RWTH AACHEN UNIVERSITY

CES Seminar Write-up

Application of Chimera Grids in Rotational Flow

Marc Schwalbach
292414
marc.schwalbach@rwth-aachen.de

Supervisors: Dr. Anil Nemili & Emre Özkaya, M.Sc.
MATHCCES
RWTH Aachen University
July 15, 2014

Table of Contents

1 Introduction

Today's scope of applications in computational fluid dynamics is continuously increasing and, as a result, the complexity of these problems increases as well. This leads to the challenging task of creating suitable grids for increasingly complex geometries which may be constituted of multiple parts, e.g. an aircraft model made up of fuselage, rudder, wings, engines, etc.. Creating a single strucutured body-conforming grid for such cases is extremely difficult.

Unstructured grids have proven to be a suitable, simpler alternative for generating grids for such complex problems. However, this method has its caveats as well. The numerical methods for unstructured grids to date are not as effective as those for structured grids, leading to longer runtimes and higher storage requirements. Additionally, generating an exceedingly clustered unstructured near a solid wall, which is required for high Reynolds number flows, is an arduous task and comparatively straightforward with strucutured grids.

An alternative to unstructured grids is the composite overset grid method, also known as the chimera method, which was first introduced by Benek, Dougherty, and Steger [2]. The idea behind chimera grids is to generate multiple simple, overlapping subgrids rather than one single complex grid. This leads to a more natural grid generation process, in particular for complex geometries with multiple parts. Each subdomain can have different grid properties, e.g. curvilinear or cartesian, and solves its own set of equations, e.g. viscous or inviscid Euler equations, which gives one more control over location-dependent modelling. The challenge with this method is the treatment of the interfaces between two subgrids, especially because global mass conservation must be ensured.

An interesting application is in the field of rotational flows, e.g. turbomachinery, wind turbines, or helicopters, where one can use a chimera grid to define a rotating subdomain within a stationary domain or vice versa. As an example, the rotating subdomain would solve the Navier-Stokes equations for a rotating reference frame, while the stationary subdomain solves the Navier-Stokes equations for an inertial reference frame. This is especially useful if the geometry involves multiple parts which could either be stationary or rotating.

The main focus of this write-up, which accompanies the CES Seminar presentation, is the chimera grids method and its application on rotational flows. First, the governing equations for stationary and rotating grids will be introduced. Then, the chimera grids method will be discussed in further detail, especially focusing on its implementation and the treatment of the interface boundaries. Lastly, the chimera grids method will be summarized followed by a conclusion.

2 Governing Equations

The Navier-Stokes equations describe the physical properties, i.e. mass, momentum, and energy, of a fluid. In this work, the simplified incompressible Navier-Stokes equations are considered. They will first be introduced for inertial reference frames, followed by their transformation to rotating reference frames. The formulations in this section can also be taken from [1] and [4].

2.1 Incompressible Navier-Stokes Equations for Inertial Reference Frames

The incompressible Navier-Stokes equations for an inertial reference frame are given by:

$$\nabla \cdot \vec{v} = 0 \tag{2.1}$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla)\, \vec{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{v} \tag{2.2}$$

Equation (2.1) describes mass conservation and equation (2.2) describes momentum conservation. The energy conservation equation is decoupled from the mass and momentum equations for the incompressible flow case. Thus, it is not included here.

2.2 Incompressible Navier-Stokes Equations for Rotating Reference Frames

For the transformation of the Navier-Stokes equations from an inertial to a rotating reference frame, consider first the time derivative of an arbitrary vector $\vec{A}$:

$$\frac{d\vec{A}}{dt} = \frac{dA_i}{dt}\hat{x}_i + A_i \frac{d\hat{x}_i}{dt} \tag{2.3}$$

with a basis unit vector $\hat{x}$ and its time derivative

$$\frac{d}{dt}\hat{x}\,(t) = \Omega \times \hat{x}. \tag{2.4}$$

Equations (2.3) and (2.4) lead us to

$$\left(\frac{d\vec{A}}{dt}\right)_f = \left(\frac{d\vec{A}}{dt}\right)_r + \Omega \times \vec{A} = \left[\left(\frac{d}{dt}\right)_r + \Omega\times\right] \vec{A} \tag{2.5}$$

