

Roadmap to Lecture 0

- 1. Quick review of solution methods in CFD**
2. What is Ansys Fluent? – Executive summary
3. Turbulence in action

Quick review of solution methods in CFD

- In Computational Fluid Dynamics (CFD), we deal with the Navier-Stokes equations (NSE), plus additional closure models.

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

Additional equations deriving from models, such as, turbulence modeling, multiphase flows, heat transfer, radiation, chemical reactions, combustion, multi-species, thermodynamics, particle interaction, acoustics, mass transfer, and so on.

- This is a coupled system of nonlinear partial differential equations (PDE).
- That looks difficult to solve, and it is difficult to solve!

Quick review of solution methods in CFD

- In Computational Fluid Dynamics (CFD), we deal with the Navier-Stokes equations (NSE), plus additional closure models.

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

Additional equations deriving from models, such as, turbulence modeling, multiphase flows, heat transfer, radiation, chemical reactions, combustion, multi-species, thermodynamics, particle interaction, acoustics, mass transfer, and so on.

- Except for a few cases with large simplifications, no general analytical solution have been found to this set of equations.
- To find the approximated solution of this set of equations we use numerical methods.

Quick review of solution methods in CFD

- Many numerical methods exist to solve the Navier-Stokes equations, just to name a few:
 - Pressure-correction methods (Predictor-Corrector type).
 - SIMPLE, SIMPLEC, SIMPLER, PISO.
 - Projection methods.
 - Fractional step (operator splitting), MAC, SOLA.
 - Density-based methods and preconditioned solvers.
 - Riemann solvers, ROE, HLLC, AUSM+, ENO, WENO.
 - Artificial compressibility methods.
 - Artificial viscosity methods.
 - Methods Based on Derived Variables,
 - Stream Function-Vorticity
 - Vorticity-Velocity Method
 - And many more.

Quick review of solution methods in CFD

- Let us briefly review the following two types of approaches for solving the NSE:
 - **Pressure-based approach (predictor-corrector).**
 - **Density-based approach.**
- Historically speaking,
 - The **pressure-based approach** was developed for low-speed incompressible flows.
 - While the **density-based approach** was mainly developed for high-speed compressible flows.
- However, both methods have been extended and reformulated to solve and operate for a wide range of flow conditions beyond their original intent.
 - Both methods can be considered all-speed solvers.

Pressure-based approach

Quick review of solution methods in CFD

Pressure-based approach

- In the pressure-based approach,
 - The velocity field is obtained from the momentum equations.
 - The pressure is obtained by solving a pressure correction equation or the pressure-Poisson equation.
 - Therefore, there is some mathematical manipulation involved.
- Pressure-based methods are intrinsically implicit.
- They are the default option in most CFD solvers.

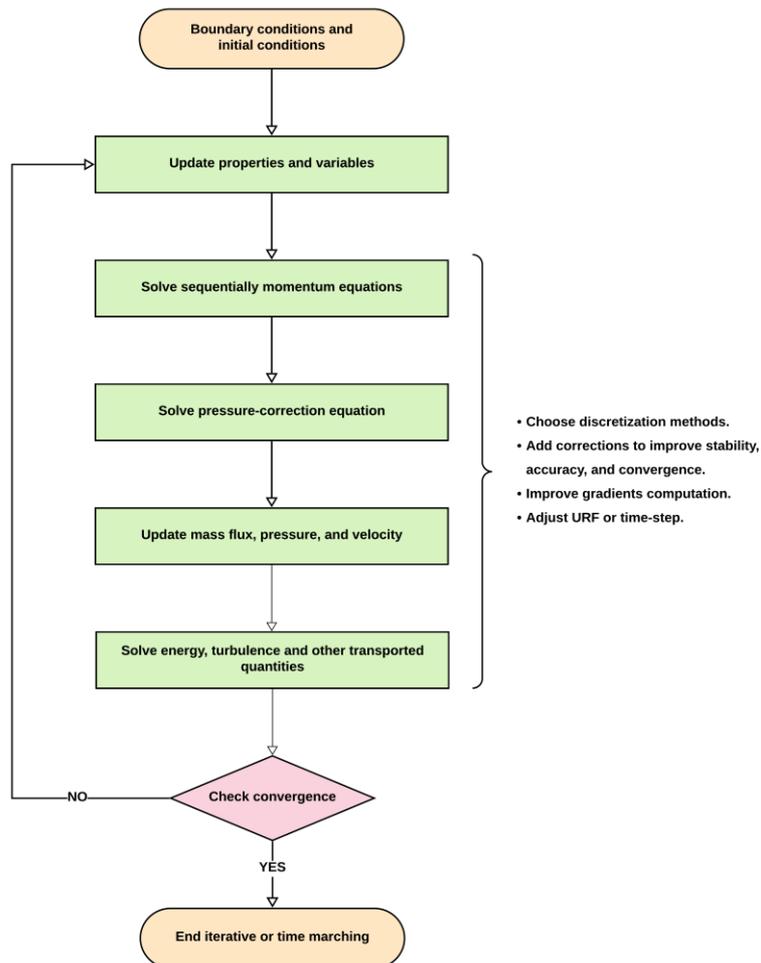
Quick review of solution methods in CFD

Pressure-based approach

- Two pressure-based solution methods are generally available, namely:
 - Segregated method.
 - Coupled method.
- In the segregated algorithm, the individual governing equations for the primitive variables are solved one after another.
- The coupled approach solves the continuity and momentum equations, simultaneously, that is, coupled together.
 - Additional transport equations (scalar transport, turbulence equations, energy, species, and so on), are solved in a segregated way.

Quick review of solution methods in CFD

Pressure-based approach – Segregated method



Each iteration consists of the steps illustrated in the figure and outlined below:

1. Update fluid properties (for example, density, viscosity, specific heat, turbulent viscosity, and so on) based on the initial conditions or current solution.
2. Solve the momentum equations, one after another, using the recently updated values of pressure and face mass fluxes.
3. Solve the pressure correction equation using the recently obtained velocity field and the mass-flux.
4. Correct face mass fluxes, pressure, and the velocity field using the pressure correction obtained from Step 3.
5. Solve the energy equation using the current values of the solution variables.
6. Solve additional transport equations, such as turbulent quantities, volume fraction, species, and so on, using the current values of the solution variables.
7. Check for the convergence of the equations.

These steps are continued until the convergence criterion is met or the user decides to stop the simulation.

Quick review of solution methods in CFD

Pressure-based approach – Segregated method

- In the pressure-based approach the velocity field is obtained from the momentum equations.
- The pressure field is extracted by solving a pressure or pressure correction equation which is obtained by manipulating continuity and momentum equations.
- If it is required, the energy equation is solved.
- Then, equations for other scalars such as turbulence, volume fraction, chemical species, etc., are solved.
- The solution process keeps iterating over the entire set of governing equations until the solution converges to a given criterion or the user decides to stop the simulation.
- In this approach, each governing equation while being solved, is **decoupled** or **segregated** from other equations, hence its name.
- The segregated algorithm is memory-efficient, since the discretized equations need only be stored in the memory one at a time.
- However, the solution convergence is relatively slow (in comparison to coupled solvers) as the equations are solved one at a time.

Quick review of solution methods in CFD

Pressure-based approach – Segregated method

- To achieve pressure-velocity coupling using segregated solvers several methods are available.
- To name a few:
 - **SIMPLE** (Semi-Implicit Method for Pressure-Linked Equations)
 - **SIMPLEC** (SIMPLE Corrected/Consistent)
 - **SIMPLER** (SIMPLE Revised)
 - **PISO** (Pressure Implicit with Splitting Operators)
- All the aforementioned methods are based on the predictor-corrector approach.
- Each one have different properties, options, and loop in slightly different ways.
- But at the end of the day, all of them will iterative until reaching the convergence criterion.

Quick review of solution methods in CFD

Pressure-based approach – A few remarks about the SIMPLE and PISO methods

- The **SIMPLE** method [1] was initially developed in the 1970s for steady state flows and extended later to unsteady flows with iterative marching at each time-step.
- In the **SIMPLE** method, under-relaxation must be used for stability reasons. This means that many iterations are required.
- The under-relaxation factors (URF) need to be tuned according to the application.
- For time dependent flows, iterative marching is necessary at each time-step, making the **SIMPLE** method inefficient.
- The **PISO** method [2] was developed in the early 1980s for steady and unsteady flows, and to address some of the drawbacks of the **SIMPLE** method.
- The **PISO** method is very efficient for unsteady flows since there is no need for iterative marching at each time-step.
- In the **PISO** method, the pressure field is free from continuity errors after the second corrector. Therefore, two correctors are normally sufficient.

References:

[1] S. V. Patankar. D. B. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", Int. J. Heat Mass Transfer, 15, 1787-1806 (1972).

[2] R. I. Issa, "Solution of the implicitly discretized fluid flow equations by operator-splitting", J. Comput. Phys., 62, 40-65 (1985).

Quick review of solution methods in CFD

Pressure-based approach – A few remarks about the SIMPLE and PISO methods

- The **PISO** method is stable for CFL numbers much greater than 1, and it does not need adjustable under-relaxation parameters.
 - However, it is recommended to do a few corrector steps to improve the stability and accuracy.
- The **PISO** method often needs more than two correctors when large pressure gradients exist.
- The **SIMPLE** and **PISO** methods are both implicit.
- Both methods were initially developed for staggered grids, but now they can be used with collocated meshes (standard practice in most modern CFD solvers).
- In the **SIMPLE** and **PISO** methods, to avoid the pressure-velocity decoupling that occurs when using collocated meshes, the cell-face velocity value is computed using Rhie-Chow interpolation [1].
- For unsteady problems, the **PISO** method is much faster than the **SIMPLE** method.
- For steady problems, both methods shows similar convergence rates.
 - However, the **SIMPLE** method is less computational expensive.
- The drawback of the **SIMPLE** method for steady simulations is that the URF are problem dependent.

References:

[1] C. M. Rhie, W. L. Chow, "Numerical study of the turbulent flow past an airfoil with trailing edge separation", AIAA Journal, Vol. 21, 1525-1532 (1983).

Quick review of solution methods in CFD

Pressure-based approach – On the origins of the SIMPLE and PISO methods

- **SIMPLE**

- S. V. Patankar and D. B. Spalding, “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows”, *Int. J. Heat Mass Transfer*, 15, 1787-1806 (1972).

- **SIMPLE-C**

- J. P. Van Doormaal and G. D. Raithby, “Enhancements of the SIMPLE method for predicting incompressible fluid flows”, *Numer. Heat Transfer*, 7, 147-163 (1984).

- **SIMPLE-R**

- S.V. Patankar, “A calculation procedure for two dimensional elliptic situations”, *Numerical Heat Transfer*, Vol. 14, pp. 409-425 (1984).

