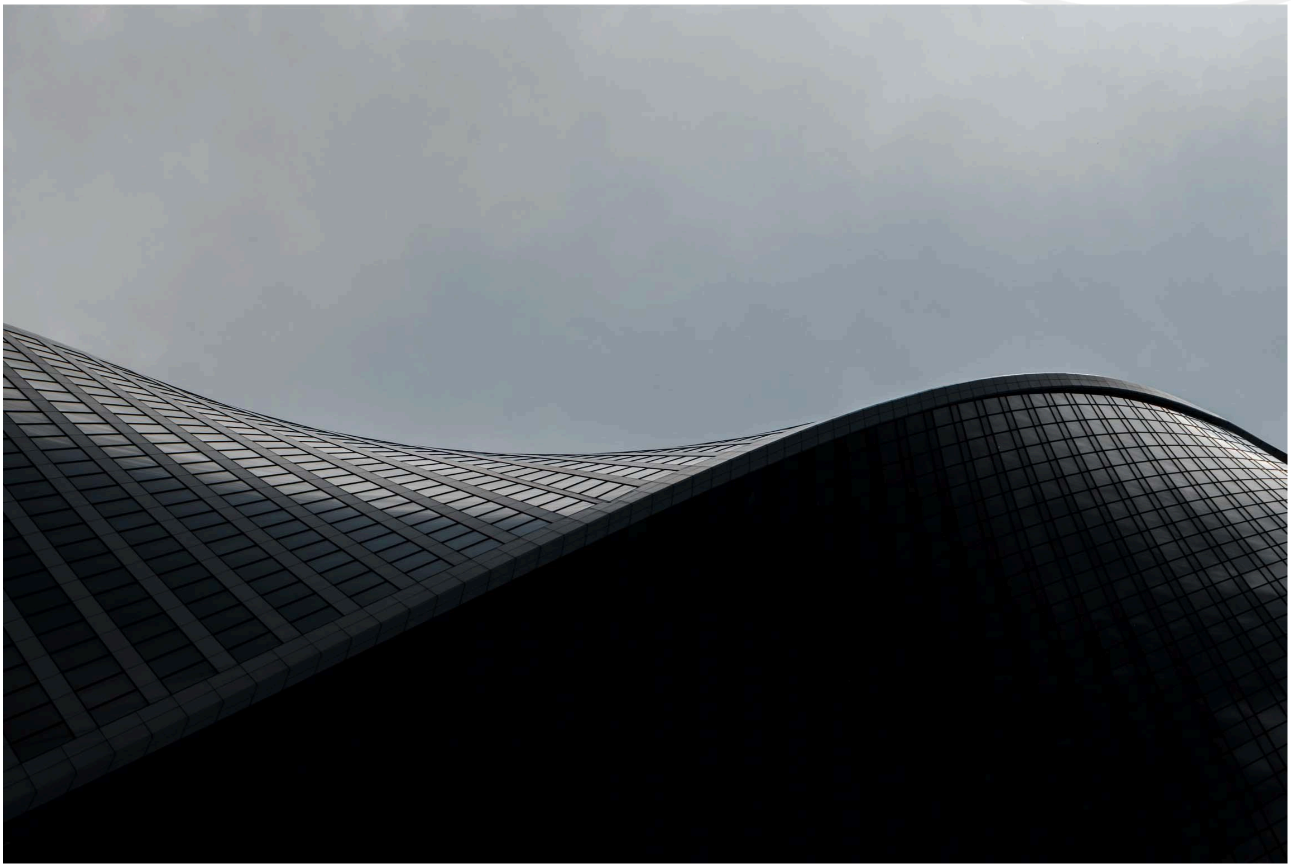
The Non-Conformal Grid technique

The first disruptive innovation in CFD



The birth of the CFD (Computational Fluid Dynamics) industry, as we know it today, can date back to the mid-80s. At that time Anthony Jameson was able to simulate the flow over a complete aircraft for the first time. Yet, it is difficult to find new real breakthroughs since then.

Some years ago, a group of experts in the CAE (Computer Aided Engineering) industry shared their ideas on where there was room for innovation in the CFD industry and what was holding it back from innovating. In the round table discussion, among many other drawbacks, they focused on the meshing phase which was defined as "*the biggest problem plaguing the CFD market*" (https://bit.ly/4a8QiDn).