with the subscripts f and r denoting the fixed (inertial) and rotating reference frames, respectively. Using this equation, the relationship between fixed and rotating reference frames for the velocity is

$$\vec{v}_f = \left(\frac{d\vec{r}}{dt}\right)_f = \left(\frac{d\vec{r}}{dt}\right)_r + \Omega \times \vec{r} \tag{2.6}$$

and for acceleration:

$$\vec{a}_f = \left(\frac{d^2\vec{r}}{dt^2}\right)_f = \left[\left(\frac{d}{dt}\right)_r + \Omega\times\right]\left[\left(\frac{dr}{dt}\right)_r + \Omega\times\vec{r}\right]$$

$$\vec{a}_f = \vec{a}_r \underbrace{+2\Omega\times\vec{v}_r}_{\text{Coriolis}} \underbrace{+\Omega\times(\Omega\times\vec{r})}_{\text{Centrifugal}} \underbrace{+\frac{d\Omega}{dt}\times\vec{r}}_{\text{Euler}} \tag{2.7}$$

In this work, the rotational velocity Ω is assumed to be constant, which causes the *Euler* term in equation (2.7) to vanish. Euler's first law of motion for a rotating reference together with equation (2.7) delivers

$$\begin{aligned}\vec{F} &= \int_V \vec{a}_r \rho \mathrm{d}V \\ &= \underbrace{\int_V \vec{a}_f \rho \mathrm{d}V}_{\vec{F}_f} + \int_V -\left(2\Omega\times\vec{v}_r + \Omega\times(\Omega\times\vec{r})\right)\rho\mathrm{d}V.\end{aligned} \tag{2.8}$$

Recall the momentum equation for the inertial reference frame (2.2). The right-hand side of this equation is $\vec{F}_f$ of equation (2.8). Plugging this in results in the momentum conservation equation for rotating reference frames

$$\frac{\partial\vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\,\vec{v} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\vec{v} - \underbrace{2\Omega\times\vec{v}}_{\text{Coriolis}} - \underbrace{\Omega\times(\Omega\times\vec{r})}_{\text{Centrifugal}}. \tag{2.9}$$

With the vector identity

$$A\times(B\times C) = (A\cdot C)\,B - (A\cdot B)\,C \tag{2.10}$$

the centrifugal term can be further simplified to

$$\Omega\times(\Omega\times\vec{r}) = -(\Omega\cdot\Omega)\,\vec{r}_\perp = -\Omega^2\vec{r}_\perp. \tag{2.11}$$

Finally, the incompressible Navier-Stokes equations in a rotating reference frame are given as

$$\begin{aligned}\nabla\cdot\vec{v} &= 0 \\ \frac{\partial\vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\,\vec{v} &= -\frac{1}{\rho}\nabla p_{eff} + \nu\nabla^2\vec{v} - 2\Omega\times\vec{v}\end{aligned} \tag{2.12}$$

with the *effective pressure*

$$p_{eff} = p - \frac{1}{2}\rho\Omega^2 r_\perp^2. \tag{2.13}$$

3 Chimera Grids

The idea behind the chimera grids (also known as overset grids) method is to decompose a single complex grid into multiple, simple , overlapping subdomains. As already mentioned in the introduction, the trade-off for the advantages of

