuse surfaces**. The radiative heat exchange depends on the surface temperature, emissivity, and geometric visibility between surfaces. The emitted radiative heat flux from a surface is proportional to the fourth power of its absolute temperature:

$$q_{\text{emit}} = \varepsilon\sigma T^4, \quad (45)$$

where ε is the surface emissivity, σ is the Stefan–Boltzmann constant, and T is the absolute surface temperature. The net radiative heat exchange also depends on view factors between surfaces, which describe how much of the radiation leaving one surface reaches another surface [9].

In CFxD, S2S radiation is applied on **wall boundaries**. The user specifies the wall emissivity, while the radiation model calculates the radiative exchange between participating wall surfaces. This model is appropriate for enclosure-type problems, heated cavities, thermal boxes, and other applications where surface radiation is important and the medium between surfaces can be considered transparent.

The S2S model does not account for participating-media radiation, such as absorption or emission by gases, soot, particles, droplets, or combustion products. For such cases, more advanced radiation models would be required.

1.7 Porous resistance models

Porous resistance models are used to represent flow resistance introduced by thin or volumetric structures that are impractical to resolve explicitly with the mesh. Typical examples include screens and meshes, perforated plates, filters, heat-exchanger cores, and similar flow restrictions. Instead of resolving the detailed geometry, the porous model introduces an additional momentum loss (pressure drop) that reproduces the dominant hydraulic effect while keeping the mesh size and runtime manageable.

CFxD supports porous-media modeling in two forms: **thin porous structures** using a **porous jump** interface, and **volumetric porous regions** using a **porous zone**. The porous jump is intended for thin resistances such as screens, meshes, or perforated plates, where the pressure loss occurs primarily across a surface. The porous zone is intended for thick or distributed media (e.g., filters or porous blocks), where the momentum loss is applied throughout a volume.

The pressure loss is modeled using a loss coefficient formulation

$$\Delta p = \xi_i \frac{\rho u_i^2}{2} L, \tag{46}$$

where ξ_i is a directional loss coefficient per unit length, u_i is the velocity component in direction i, ρ is the fluid density, and L is the porous-media thickness. For a porous jump interface CFxD uses $L = 1\ m$ as a normalized thickness.

1.8 Rotation modeling

Rotating-flow problems are common in engineering applications such as fans, impellers, mixers, and rotating machinery. Directly resolving rotor motion with a moving or sliding mesh is computationally expensive and typically requires a transient approach. As an efficient alternative, CFxD supports the **Multiple Reference Frame (MRF)** method.

The MRF approach models a rotating region using a **steady rotating reference frame**. The mesh remains stationary, but the momentum equation inside the specified rotating zone is modified by adding rotation-related source terms. In the rotating frame, the apparent forces include **Coriolis** and **centrifugal** contributions, which account for the effect of rotation on the flow without physically moving the mesh. The method is well suited for quasi-steady rotor–stator problems and provides a practical approximation of the mean flow field around rotating components.

MRF is applied by defining a **rotating volumetric zone** and specifying the rotation axis and rotational speed. Outside the rotating zone, the flow is solved in a stationary reference frame.

MRF does not capture time-dependent effects such as blade passing, transient rotor–stator interaction, or rotating wakes that require a sliding mesh and transient simulation. It is intended as a steady-state approximation for engineering design and performance estimates.

1.9 Mesh considerations

At the core of CFD are the **conservation laws** of **mass**, **momentum**, and **energy**, which must be solved across a spatial domain. To accomplish this, the domain is divided into small subregions — called **cells** or **elements** — through a process known as **meshing**.

The complete set of these cells forms the **computational mesh** (also referred to as a **grid**), which provides the geometric framework for solving the discretized equations.

In CFxD, which supports three-dimensional simulations, **hexahedral cells** are primarily used. Hexahedral meshes offer good numerical stability and performance, especially for structured and semi-structured domains.

The **quality** and **resolution** of the mesh significantly affect the accuracy, stability, and computational efficiency of a CFD simulation. Key impacts include:

- **Accuracy**: A finer mesh captures geometric details and sharp gradients more accurately, especially near walls, interfaces, or shock layers.
- **Stability**: Poor mesh quality (e.g., skewed or highly stretched elements) can cause solver divergence or numerical oscillations.
- **Performance**: Smaller cells increase computational costs. A balanced mesh strategy — fine where needed, coarse elsewhere — helps reduce runtime without sacrificing accuracy.

A poorly constructed mesh can lead to **unphysical results**, **non-converging solutions**, or **excessive computation times**.

1.10 Boundary conditions

In CFD, **boundary conditions (BCs)** define the behavior of the fluid at the edges of the computational domain. They are a critical part of the simulation setup, as they influence how fluid enters, exits, or interacts with surfaces in the domain.
CFxD allows users to define boundary conditions for:

- **Velocity**
- **Pressure**
- **Temperature**
- **Turbulence variables** (e.g., k, eps, ω)

Boundary Types in CFxD

Inlet

The **Inlet** boundary condition is used to specify the incoming flow entering the computational domain. CFxD provides the following inlet options:

Velocity Inlet

The flow velocity is prescribed directly using its components (**Dirichlet condition**).

- The user specifies the velocity vector.
- Turbulence quantities (e.g., k, ω, or ε) can be defined in the turbulence subpanel.
- Temperature can be specified when heat transfer is enabled.

This option is typically used when the inflow velocity profile is known.

Mass Flow Inlet

Instead of velocity components, the user specifies the **mass flow rate**.

- CFxD internally computes the corresponding velocity based on fluid properties and inlet area.
- Temperature can be prescribed at the inlet.
- Turbulence quantities can be specified when a turbulence model is active.

This condition is appropriate when the total mass flow entering the domain is known.

Freestream Inlet

The **Freestream** condition is intended for external flow simulations (e.g., flow around bodies).

- The user specifies freestream velocity and other flow properties.
- The boundary allows a natural adjustment of flow variables consistent with far-field conditions.
- It is commonly used for aerodynamic simulations where the inlet represents undisturbed ambient flow.

For compressible subsonic flow (M < 1), freestream conditions are interpreted accordingly using the specified velocity and thermodynamic properties.

Pressure Inlet

The pressure inlet condition is used only in combination with a **Mass flow** Outlet boundary condition.

- The inlet pressure is specified by the user.

Outlet

CFxD provides several outlet boundary conditions.

Pressure Outlet

This option imposes a fixed static pressure at the outlet boundary.

- In the user interface, the specified pressure value is treated as **gauge pressure** for both incompressible and compressible simulations.
- For **compressible subsonic flow (M < 1)**, CFxD internally converts the gauge value to **absolute pressure** using the reference (ambient) pressure defined in the Operating conditions.
- Velocity is typically assigned to a **zero-gradient** condition.
- Temperature is treated according to the selected thermal model and boundary configuration.

This condition is appropriate when the downstream pressure is known (e.g., discharge into the atmosphere or into a chamber with controlled pressure).

Open Outlet

The **Open** outlet is designed to allow flow to exit the domain with minimal artificial constraint.

- Pressure is defined by the user as a gauge value (converted internally to absolute pressure in compressible simulations).
- Velocity and other transported variables use an open condition, allowing reverse flow if it occurs. The user may specify a backflow velocity value, which is applied when the flow reverses at the outlet boundary.
- For temperature, the user may specify an outlet value that is applied in case of backflow, consistent with the open boundary condition.

This condition is useful when the exact outlet pressure is not strictly prescribed and a more flexible outflow treatment is required.

Mass Flow Outlet

The **Mass Flow** Outlet is used only in combination with Pressure Inlet boundary condition. It represents the target mass flow rate at the outlet.

- The target outlet mass flow rate at the outlet is specified by the user.

Wall

Walls represent **solid boundaries** in the computational domain. A typical wall condition includes:

- **No-slip velocity** condition: fluid velocity at the wall is set to zero
- **Thermal conditions**, which may include:
  - **Fixed temperature** (*Dirichlet condition*): the temperature at the wall is specified
  - **Zero-gradient** (*adiabatic or insulated*): no heat flux across the wall.

Symmetry

Symmetry conditions are used for idealized or simplified boundaries with no normal flux or no shears.

Interface

The interface is used for the porous jump model and for CHT. A porous jump represents a localized pressure drop across thin resistances such as screens, meshes, perforated plates or similar flow restrictions. For CHT, the interface defines the thermal contact between a fluid and a solid and is used to couple heat transfer across the shared internal surface.

1.11 Limitations

CFxD is designed for general-purpose CFD simulations under common engineering conditions. However, it has certain limitations that described below.

Continuum Flow Assumption

CFxD is based on **continuum mechanics**, which assumes that fluid properties vary smoothly and continuously in space. This assumption breaks down when the fluid is too rarefied (e.g., at very low pressures or in micro/nanoscale geometries). To verify that the continuum assumption is valid, the **Knudsen number** is used:

$$Kn = \frac{l}{L}, \tag{4}$$

where l is a free path (average distance a molecule travels before colliding) and L is a characteristic length scale. The flow is considered **continuum** when the *Knudsen number* is much smaller than 1: $Kn \ll 1$. In such cases, the Navier–Stokes equations used in CFxD remain valid. For larger Knudsen numbers (Kn$\gtrsim$0.1), rarefied gas effects become significant, and CFD models based on continuum mechanics may no longer provide accurate results.

Newtonian Fluids

CFxD supports only the simulation of **Newtonian fluids**, where viscosity remains constant regardless of the flow conditions. Non-Newtonian effects such as shear-thinning, viscoelasticity, or time-dependent viscosity are not currently supported.

Single-Phase Flows

Only **single-phase** simulations are supported. CFxD does not currently handle multiphase models such as:

- Gas–liquid mixtures,
- Particle-laden flows (DPM),
- Free-surface flows (VOF),
- Cavitation or phase change phenomena.

No Chemical Reactions

CFxD does **not support reacting flows**. This includes combustion, oxidation, and species transport with source terms or reaction kinetics. All scalar fields (e.g., temperature) are treated as passive or purely diffusive.

The limitations and capabilities of CFxD are summarized in Table 1.

Table 1: CFxD capabilities vs unsupported features.

Feature Category	Supported in CFxD	Not Supported
Mode	Steady, transient	-
Phases	Single-phase flow, gas mixture	Multiphase (VOF, DPM, cavitation, etc.) flows
Fluid Type	Newtonian fluids	Non-Newtonian fluids (e.g., shear-thinning)
Compressibility	Incompressible, compressible (subsonic $M<1$, supersonic $M\geq1$)	-
Flow Regimes	Laminar, turbulent (k–ε, SST)	LES, DNS
Heat Transfer	Conduction, natural/forced convection, CHT, surface-to-surface	Participating-media radiation, gas radiation
Chemical Reactions	-	Combustion, reacting species
Acoustic	-	Empirical, integral, CAA
Porosity	Porous jump/Porous zone	-
Rotation	Rotating domain	-
Geometry Dimensions	3D	2D
Meshing	Hexahedral/hex-dominant	Tetrahedral, polyhedral
Solver Interface	OpenFOAM integration	Custom solvers, non-OpenFOAM-based solvers
Post-Processing	3D visualization, probes, charts, contours, streamlines	Advanced visualization (e.g., particle tracks, FFTs)
Material Models	Constant properties, ideal gases	Real gases, temperature-dependent properties

Chapter II. Getting started

2.1 Installation

CFxD is distributed as a standalone installer for supported operating systems. The **demo version** can be downloaded from the official CFxD website: https://cfxd.com/download/CFxD-win.exe.

Windows Installation

1. Download the installer file CFxD-win.exe from the link above.
2. Double-click the installer to start the installation process.
3. Follow the on-screen instructions of the installation wizard:
    - Choose the installation directory (recommended to use a default installation folder: C:\Program Files\CFxD)
    - Select the components (CFxD Application: core program, CFxD Runtime: required libraries, such as OCCT, OpenFOAM, VTK, Microsoft MPI: required for a multi-core simulation)
    - Accept the license agreement
4. Click **Install** and wait until the installation process is complete. If prompted by Windows security, confirm that you want to allow the application to make changes to your system.
5. Once finished, CFxD can be launched from:
    - the **Start Menu**, or
    - the **desktop shortcut** (if selected during installation).

First launch

On the first launch:

- CFxD initializes its internal folders and settings.
- The software starts in **Demo mode**, which includes a mesh-size limitation of 250000 cells and single CPU core usage.
- An active internet connection is required to check for updates and license status.

To remove the demo limitations, a valid license must be activated through the application interface.

System Requirements (Minimum)

- **Operating System**: Windows 10/11 (64-bit)
- **Processor**: Multi-core CPUs recommended
- **Memory**: 8 GB RAM (16 GB or more recommended for large meshes)
- **Disk Space**: At least 2 GB free space
- **Graphics**: OpenGL-compatible GPU

Actual hardware requirements depend on geometry complexity and mesh density.

Troubleshooting

- If the installer does not start, ensure you have administrator privileges.
- Temporarily disable antivirus software if installation is blocked.
- Make sure all required system updates for Windows are installed.

2.2 Graphical User Interface (GUI)

CFxD includes all essential elements for flow simulation in engineering applications, such as geometry creation and import/export, mesh generation, boundary condition setup, solution computation, and visualization. As an example, the workspace for geometry creation is shown in Figure 4.