- **PISO**

- R. I. Issa, “Solution of the implicitly discretized fluid flow equations by operator-splitting”, *J. Comput. Phys.*, 62, 40-65 (1985).

- **Rhie-Chow interpolation**

- C. M. Rhie and W. L. Chow, “Numerical study of the turbulent flow past an airfoil with trailing edge separation”, *AIAA Journal*, Vol. 21, 1525-1532 (1983).

Quick review of solution methods in CFD

Pressure-based approach – Derivation of the pressure-correction equation

- The pressure equation is derived starting from the momentum equation,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} = -\nabla p$$

- Then, by taking the divergence of the momentum equation and setting $\nabla \cdot \mathbf{U} = 0$, we obtain,

$$\cancel{\nabla \cdot \left(\frac{\partial \mathbf{U}}{\partial t} \right)} + \nabla \cdot (\nabla \cdot (\mathbf{U}\mathbf{U})) - \cancel{\nabla \cdot (\nu \nabla^2 \mathbf{U})} = -\nabla^2 p$$

- Then, the final form of the pressure equation is as follows,

$$\nabla^2 p = \nabla \cdot \mathbf{H}(\mathbf{U}) \quad \text{where} \quad \mathbf{H}(\mathbf{U}) = -\nabla \cdot (\mathbf{U}\mathbf{U})$$

- Notice that the continuity is enforced when deriving the pressure equation and in all boundaries of the domain.

Quick review of solution methods in CFD

Pressure-based approach – Derivation of the pressure-correction equation

- In the pressure-based approach, the actual equations that are being solved are,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nu \nabla^2 \mathbf{U} = -\nabla p$$

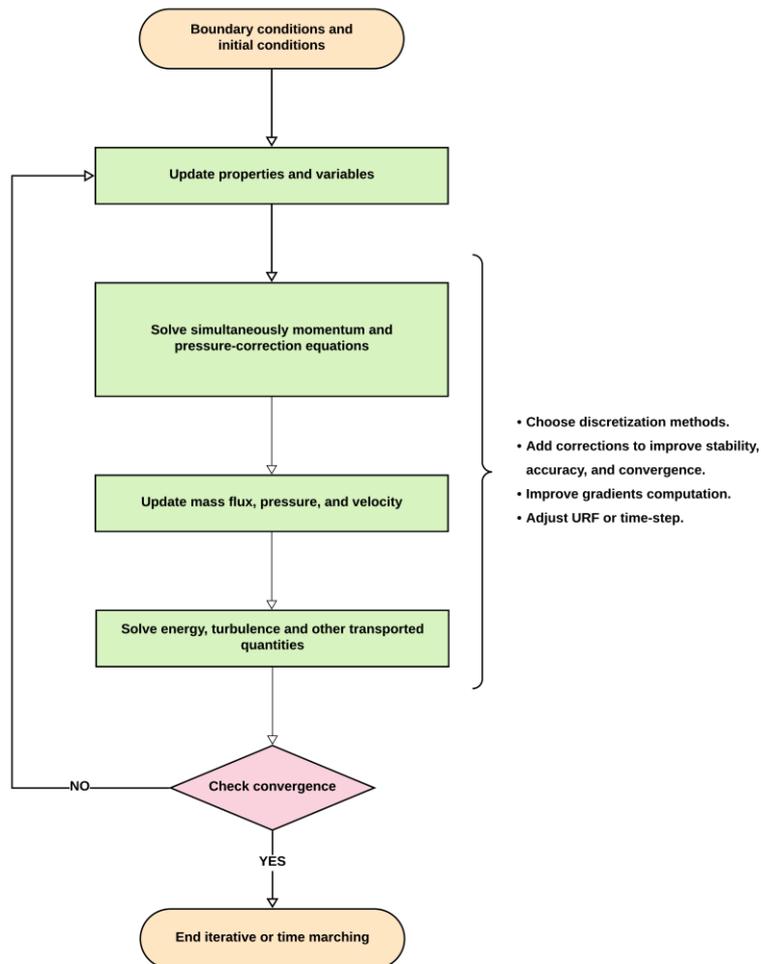
$$\nabla^2 p = \nabla \cdot \mathbf{H}(\mathbf{U}) \quad \text{where} \quad \mathbf{H}(\mathbf{U}) = -\nabla \cdot (\mathbf{U}\mathbf{U})$$

This system of equations is mathematically equivalent to the original Navier-Stokes equations.

- The previous equations are solved in a given domain, with boundary conditions **BCs**, and initial condition **ICs**.
- In this set of equations, continuity $\nabla \cdot \mathbf{U} = 0$ is enforced while deriving the pressure equation (referred to as pressure-Poisson equation) and in all boundaries of the domain.
- We use these equations because in the original incompressible Navier-Stokes equations, pressure does not appear in the continuity equation, so is not possible to link the equations.
- Therefore, we derive an alternative set of equations where pressure appears, albeit in the form of a gradient and large pressure gradients may cause stability and accuracy problems.
- The velocity obtained in the momentum equation (momentum predictor step) is used to compute the pressure using the pressure-Poisson equation (pressure corrector step), and then the velocity is corrected with the new pressure value (momentum corrector step).
- This is referred to as pressure-velocity coupling (P-V coupling).

Quick review of solution methods in CFD

Pressure-based approach – Coupled method



Each iteration consists of the steps illustrated in the figure and outlined below:

1. Update fluid properties (for example, density, viscosity, specific heat) including turbulent viscosity based on the initial conditions or current solution.
2. Solve the momentum equations and pressure-correction equation in a coupled manner.
3. Correct face mass fluxes, pressure, and the velocity field using the pressure correction obtained from Step 2.
4. Solve the energy equation using the current values of the solution variables.
5. Solve additional transport equations, such as turbulent quantities, volume fraction, species, and so on, using the current values of the solution variables.
6. Check for the convergence of the equations.

These steps are continued until the convergence criterion is met or the user decides to stop the simulation.

Quick review of solution methods in CFD

Pressure-based approach – Coupled method

- The pressure-based coupled algorithm solves a coupled system of equations comprising the momentum equations and the pressure-based continuity equation.
- The remaining equations are solved in a decoupled fashion as in the segregated algorithm.
- Since the momentum and continuity equations are solved in a closely coupled manner, the rate of convergence significantly improves when compared to the segregated algorithm.
- However, compared to the segregated algorithm, the memory requirements are much larger.
 - By at least 2 to 3 times.
- As memory requirements are very high, they are not very efficient for unsteady computations.
- This method has a high computational cost per iteration.
- Disregarding the computational drawbacks of the coupled method, when dealing with steady simulations the convergence rate is must faster when compared to the segregated algorithm.
- The coupled method is strongly recommended for steady simulations and stiff problems.

Density-based approach

Quick review of solution methods in CFD

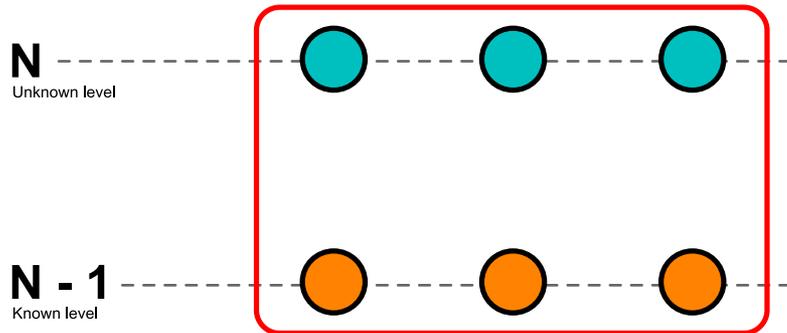
Density-based approach

- The density-based approach solves the continuity, momentum, energy, and species equations simultaneously, that is, coupled together.
- Conversely to the pressure-based approach, there is no mathematical manipulation on the governing equations.
- Pressure is obtained through an equation of state and additional thermodynamics relationships.
- Governing equations for additional scalars are solved afterward and sequentially, that is, segregated from one another and from the coupled set.
- The density-based solvers are recommended when there is a strong coupling between density, energy, momentum, and/or species.
- To deal with low Mach number flows and incompressible flows, the density-based approach use equation preconditioning.
- Because the governing equations are non-linear and coupled, several iterations of the solution loop must be performed before a converged solution is obtained.
- Typical solution methods used in the density-based approach are:
 - ROE, AUSM+, HLLC

Quick review of solution methods in CFD

Density-based approach

- In the density-based solution method, you can solve the coupled system of equations using either an explicit formulation or an implicit formulation.
 - **Implicit formulation:**
 - For a given variable, the unknown value in each cell is computed using a relation that includes both existing and unknown values from neighboring cells.
 - Therefore, each unknown will appear in more than one equation in the system, and these equations must be solved simultaneously to give the unknown quantities.

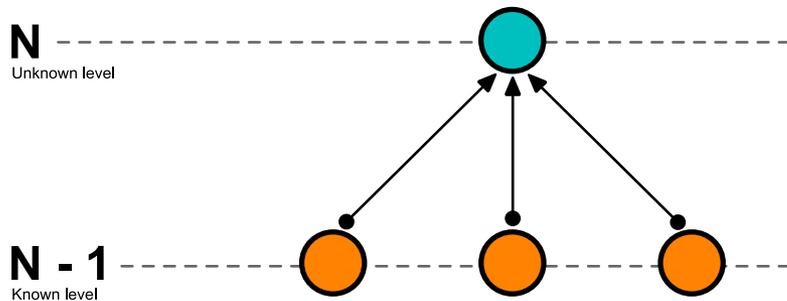


- In the figure, **N** is the current time level. We do not know the solution in this level.
- **N - 1** is the previous time level, where we know the solution in all control volumes.

Quick review of solution methods in CFD

Density-based approach

- In the density-based solution method, you can solve the coupled system of equations using either an explicit formulation or an implicit formulation.
 - **Explicit formulation:**
 - For a given variable, the unknown value in each cell is computed using a relation that includes only existing values.
 - Therefore, each unknown will appear in only one equation in the system and the equations for the unknown value in each cell can be solved one at a time to give the unknown quantities.



