Unsteady and steady simulations

- The integral form of the general transport equation is the cornerstone of the finite volume 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 the general transport equation we can write down the Navier-Stokes equations (NSE). For example, by setting the variables to,

$$\begin{aligned} \phi &= 1 \\ \Gamma_\phi &= 0 \\ S_\phi &= 0 \end{aligned}$$

- We can obtain the continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

Unsteady and steady simulations

- The integral form of the general transport equation is the cornerstone of the finite volume 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 the general transport equation we can write down the Navier-Stokes equations (NSE). For example, by setting the variables to,

$$\phi = u \qquad \Gamma_\phi = \mu \qquad S_\phi = S_u - \frac{\partial p}{\partial x}$$

$$\phi = v \qquad \Gamma_\phi = \mu \qquad S_\phi = S_v - \frac{\partial p}{\partial y}$$

$$\phi = w \qquad \Gamma_\phi = \mu \qquad S_\phi = S_w - \frac{\partial p}{\partial z}$$

- We can obtain the momentum equations,

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho \mathbf{u} u) = \nabla \cdot (\mu \nabla u) - \frac{\partial p}{\partial x} + S_u$$

$$\frac{\partial \rho v}{\partial t} + \nabla \cdot (\rho \mathbf{u} v) = \nabla \cdot (\mu \nabla v) - \frac{\partial p}{\partial y} + S_v$$

$$\frac{\partial \rho w}{\partial t} + \nabla \cdot (\rho \mathbf{u} w) = \nabla \cdot (\mu \nabla w) - \frac{\partial p}{\partial z} + S_w$$

Unsteady and steady simulations

- The integral form of the general transport equation is the cornerstone of the finite volume 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 the general transport equation we can write down the Navier-Stokes equations (NSE). For example, by setting the variables to,

$$\begin{aligned} \phi &= h \\ \Gamma_\phi &= k/C_p \\ S_\phi &= S_h \end{aligned}$$

- We can obtain the momentum equations,

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) = \nabla \cdot \left(\frac{k}{C_p} \nabla T \right) + S_h$$

Unsteady and steady simulations

- The integral form of the general transport equation is the cornerstone of the finite volume method.

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

- We want to solve the general transport equation for the transported quantity ϕ in a given domain, with given boundary conditions BC and initial conditions IC.
- The general transport equation is a second order equation.
- 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.
- By the way, the Navier-Stokes equations (NSE) are also second order.
- After all, they can be derived from the general transport equation.

Unsteady and steady simulations

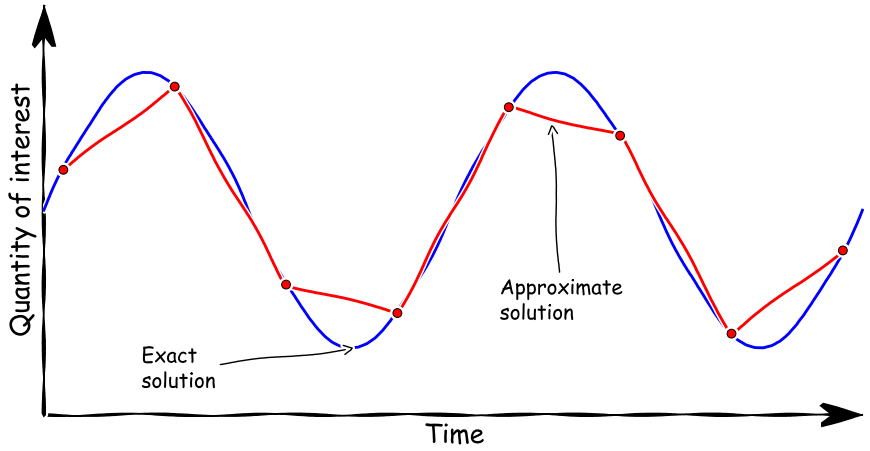
How to run unsteady simulations using a general CFD solver?

- Select the temporal discretization scheme.
- Set the time step.
- Remember, the time-step must be chosen in such a way that it resolves the time-dependent features and maintains solver stability.
- Set the tolerance (absolute and/or relative) of the linear solvers. However, most of the times the default parameters are fine.
- Monitor the CFL number.
- Monitor the stability and boundedness of the solution.
- Monitor integral quantities, e.g., lift, drag, velocity at a point, average temperature at the outlet, and so on.
- And of course, you need to save the solution with a given frequency.
- Have in mind that unsteady simulations generate a lot of data.
- End time of the simulation?, it is up to you.