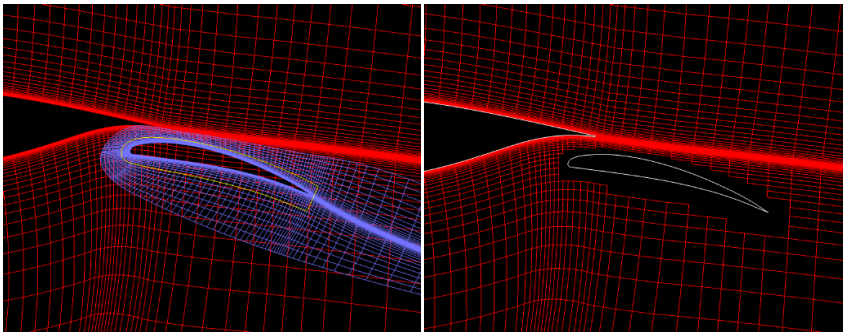
this method is the difficulty of dealing with the overlapping interfaces between subdomains. This section will mainly deal with the treatment of these interfaces. First, the two types of cells that result from the chimera method are introduced. Next, grid connectivity and global mass conservation are discussed. The interface boundary treatment and two possible methods are then considered and finally, the entire method is outlined. The main reference for this section is [3].

3.1 Fringe and Hole Cells

At the overlapping regions of chimera subdomains, two types of cells are defined: fringe cells and hole cells.

Fringe cells refer to the overlapping boundary cells of a subdomain. For example, in figure (3.1a), the fringe cells of the blue subdomain are the two outer boundary cells, since the blue subdomain lies within the red subdomain. The fringe cells of the red subdomain are then those which overlap the outer boundary of the blue subdomain and are used for the interpolation at the interface. These can be identified as the cells bordering the hole left by the flap in figure (3.1b). This directly leads to the other type: hole cells.

Hole cells are cells which are located within a solid body or a region of higher resolution of a subdomain. Computations in hole cells are not required since this region is already handled by a finer subdomain. Thus they can be blanked out by multiplying the respective coefficients by zero. This is referred to as hole cutting, which is illustrated in figure (3.1). Several hole-cutting methods exist, but their discussion is beyond the scope of this work.



(a) (b)

Figure 3.1: Before(a) and after(b) hole cutting. Source: (grc.nasa.gov)

3.2 Grid Connectivity

Before one can start interpolating variables from one interface onto another interface, the node of fringe cells of one domain must be properly paired with so-called host cells from the overlapping domain. Additionally, the coordinates r, s, t of fringe nodes with their respective host cells are required for the interface boundary treatment, which will be discussed in detail further below.

This process is referred to as grid connectivity. In case one deals with stationary grids, this can be entirely done as a pre-processing step.

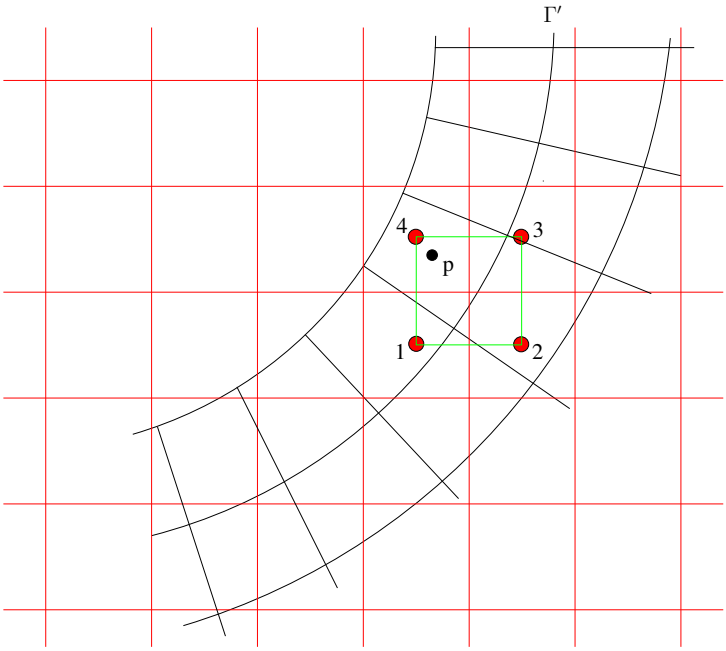


Figure 3.2: Visualization of a fringe node and its host cell

For the follow explanation, consider figure 3.2 which illustrates an interpolation from the red domain to the black domain. p is a node on the outer boundary of the black domain. The question now is: which surrounding red cells should be defined as the hosts of the node p? As one can see in the figure, the red nodes 1, 2, 3, and 4 form the closest surrounding host cell, which is colored in light green, around p. To determine the host cells, the following algorithm is used:

1. Loop over all fringe nodes p
2. For each node p, loop over all potential host cells
3. For each host cell, check if the interface node p is located inside the host cell

Step 3 can be checked by going over all surfaces of each host cell and using the dot product to determine the angular relationship. For example, referring to figure 3.3, the surface spanned by 1-4-7-5 must fulfill the inequality

$$r_{71} \times r_{54} \cdot r_{cp} \geq 0. \tag{3.1}$$

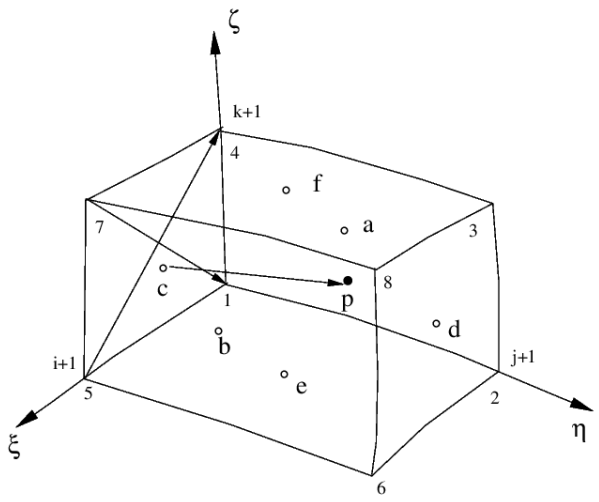


Figure 3.3: Node within a host cell [3]

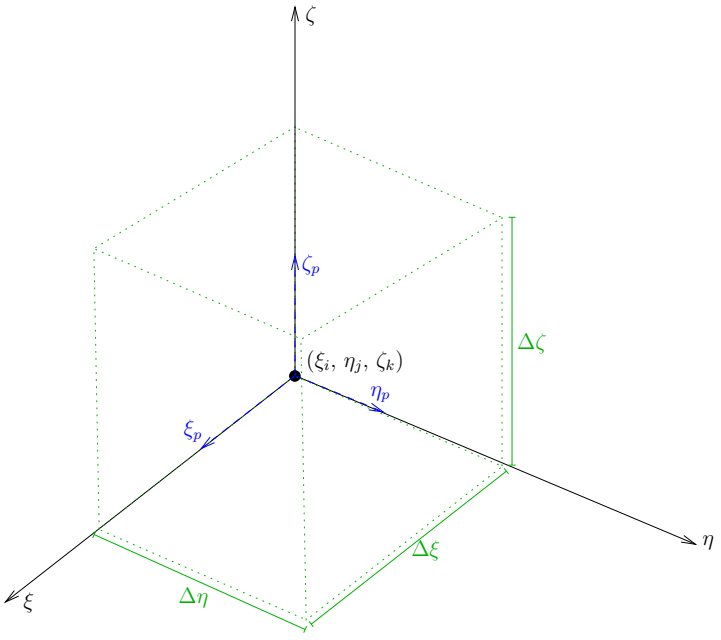


Figure 3.4: Host cell and the coordinates ξ_p, η_p, ζ_p of node p

After the host cells have been determined, the coordinates of the black fringe nodes within their respective host cells (refer to figure 3.4) must be determined. These are defined as

$$r = \frac{\xi_p - \xi_i}{\Delta\xi}, \qquad s = \frac{\eta_p - \eta_j}{\Delta\eta}, \qquad t = \frac{\zeta_p - \zeta_k}{\Delta\zeta} \tag{3.2}$$

and

$$r, s, t \in [0,1] \times [0,1] \times [0,1]\,.$$

Now let f be the phyiscal coordinates x_p, y_p, z_p of node p. Then the trilinear interpolation equation can be used to give a nonlinear system of equations for r, s, t:

$$f = a_1 rst + a_2 rs + a_3 rt + a_4 st + a_5 r + a_6 s + a_7 t + a_8 \tag{3.3}$$