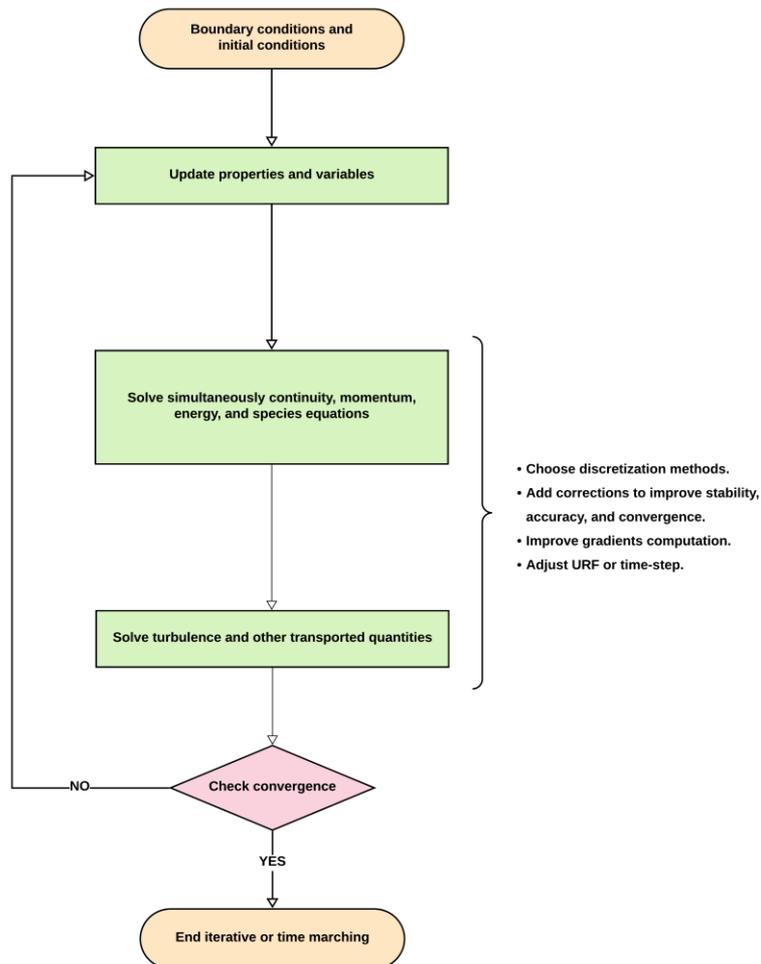
- In the figure, N is the current time level. We do not know the solution in this level.
- $N-1$ is the previous time level, where we know the solution in all control volumes.

Quick review of solution methods in CFD

Density-based approach

- In the density-based solution method, you can solve the coupled system of equations using either an explicit formulation or an implicit formulation.
- The explicit formulation is very memory efficient and fast, but it is conditionally stable.
 - This formulation has a terrible constrain on the maximum allowable CFL number.
 - Usually, the maximum allowable **CFL** number is limited to 1.0.
 - Some explicit methods have a **CFL condition** of 0.5, and some of them can go up to 2.0
 - Also, the time step is an order of magnitude smaller than that required in the implicit formulation using the same CFL number.
- However, the implicit formulation is preferred as:
 - It is unconditionally stable, that is, there is no constrain on the maximum allowable CFL number.
 - It tends to be more stable and faster.
- Using good standard practices, both formulations have the same degree of accuracy.

Density-based approach



Each iteration consists of the steps illustrated in the figure and outlined below:

1. Update fluid properties (for example, density, viscosity, specific heat) including turbulent viscosity based on the initial conditions or current solution.
2. Solve the continuity, momentum, energy, and species equations in a coupled manner.
3. Solve additional transport equations, such as turbulent quantities, volume fraction, and so on, using the current values of the solution variables.
4. Check for the convergence of the equations.

These steps are continued until the convergence criterion is met or the user decides to stop the simulation.

Quick and dirty introduction to the finite volume method FVM

Quick review of solution methods in CFD

- In CFD and multiphysics simulations there are many discretization approaches, just to name a few:
 - Finite Difference Method → FDM
 - Finite Element Method – Galerkin → G-FEM
 - Finite Element Method – Discontinuous Galerkin → DG-FEM
 - Finite Volume Method → FVM
 - Immersed Boundary Method → IBM
 - Lattice Boltzmann Method → LBM
 - Spectral Element Methods → SEM
 - Boundary Element Method → BEM
- Each method will find the approximated numerical solution of the governing equations
- The main difference among all methods is the way how they arrive to the system of discrete algebraic equations.
- Most multiphysics frameworks and CFD solvers are based on the FVM.
- The popularity of the FVM relies on the fact that it can be used with arbitrary cell types, it is easy to implement, and it enforces conservation in every single cell of the mesh (thus in the whole domain).
- OpenFOAM, SU2, code Saturne, FUN3D, USM3D, CFDShip-Iowa, CFX, FLUENT, Star-CCM, NUMECA, and CFD-ACE+ (among many CFD solvers), are all based on the FVM.

Quick review of solution methods in CFD

- Let us use the integral form of the general transport equation as the starting point to explain the basic theory behind the FVM method,

$$\underbrace{\int_{V_P} \frac{\partial \rho \phi}{\partial t} dV}_{\text{Temporal derivative}} + \underbrace{\int_{V_P} \nabla \cdot (\rho \mathbf{u} \phi) dV}_{\text{Convective term}} - \underbrace{\int_{V_P} \nabla \cdot (\rho \Gamma_\phi \nabla \phi) dV}_{\text{Diffusion term}} = \underbrace{\int_{V_P} S_\phi(\phi) dV}_{\text{Source term}}$$

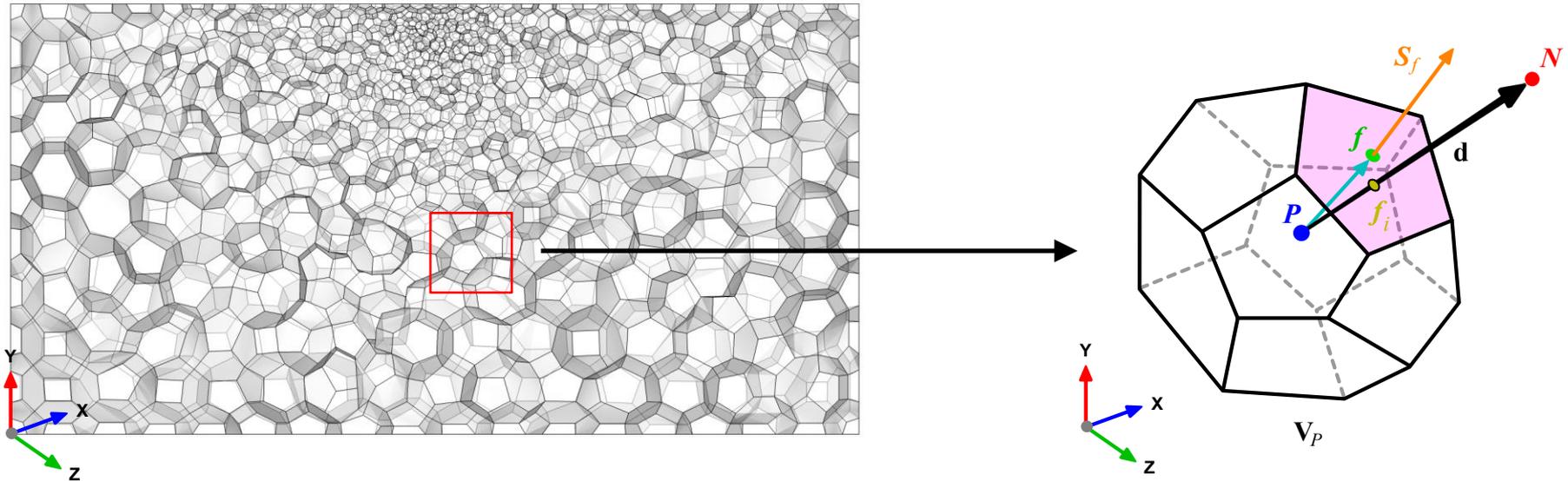
- Using either of the methods that we just reviewed (pressure based or density based), any general CFD solver based on the FVM will find the approximated numerical solution of the general transport equation for the transported quantity ϕ in a given domain, with given boundary conditions **BC** and initial conditions **IC**.
- This is a second order equation.
 - Therefore, for good accuracy, it is necessary that the order of the discretization is equal or higher than the order of the equation that is being discretized.
- Remember, starting from this equation we can write down the Navier-Stokes equations (NSE).
- Therefore, everything we are going to address also applies to the NSE.

Quick review of solution methods in CFD

- Independently of the method used (pressure based or density based), in the FVM method we follow these steps,
 - **Domain discretization or mesh generation:** division of the domain into discrete control volumes.
 - **Equation discretization:** integration of the governing equations on the individual control volumes to construct algebraic equations for the discrete dependent variables (unknowns) such as velocities, pressure, temperature, and conserved scalars.
 - **Linear system solution:** solution of the linear system arising from the integration of the governing equations.

Quick review of solution methods in CFD

- Let us divide the solution domain into a finite number of arbitrary control volumes or cells, such as the one illustrated below.
- Inside each control volume the solution is sought.
- We know all the geometrical information of all cells.
- That is, cell centers, face centers, cells neighbors, face connectivity, cells volume, faces area, vectors connecting cells centers, face centers, and so on.



Quick review of solution methods in CFD

- After dividing the solution domain into a finite number of arbitrary control volumes or cells, we need to discretize the governing equation in the domain of interest.
- That is, we need to convert the equation from the continuum world to the discrete world.
- Let us introduce the algebraic equations arising from the discretization of the governing equations.
- To arrive to the algebraic set of equations that we will show next, we followed these steps,
 - Integrating in space and time each term of the general transport equation.
 - Using Gauss theorem to convert the volume integrals into surface integrals.
 - Using the mid point rule to approximate the integrand (by the way, there are many ways to approximate the integrand).
- It is important to mention that all the approximations we have introduced so far are at least second-order accurate.
- Also, we are assuming that we are using a collocated arrangement, that is, all quantities are computed and stored at the cell centers.

Quick review of solution methods in CFD

- Integrating in space each term of the general transport equation and by using Gauss theorem, yields to the following discrete equations for each term,

Convective term:

$$\int_{V_P} \underbrace{\nabla \cdot (\rho \mathbf{u} \phi)}_{\text{convective term}} dV = \oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\rho \mathbf{u} \phi)}_{\text{convective flux}} = \sum_f \int_f d\mathbf{S} \cdot (\rho \mathbf{u} \phi)_f \approx \sum_f \mathbf{S}_f \cdot (\overline{\rho \mathbf{u} \phi})_f = \sum_f \mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f$$

Diffusive term:

$$\int_{V_P} \underbrace{\nabla \cdot (\rho \Gamma_\phi \nabla \phi)}_{\text{diffusion term}} dV = \oint_{\partial V_P} \underbrace{d\mathbf{S} \cdot (\rho \Gamma_\phi \nabla \phi)}_{\text{diffusive flux}} = \sum_f \int_f d\mathbf{S} \cdot (\rho \Gamma_\phi \nabla \phi)_f \approx \sum_f \mathbf{S}_f \cdot (\overline{\rho \Gamma_\phi \nabla \phi})_f = \sum_f \mathbf{S}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f$$

Source term:

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

Gradient term:

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

Quick review of solution methods in CFD

Space discretization

- Using the previous equations to evaluate the general transport equation over all the control volumes in the domain, we obtain the following semi-discrete equation,

$$\int_{V_P} \underbrace{\frac{\partial \rho \phi}{\partial t} dV}_{\text{temporal derivative}} + \sum_f \underbrace{\mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f}_{\text{convective flux}} - \sum_f \underbrace{\mathbf{S}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f}_{\text{diffusive flux}} = \underbrace{(S_c V_P + S_p V_P \phi_P)}_{\text{source term}}$$

where $\mathbf{S} \cdot (\rho \mathbf{u} \phi) = F^C$ is the convective flux and $\mathbf{S} \cdot (\rho \Gamma_\phi \nabla \phi) = F^D$ is the diffusive flux.