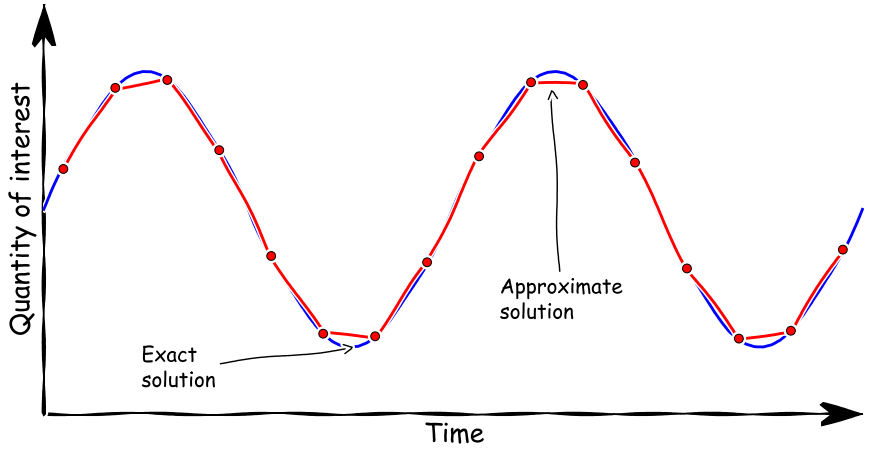
Unsteady and steady simulations

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

- 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

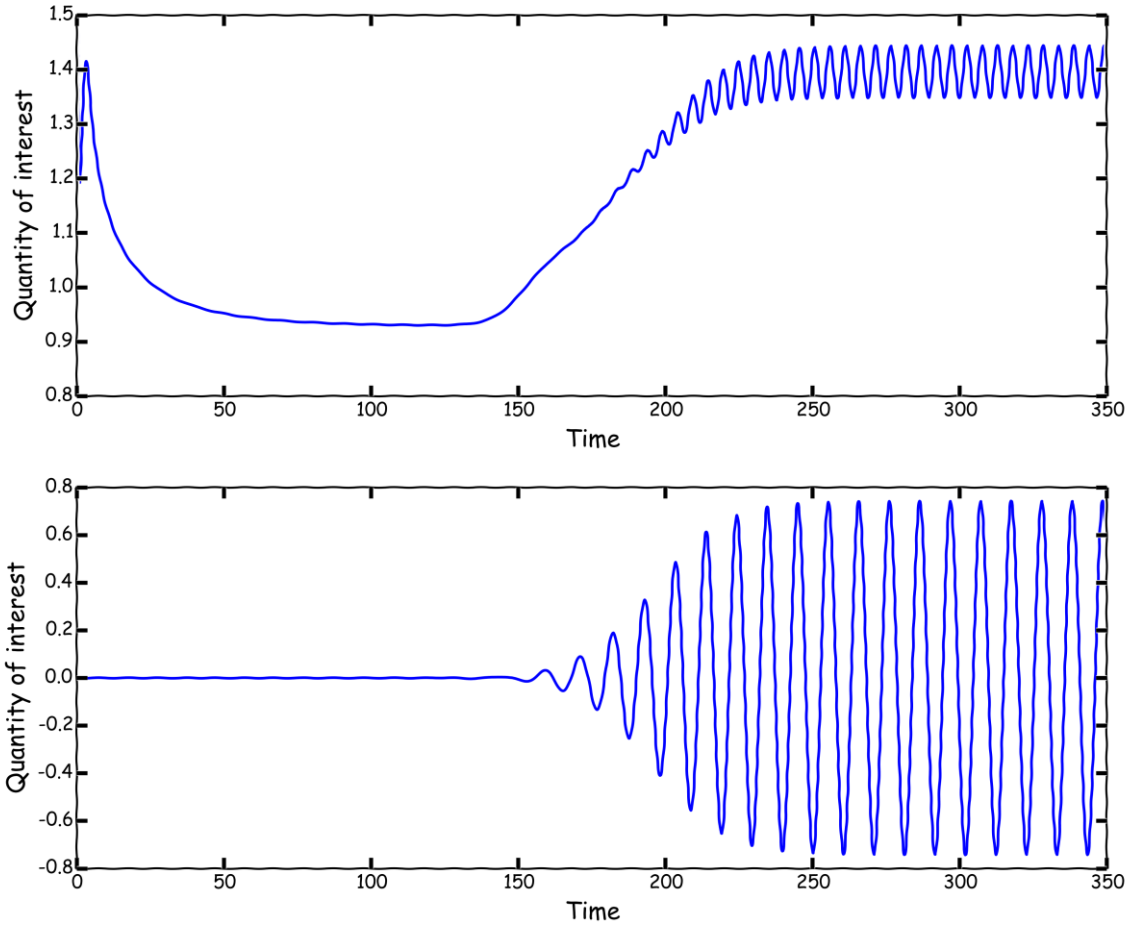


By using a smaller time step you resolve better the physics and you gain stability