with

$$\begin{aligned}
a_1 &= f_{i+1,j+1,k+1} - a_2 - a_3 - a_4 - a_5 - a_6 - a_7 - a_8 \\
a_2 &= f_{i+1,j+1,k} - a_5 - a_6 - a_8 \\
a_3 &= f_{i+1,j,k+1} - a_5 - a_7 - a_8 \\
a_4 &= f_{i,j+1,k+1} - a_6 - a_7 - a_8 \\
a_5 &= f_{i+1,j,k} - a_8 \\
a_6 &= f_{i,j+1,k} - a_8 \\
a_7 &= f_{i,j,k+1} - a_8 \\
a_8 &= f_{i,j,k}.
\end{aligned} \tag{3.4}$$

This system of equations can be solved using the Newton method. As an initial guess, Tang [3] suggests

$$r_0 = \frac{|r_{a'p}|}{|r_{a'b'}|}, \qquad s_0 = \frac{|r_{c'p}|}{|r_{c'd'}|}, \qquad t_0 = \frac{|r_{e'p}|}{|r_{e'f'}|}. \tag{3.5}$$

Here, the points denoted by a prime refer to the projection point of p on the corresponding surface. For example, a' refers to the projection point on the surface with midpoint a (see figure 3.3). With known coordinates r, s, t, the trilinear interpolation (3.3) can be used to solve for the interpolated velocities and pressure by substituting f by the corresponding variable (e.g. velocity u).

3.3 Global Mass Conservation

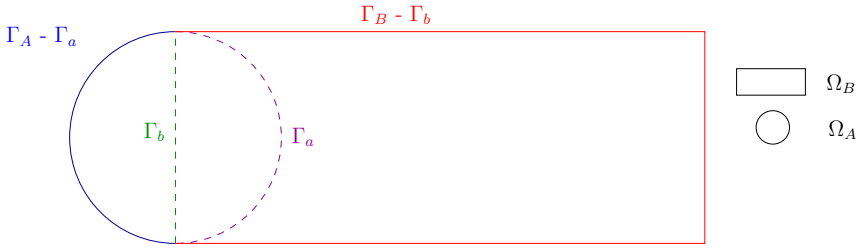


Figure 3.5: Global mass conservation

As with all numerical schemes, it is important that mass is conserved globally during the interface boundary treatment. To illustrate global mass conservation in the context of chimera grids, consider a simple case of overlapping subdomains

as in figure 3.5. A global mass conservation law for incompressible flow in the domain $\Omega = \Omega_A \cup \Omega_B$ is given by:

$$\int_\Gamma \vec{v}\cdot\vec{n}\mathrm{d}S = \int_{\Gamma_A} \vec{v}^A\cdot\vec{n}\mathrm{d}S - \int_{\Gamma_a} \vec{v}^A\cdot\vec{n}\mathrm{d}S + \int_{\Gamma_B} \vec{v}^B\cdot\vec{n}\mathrm{d}S - \int_{\Gamma_b} \vec{v}^B\cdot\vec{n}\mathrm{d}S = 0 \quad (3.6)$$

Assuming conservative numerical schemes are used in each subdomain Ω_A and Ω_B:

$$\int_{\Gamma_A} \vec{v}^A\cdot\vec{n}\mathrm{d}S = 0 \qquad \int_{\Gamma_B} \vec{v}^B\cdot\vec{n}\mathrm{d}S = 0 \quad (3.7)$$

Plugging this into equation (3.6), one receives

$$\int_{\Gamma_a} \vec{v}^A\cdot\vec{n}\mathrm{d}S + \int_{\Gamma_b} \vec{v}^B\cdot\vec{n}\mathrm{d}S = 0. \quad (3.8)$$

Additionally, the following conservation equations hold since the overlapping region belongs to both subdomains:

$$\begin{aligned}\int_{\Gamma_a} \vec{v}^A\cdot\vec{n}\mathrm{d}S + \int_{\Gamma_b} \vec{v}^A\cdot\vec{n}\mathrm{d}S = 0\\ \int_{\Gamma_a} \vec{v}^B\cdot\vec{n}\mathrm{d}S + \int_{\Gamma_b} \vec{v}^B\cdot\vec{n}\mathrm{d}S = 0\end{aligned} \quad (3.9)$$