- The face values appearing in the convective and diffusive fluxes have to be computed by some form of interpolation from the centroid values of the control volumes at both sides of face f .
- Also, the gradients appearing in the governing equations need to be computed with high accuracy.
- Many methods are available for space discretization of the convective fluxes, diffusive fluxes, and gradients.

Quick review of solution methods in CFD

Time discretization

- Using the previous equations to evaluate the general transport equation over all the control volumes in the domain, we obtain the following semi-discrete equation,

$$\int_{V_P} \underbrace{\frac{\partial \rho \phi}{\partial t} dV}_{\text{temporal derivative}} + \sum_f \underbrace{\mathbf{S}_f \cdot (\rho \mathbf{u} \phi)_f}_{\text{convective flux}} - \sum_f \underbrace{\mathbf{S}_f \cdot (\rho \Gamma_\phi \nabla \phi)_f}_{\text{diffusive flux}} = \underbrace{(S_c V_P + S_p V_P \phi_P)}_{\text{source term}}$$

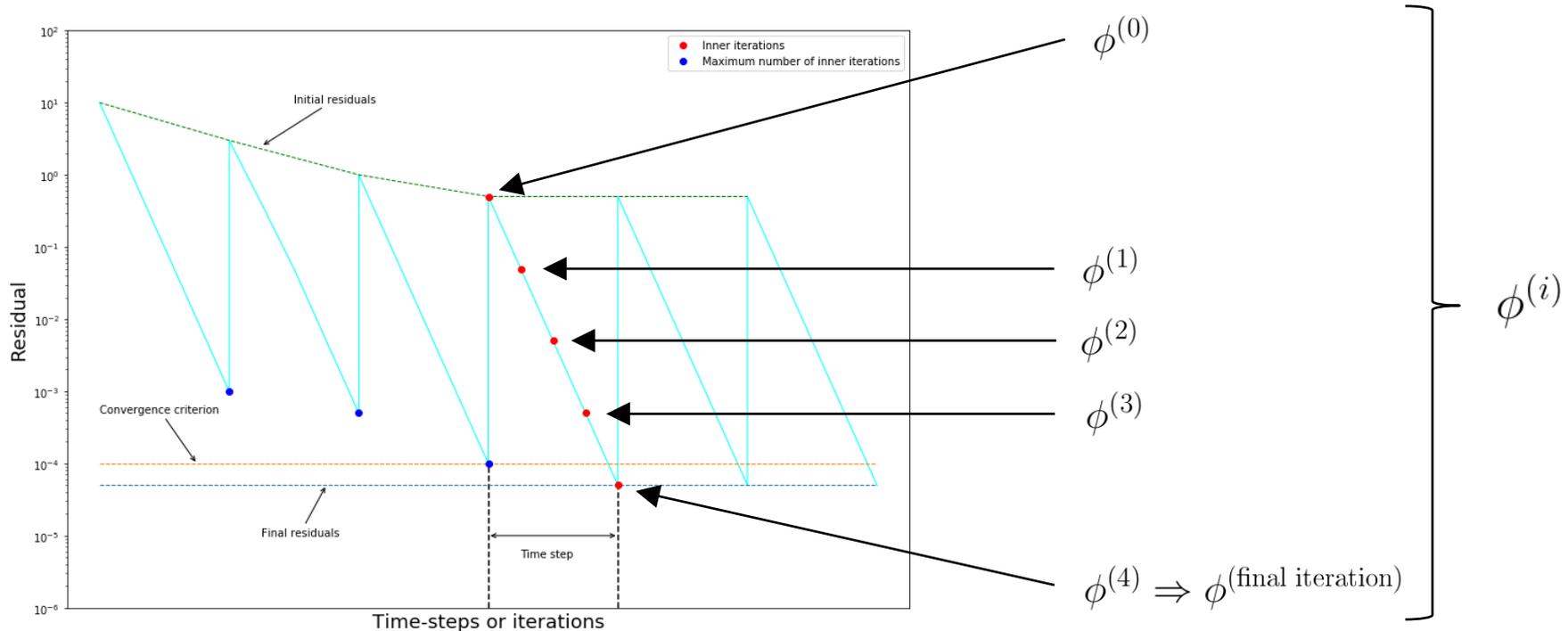
where $\mathbf{S} \cdot (\rho \mathbf{u} \phi) = F^C$ is the convective flux and $\mathbf{S} \cdot (\rho \Gamma_\phi \nabla \phi) = F^D$ is the diffusive flux.

- After spatial discretization, we can proceed with the time discretization.
- By proceeding in this way, we are using the Method of Lines (MOL).
- The main advantage of the MOL method, is that it allows us to select numerical approximations of different accuracy for the spatial and time terms.
- Each term can be treated differently to yield to different accuracies.

Quick review of solution methods in CFD

Linear system solution

- To get a better idea of how iterative methods work, and what are initial residuals and final residuals, let us take another look at a residual plot.



- If the following condition is fulfilled, the linear solver will stop iterating and will advance to the next time-step.

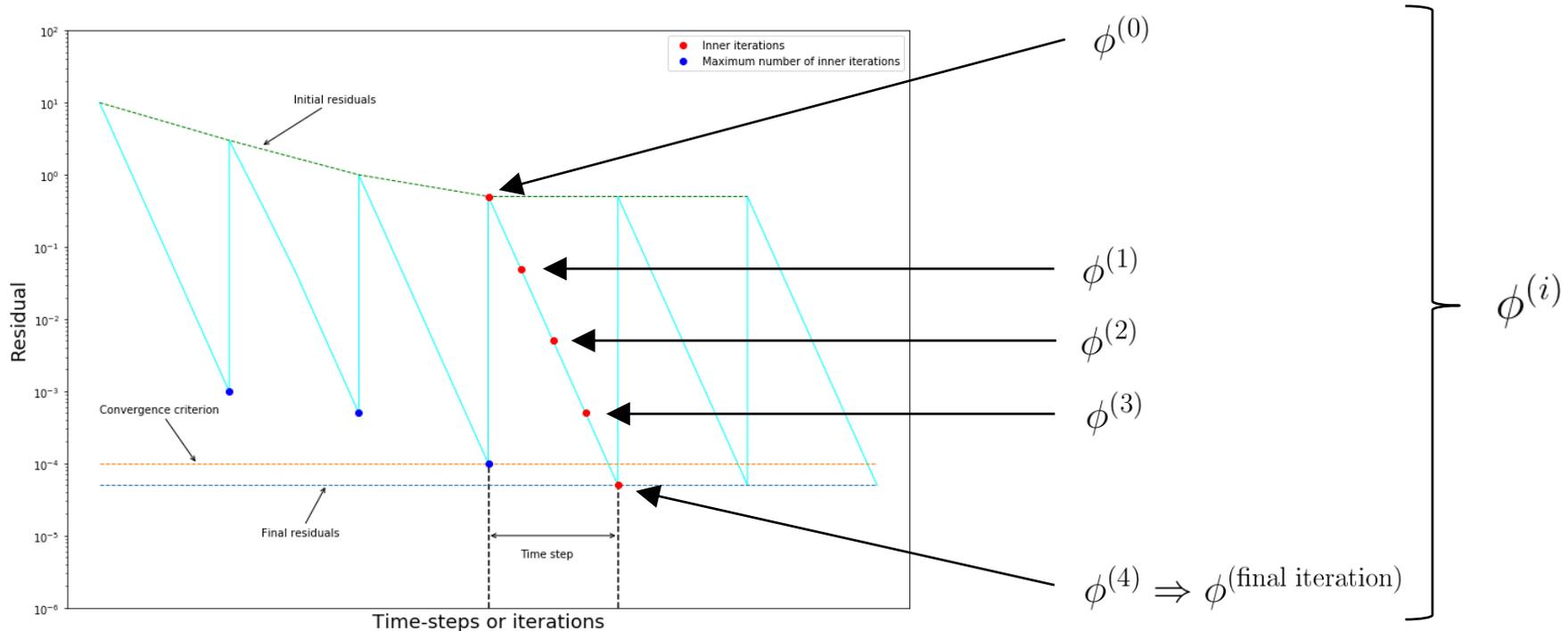
$$|\mathbf{A}\phi^i - \mathbf{b}| \leq |\mathbf{r}|$$

- This condition defines the final residual, where \mathbf{r} is the tolerance or convergence criterion (defined by the user).

Quick review of solution methods in CFD

Linear system solution

- To get a better idea of how iterative methods work, and what are initial residuals and final residuals, let us take another look at a residual plot.



- By working in an iterative way, every single iteration $\phi^{(i)}$ is a better approximation of the previous iteration.
- It is clear that if the initial residual $|\mathbf{A}\phi - \mathbf{b}|^0$ is the same as the final residual $|\mathbf{A}\phi - \mathbf{b}|^{\text{Final iteration}}$ (we are converging in one iteration), we can say that we have reached a steady solution (this does not happen very often).
- Every iterative linear solver has different properties.