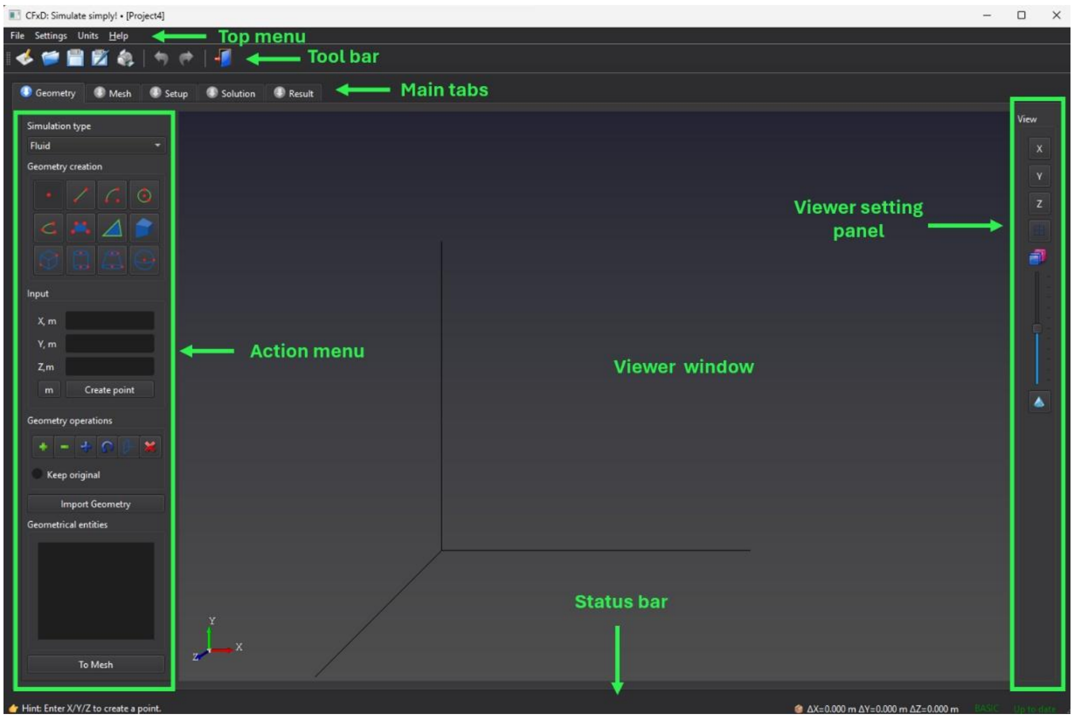


Figure 4. Working window for geometry creation.

The **Top menu** provides access to common operations, including file management, units change, report printing, help, and general program information.

The **Toolbar**, located below the top menu, contains icons that replicate most of the operations available in the top menu.

Below the toolbar are the **Main tabs** (Figure 5) required for flow simulation: **Geometry, Mesh, Setup, Solution, and Result**. The icon indicates completed tabs. The icon indicates incomplete tabs.

Figure 5. Main tabs.

Each tab is divided into two main sections: a vertical **Action menu** on the left and **Viewer window** for visualization on the right (Figure 4). The tabs with **Viewer window** additionally have the **View setting panel** on the right side.

The **Status bar** is located at the bottom of the window and displays hints, overall geometry dimensions, license type and update information button. An internet connection is required to check for updates. If an update is available, it can be installed using this button.

The general workflow starts with the vertical action menu, following a top-to-bottom sequence, and then progresses horizontally to the next tab. A detailed description of each tab and buttons is provided in the following chapters.

2.3 File structure

The project folder structure is presented in Figure 6. The projects are saved by default in **Documents** folder. The folder structure represents the workflow described previously. The geometrical entities are saved in the **Geometry** folder, mesh settings and mesh in the **Mesh** folder, the settings and boundary conditions in the **Setup** folder, solver files in the **Solution** folder and output files and report in the **Result** folder. The general data is saved in main file with .cfxd extension.

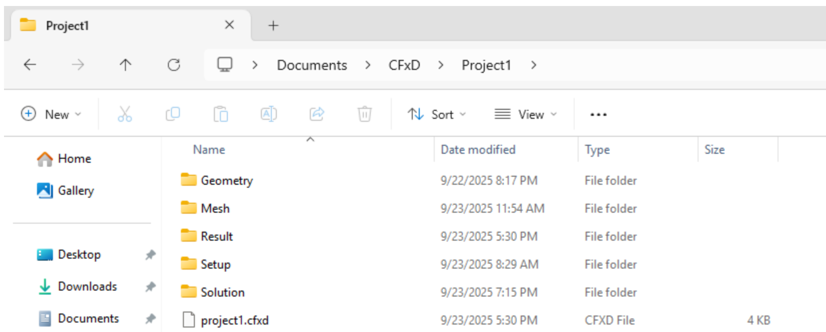


Figure 6. Folder structure.

The file structure is described in Table 2. Project file and main folder names are automatically generated with numbers: project1.cfxd in the folder Project1, project2.cfxd in the folder Project2, etc. The project file is written in JSON format and includes all information about files and user inputs.

Table 2: File structure.

Folder	Sub folder	Files (description)
		projectX.cfxd – main file
Geometry/		*.brep - native OCCT file *.stl - exported geometry
Mesh/	0/	default empty folder
	constant/	extendedFeatures – surface feature data edgeMesh – edges mesh data polyMesh – mesh file triSurface - surface geometry files
	system/	blockMeshDict – structured mesh control file snappyHexMeshDict – unstructured hexa-mesh control file surfaceFeatureDict – sharp edges detection control file
Setup/	0/	U – initial and boundary values of velocity T – initial and boundary values of temperature p/p_rgh – initial and boundary values of pressure k – initial and boundary values of turbulent kinetic energy omega – initial and boundary values of turbulence dissipation rate nut – turbulent kinematic viscosity, derived or explicitly initialized alphat – turbulent thermal diffusivity, derived or explicitly initialized
	constant/	transportProperties - transport properties turbulenceProperties - turbulence properties
	system/	controlDict – simulation control file decomposeParDict – multicore computing control file fvSchemes – discretization scheme settings file fvSolution – solution numerical settings file
Solution/	0/ X/ constant/ system/ postProcessing/	U, T, p, k, epsilon, omega, nut, etc. output statistics files files for post-processing files
Result/		images, export data report.pdf – report file with simulation results

2.4 History log

The history log window (Figure 7) can be opened from the top menu by choosing Help → Log Window. The history log window records important actions made by the user. The log is also useful for troubleshooting and debugging.

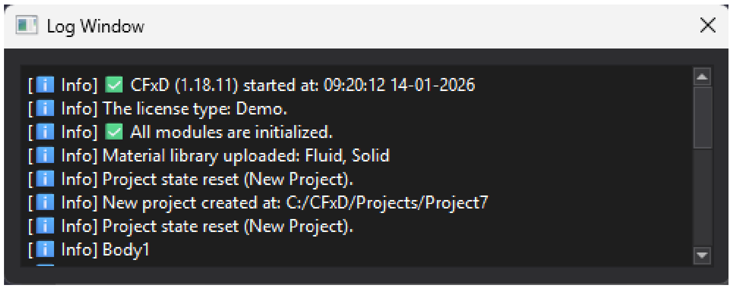


Figure 7. History log.

2.5 Interactive help

The interactive help can be opened from the top menu by choosing **Help → Manual**. The help window provides a structured description of how to use the program. Additionally, you can open context-sensitive help for the related topic by pointing the mouse cursor at an element and pressing **F1** (Figure 8).

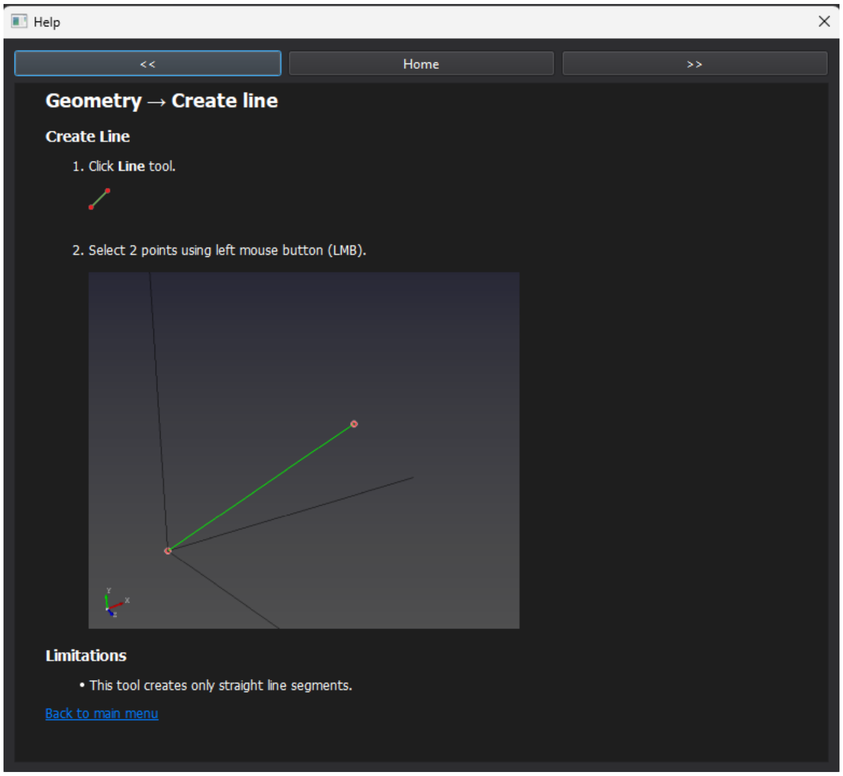


Figure 8. Interactive help opened with F1.

2.6 Licenses

The license structure is summarized in Table 3.

Table 3: License structure.

Licence type	Demo	Basic	Pro
Maximum number of cells	250,000	10 million	100 million
Maximum CPU cores	1	10	100
Advanced physics	No	No	Yes

The advanced physics includes a conjugate heat transfer and rotating flow modeling (MRF). CFxD can be installed and used in Demo mode without purchasing a license. Basic and Pro licenses can be purchased and activated from within the application.

2.7 Typical workflows

A typical CFD simulation in CFxD consists of five steps: Geometry, Mesh, Setup, Solution and Result. The following sections summarize common workflows for different simulation types.

General flow

A general flow case is a **single-phase** simulation without heat transfer and without additional models such as porous resistance or rotating flow. The main steps are summarized in Table 4.

Table 4: General flow.

Geometry	• Create or import geometry in CFxD. • Ensure the flow domain is a single closed body. • Name boundary faces (e.g., inlet, outlet, wall, symmetry).
Mesh	• Generate the base mesh. • Apply local surface and/or volume refinement if required.
Setup	• Assign a material to the fluid domain. • Define boundary conditions and operating conditions.
Solution	• Run the solver and monitor residuals and key quantities.
Result	• Visualize results (contours, vectors, plots) and generate a report if needed.

Examples of general flows can be found in Tutorial 1 (incompressible internal flow), Tutorial 2 (incompressible external flow) and Tutorial 5 (low Mach compressible flow).

General flow with heat transfer

The general flow with heat transfer includes cases with wall heat transfer (fixed temperature), thermal mixing (the same fluid at different inlet temperatures) and natural convection (buoyancy driven flow). The additional setup steps are summarized in Table 5.

Table 5: General flow with heat transfer.

Setup	• Assign a material with thermal properties to the fluid domain. • Enable the **Heat Model**. • Define boundary conditions with thermal inputs (e.g., inlet temperature, wall temperature/heat flux) and set operating conditions (reference pressure). • For natural convection, enable **gravity/buoyancy** in **Operating conditions**.

Examples of general flows can be found in Tutorial 3 (flow with heat transfer) and Tutorial 4 (natural convection).

Flow with internal wall (baffle)

This workflow applies to thin internal features (e.g., plates, screens, or baffles) that influence the flow but do not require modeling heat conduction within a solid region. The feature is represented as an **internal wall surface** inside the fluid domain. Thermal boundary conditions can be applied on this surface (e.g., adiabatic or fixed temperature) to model heat transfer between the fluid and the wall. The same internal surface can also be used to define a **porous jump**. The specific steps are summarized in Table 6.

Table 6: Flow with internal wall (baffle).

Geometry	• Create an internal surface. • Optionally, assign a name to the surface (interface).
Mesh	• Apply Surface Refinement around the internal surface if required.
Setup	• Assign the **Interface** as a **Wall** boundary, or as a **Porous** to model porous jump.

Flow with conjugate heat transfer (CHT)

This workflow includes thin solid inserts inside a fluid domain (e.g., fins, thin plates, heat spreaders) where heat conduction through the solid is important. The model is configured as **conjugate heat transfer (CHT)** with separate **fluid** and **solid** domains coupled through an internal interface. This allows the solid temperature field and the heat flux through the solid thickness to be resolved. The specific steps are summarized in Table 7.

Table 7: Flow with conjugate heat transfer (CHT).

Geometry	• Select **Conjugated heat transfer** in **Simulation type**. • Create or import two bodies (fluid and solid). • Optionally, assign a name to the shared surface (interface).
Mesh	• Apply Surface Refinement around the shared surface if required.
Setup	• Assign a material to the fluid domain. • Assign a material to the solid domain. • Define the interface as Perfect contact or Resistance.

Flow with porous media

CFxD models porous resistance using either a **porous jump** (thin resistance across an internal interface) or a **porous zone** (volumetric resistance applied within a defined region). In both cases, resistance is specified using directional loss coefficients (X/Y/Z). The porous jump workflow is described in **Flow with internal wall (baffle)**. The specific steps for porous zone are summarized in Table 8.

Table 8: Flow with porous media.

Mesh	• Create a porous zone in **Refinement | Zones** dialog.
Setup	• Set the directional loss coefficients in the **Porous zone** dialog opened from **Domains tree** under **Porous**.

Flow with rotation

CFxD models rotating flow using **MRF**, where a rotating zone is defined inside the domain and solved in a rotating reference frame while the mesh remains stationary. The specific steps for rotation modeling are summarized in Table 9.

Table 9: Flow with rotating frame.

Mesh	• Create a **Rotating zone (Cylinder)** in **Refinement | Zones** dialog.
Setup	• Set the rotation axis and rotational speed in the **Rotating zone** dialog opened from **Domains tree** under **Rotation**.

Flow with gas mixture

This workflow applies to mixing of two gases in a single fluid domain, with or without temperature differences between the inlet streams. In CFxD, gas-mixture cases are modeled as compressible species-transport simulations, where the composition is defined at the inlets using mixture fractions. The specific steps are summarized in Table 10.

Table 10: Flow with gas mixture.

Setup	• Enable **Gas mixture**. • Set two gases in the **Domain properties** dialog opened from the **Domains tree** under **Fluid**. • Define mixture fractions of the two gases in the **Inlet** dialog opened from the **Boundaries tree** under **Inlet**.

Chapter III. Geometry

The **Geometry** tab includes a vertical **Action menu** with options for basic geometry creation, coordinate and movement input, geometry operations, and a **Geometry entities** tree displaying the created elements. Geometrical elements are visualized in the **Viewer window** (Figure 9). The **Viewer settings** panel is located on the right side of the **Viewer window**.

In addition to the geometry tools, the **Action menu** includes a **Simulation Type** selector at the top. It provides currently two options - **Fluid** and **Conjugate Heat Transfer** - which streamlines the subsequent workflow.