The Conformal Mesh

Since the birth of CFD, the standard way to create a 3D mesh around the objects (aeroplanes, machineries, buildings, …) was based, and still is, on a 2-step procedure:

1. creation of a 2D surface mesh on all objects
2. creation of the 3D mesh that, starting from the nodes of the object's surface meshes, fills all of the space of the computational domain with hexahedral, or tetrahedral cells, or a mix of them.

This type of mesh is said to be Conformal because the nodes that define the objects, created in the first step, are also nodes of the 3D mesh.

When creating the 3D conformal mesh, most of the cells, particularly those near the objects, will be distorted from a regular hexahedral or tetrahedral shape. The challenge of this mesh generation procedure will be to minimise such distortion. The truncation errors of the numerical analysis increase, in fact, with cell distortion and the order of accuracy of the numerical schemes degrades from the nominal 2nd order to the 1st order, or somewhere in between depending on the level of distortion.

A variety of very sophisticated 3D mesh generators have been developed over the years to cope with these difficulties. However, they are time-consuming, and the mesh generation software needs to be used by skilled personnel. Moreover, if the 3D CAD models are geometrically complex, with a wide range of characteristic lengths, from large to small geometrical details, the 2-step process becomes even longer and more difficult, and often the user must first spend time cleaning the objects models from the very many small details, smoothing the surface out ("*pre-processing takes up to 60-70% of the engineer's time*" cit. reference).

Over the last few years the situation has hardly improved. The only new technique that has appeared since then, which is worth mentioning, is the so-called "Cut-cell" technique.

This technique makes use of a background Cartesian mesh which is automatically refined near the solid objects and finally forced to be body-fitted (i.e. Conformal) by adjusting, cutting, merging and modifying the cells adjacent to the solid surfaces. It is indeed automatic and quick, and it is Cartesian almost everywhere except close to the objects. Furthermore, it is not free from many defects.

The Fig.1 shows a cut-cell mesh on the two sides of an object. The effect of "cutting, adjusting and merging" is clearly visible, though the object is indeed very smooth. When the geometry definition is more complex, the distortion effects are much larger and may even become too large to cope with.

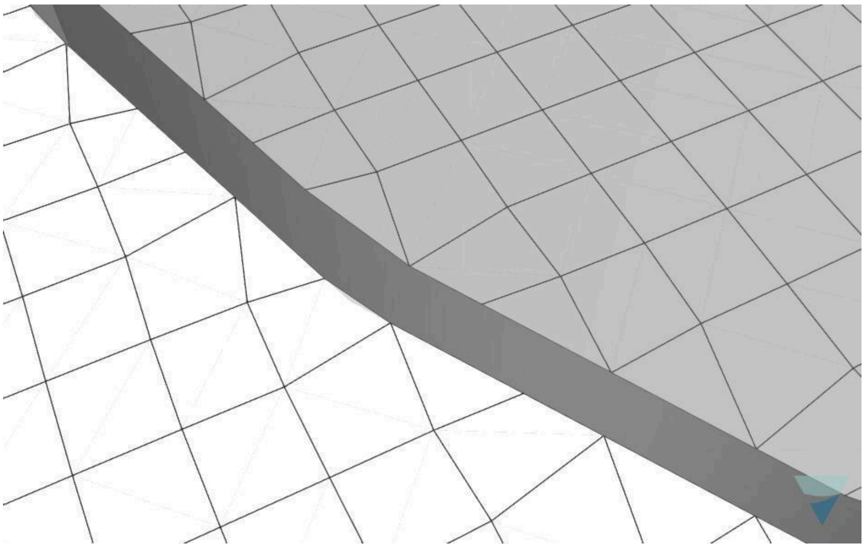


Fig 1: cut-cell conformal grid

Beside the distortion of the cells near the surfaces, another important defect of the Cut-cell technique is that it is not possible to control the mesh size: if the 3D model is complex (i.e. it contains many small scale geometrical details, which are irrelevant for the purpose of the simulation) the cleaning phase becomes mandatory, which is a tedious and time-consuming activity. If all small details are kept, the user loses control of the final mesh size because everytime a mesh line intercepts one of the irrelevant small scale objects, a new mesh node is added. This behaviour of the Cut-cell method may easily lead to meshes with many millions nodes and even tens of millions cells, making it impossible to carry out preliminary tests on coarse meshes.

The Non-Conformal Grid technique by Vento

The Non-Conformal Grid (NCG) technique was born in the early 2000s thanks to the pioneering work of Roberto Verzicco, with the name of “Immersed Boundary” technique. This technique represents, we believe, the first breakthrough in the history of CFD. It remained, however, hidden in the R&D departments of universities and research centres.