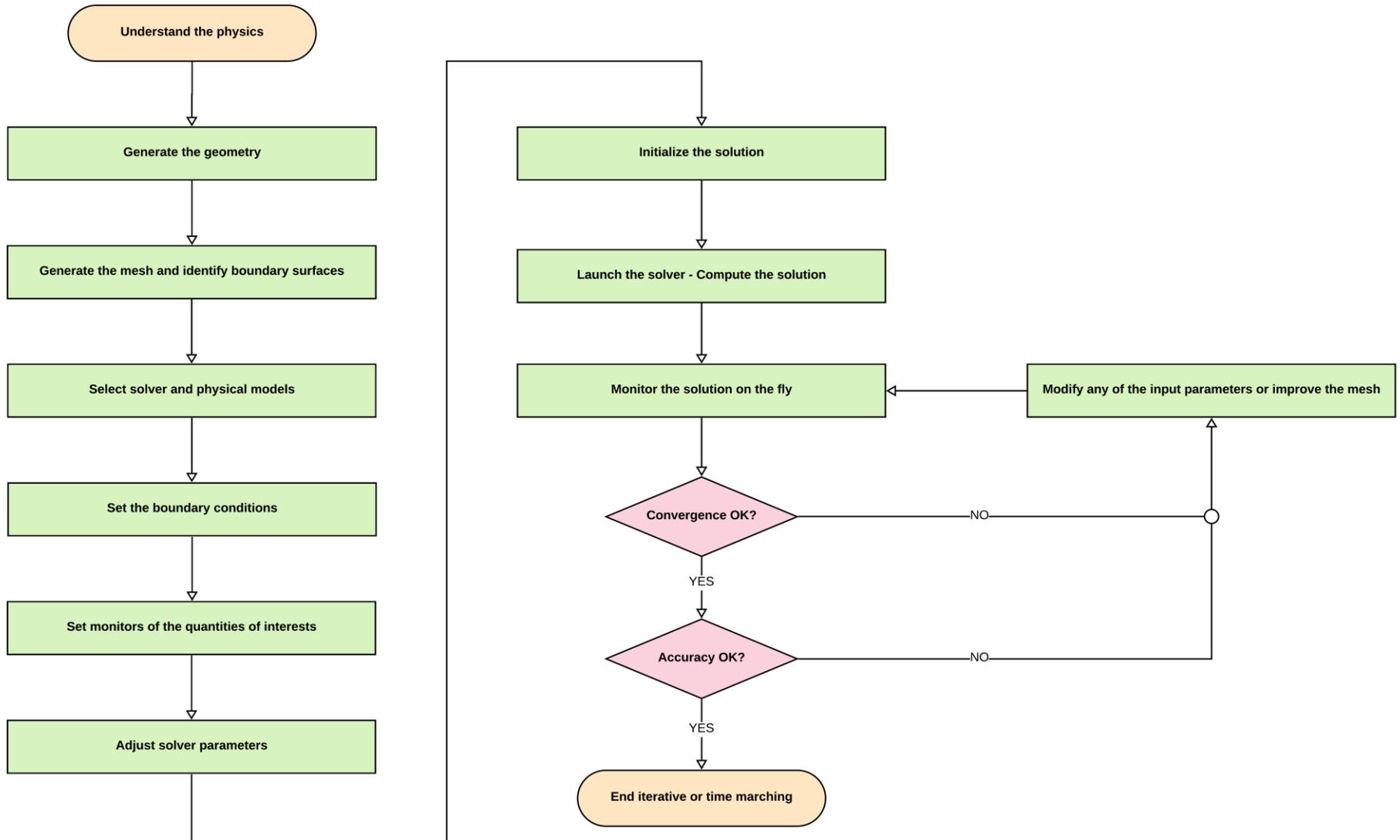
Quick review of solution methods in CFD

So, what does an unstructured FVM solver do?

- It simply discretizes in space and time the governing equations in arbitrary polyhedral control volumes over the whole domain. Assembling in this way a large set of linear differential algebraic equations (DAE), and then it solves this system of DAE to find the solution of the transported quantities.
- Therefore, the following information must be readily available to the solver:
 - The mesh.
 - Boundary conditions and initials conditions.
 - Physical properties such as density, gravity, diffusion coefficient, viscosity, etc.
 - Physical models, such as turbulence, mass transfer, etc.
 - How to discretize in space each term of the governing equations (diffusive, convective, gradient and source terms).
 - How to discretize in time the obtained semi-discrete governing equations.
 - How to solve the linear system of equations (crunching numbers).
 - Set runtime parameters and general instructions on how to run the case (such as time step, under-relaxation factors, and maximum CFL number).
 - Additionally, we may set monitors for post-processing.
- Every CFD solver has a different way to ask for this information.
- Some solvers use a GUI (e.g., Fluent, StarCCM+, CFX, NUMECA), and others interacts via ascii files using the command line interface (e.g., OpenFOAM, SU2, FLUBIO-PETSC).

Quick review of solution methods in CFD

Workflow of a simulation



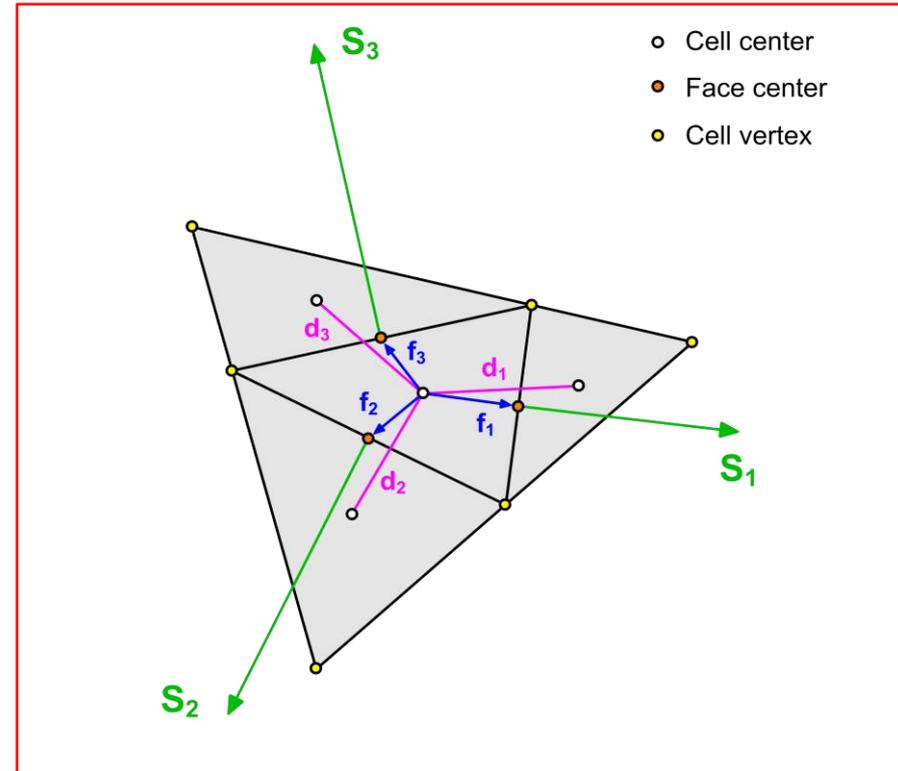
Quick review of solution methods in CFD

Final remarks – In CFD, the mesh is everything

- The matrix of coefficients of the discretized system of algebraic equations depends on the geometrical quantities shown in the figure.
- Specifically, on the dot product of \mathbf{S} (the vector normal to the face passing by the face center) and \mathbf{d} (the vector that connects two cell centers).
- This dependence on the dot product $\mathbf{S} \cdot \mathbf{d}$ is due to the fact that the coefficients contain the following term,

$$\frac{\mathbf{S} \cdot \mathbf{S}}{\mathbf{S} \cdot \mathbf{d}}$$

- This dependence arises from the discretization of the diffusive fluxes and the need of correcting secondary gradients.
- Have in mind that many methods have been developed to deal with the discretization of the diffusive fluxes and treatment of the secondary gradients.
- The dependence shown above corresponds to the most successful approach, which is the one used in Ansys Fluent and many other FVM solvers.
- For a review of the different methods, the interested reader can refer to the reference [1].



In the figure:

- \mathbf{S} is the vector normal to face and anchored at the face center
- \mathbf{d} is the vector connecting two cell centers.
- \mathbf{f} is the vector from the cell center to the face center.
- If all these vectors are aligned, we are in the presence of a perfect mesh. In practice, this does not happen very often.

Quick review of solution methods in CFD

Final remarks – In CFD, the mesh is everything

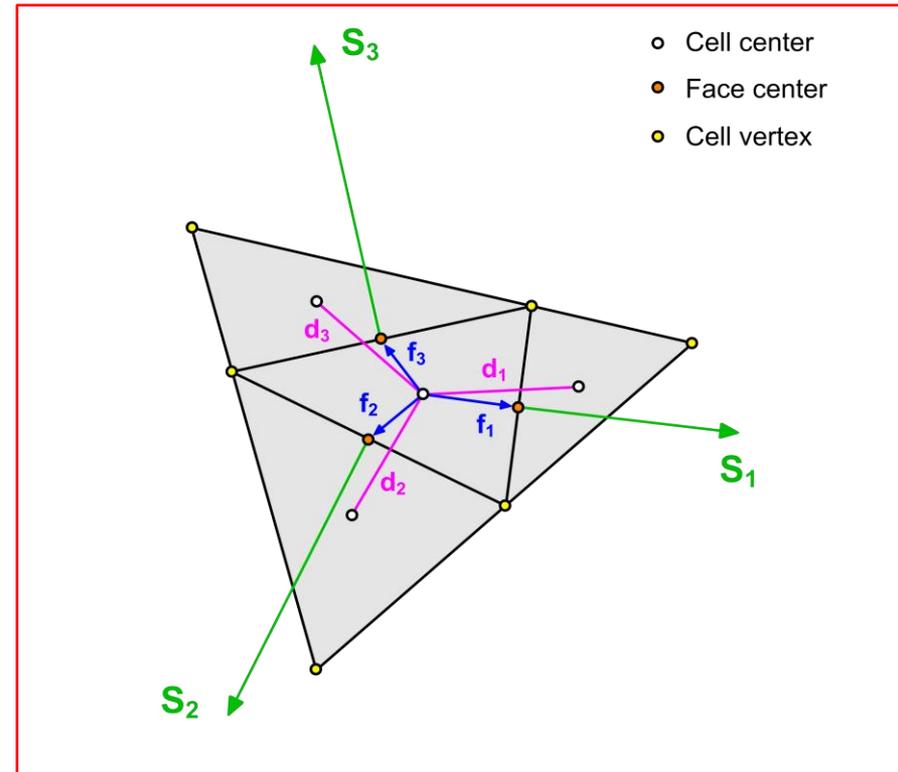
- In reference to the following term,

$$\frac{\mathbf{S} \cdot \mathbf{S}}{\mathbf{S} \cdot \mathbf{d}}$$

- Recall from linear algebra that the dot product of two Euclidian vectors \mathbf{a} and \mathbf{b} is defined as,

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\theta)$$

- Therefore, if the two vectors \mathbf{a} and \mathbf{b} are orthogonal, the dot product $\mathbf{a} \cdot \mathbf{b}$ is equal to zero as $\cos(90)$ is equal to zero.
- And this has important implications when assembling the matrix of coefficients during space discretization of the governing equations.
- From the figure and the dot product definition, we can say the following about the quality of the mesh:
 - A good quality mesh is a mesh where the vectors \mathbf{S} and \mathbf{d} are parallel.
 - A bad quality mesh is a mesh where the vectors \mathbf{S} and \mathbf{d} are orthogonal.



In the figure:

- \mathbf{S} is the vector normal to face and anchored at the face center
- \mathbf{d} is the vector connecting two cell centers.
- \mathbf{f} is the vector from the cell center to the face center.
- If all these vectors are aligned, we are in the presence of a perfect mesh. In practice, this does not happen very often.