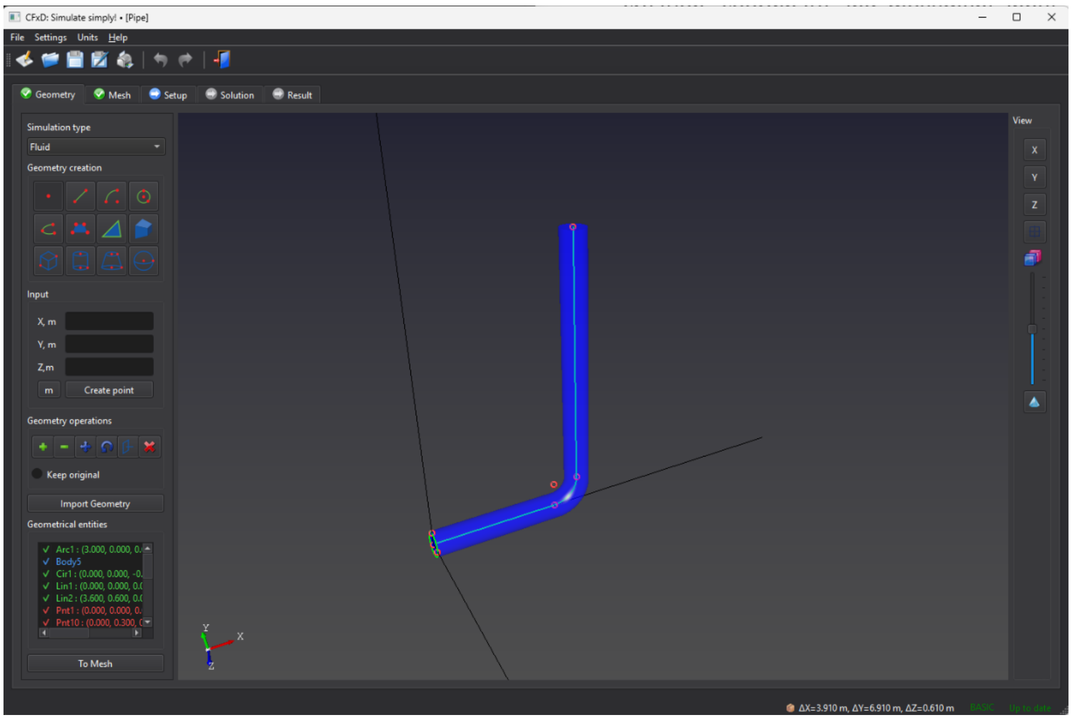


Figure 9. Geometry Tab.

The origin of the coordinate system (0, 0, 0) is located at the center of the graphical window. The axes are visually represented by the black axis in the graphical window with designated X, Y, Z coordinates shown at the bottom-left corner.

The status bar in the bottom provides helpful hints for geometry creation at the left, while overall geometry dimensions are displayed on the right.

3.1 Mouse manipulation

The software supports a standard three-button mouse with a scroll-wheel (Figure 10) for geometry creation and manipulation.

Figure 10. Three-button mouse.

The following notation will be used: **LMB** - the left mouse button, **MMB** - the middle mouse button, **RMB** - the right mouse button, **SWM** a scrolling wheel of a mouse. The mouse commands used in CFxD are introduced in Table 11.

Table 11: Commands used for the mouse buttons in CFxD.

Buttons	Action
LMB	select elements
MMB	rotate
SWM	in/out zoom
RMB	select for naming and pan

3.2 Geometry Creation

The basic geometries can be created using the following basic geometric primitives: point, line, arc, circle, curve, surface, cube, cylinder, cone and sphere (Figure 11).

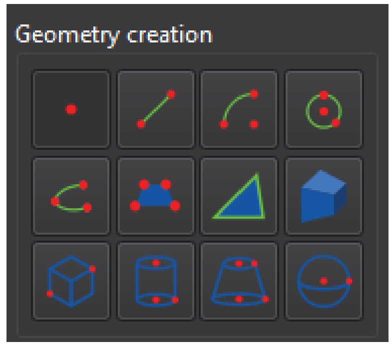
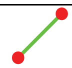
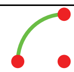
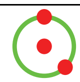
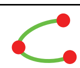
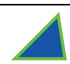
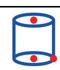
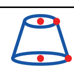
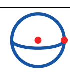


Figure 11. The geometry creation buttons.

A summary of all geometric primitives, their creation methods, and limitations is provided in Table 12.

Table 12: Geometrical primitives with properties.

Geometry	Icon	Creation	Limitation
Point		spatial coordinates: x, y, z	---
Line		by two (2) points selection	distinct points
Arc		by three (3) points selection 1: centre, 2: on radius, 3: on radius	two points on same radius, not collinear
Circle		by three (3) points selection 1: centre, 2: on radius, 3: on radius	two points on same radius, not collinear
Curve		by three (3) points selection	distinct points
Surface		by four (4) points selection	coplanar, ordered not collinear
		by three (3) curve selection	closed loop
Body		by six (6) surfaces selection	closed volume
Box		by two (2) points selection	opposite corners, axis-aligned
Cylinder		by three (3) points selection	radius ⊥ height
Cone		by four (4) points selection	both radii ⊥ height
Sphere		by two (2) points selection	distinct points

The following colours are used for geometry entities: red circles for points, green for edges, blue for surfaces and gray for solids, light blue for geometry highlighted by hovering mouse pointer, white for selected geometry.

Point creation

The **X**, **Y**, and **Z** fields in the **Input panel** define a point in 3D space (Figure 12). Units can be switched using the **Units** button (e.g., **mm** for SI or **in** for Imperial). Click **"Create Point"** button to add the point to the graphic window. Points are displayed as small **red** markers; a **selected** point is highlighted in **light blue**. Coordinate-based point entry is available also while creating other geometry primitives.

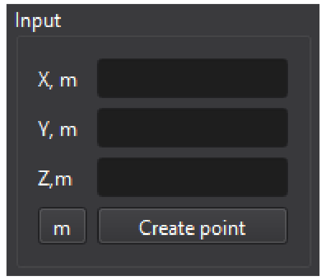


Figure 12. Input panel.

The point can be selected using LMB and deselected by a mouse click anywhere on the screen outside of the point. An example of some created points is shown in Figure 13.

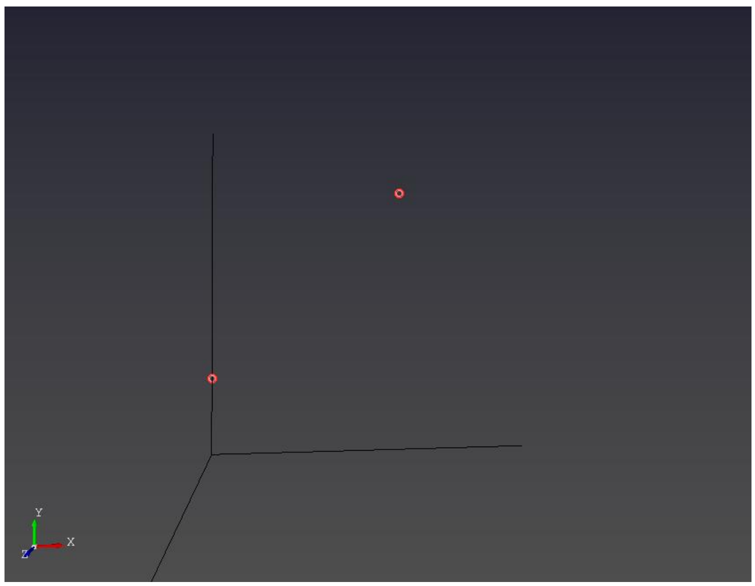


Figure 13. Created points.

Line creation

A **line**  can be created by selecting two points using the **LMB**. An example showing a created line is presented in Figure 14. Like points, a line can be selected using **LMB** and deselected by mouse click anywhere on the screen outside the selected element.

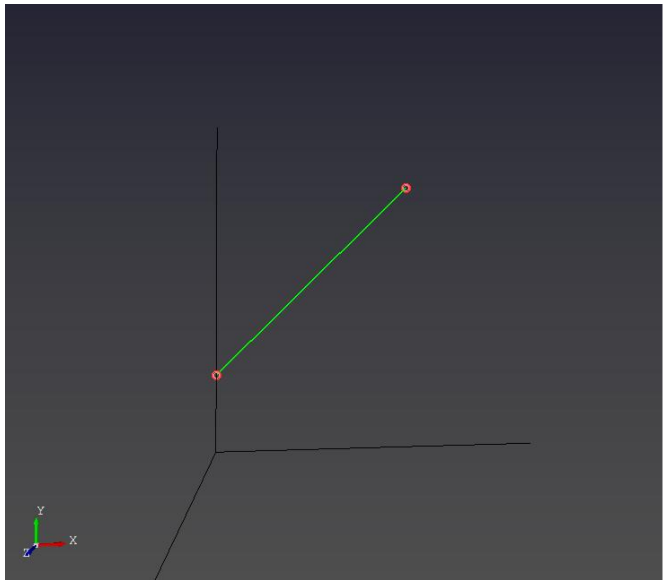


Figure 14. Line created with two points selection.

Arc creation

An **arc** can be created using three points**,** which must be selected in the following order**:** centre point (1), first point on the radius (2), and second point on the radius (3). The selected points must not be collinear, and two radius points must be on the same circle. An example of a created arc is shown in Figure 15.

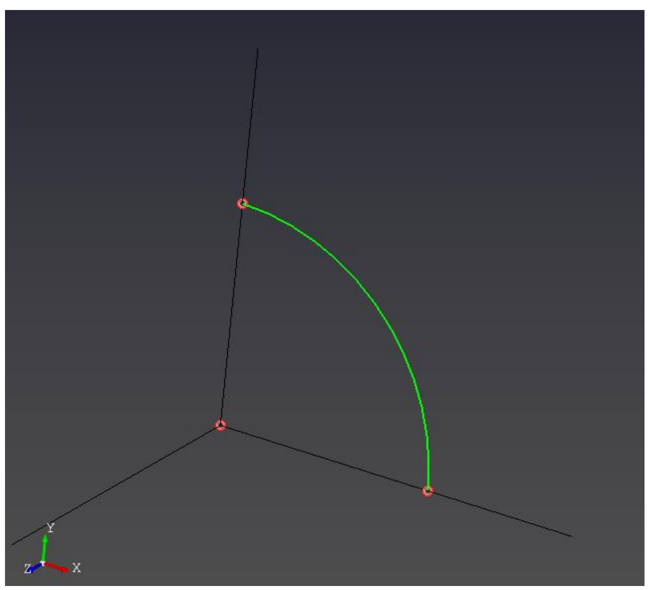


Figure 15. Arc created with three points selection.

Circle creation

Similar to an arc, a **circle** can be created using three points. The points must be selected in the following order: centre point (1), first point on the radius (2), and second point on the radius (3). The selected points must not be collinear, and the two radius points must be on the same circle. An example of a circle is shown in Figure 16.

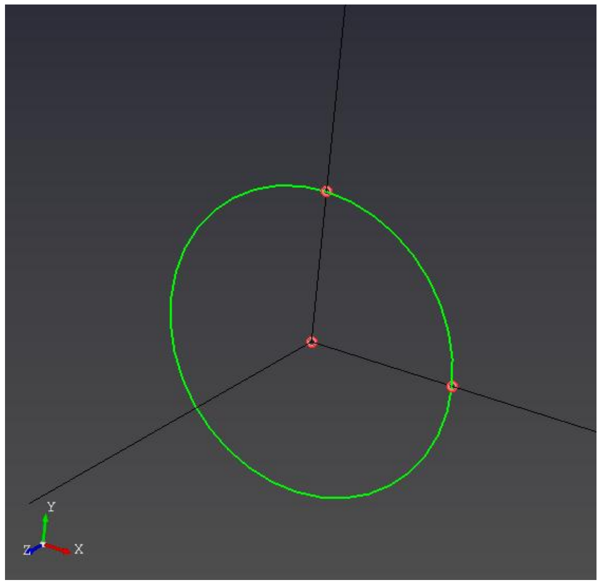


Figure 16. Circle created with three points selection.

Curve creation

A **curve** creation is based on three points (Figure 17). B-spline is used to get a curve going through selected three points.

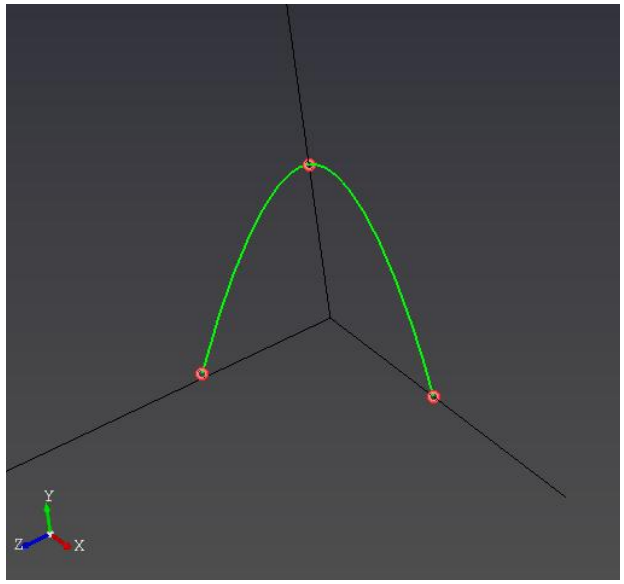


Figure 17. Curve created with three points selection.

Surface creation

Surface creation supports two methods: using four (4) points, or selecting three (3) lines or curves.

For the first method, exactly four distinct points must be selected, and they must define a closed boundary. Select the points sequentially—clockwise or counterclockwise—so that four connecting edges are implicitly formed. A surface cannot be created by selecting two diagonal pairs of points, because this does not define a valid boundary; the four points must follow one another along the perimeter of the surface to ensure proper surface generation. When creating a surface from points, all points must lie on the same plane. If the four points are not coplanar, surface creation will fail.

When creating a surface from lines or curves, the boundary must be defined by exactly three (3) lines or curves. A circle is a special case: a surface can be created from a single circular curve because it already forms a closed loop. In all cases, the selected lines and curves must form a closed loop and lie on the same plane. The examples of created surfaces are shown in Figure 18 (left: created by points, right: created by lines).