Combining the results from equations (3.8) and (3.9), one arrives at the conditions for global mass conservation:

$$\begin{aligned}\int_{\Gamma_a} \vec{v}^A\cdot\vec{n}\mathrm{d}S = \int_{\Gamma_a} \vec{v}^B\cdot\vec{n}\mathrm{d}S\\ \int_{\Gamma_b} \vec{v}^A\cdot\vec{n}\mathrm{d}S = \int_{\Gamma_b} \vec{v}^B\cdot\vec{n}\mathrm{d}S\end{aligned} \quad (3.10)$$

These resulting conditions state that the mass fluxes from subdomain Ω_A and from Ω_B must be equal at the overlapping boundaries Γ_a and Γ_b. This condition will be the basis for the following interface boundary treatment methods to ensure global mass conservation.

3.4 Interface Boundary Treatment

In this section, a method for determining the interface velocities which takes into account the mass flux condition for global mass conservation is discussed. To illustrate this method, consider again figure 3.2 from the previous section. Let the red cartesian subdomain be defined as B and the black curvilinear subdomain as A. The goal is to formulate a boundary condition at the overlapping interface to determine the velocities in A dependent on the velocities in B.

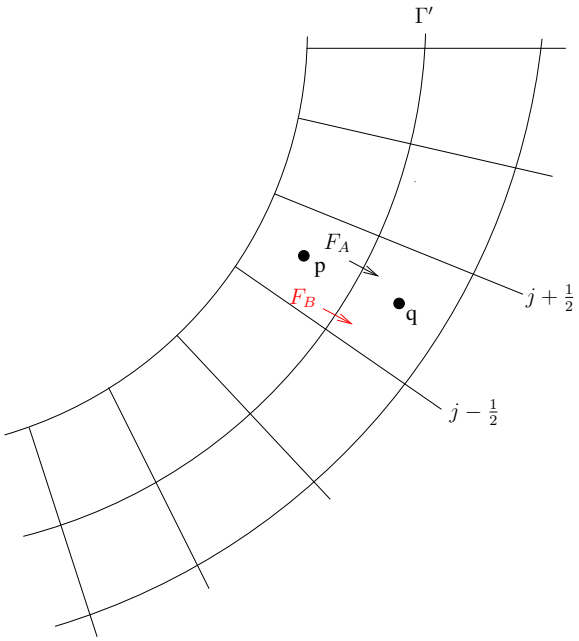


Figure 3.6: Interface mass fluxes $F_A \stackrel{!}{=} F_B$

To ensure global mass conservation, the numerical mass flux from A should be equal to the numerical mass flux from B across the boundary Γ' (figure 3.6). Locally in an interval $\left[j-\frac{1}{2}, j+\frac{1}{2}\right]$,

$$F_{\Gamma',j}\left(U_p^A, U_q^A\right) = F_{\Gamma',j}\left(U_p^B, U_q^B\right) \tag{3.11}$$

holds with contravariant velocities U. Two different methods are mentioned by Tang [3].

3.4.1 Method 1

In the first method, interpolated values are used for U_p^B and U_q^B and plugged into eqation (3.11) directly, leading to:

$$\begin{aligned} F_{\Gamma',j}\left(U_p^A, U_q^A\right) &= F_{\Gamma',j}\left(U_p^I, U_q^I\right) \\ U^I &= u^I\xi_x + v^I\eta_x + w^I\zeta_x. \end{aligned} \tag{3.12}$$

u^I, v^I, and w^I are determined by 2nd order accurate trilinear interpolation (equation (3.3)). In [3], central differencing was used to discretize the continuity equation and thus eq. (3.12) can be rewritten as

$$\left(\frac{U^A}{J}\right)_p + \left(\frac{U^A}{J}\right)_q = \left(\frac{U^I}{J}\right)_p + \left(\frac{U^I}{J}\right)_q. \tag{3.13}$$