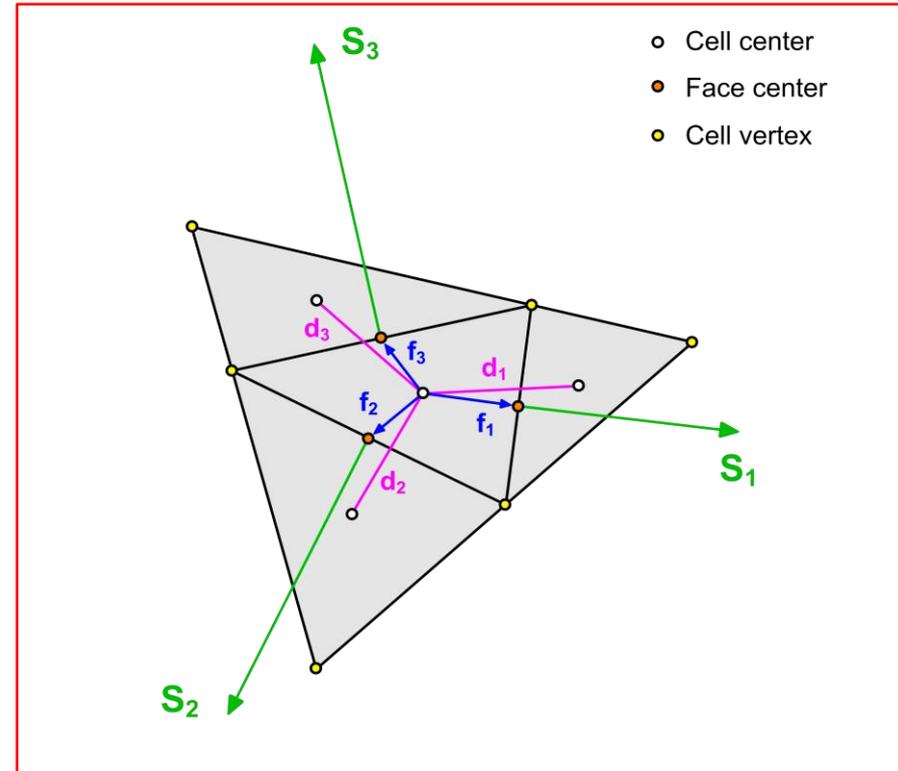
Quick review of solution methods in CFD

Final remarks – In CFD, the mesh is everything

- For perfect cells (orthogonal meshes), the trigonometric term of the dot product is equal to one (there is no deviation between the vectors \mathbf{S} and \mathbf{d}).

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \underbrace{\|\mathbf{b}\| \cos(\theta)}$$

- The more a cell deviates from its perfect shape, the smaller the trigonometric term in the dot product definition becomes.
- And this results in large values of the matrix coefficients which increases the system stiffness.
- For very bad quality cells (e.g., very skew cells or cells with zero volume), this vector product may become zero, producing an undefined system (throwing a division by zero error).
- The dot product $\mathbf{S} \cdot \mathbf{d}$, affects in particular the Laplacian and gradient terms appearing in the governing equations.
- It introduces secondary gradients that need to be corrected.
- It is worth noting that you can mesh meshes with zero non-orthogonality with any cell type.
 - 2D: quads, triangles, polyhedrons, and so on.
 - 3D: hexes, tetras, pyramids, prisms, polyhedrons, and so on.



In the figure:

- \mathbf{S} is the vector normal to face and anchored at the face center
- \mathbf{d} is the vector connecting two cell centers.
- \mathbf{f} is the vector from the cell center to the face center.
- If all these vectors are aligned, we are in the presence of a perfect mesh. In practice, this does not happen very often.

Quick review of solution methods in CFD

Final remarks – Useful references

- During the derivation of the algebraic set of equations we fast-forwarded many steps, for a detailed derivation, please refer to the following references,
 - S. Patankar. Numerical Heat Transfer and Fluid Flow. Taylor & Francis, 1980.
 - S. Mazumder. Numerical Methods for Partial Differential Equations: Finite Difference and Finite Volume Methods. Academic Press, 2015.
 - H. K. Versteeg, W. Malalasekera. An Introduction to Computational Fluid Dynamics. Prentice Hall, 2007.
 - F. Moukalled, L. Mangani, M. Darwish. The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction With OpenFOAM and Matlab. Springer-Verlag, 2015.
 - J. H. Ferziger, M. Peric, R. Street. Computational Methods for Fluid Dynamics. Springer-Verlag, 2020.
 - Ansys Fluent Help system.

On the CFL number

Quick review of solution methods in CFD

- Before reviewing how to control unsteady and steady simulations, let us address the **CFL number**.
- So, what is the **CFL number** or **Courant number**?
- In one dimension, the **CFL number** is defined as,

$$CFL = \frac{u \Delta t}{\Delta x}$$

- The **CFL number** is a measure of how much information (u) traverses a computational grid cell (Δx) in a given time-step (Δt).
- The **CFL number** is a very important metric in CFD.

Quick review of solution methods in CFD

- Similarly, we can define the **CFL number condition**, which is related to the **CFL number**.
- For the **N** dimensional case, the **CFL number condition** becomes,

$$CFL = \Delta t \sum_{i=1}^n \frac{u_i}{\Delta x_i} \leq CFL_{max}$$



Maximum CFL number allowed
by the numerical method

Quick review of solution methods in CFD

- The **CFL number** is a necessary condition to guarantee the stability of the numerical scheme.
- But not all numerical schemes have the same stability requirements or CFL_{\max} requirement (maximum allowable **CFL number**).
- Implicit numerical methods are **unconditionally stable**.
 - These are the preferred methods.
- In other words, implicit methods are not constrained to the **CFL number condition** or maximum allowable **CFL number**.
- However, the fact that implicit methods are unconditionally stable does not mean that we can use a time step of any size.
- The time-step must be chosen in such a way that it resolves the time-dependent features, and it maintains the solver stability.

Quick review of solution methods in CFD

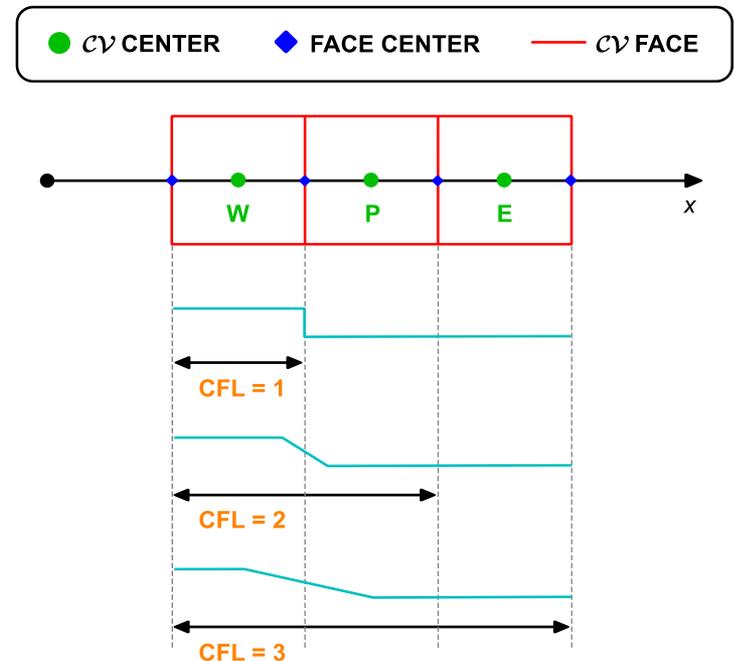
- On the other hand, explicit methods are **conditionally stable**.
- They have a constraint on the maximum allowable CFL number (**CFL number condition**).
- If you choose a **CFL number** larger than the maximum allowable by the explicit method, your numerical solution will become unstable, and it will diverge.
- Some explicit methods have a **CFL condition** of 0.5, and some of them can go up to 2.0.
- But usually, the maximum allowable **CFL number** is limited to 1.0.
- Additionally, the time step is an order of magnitude smaller than that required in the implicit formulation using the same **CFL number**.
- On the positive side, explicit methods are extremely fast, memory efficient, easy to parallelize, and very accurate.

Quick review of solution methods in CFD

I like to see the **CFL number** as follows,

$$CFL = \frac{u \Delta t}{\Delta x} = \frac{u}{\Delta x / \Delta t} = \frac{\text{speed of the PDE}}{\text{speed of the mesh}}$$

It is an indication of the amount of information that propagates through one cell (or many cells), in one time-step.

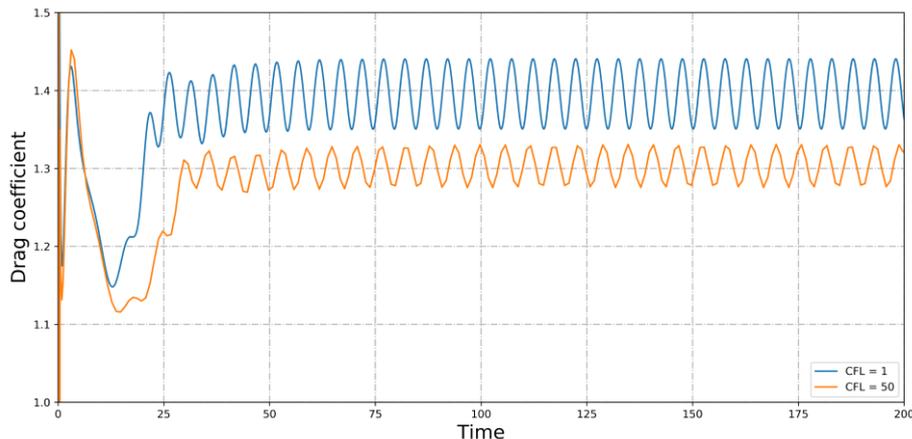


- The **CFL condition** is a necessary condition for stability (and hence convergence).
- But it is not always sufficient to guarantee stability.
- Other properties of the discretization schemes that you should observe are: conservation, boundedness, transportiveness, and accuracy.
- The **CFL number** is not a magical number. There is a whole stability analysis theory behind it.

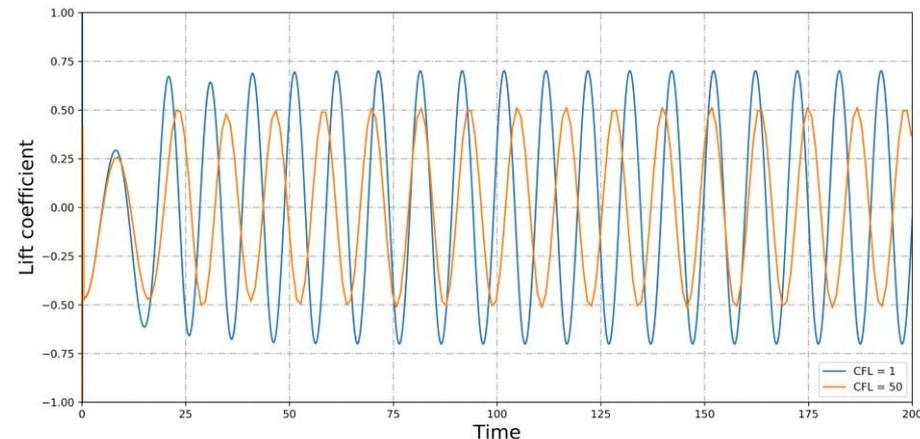
Quick review of solution methods in CFD

How large can the CFL number be?