Unsteady and steady simulations

Monitoring and sampling unsteady simulations

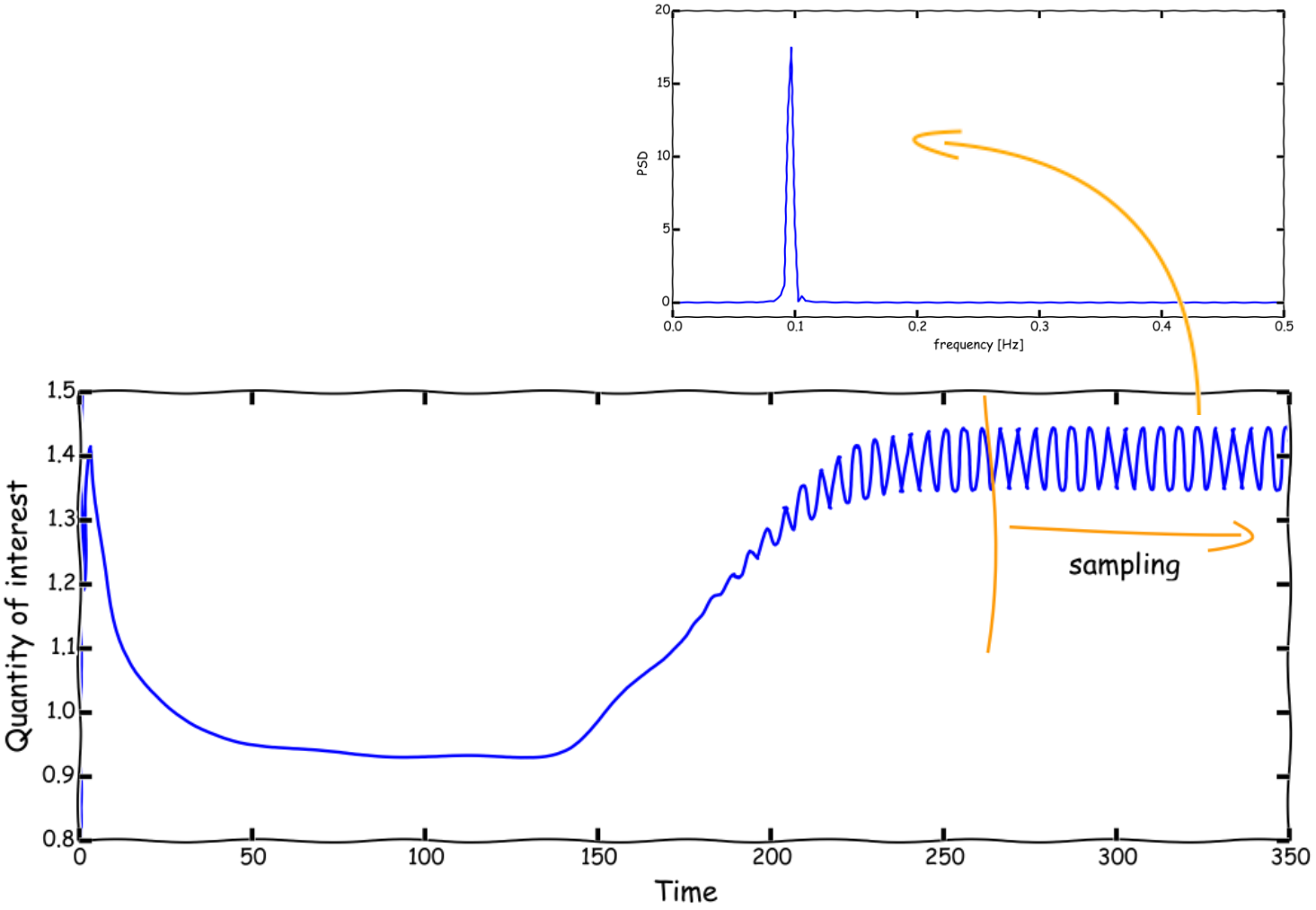
- When running unsteady simulations, it is highly advisable to monitor a quantity of interest.
- The quantity of interest can fluctuate in time, this is an indication of unsteadiness.



Unsteady and steady simulations

Monitoring and sampling unsteady simulations

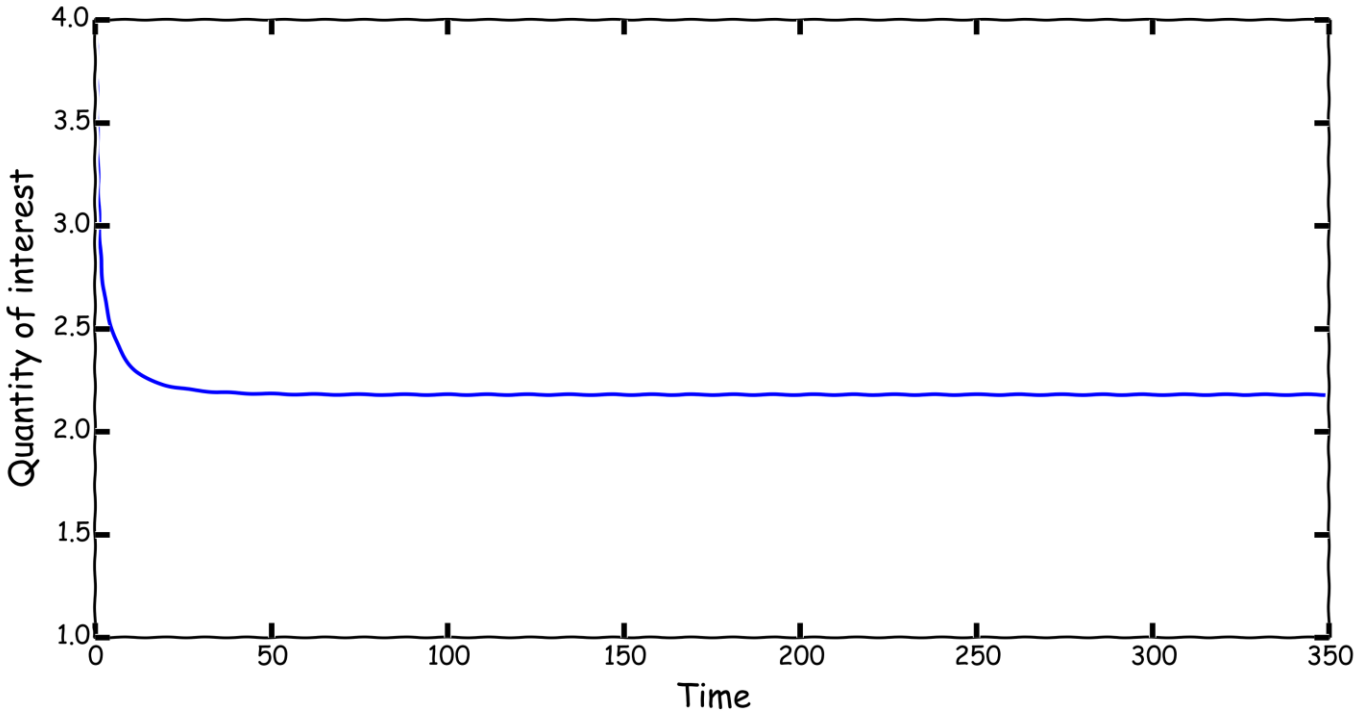
- Remember to choose wisely where to do the sampling.



Unsteady and steady simulations

I am running an unsteady simulations and the QOI does not change

- When you run unsteady simulations, flow variables can stop changing with time. When this happens, we say we have arrived at a steady state.
- Remember, this is the exception rather than the rule.
- If you use a steady solver, you will arrive to the same solution (maybe not), in much less iterations.