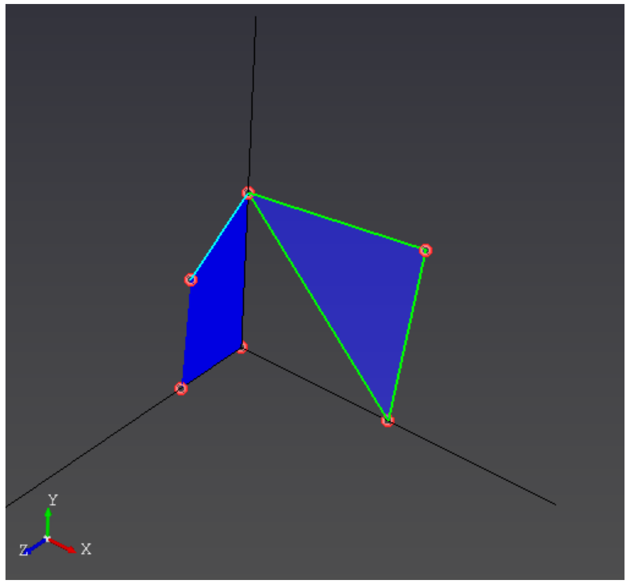


Figure 18. Created surfaces by points (left) and by lines (right).

Box

A **Box** is created from two opposite corners. Faces are aligned with the global axes (parallel to XY, XZ, YZ planes). If the two points have the same X, Y, or Z coordinate (zero size in one direction), an error is shown. An example of a created box is shown in Figure 19.

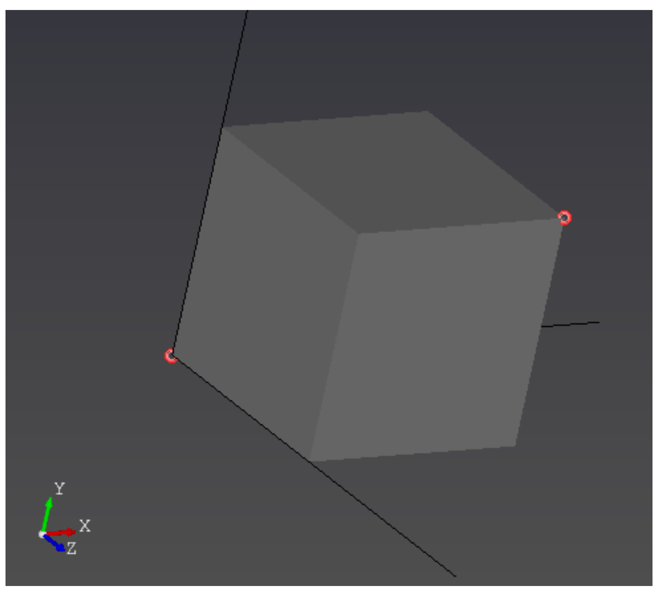


Figure 19. Box created with two points selection.

Cylinder

A **cylinder** can be created by selecting three points: (1) base center, (2) a point on the radius, and (3) a point defining the height direction and length. The radius direction (1→2) must be perpendicular to the height direction (1→3). An example of cylinder is shown in Figure 20.

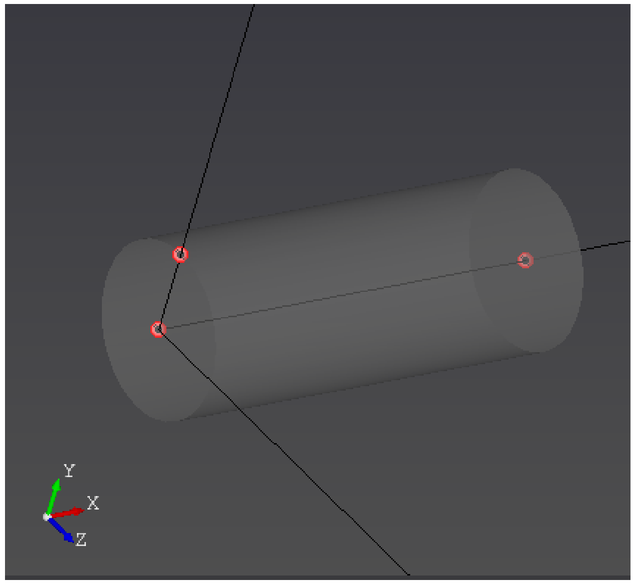


Figure 20. Cylinder created with three points selection.

Cone

A cone (frustum) can be created by selecting four points: (1) bottom center, (2) bottom radius point, (3) top center, (4) top radius point. The height direction is (1→3). Both radius directions (1→2 and 3→4) must be perpendicular to the height direction. An example of cone is shown in Figure 21.

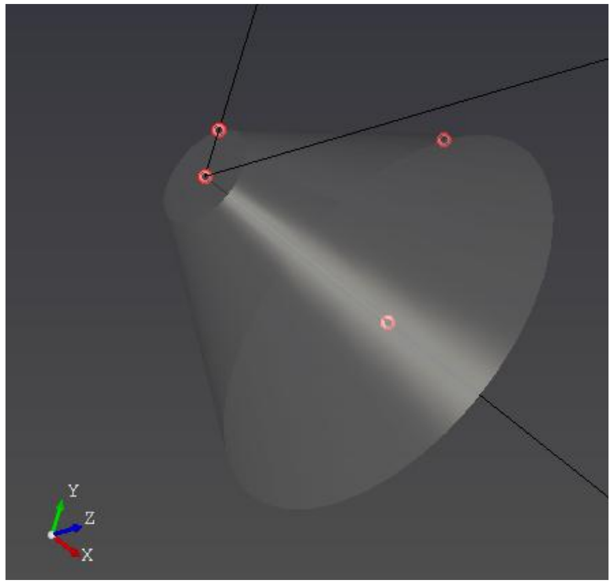


Figure 21. Cone created with four points selection.

Sphere

A sphere can be created by two points selections: central point and point on radius. An example of sphere is shown in Figure 22. Sphere is generated with upper and bottom faces.

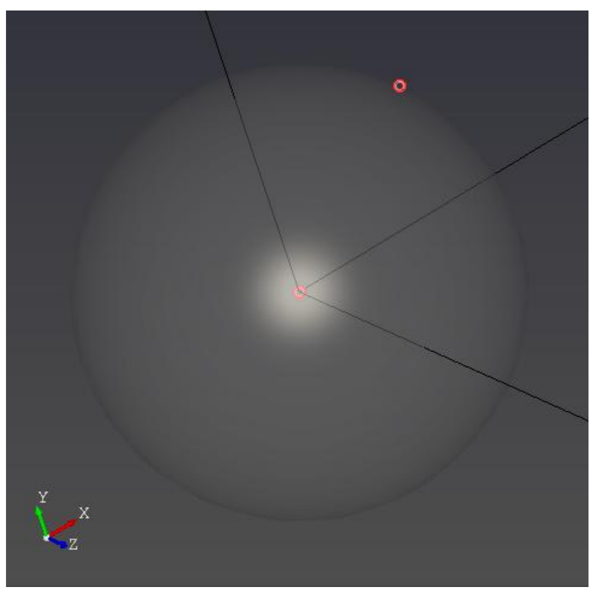


Figure 22. Sphere created with two points selection.

Body

Body can be created by selecting six (6) faces that form the closed body. The example of body is shown in Figure 23. It has been created by following operations: 8 points → 6 surfaces → body.

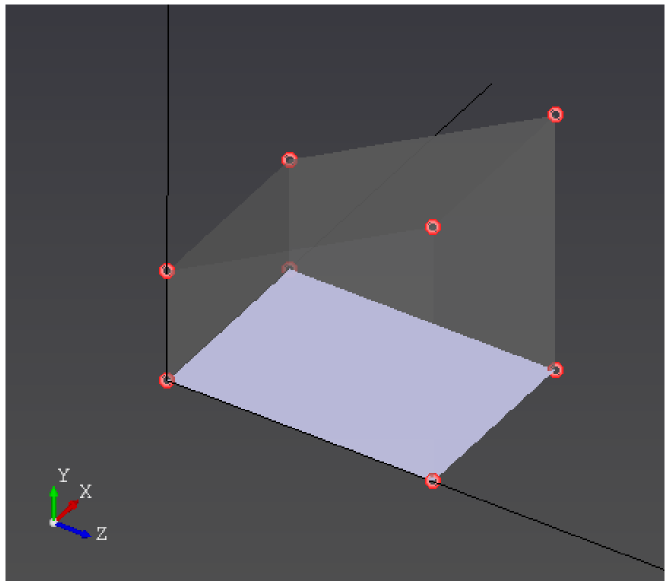


Figure 23. Body created with six surface selection.

3.3 Geometry operations

The **Geometry Operations** panel provides tools for

- Boolean operations (**Combine** and **Subtract**),
- Geometry relocation (**Translate** and **Rotate**),
- **Pull** ,
- **Delete** .

The Boolean operations require the selection of two bodies. The **Input panel** (Figure 24) is used to specify translation distances, pull distance and rotation angles. The **Pull** operation works only with planar surfaces. In addition to pulling in the normal distance to surface, it also provides an operation to pull along a selected curve.

For Boolean and relocation operations, the original geometry can be preserved by enabling the **Keep original** option. If this option is not enabled the original geometry is removed after the operation completes.

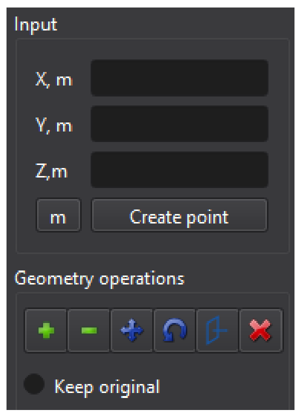


Figure 24. Geometry operations panel.

3.4 Geometry import

Geometry created in other programs can be imported into CFxD using the **Import** button located below the **Geometry Operation** panel. Currently, only the **STEP** format is supported. CFxD automatically converts the geometry dimensions from the STEP file. An example of imported geometry is shown in Figure 25.

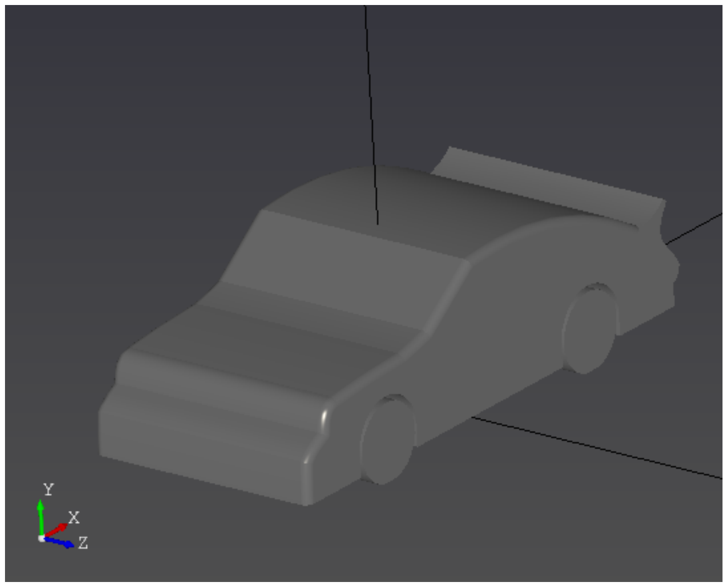


Figure 25. Imported geometry.

3.5 Geometrical entities window

The created geometric entities appear in the Geometrical entities window (Figure 26) located below Geometry operation panel. This window is updated automatically when entities are created or deleted. Pressing the box with the checkmark will hide/unhide geometry. Geometry can also appear highlighted in the graphical window by clicking the desired geometry in the geometrical entities window.

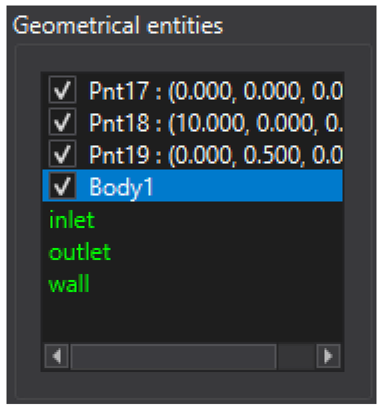


Figure 26. Geometrical entities window.

Deleting geometry or name

Geometry or name may be deleted from the geometrical entities window by clicking on desired entity with RMB and selecting **Delete** (Figure 27).

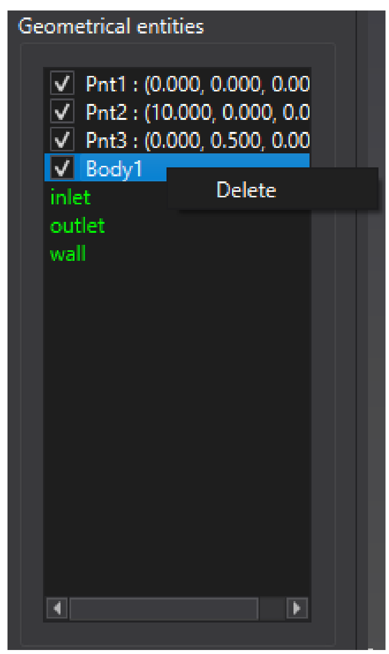


Figure 27. Deleting Body1 in geometrical entities window.

3.6 View settings

Basic **view settings**, such as **views from X, Y, Z** directions, **fit all**, **wireframe/solid**, **and transparency control**, are located to the right of the graphic window (Figure 28).

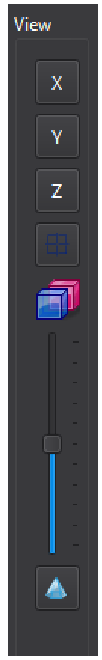


Figure 28. View settings.

3.7 Naming

The naming operation can be activated in the Graphic window with RMB. RMB is used to select a face to assign a name in the appeared dialog (Figure 29). By default, the names beginning with “inl” will be recognized as inlets during the Setup step. Similarly, “out” is recognized as an outlet, “wall” as a wall, “sym” as a symmetry boundary, and “inter” as an interface. The interface should be an internal surface.

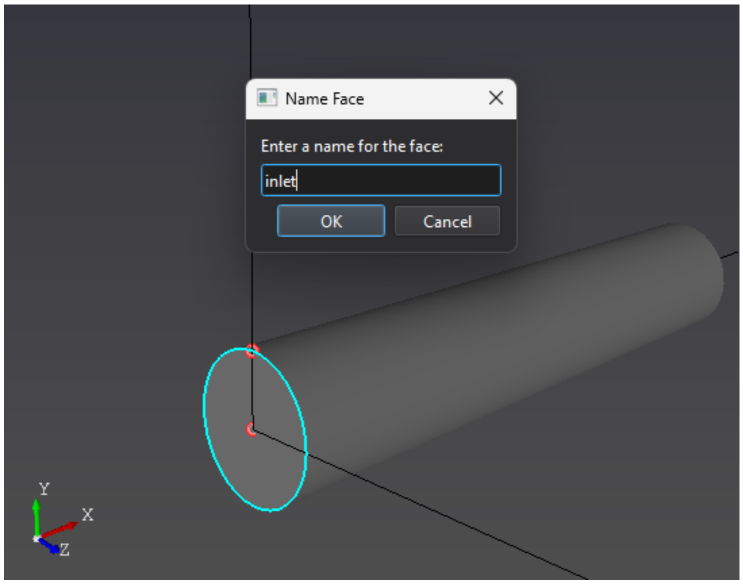


Figure 29. Face naming.