- With implicit solvers, it can be as large as you like.
- As a rule, the CFL number used in the simulation should be a compromise between accuracy, stability, and solution speed.
 - Up to a CFL number of ten, the solution can be deemed accurate.
 - But the ideal CFL number is one.
- In the figures below, we illustrate a solution running using a low and high CFL number.
- As you increase the CFL number, you start to lose accuracy.



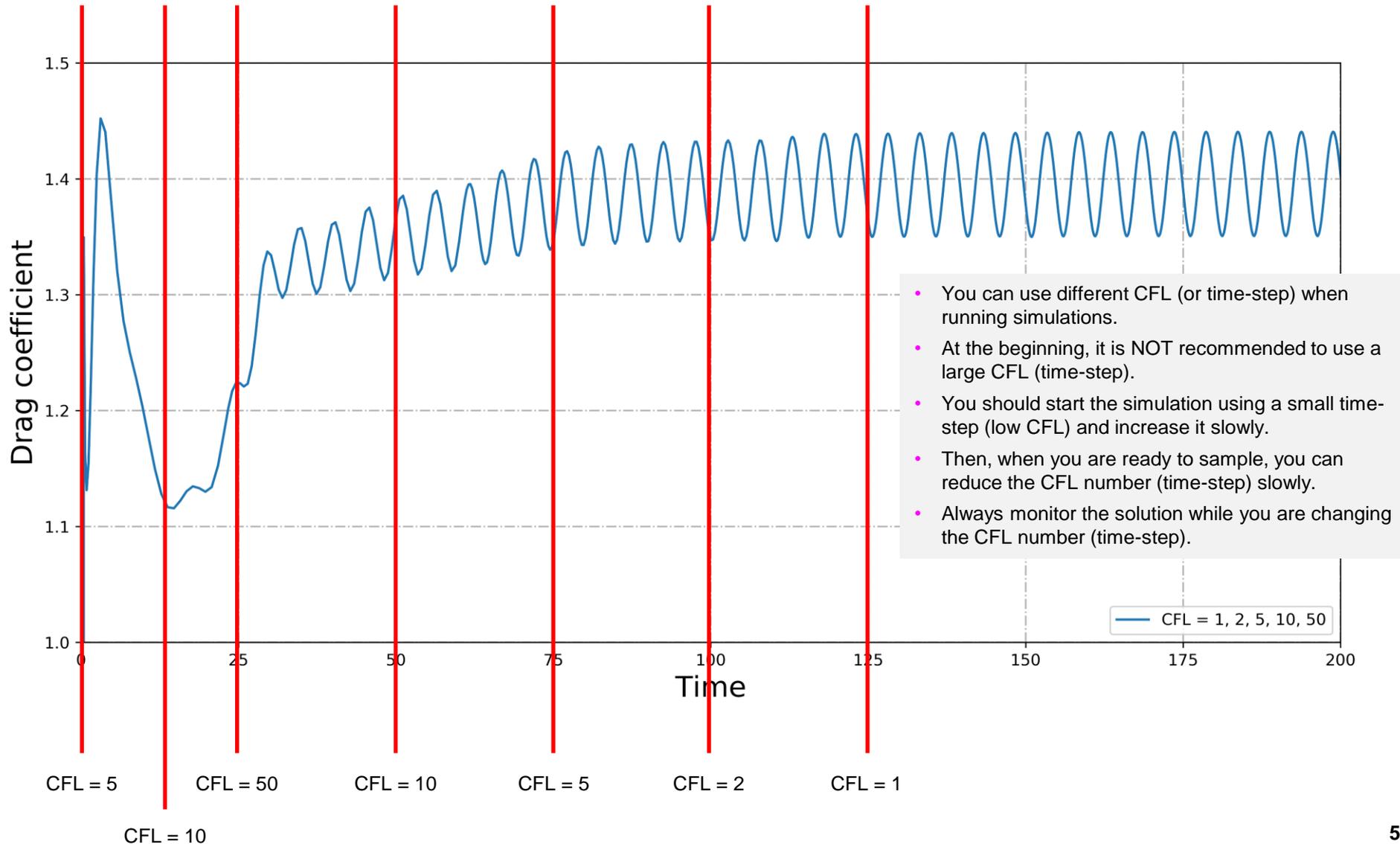
Drag coefficient time series
CFL = 1 and CFL = 50



Lift coefficient time series
CFL = 1 and CFL = 50

Quick review of solution methods in CFD

How large can the CFL number be?



Controlling unsteady and steady simulations

Quick review of solution methods in CFD

How to control unsteady simulations using a general CFD solver?

- In unsteady simulations, as we are solving the time derivative, we need to look also at the time domain.
- Therefore, to control unsteady simulation you need to,
 - Select the time discretization scheme.
 - Set the time step.
 - The time-step must be chosen in such a way that it resolves the time-dependent features and maintains solver stability.
 - End time of the simulation?, it is up to you.
 - But usually, you will stop when,
 - You get a statistically steady or periodic behavior of a monitored quantity of interest.
 - When you have gathered enough unsteady statistics.
 - If the monitored quantity of interest stop changing in time.
 - If this happens, you can say that you have arrived at a steady state (this is the exception rather than the rule).

Quick review of solution methods in CFD

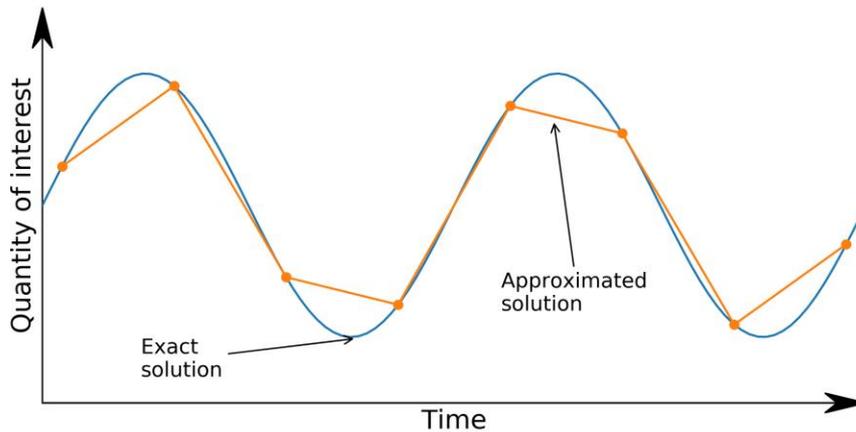
How to control unsteady simulations using a general CFD solver?

- Additionally, you also need to,
 - Monitor the CFL number.
 - Monitor the stability and boundedness of the solution.
 - Reducing the time-step will improve the solution stability at the cost of longer computational times.
 - Monitor one or several quantities of interest.
 - Plot on the fly the contours of a field variable, e.g., pressure, velocity, temperature, concentration, and so on.
 - Monitor the simulation wall clock.
 - Save the solution with a given frequency.
 - Have in mind that unsteady simulations generate a lot of data.

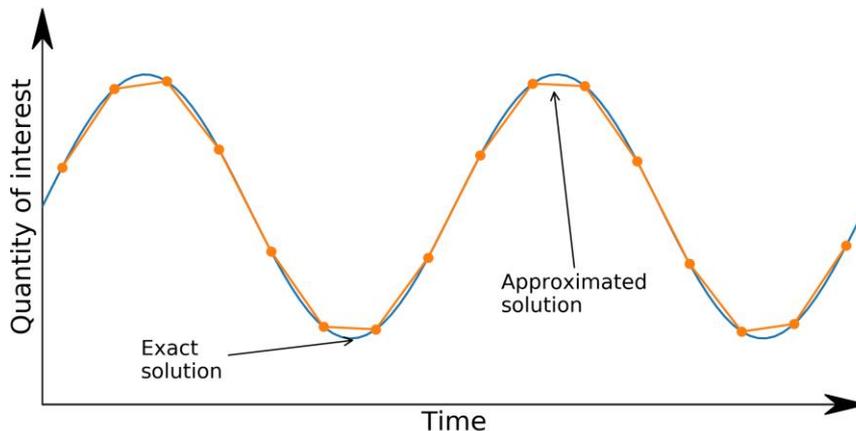
Quick review of solution methods in CFD

How to choose the time-step in unsteady simulations and monitor the solution

- Remember, when running unsteady simulations, the time-step must be chosen in such a way that it resolves the time-dependent features and maintains solver stability.