Unsteady and steady simulations

What about steady simulations?

- First of all, steady simulations are a big simplification of reality.
- Steady simulations is a trick used by CFDers to get fast outcomes with results that might be very questionable.
- Most of the flows you will encounter in industrial applications are unsteady.
- In steady simulations, we made two assumptions:
    - We ignore unsteady fluctuations. That is, we neglect the temporal 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 look at the last saved solution.

What about steady simulations?

- To run steady simulations using a general CFD solver, you need to use the appropriate solver and set the discretization scheme to deal with a steady simulation.
- As you are not solving the temporal derivative, you do not need to set the time step.
- However, you need to tell to the CFD solver how many iterations you would like to perform.
- You can also set the residual controls. If you do not set the residual controls, the simulation will run until reaching the maximum number of iterations.
- Additionally, you will need to set the under-relaxation factors.
- Under-relaxation works by limiting the amount 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).
- Under-relaxation will make the coefficient matrix more diagonally dominant.

Unsteady and steady simulations

What about steady simulations?

- Under-relaxation factors (URF), work in the following way.
- URFs control the change of the variable ϕ ,

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

- 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 implicit or explicit.

Unsteady and steady simulations

What about steady simulations?

- 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,

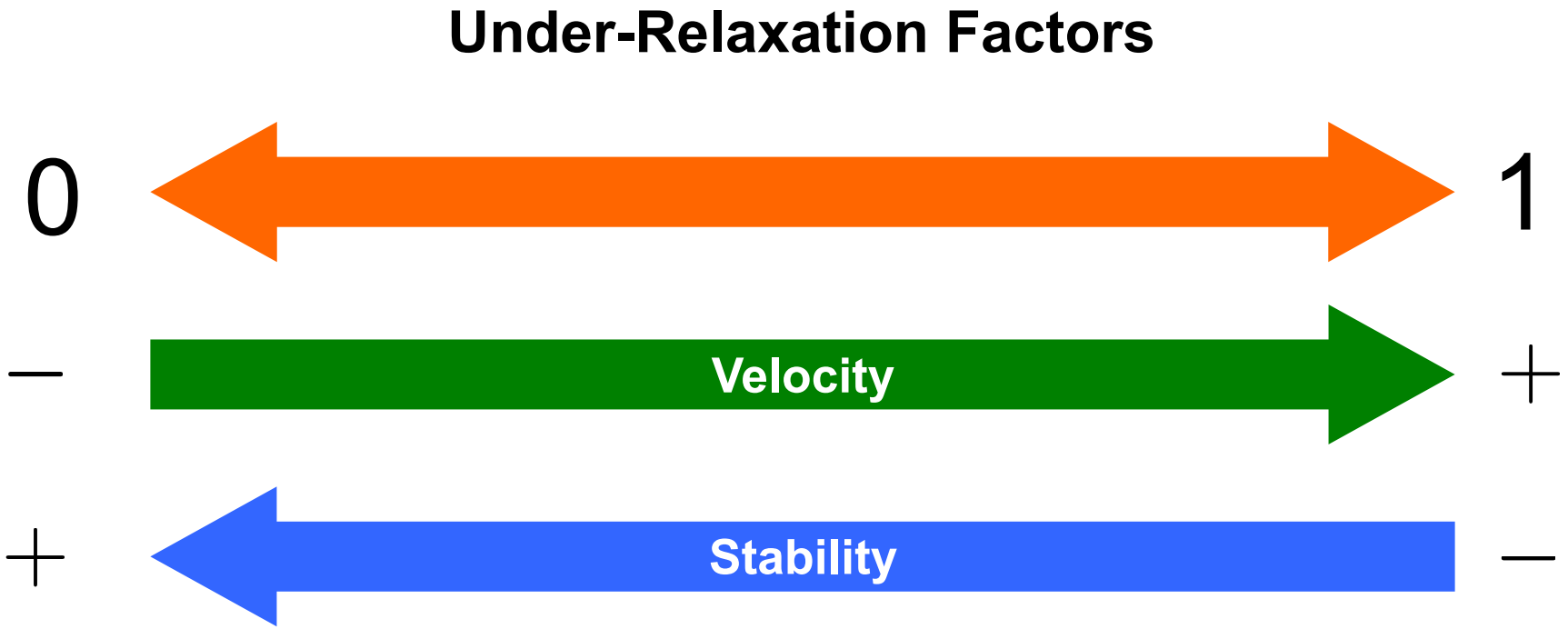
$$CFL = \frac{\alpha}{1-\alpha} \qquad \alpha = \frac{CFL}{1+CFL}$$

- A large CFL number is equivalent to small URF.

Unsteady and steady simulations

What about steady simulations?

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



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

What about steady simulations?

- Finding the right under-relaxation factors involved 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).
- Generally speaking, is not recommended to reduce implicit under-relaxation factors to values below 0.5 as it can stalled the convergence rate, add an oscillatory behavior or slow down the convergence rate.
- If you reach 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).
- 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 recommended to use the values mentioned in literature (referred to as industry standard).

Unsteady and steady simulations

What about steady simulations?