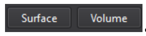
3.8 Export geometry

The last action in the **Geometry Tab** is export geometry to Mesh. The **“To Mesh”** button located at the bottom left menu will convert the geometry to the format (STL) comparable to the mesh generator. It will be saved in the Mesh folder. Only a solid element that forms a closed volume can be converted to the format for future meshing. The geometry in native format (BREP) will be saved in the folder Geometry as well. After successful export the **Geometry Tab** 

will be changed to **Mesh Tab** 

automatically.

Chapter IV. Mesh

The mesh settings, information, and graphical representation of the mesh are shown in Figure 30. Currently, CFxD supports **hexahedral-dominant** mesh. The mesh generation process includes the following settings: base **cell size**, local controls for **surface/volume refinement** and **zones**. **Surface** and **Volume refinement** dialogs can be called with RMB on **Mesh viewer** screen or from left menu 

. Initially, a recommended cell size is **automatically calculated** based on the overall dimensions of the geometry.

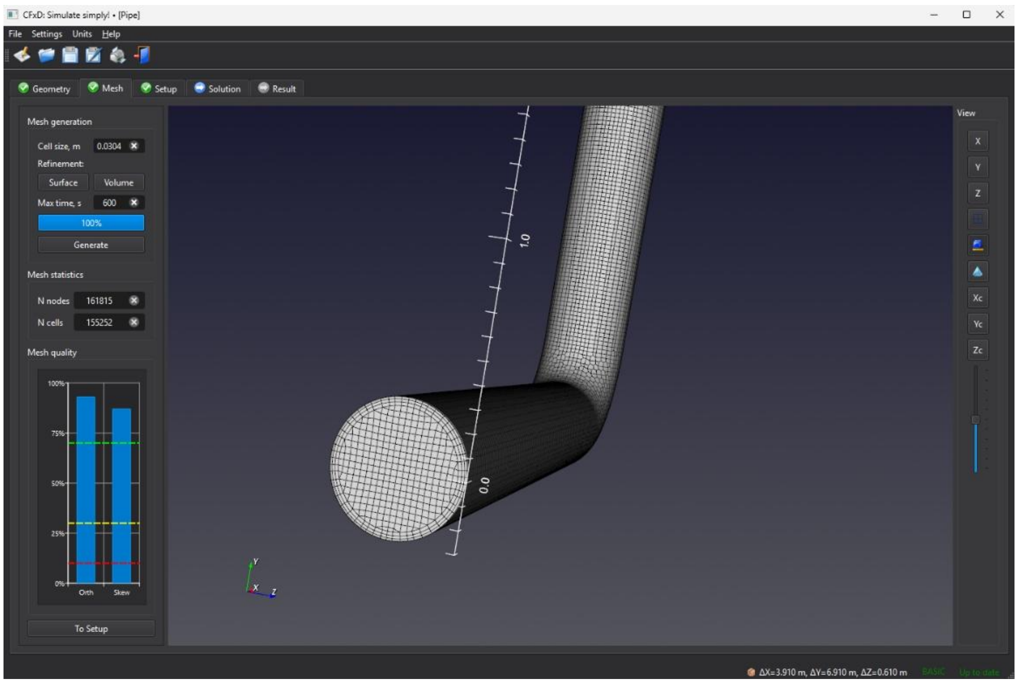


Figure 30. Mesh tab.

4.1 Refinement

Surface refinement allows the cell size to be specified on a selected surface. In addition, prism (boundary-layer) layers can be enabled to better resolve near-wall gradients. The selected surface is highlighted, and a blue cell-size cube indicating the specified local cell size is shown in the **Mesh viewer** (Figure 31). The added surface can be deleted with RMB in the table in **Surface refinement** dialog.

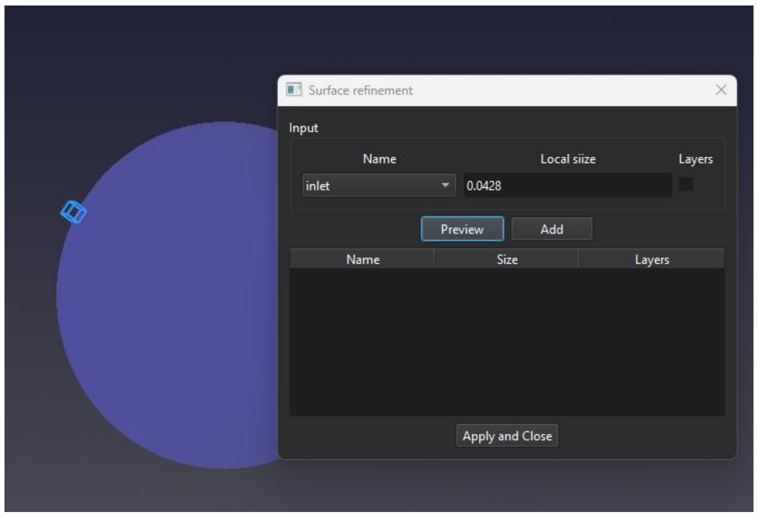


Figure 31. Surface mesh refinement dialog.

In addition to surface refinement, **volume refinement** can be used. Two predefined volume shapes are available: **box** and **cylinder**. The box is defined by two diagonal points, and a local cell size can be specified for the cells inside the box. A red box with a blue cell-size cube is shown when **Preview** is clicked (Figure 32).

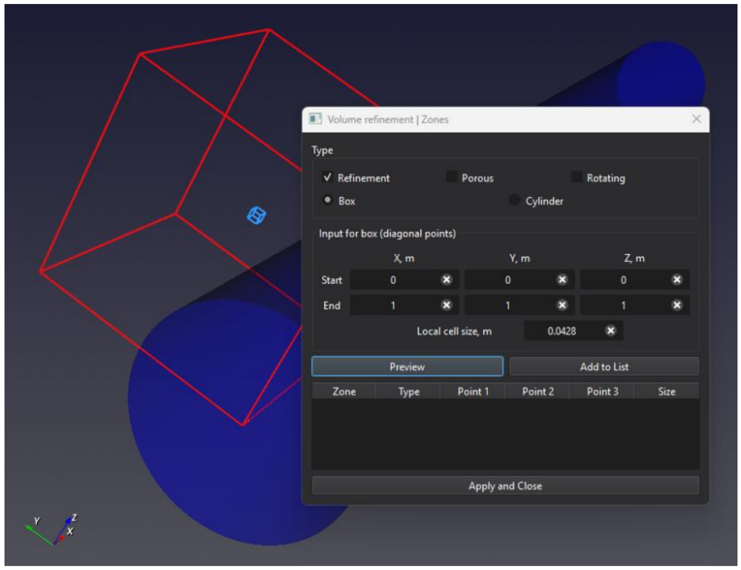


Figure 32. Volume mesh refinement dialog with a box.

Similarly, refinement inside a cylinder can be added. The cylinder is defined by three points: the center, a point defining the radius, and a point defining the height. Added volumes (box/cylinder) can be deleted with the right mouse button (RMB) in the table in the **Volume refinement** dialog. The volume refinement dialog can be used to mark the zone that can be used for porous structure (Figure 34) or rotating domain.

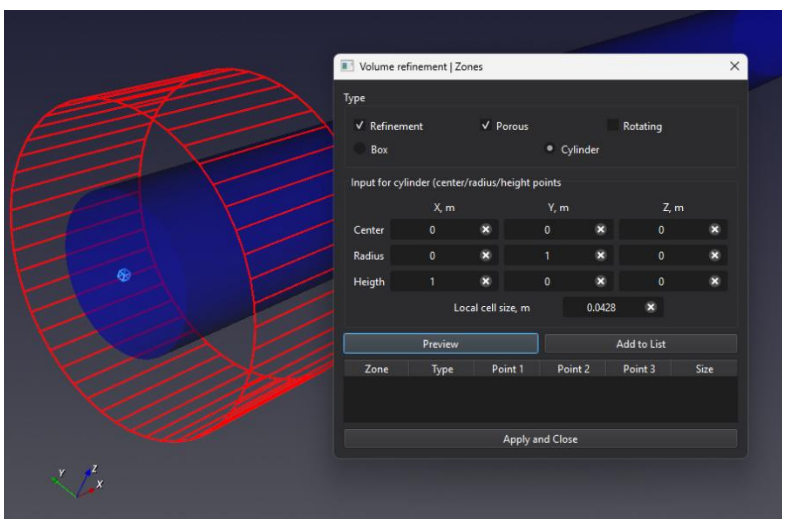


Figure 33. Set a cylinder for refinement and as a porous zone.

The example of mesh refinement with box and cylinder is shown in Figure 34.

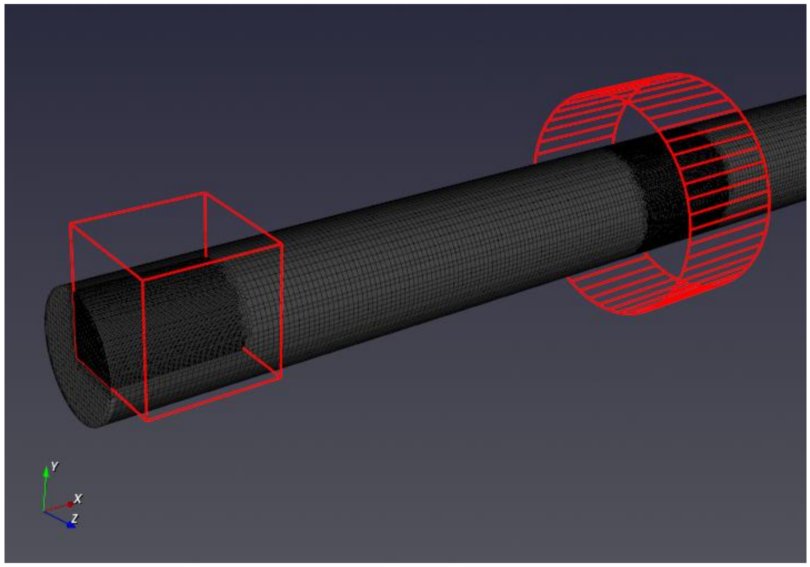


Figure 34. Example of volume refinement.

The mesh generation process can be limited by a **maximum execution time**. Mesh generation starts using the **Generate** button, and the progress of the process is displayed in a **progress bar**.

4.2 Mesh statistics, Quality and Export

The **Mesh statistics** panel is located below the **Mesh generation** panel. It provides summary information for the generated mesh, including the total number of nodes and cells (Figure 35).

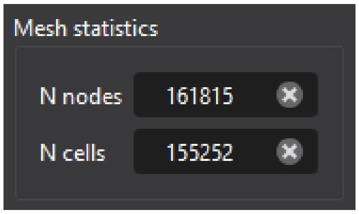


Figure 35. Mesh statistics panel.

The **Mesh Quality** panel shows a histogram of mesh quality metrics such as **orthogonality** and **skewness** (Figure 36). Three dashed lines indicate recommended quality thresholds: **good** (green), **sufficient** (yellow), and **minimal** (red). Mesh quality can also be evaluated visually in the graphics window.

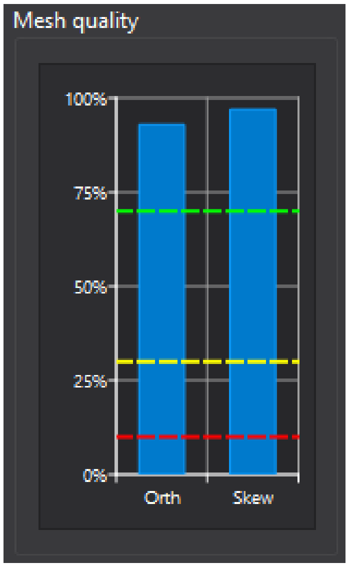


Figure 36. Mesh quality panel.

Similar to the **Geometry** tab, the **Mesh** tab includes a **View** panel with the same navigation and display options. In addition, it provides mesh-specific features, including a coordinate scale, that can be shown or hidden using the **Scale** button  in the **View** panel (Figure 37).

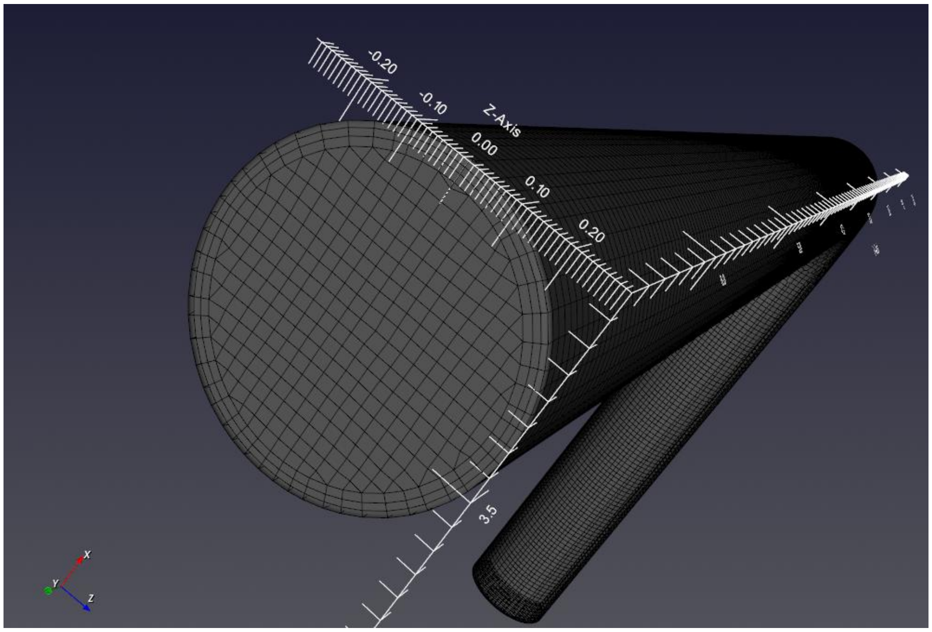


Figure 37. Coordinate scale.