When you use large time steps you do not resolve well the physics

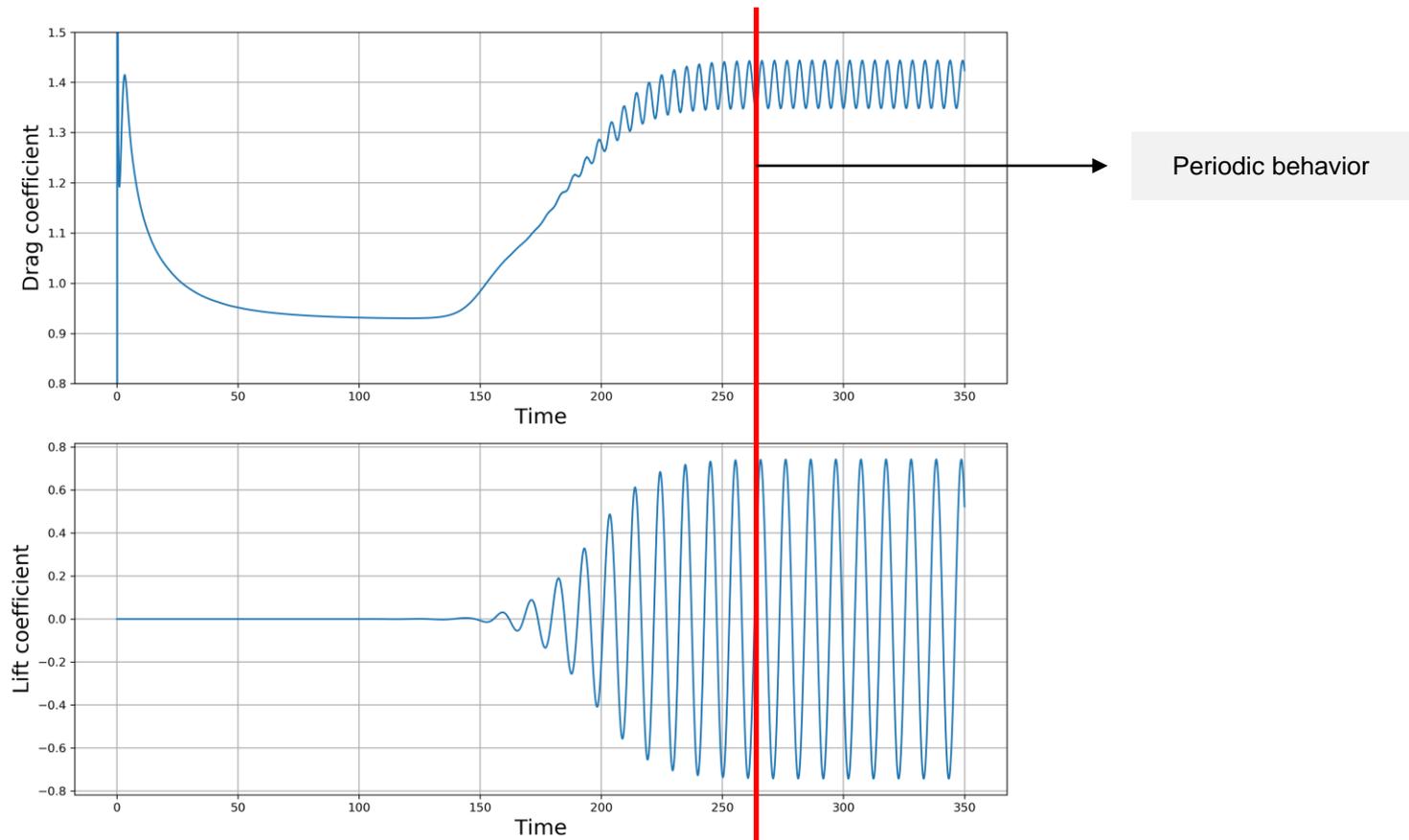


By using a smaller time step you resolve better the physics, and you gain stability

Quick review of solution methods in CFD

Monitoring and sampling unsteady simulations

- When running unsteady simulations, it is highly advisable to monitor a quantity of interest.
- The quantity of interest (QoI) can fluctuate in time, this is an indication of unsteadiness.
- You can stop the simulation if the QoI shows a statistically steady or periodic behavior.



Quick review of solution methods in CFD

How to control steady simulations using a general CFD solver?

- First of all, steady simulations are a big simplification of reality.
- Steady simulations are a trick used by CFDers to get fast outcomes with results that might be very questionable.
- In steady simulations we made two assumptions:
 - We ignore unsteady fluctuations. That is, we neglect the time derivative in the governing equations.
 - We perform time or iterative averaging when dealing with stationary turbulence (RANS modeling)
- The advantage of steady simulations are:
 - They require low computational resources.
 - They give fast outcomes.
 - They are easy to post-process and analyze.
 - We usually take a lot at the last saved solution.

Quick review of solution methods in CFD

How to control steady simulations using a general CFD solver?

- To run steady simulations using a general CFD solver, you need to,
 - Use the appropriate solver or solution method (as not all solution methods can deal with steady simulations) .
 - Set the discretization scheme to deal with a steady simulation.
- Set the residual controls and maximum number of iterations.
 - If you do not set the residual controls, the simulation will run until reaching the maximum number of iterations.
- As there is no time derivative (the time step is infinite), there must be a way to control the iterative marching in steady simulations.
 - This is done by adjusting the under-relaxation factors (URF).
- Under-relaxation works by limiting the amount by which a variable changes from one iteration to the next.
 - Either by modifying the solution matrix and source (implicit under-relaxation) prior to solving for a field
 - Or by modifying the field directly (explicit under-relaxation).

Quick review of solution methods in CFD

Under-relaxation factors

- Under-relaxation works by limiting the amount by which a variable ϕ changes from one iteration to the next,

$$\phi_P^n = \phi_P^{n-1} + \alpha(\phi_P^{n*} - \phi_P^{n-1})$$

- If $\alpha < 1$ we are using under-relaxation.
- Under-relaxation is a feature typical of steady solvers using the **SIMPLE** family of methods.
- Many times, steady simulations diverge because of wrongly chosen URF.
- In CFD, under-relaxation can be implicit or explicit.
- By the way, you can also over-relax but that is not recommended in CFD.

Quick review of solution methods in CFD

Under-relaxation factors

- In explicit under-relaxation we relax the field variable,

$$\phi = \phi_{n-1} + \alpha \Delta \phi$$

- In implicit under-relaxation we relax the discretized algebraic equation variable,

$$\frac{a_P \phi}{\alpha} = \sum_N a_N \phi_N + b + \frac{1 - \alpha}{\alpha} a_P \phi_{n-1}$$

- Choosing the right under-relaxation factors (URF) is equivalent to choosing the right time step.
- You can relate URF to the CFL number as follows,

$$CFL = \frac{\alpha}{1 - \alpha} \qquad \alpha = \frac{CFL}{1 + CFL}$$

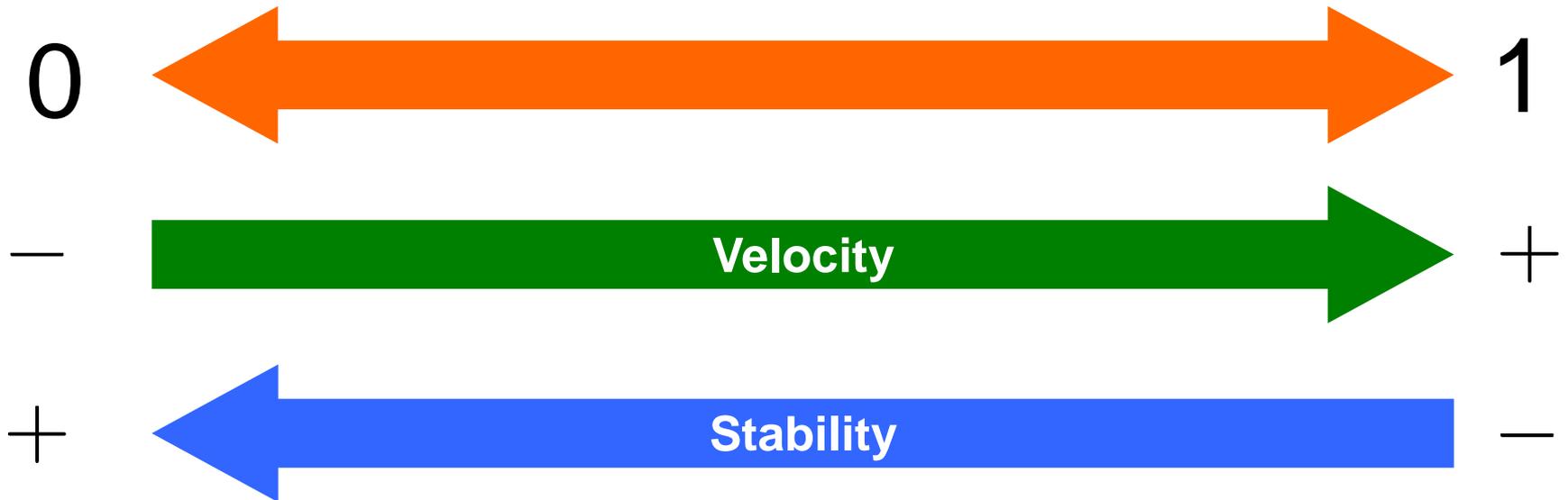
- A large CFL number is equivalent to small URF.

Quick review of solution methods in CFD

Under-relaxation factors

- The under-relaxation factors are bounded between 0 and 1.

Under-relaxation factors URF



- Selecting the under-relaxation factors is kind of equivalent to selecting the right time step.

Quick review of solution methods in CFD

Under-relaxation factors

- Finding the right under-relaxation factors involve experience and a lot of trial and error.
- Choosing the wrong under-relaxation factors can stall the convergence or give you oscillatory/noisy convergence rate (residuals and monitored quantities).
- **Controlling the implicit URF:**
 - It is not recommended to reduce implicit URF to values below 0.5 as it can stalled the convergence rate, add an oscillatory behavior, or slow down too much the convergence rate.
 - If you hit the 0.5 mark when using implicit under-relaxation factors, it is better to stabilize the solution in a different way (increase viscosity, ramp boundary conditions, use upwind, increase corrections, and so on).
- **Controlling the Explicit URF:**
 - Instead, explicit under-relaxation factors can be reduced to as low as 0.1 and still obtain convergence in a reasonable number of iterations.
- It is not recommended to use URF too close to one as it can affect the stability of the solution.
- It is recommended to use the values mentioned in literature, referred to as industry standard.

Quick review of solution methods in CFD

Under-relaxation factors

- An optimum choice of under-relaxation factors is one that is small enough to ensure stable computation but large enough to move the iterative process forward quickly.
- Different methods (**SIMPLE**, **SIMPLEC**, **SIMPLER**), have different URF requirements.
- These are the under-relaxation factors commonly used with **SIMPLE** and **SIMPLEC** methods (industry standard),

SIMPLE	
p	→ 0.3
U	→ 0.7
energy	→ 0.7
k	→ 0.7
omega	→ 0.7
epsilon	→ 0.7

SIMPLEC	
p	→ 1.0
U	→ 0.9
energy	→ 0.9
k	→ 0.9
omega	→ 0.9
epsilon	→ 0.9

- According to the physics involved you will need to add more under-relaxation factors.
- Finding the right under-relaxation factors involved experience and a lot of trial and error.

Quick review of solution methods in CFD

Under-relaxation factors and unsteady solvers

- It is also possible to use under-relaxation factors with unsteady solvers.
 - URF helps improving the stability of unsteady solvers.
- You should be careful not to use too low URF with unsteady solvers because you might lose time accuracy.
- You can safely use large URF (close to one) or the industry standard URF with unsteady solvers.
- If you use low values (less than 0.5 for all variables), it is recommended to run a time convergence test to determine if you are losing time accuracy.
 - The unsteady solution without URF must match the unsteady solution with URF, otherwise your solution is not time-accurate.

Industry standard URF

SIMPLE

p	→ 0.3
U	→ 0.7
energy	→ 0.7
k	→ 0.7
omega	→ 0.7
epsilon	→ 0.7

SIMPLEC

p	→ 1
U	→ 0.9
energy	→ 0.9
k	→ 0.9
omega	→ 0.9
epsilon	→ 0.9

Recommended URF

SIMPLE – SIMPLEC – PISO

p	→ 0.7 (0.3 IN SIMPLE)
U	→ 0.7
energy	→ 0.7
k	→ 0.7
omega	→ 0.7
epsilon	→ 0.7