- 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, PISO**), 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**
k	**→ 0.7**
omega	**→ 0.7**
epsilon	**→ 0.7**

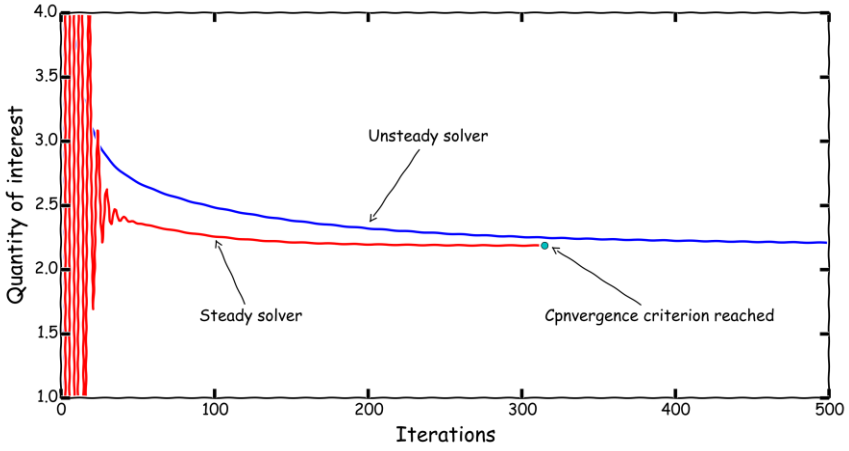
SIMPLEC	
p	**→ 1.0**
U	**→ 0.9**
k	**→ 0.9**
omega	**→ 0.9**
epsilon	**→ 0.9**

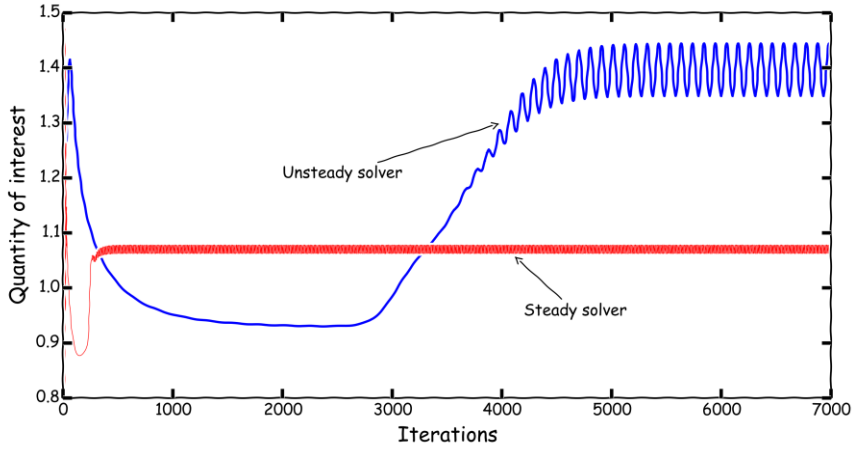
- According to the physics involved you will need to add more under-relaxation factors (density, energy, species, and so on)..
- Finding the right under-relaxation factors involved experience and a lot of trial and error.

Unsteady and steady simulations

Steady simulations vs. Unsteady simulations

- Steady simulations require less computational power than unsteady simulations.
- They are also much faster than unsteady simulations.
- But sometimes they do not converge to the right solution.
- They are easier to post-process and analyze (you just need to take a look at the last saved solution).
- You can use the solution of an unconverged steady simulation as initial conditions for an unsteady simulation.
- Remember, steady simulations are not time accurate, therefore we can not use them to compute temporal statistics or compute the shedding frequency





Unsteady and steady simulations

Under-relaxation factors and unsteady solvers

- It is also possible to use under-relaxation factors with unsteady solvers.
- You should be careful not to use too low URF with unsteady solvers because you might loose temporal accuracy.
- You can 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 temporal convergence test to determine if you are loosing time accuracy.
- The unsteady solution without URF must match the unsteady solution with URF, otherwise your solution is not time-accurate.
- When you use URF with unsteady solvers you increase the diagonal dominance of the linear system. Therefore, they improve the stability of unsteady solvers.

Industry standard URF

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

SIMPLEC	
p	→ 1
U	→ 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
k	→ 0.7
omega	→ 0.7
epsilon	→ 0.7

Note: use these guidelines with unsteady solvers

Unsteady and steady simulations

Linear system solution

- Reducing the time-step or changing the under-relaxation factors will make the coefficient matrix more diagonally dominant.
- In CFD, it is extremely important that the matrix **A** is diagonally dominant.
- A matrix is diagonally dominant if in each row the sum of the off-diagonal coefficient magnitude is equal or smaller than the diagonal coefficient,

$$a_{ii} \geq \sum_{j=1}^{N} |a_{ij}| \quad \Rightarrow \quad j \neq i$$

- And at least one *i*,

$$a_{ii} \geq \sum_{j=1}^{N} |a_{ij}| \quad \Rightarrow \quad j \neq i$$

- Diagonal dominance is a very desirable feature for satisfying the boundedness criterion.