For visual inspection of the internal mesh, sectional cuts can be displayed in the **Xc**, **Yc**, and **Zc** planes using the cut controls and slider (Figure 38).

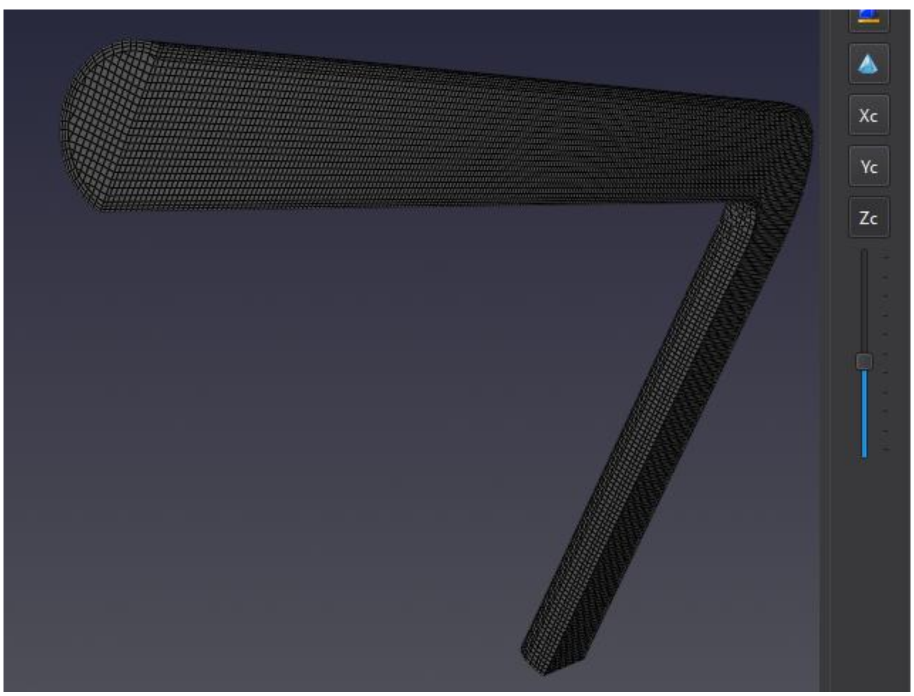


Figure 38. Cut section.

By pushing button “To Setup” the mesh will be exported to Setup step and **Mesh** Tab will be changed to **Setup** Tab.

Chapter V. Setup

The solver settings and boundary conditions are defined in the **Setup** tab (Figure 35). This tab is used to configure the physical model and numerical setup before running the simulation. It allows:

- selection of the **simulation mode** (steady or transient) and the corresponding iteration/time settings,
- selection of the **flow regime** (laminar or turbulent) and the **turbulence model** (when applicable),
- selection of the **flow type** (**incompressible** or **compressible** subsonic flow);
- enabling or disabling **heat transfer**,
- definition of **operating conditions** (e.g., reference pressure and gravity/buoyancy),
- selection of **materials** and assignment of thermophysical properties to domains,
- assignment of **boundary conditions** to named faces (inlet, outlet, wall, symmetry, interface),
- selection of the number of **CPU cores** for parallel execution.

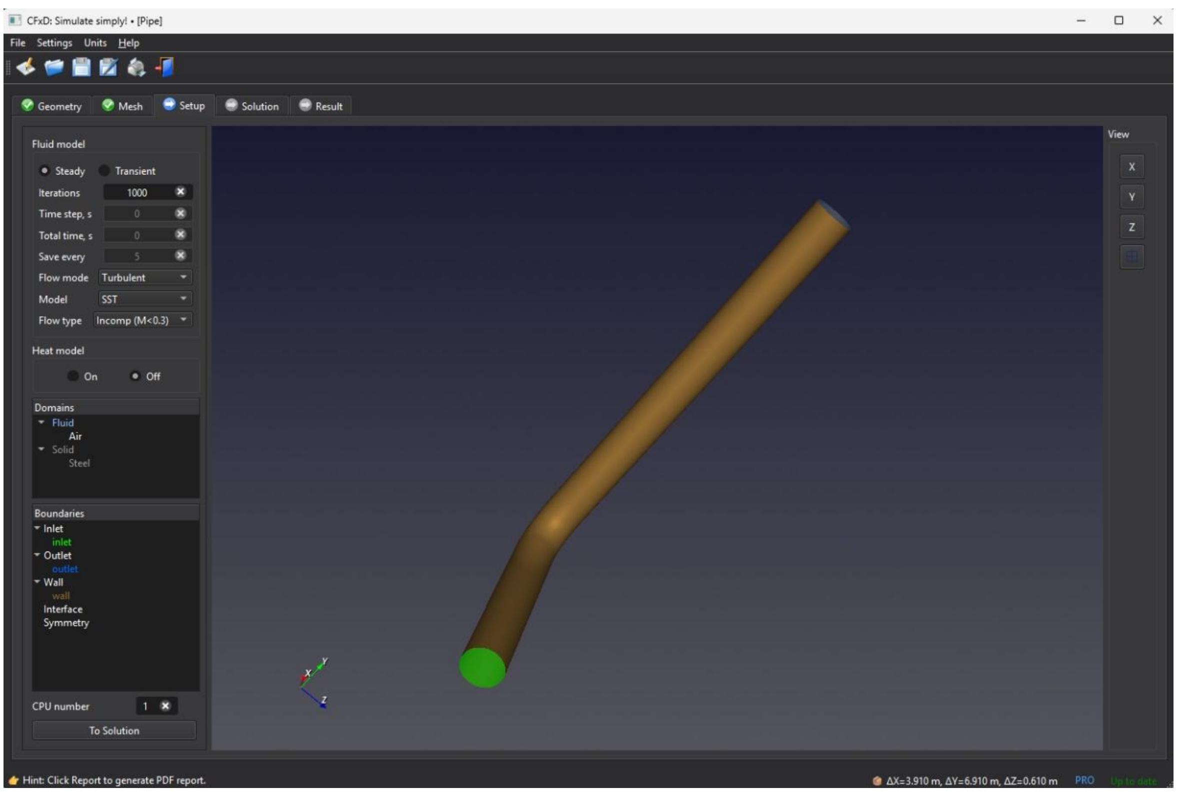


Figure 39. Setup tab.

5.1 Fluid model

The **Fluid Model** panel allows the user to:

- Select **Steady** or **Transient** mode,
- Set the total number of iterations (for steady simulations),
- Specify the time step and total number of time steps (for transient simulations, when available),
- Choose the flow regime: **Laminar** or **Turbulent**,
- Select the turbulence model (**k–ε** or **SST**),
- Select the flow type: **Incompressible** or **Compressible** (subsonic regime, M < 1).

The **Heat Model** panel allows the user to enable or disable **heat transfer modeling**.

The transient simulation inputs include **Iteration per step**, **Time step**, **Total time**, and the saving option **Save every** (Figure 40).

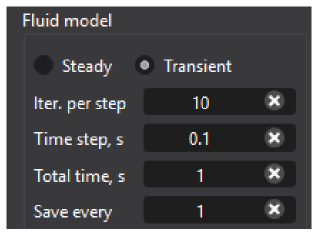


Figure 40. Transient simulation inputs.

5.2 Operating conditions and buoyancy

Operating conditions are configured via **Top Menu → Settings → Operating conditions** (Figure 41). This panel is used to define the **reference pressure** and **gravity vector**, which are required for consistent pressure specification and buoyancy-driven flow modeling.

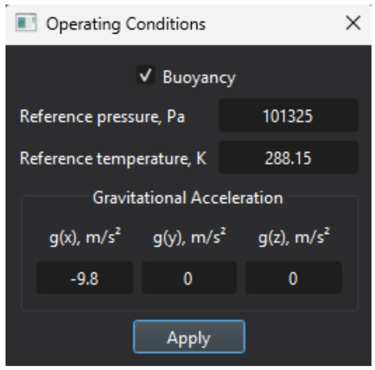


Figure 41. Operating conditions panel.

Reference pressure. The reference pressure is used as the ambient (baseline) value for cases where pressure is specified as **gauge pressure** in the boundary-condition panels. For compressible subsonic simulations, CFxD uses the reference pressure to convert gauge pressure values to absolute pressure internally.
Reference temperature. The panel also includes a **reference temperature** field. In the current version, this value is displayed for completeness and future extensions, but it is **not used** in the solver setup and does not affect the simulation results.
Buoyancy (natural convection). Natural convection can be enabled by activating the **Buoyancy** option in the same panel. When buoyancy is enabled, gravity introduces a density-driven body force in the momentum equation based on temperature-dependent density effects. Buoyancy is typically used for problems such as heated enclosures, vertical plumes, and thermally driven circulation.
Gravitational acceleration. The gravity vector components **g(x)**, **g(y)**, and **g(z)** define the direction and magnitude of gravity. For example, a vertical gravity field is commonly set by assigning -9.81 m/s^2 to the appropriate component aligned with the vertical axis of the model.

5.3 Domains

Materials can be assigned to each domain in the **Domains** tree using the **Domain properties** dialog. This dialog allows the selection of fluid or solid material properties. Several common materials are predefined in the material library (Figure 42) and new materials can be added to the library.

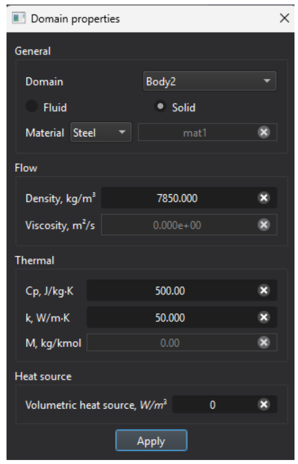


Figure 42. Material selection.

For **CHT** cases, fluid material properties should be assigned to the **fluid domain**, and solid material properties to the **solid domain**. For solid domains, a **volumetric heat source** can also be specified in the same dialog. This option applies to an uniform internal heat generation rate within the selected solid region and is commonly used to model heaters, electronic components, or other internally heated parts.

In the **porous zone** model, a new **Porous** item is added to the **Domains** tree, and all created porous zones are listed under it (Figure 43). Porous zone properties can be edited by opening the **Porous zone** dialog for the selected zone. The dialog allows the porous zone to be assigned to a specific volume region and configured by specifying the **zone thickness/length** and **directional loss coefficients** (X, Y, Z). These coefficients define the additional momentum loss applied inside the selected volume to represent flow resistance through a porous medium (e.g., filters or porous blocks). After setting the parameters, click **Apply** to store the porous-zone settings.

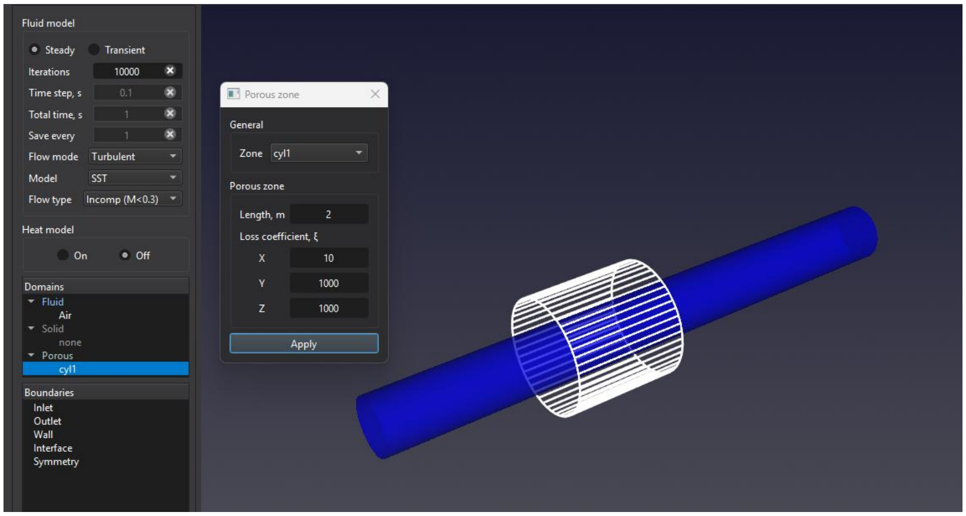


Figure 43. Porous zone setup.

5.4 Boundary conditions

The **Boundaries** tree is used to assign boundary conditions to selected faces of the geometry. It includes the general boundaries such as Inlet, Outlet, Wall and Symmetry. To visually distinguish the boundary faces in the **View** window, the following color-coding is used: inlet (green), outlet (blue), wall (gray) and symmetry (yellow). Below all boundary condition settings are described.

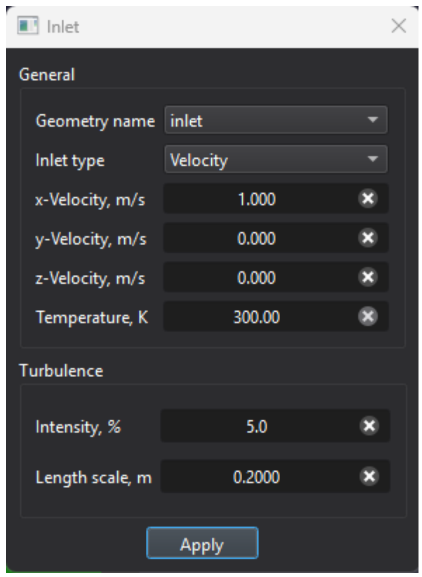


Figure 44. Inlet input panel.

The **Inlet** panel is shown in Figure 44. At the inlet, **velocity** (defined by its components), **mass flow rate**, or pressure can be specified. The inlet turbulence parameters can be configured in the **Turbulence** subpanel. In addition to the standard velocity inlet, a **Freestream** inlet option is available, which allows the user to specify the far-field velocity for external flow simulations.

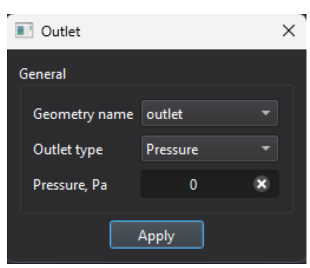


Figure 45. Outlet input panel.

The Outlet input panel is shown in Figure 45. **Pressure** is specified as gauge pressure (relative to reference pressure). A **Mass Flow Outlet** option is also available, allowing a target outlet mass flow rate to be specified. For the **Open Outlet** type, the user can specify the **backflow velocity** and **backflow temperature**, which are applied if reverse flow occurs at the boundary.