In summary, one receives

$$
\begin{aligned}
U_p^A &= J_p \left(\left(\frac{U^I}{J} \right)_p + \left(\frac{U^I}{J} \right)_q - \left(\frac{U^A}{J} \right)_q \right) \\
V_p^A &= V_p^I \\
W_p^A &= W_p^I \\
p_p^A &= p_p^I.
\end{aligned} \tag{3.14}
$$

3.4.2 Method 2

For the second method, one assumes that a 2nd order accurate differencing scheme is used to determine the numerical fluxes:

$$
F_{\Gamma',j}\left(U_p, U_q\right) = \frac{U_p + U_q}{2} + O\left(\Delta^2\right) \tag{3.15}
$$

With the mass flux condition (eq. (3.11)), this leads to

$$
\frac{U_p^A + U_q^A}{2} = \frac{U_p^I + U_q^I}{2} + O\left(\Delta^2\right) \tag{3.16}
$$

which is fulfilled by

$$
\begin{aligned}
u_p^A &= u_p^I + u_q^I - u_q^A \\
v_p^A &= v_p^I + v_q^I - v_q^A \\
w_p^A &= w_p^I + w_q^I - w_q^A \\
p_p^A &= p_p^I.
\end{aligned} \tag{3.17}
$$

3.4.3 Method Comparison

Tang [3] compared the two mentioned methods using a lid-driven cavity flow simulation. For both methods, identical results were achieved with similar convergence characteristics. Method 1 exhibited slightly faster convergence in one subdomain, while method 2 had somewhat lower residual levels for one velocity component. The conservation error converges rapidly and remains constant afterwards for both methods. However, method 2 showed to have smaller errors in one subdomain, particularly at an earlier elapsed simulation time.

Note that Tang [3] used an artifical compressibility (AC) based solver. As noted by Zhang [5], SIMPLE-based solvers should use the interpolated pressure to obtain a pressure correction BC:

$$
p'_p = p_p^I - p_p^{old} \tag{3.18}
$$

3.5 Algorithm Summary

To summarize the discussed chimera grid method, the algorithm is divided into a pre-processing stage and a solver stage.

The pre-processing stage need only be carried out once as part of the intilization of the flow solver. However, this is only the case for stationary and constant grids. In the case of moving grids or grid refining or remeshing, this stage should be carried out everytime the grid changes. The basic steps are as follows:

- decompose domain into subdomains (possibly only during grid generation)
- solve grid connectivity and store r, s, t coordinates
- "cut out" hole cells by blanking them out

The solver stage deals mainly with the actual interface treatment of the overlappping boundaries. These steps need to be carried out at each iteration as they define the boundary conditions on outer fringe cells:

- loop through outer fringe cells and for each cell:
  - use r, s, t to interpolate velocities and pressure at interface
  - set boundary conditions according to method 1 or 2

In this work, a simple case was considered to outline the basic chimera grid method. The decomposition and overlapping of subdomains can be arbitrarily complex, depending on the application. In such cases, the outlined method has to be extended accordingly to take into account multiple overlapping subdomains.

4 Conclusion

In this write-up, the Chimera grids method was introduced as a remedy to the arduous task of grid generation for complex geometries. It allows the decomposition of a complex domain into simple, overlapping subgrids which can be easily generated. The main difficulty of this method lies in the treatment of interface boundaries of overlapping subdomains, taking global mass conservation into account. Two interface treatment methods [3] were discussed and finally the entire algorithm was outlined.

References

[1] Jiri Blazek. *Computational Fluid Dynamics: Principles and Applications: Principles and Applications*. Elsevier, 2001.

[2] Joseph L Steger, F Carroll Dougherty, and John A Benek. A chimera grid scheme. 1983.

[3] Hansong Tang. Numerical simulation of unsteady three dimensional incompressible flows in complex geometries. 2001.

[4] John Robert Taylor. *Classical mechanics*. University Science Books, 2005.

[5] Xing Zhang. Computation of viscous incompressible flow using pressure correction method on unstructured chimera grid. *International Journal of Computational Fluid Dynamics*, 20(9):637–650, 2006.