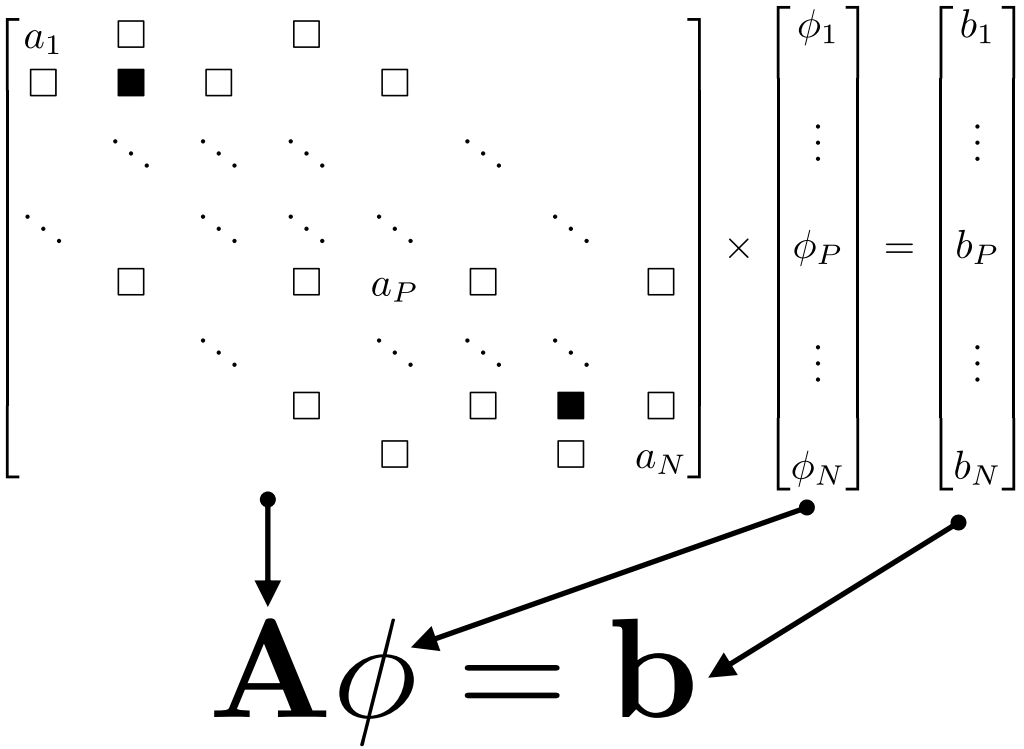
Unsteady and steady simulations

Linear system solution

- Diagonal dominance is a very desirable feature for satisfying the boundedness criterion.
- To achieve diagonal dominance we need large values of net coefficient (coefficients of the diagonal).
- This can be controlled by using under-relaxation, reducing the time-step, by assuring that any source term in the RHS is negative, and by having good quality meshes.
- If a matrix is diagonally dominant, it also satisfy the Scarborough criterion.

■ = Diagonal contribution

□ = Off-diagonal contribution



Unsteady and steady simulations

Linear system solution

- If a matrix is diagonally dominant, it also satisfy the Scarborough criterion [1].

$$\frac{\sum |a_{nb}|}{|a_p|} \begin{cases} \leq 1, \text{for all equations} \\ < 1, \text{for at least one equation} \end{cases}$$

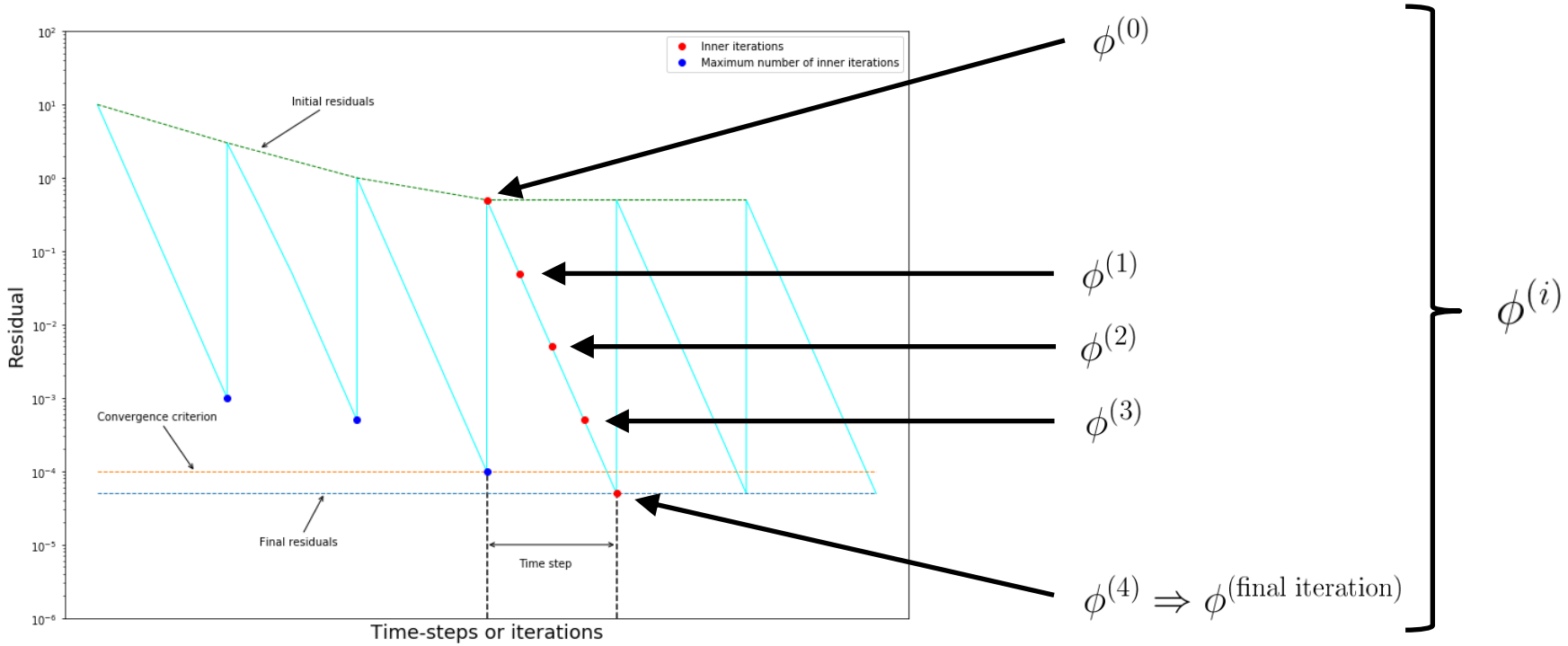
- The satisfaction of this criterion ensures that the equations will converge by at least one iterative method.
- This is a sufficient condition, not a necessary one. This means that we can get convergence, even if, at times, we violate this criterion.
- The finite volume method uses this criterion to set some basic discretization rules related to obtaining a convergent solution, implementing boundary conditions, and adding source terms.
    - When linearizing the source terms they must be negative, so when they are added to a_p in the LHS, they help increasing the diagonal dominance.
    - All coefficients in the LHS and RHS of the linear system should have the same sign (essential requirement for boundedness).
    - If the boundedness requirement is not satisfied, it is possible that the solution does not converge at all, or if it does, the solution is oscillatory (contains wiggles).