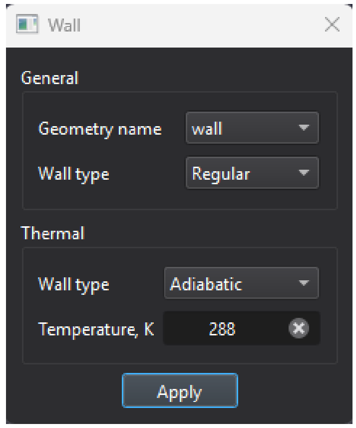


Figure 46. Wall input panel.

The **Wall** input panel (Figure 46) is used for wall setup. Current version supports **Regular** (no slip), **Frictionless** (slip) and **Rough** walls. In the **Thermal** subpanel, the wall can be set as adiabatic or with constant temperature.
The **Interface** input panel (Figure 47) is used to set an internal thin wall or porous jump by specifying the loss coefficient.

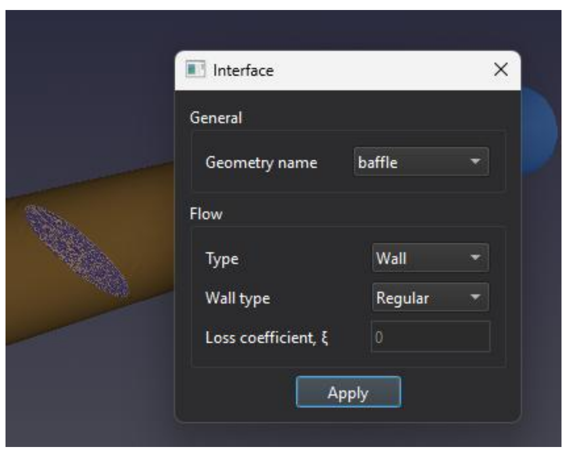


Figure 47. Interface input panel.

Boundaries can be deleted by pressing the RMB on a boundary in the "Boundaries" window and selecting delete (Figure 48).

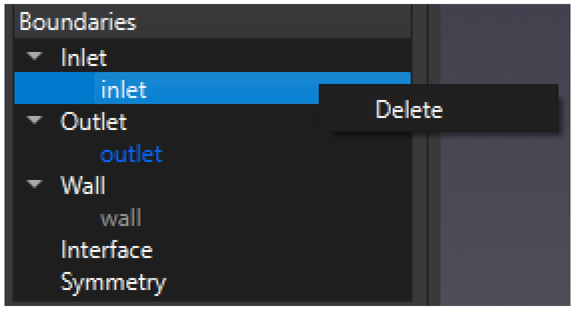


Figure 48. Deleting boundaries.

5.5 Parallel settings

At the bottom of the **Setup** tab, the number of **CPU cores** can be selected to run the simulation in parallel. Parallel execution can significantly reduce runtime for medium and large cases, depending on mesh size and solver settings.

If the requested number of cores exceeds the number of available CPU cores, CFxD automatically reduces it to the maximum supported value. The selected core count is applied during the **Solution** step when the case is decomposed and the solver is launched in parallel. After the run, CFxD reconstructs the results for post-processing.

I can For small meshes, using many cores may provide limited benefit due to parallel overhead. CFxD displays a notification if an excessively large number of CPU cores is selected for a small mesh.

Chapter VI. Solution

Solution tab allows to set and monitor residuals (convergence plot), track field values at selected points or faces, adjust a stability control, and run/pause/resume a simulation (Figure 49).

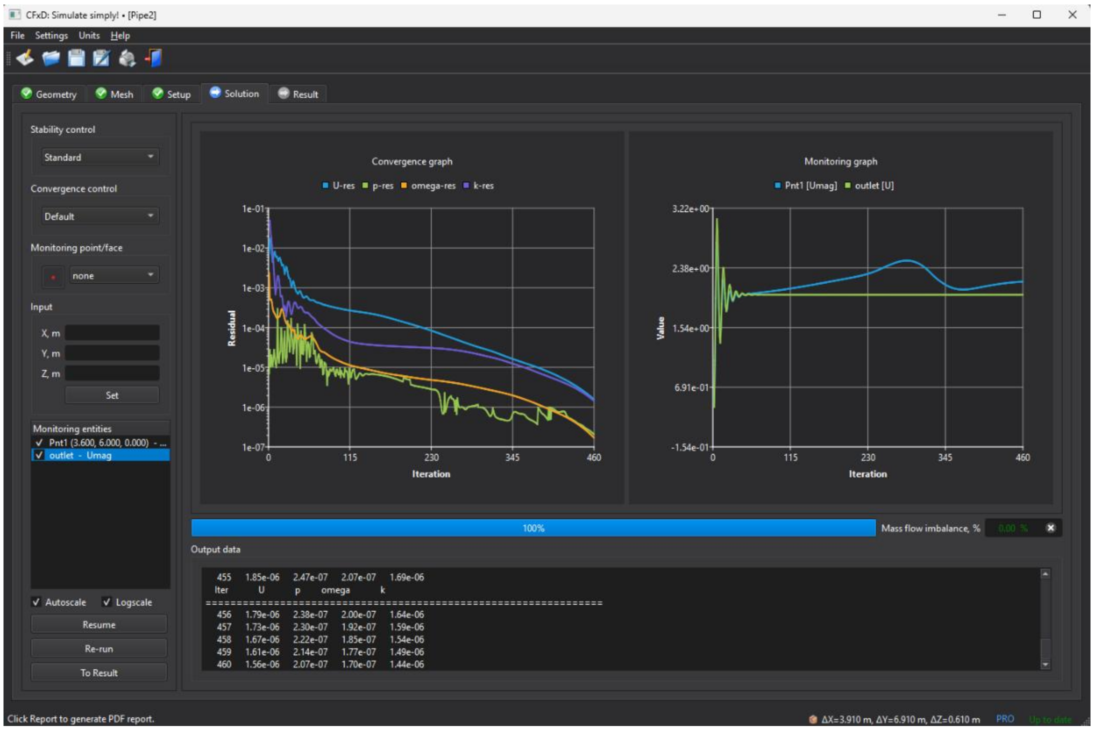


Figure 49. Solution tab.

6.1 Stability control panel

The **Stability control panel** (Figure 50) allows to adjust the relaxation factors to improve solution stability. It includes three preset modes: **Aggressive**, **Default**, and **Conservative**. **Aggressive** uses higher relaxation factors and may speed up convergence, but it can also make the solution unstable and lead to divergence. **Default** provides a balanced setting and is recommended for most cases. **Conservative** applies stronger damping (lower relaxation factors) to improve stability. It may converge more slowly but is useful for difficult meshes or highly nonlinear flows.

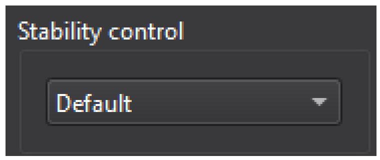


Figure 50. Stability control panel.

6.2 Monitoring point/face

The monitoring points or faces can be created in the **Monitoring point/face panel** (Figure 51). The points coordinates can be set in the **Input panel** similarly to points input in Geometry tab. The faces can be selected from the drop-down list. The list contains available boundary faces (patches).

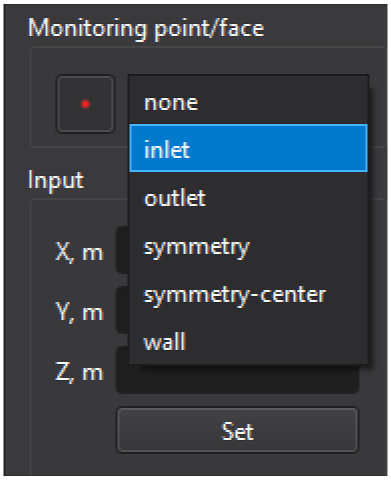


Figure 51. Monitoring point/face panel.

Monitoring entities can be deleted by pressing the RMB on a monitoring entity in the **Monitoring entities tree** and selecting **Delete** (Figure 52). Monitoring entities may also be hidden or unhidden by pressing the check mark box.

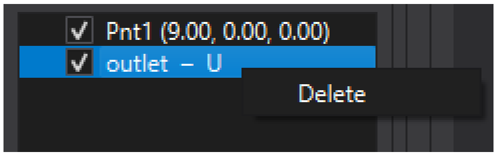


Figure 52. Deleting monitoring entities.

The monitoring entities are set in the Coordinates panel. Point can be defined via coordinate input, while face can be chosen from the **Selection** menu using their assigned names. By clicking on element in **Monitoring** entities tree the variable at point or face can be set (Figure 53).

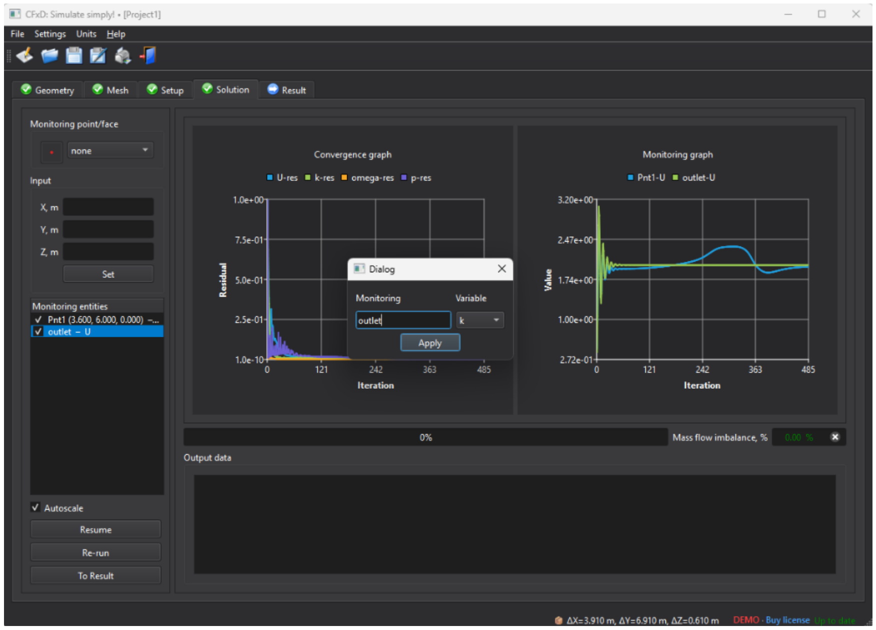


Figure 53. Setup of monitoring entity.

Convergence and monitoring curves can be viewed in the graphs on the right side of the program. The convergence graph has two display modes: **Linear** (default) and **Log scale**. Both graphs support **Autoscale**. When **Autoscale** is disabled, the graphs can be zoomed in and out manually. Below the graphs, the **Progress bar** and the **Mass flow imbalance** monitor are displayed. Detailed numerical values and important solver output messages are shown in the **Output data** window.

6.3 Start, Pause, Resume

Simulation execution is controlled using two buttons. Initially, only **Run** is enabled (Figure 54).

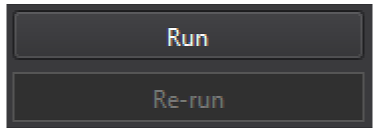


Figure 54. Initial buttons state.

After the simulation is started by clicking **Run**, the **Run** button changes to **Pause**, allowing the simulation to be paused (Figure 55).

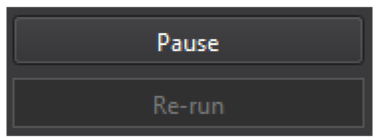


Figure 55. Button state after starting the run.

After clicking **Pause**, the second button becomes active, and the buttons change to **Resume** and **Re-run** (Figure 56). Clicking **Resume** continues the simulation from the paused state.

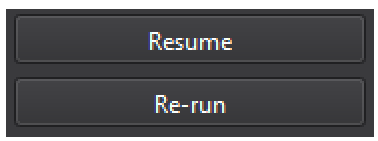


Figure 56. Button state after pausing.

The **Re-run** button restarts the simulation from the beginning. After the **Solution** step is completed, the next step, **Result**, can be opened using the **To Result** button.

Chapter VII. Result

The simulation results can be visualized in **Result** tab (Figure 57).

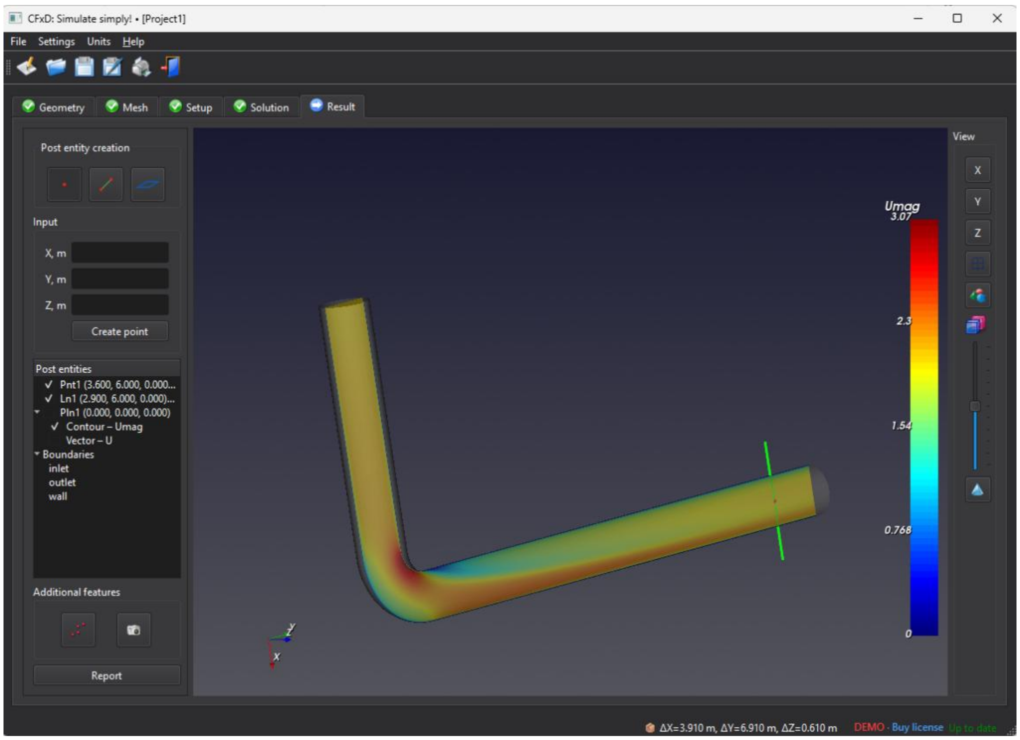


Figure 57. Result tab.