[1] James Blaine Scarborough (1958). Numerical Mathematical Analysis. Johns Hopkins Press.

Unsteady and steady simulations

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.

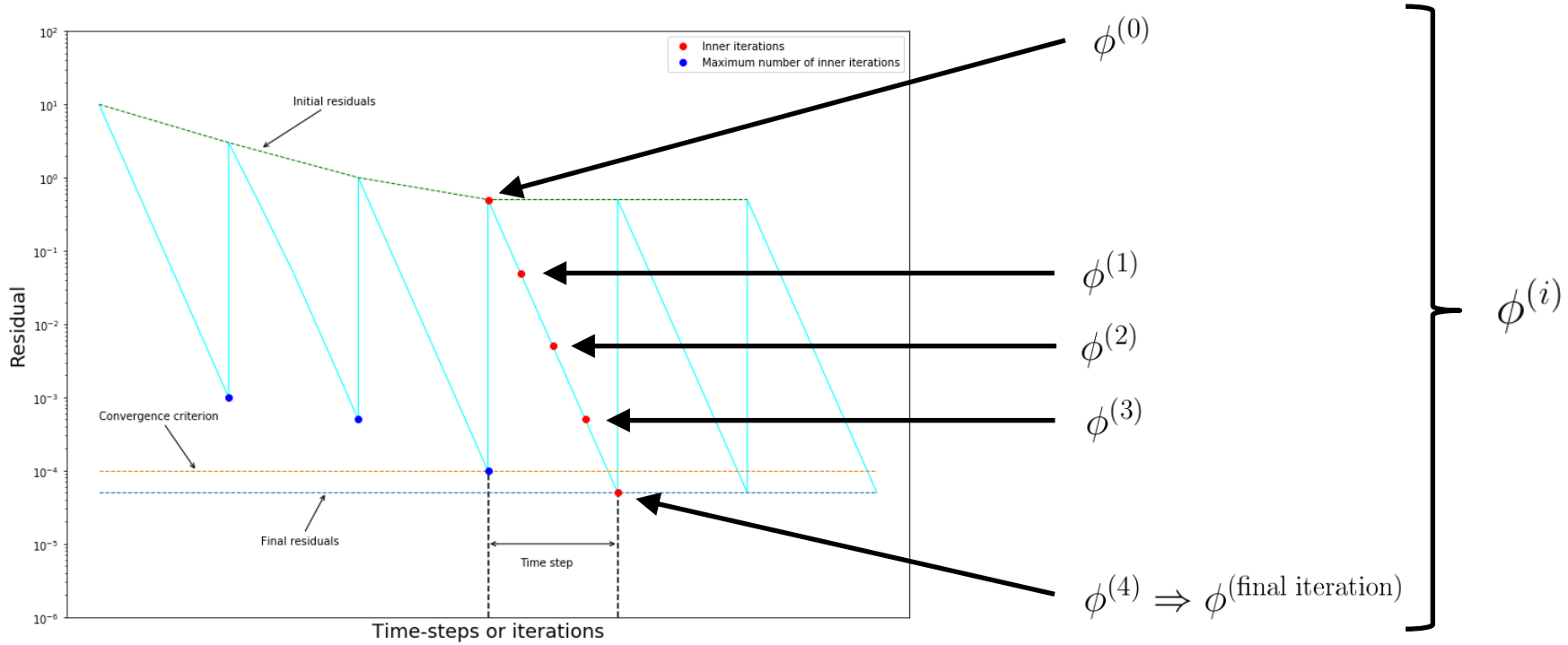


- $\phi^{(0)}$ is the initial guess used to start the iterative method.
- Iteration 0 defines the initial residual, and greatly influence the convergence rate.
- You can use any value at iteration 0, but usually is a good choice to take the previous solution vector.
- Remember, the closest you are to the actual solution, the faster the convergence rate will be.

Unsteady and steady simulations

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.

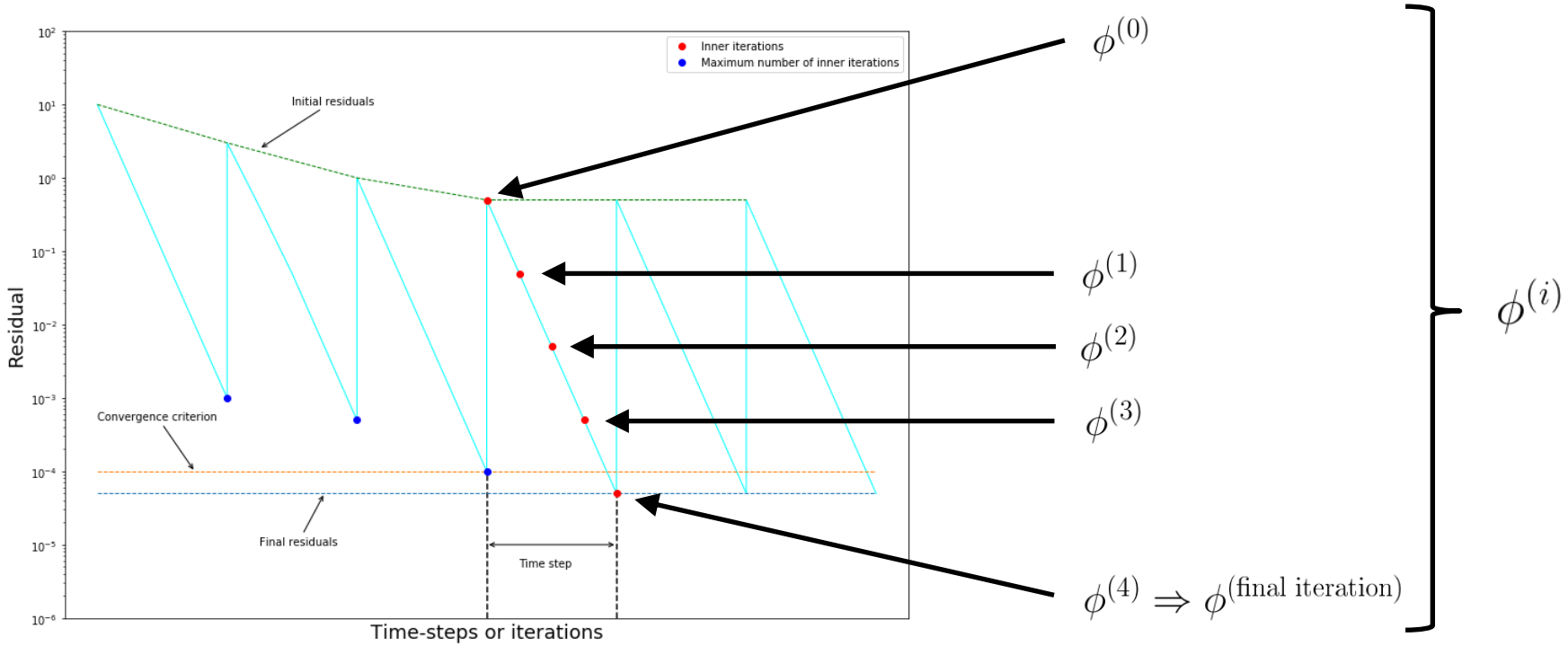
$$\left|\mathbf{A}\phi^{i} - \mathbf{b}\right| \leq |\mathbf{r}|$$

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

Unsteady and steady simulations

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.

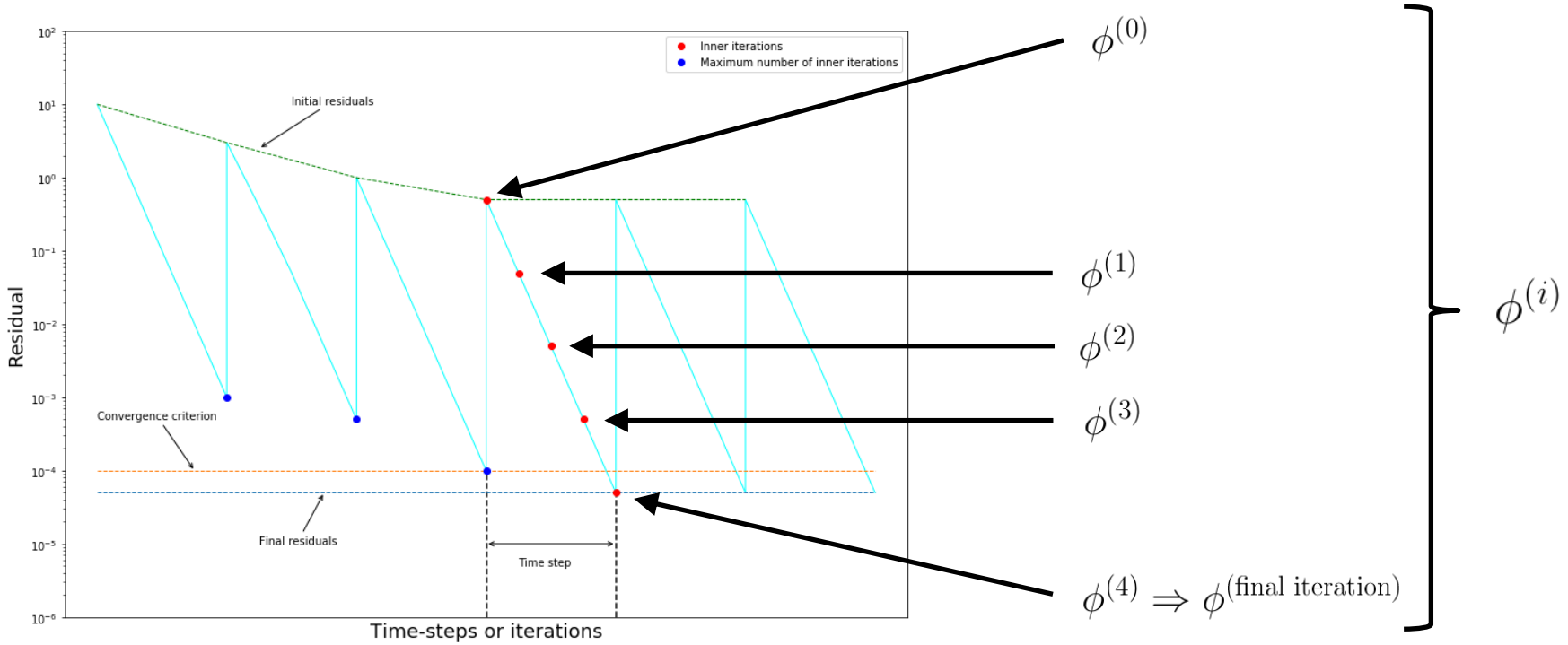


- By working in an iterative way, every single iteration $\phi^{(i)}$ is a better approximation of the previous iteration $\phi^{(i-1)}$
- Sometimes the linear solver might stop iterating because it has reached the maximum number of iterations, you should be careful of this because we are talking of unconverged iterations.
- Also, it is recommended to do at least one iteration as it helps at linearizing the equations.

Unsteady and steady simulations

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.



- 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. Also, depending on the matrix type (symmetric or asymmetric), they might have different convergence rates.