The results can be post-processed at a point (single value), along a line (chart) and on a plane (contour/vector).

7.1 Postprocessing at a point

The value of any variable at a point can be viewed in the dialog opened from the **Post entities** panel by clicking a point in the tree (Figure 58).

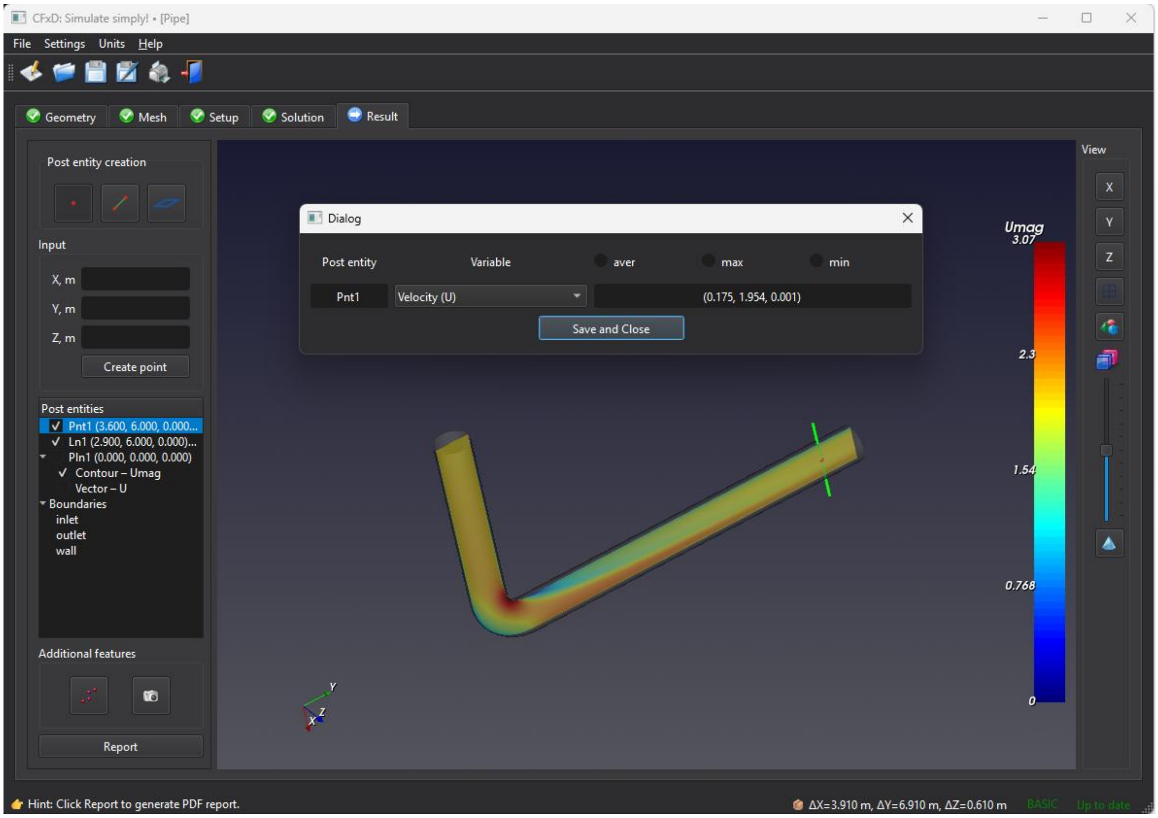


Figure 58. Velocity components value at a point.

7.2 Postprocessing along a line

Similarly, the values of any variable along a line (Chart) can be viewed in the dialog opened from the **Post entities** panel by clicking a line in the tree (Figure 59). In addition, the **average**, **minimum**, and **maximum** values along the line can be extracted. For comparison purposes, **external reference data** (e.g., experimental measurements or results from other simulations) in .csv format can also be added to the same chart. This allows CFxD results to be compared directly against a baseline dataset within a single plot.

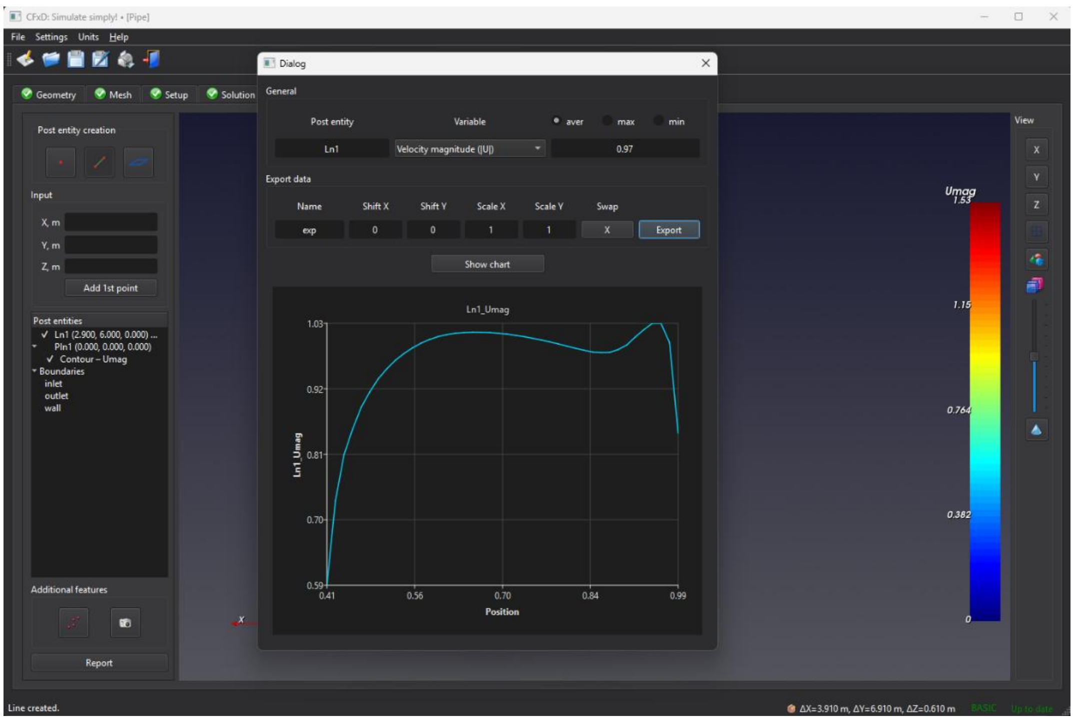


Figure 59. Chart at line.

7.3 Postprocessing on plane/boundary

The contour or vector plot can be visualized on a plane. It can be configured in the dialog opened from the **Post entities** panel by clicking a plane in the tree. The contour or vector plot is displayed in the **Graphics** window (Figure 60).

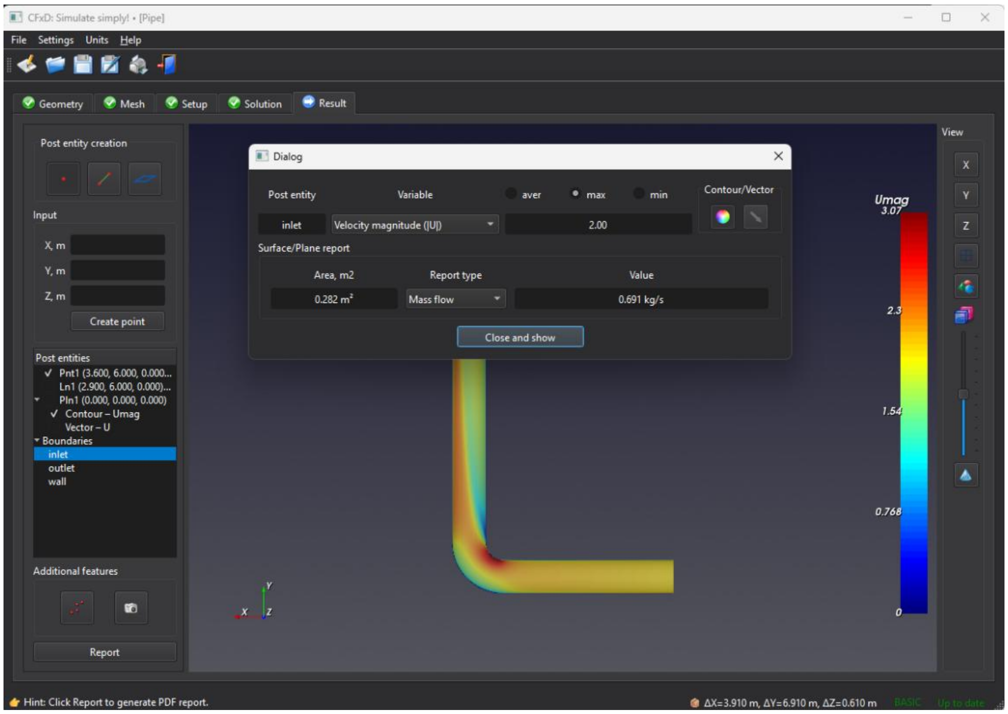


Figure 60. Velocity contour plot.

When **Velocity** is selected, the **Vector Settings** panel is enabled (Figure 61). It allows the vector display to be customized, including:

- **Vector scale**: set a manual scale or use **Auto** scaling.
- **Vector density**: increase or decrease the number of displayed arrows.
- **Uniform length**: display arrows with a constant length (direction only) or use auto magnitude-scaled arrows.

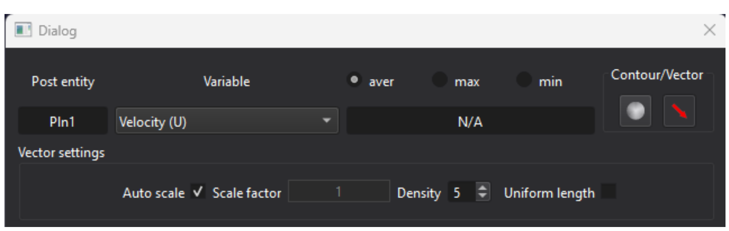


Figure 61. Vector settings panel.

The color bar on the right can be customized with a left or right mouse click. The minimum and maximum limits, as well as the number of ticks, can be adjusted (Figure 62). The color bar can also be reset to the default automatic settings.

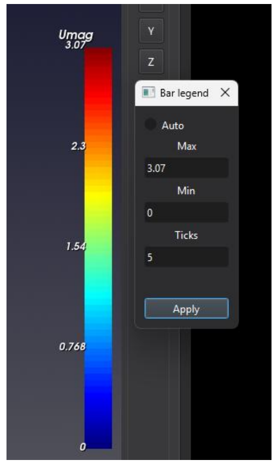


Figure 62. Customizing color bar.

Similar to line reports, average, maximum and minimum values can be extracted. In addition, mass flow rate, volumetric flow rate, and wall pressure force can be calculated.

7.4 Deleting and hiding post entities

Post entities can be deleted by pressing RMB over desired post entity in the "post entities window" and selecting delete (Figure 63). Monitoring entities may also be hidden or unhidden by pressing the check mark box.

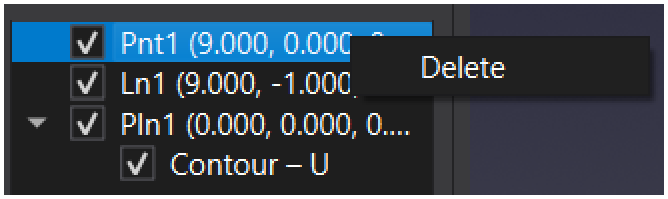


Figure 63.Deleting post entities.

Streamlines can be visualized using the Streamlines button. Streamlines are seeded from the inlet face.

7.5 Transient simulation result

For transient simulations, the **Post transient** panel is displayed. It allows results to be visualized at a selected time step. In addition, an animation dialog can be opened to create and play an animation (Figure 64). The animation is saved in the **Result** folder.

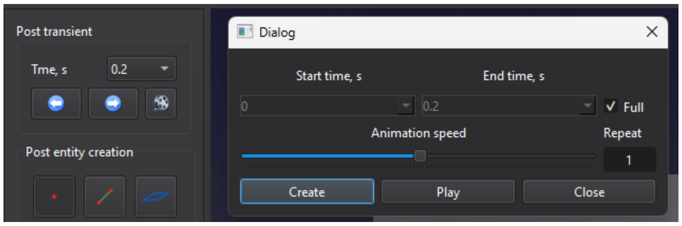
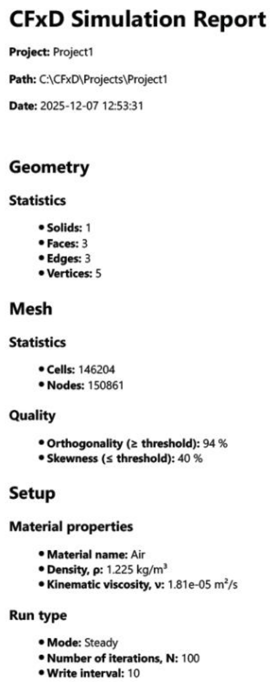


Figure 64. Post-processing for the transient simulations.

7.6 Report

The generated images can be saved to files. All important data can be compiled to PDF document using the **“Report”** button. An example of a generated report is shown in Figure 65.



CFxD Simulation Report

Project: Project1

Path: C:\CFxD\Projects\Project1

Date: 2025-12-07 12:53:31

Geometry

Statistics

- **Solids:** 1
- **Faces:** 3
- **Edges:** 3
- **Vertices:** 5

Mesh

Statistics

- **Cells:** 146204
- **Nodes:** 150861

Quality

- **Orthogonality (≥ threshold):** 94 %
- **Skewness (≤ threshold):** 40 %

Setup

Material properties

- **Material name:** Air
- **Density, ρ:** 1.225 kg/m^3
- **Kinematic viscosity, ν:** 1.81e-05 m^2/s

Run type

- **Mode:** Steady
- **Number of iterations, N:** 100
- **Write interval:** 10

Figure 65. Example of report.

References

1. Ferziger J. H., Peric M. Computational Methods for Fluid Dynamics. 3rd edition. Springer. 2002.
2. Schneider P. J., Eberly D. H. Geometric Tools for computer graphics. Elsevier Science. 2003.
3. Versteeg H. K., Malalasekera W. An introduction to Computational Fluid Dynamics.
4. VTK. Manual.
5. Open Cascade. Manual.
6. OpenFOAM. Manual.
7. OpenGL. Manual.
8. Qt. Manual.
9. Yun A. Computational Fluid Dynamics: from zero to guru. Creative Space, 2019.