At Vento we developed our own NCG technique thanks to Roberto’s suggestions and advice, and we made it available for the first time in a commercial CFD product. Our technology is proprietary and ahead of the state-of-the-art described in the scientific literature[1] and never published.

The purpose of the present White Paper is to show the immense potential of the NCG technique.

Since the mesh and the object’s nodes are two disjoint sets, and so they will remain, no new nodes will be added on the object surfaces, and the mesh size remains completely under the user’s control. This feature opens the doors to the possibility to simulate a set of quick analyses, even on super complex geometries, for obtaining preliminary results, useful to understand the trends when trying various design variants.

The Fig.2 shows a schematic representation of the grid (black lines), the “immersed” object (the blue line) and the boundary of the computational fluid domain (the red lines). The latter is made by all cells whose centre lies in the fluid (upper) region.

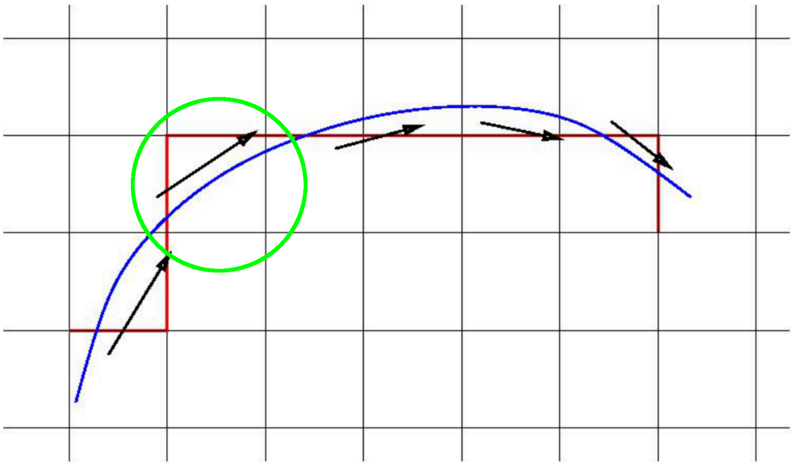


Fig 2: Non-Conformal grid

[1] Finite-Volume Ghost-Cell formulation

It can be seen that the mesh nodes and object’s nodes do not match, the essence of the NCG technology is that the position of the object inside the mesh is determined by the calculation of the intersected points and of the normal to the object’s surface. In summary:

- *the shape and the position of the object are reconstructed by the NCG algorithms without relying on the mesh topology*;
- *the cells that intersect the objects, or near them, are neither cut, nor modified. They remain untouched:* the “fluid” (or computational) domain is made by all cells whose cell centre lies on the fluid side (upper side in this example). It is important to notice that the cell encircled is roughly divided into two triangles by the blue interception with the object: the upper triangle is occupied by the flow, but the cell is not part of the computed fluid domain, for its cell centre lies below the object.
- *so, in the NCG technique, the fluxes of mass, momentum and energy are allowed through the (red) boundaries of the Domain:* see the arrows in the image
- *finally, it is worth recalling that, though the fluid domain appears to be stair-step, it is not*

The image 3 below shows the detail highlighted in the previous image.

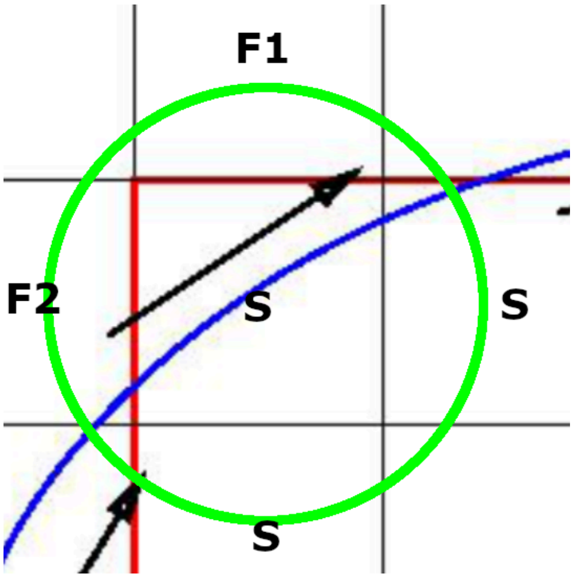


Fig 3: detail of the Fig.2, Non-Conformal grid

The encircled cell, as said before, is not part of the computational domain because its cell centre lies on the solid region, and therefore it is not computed. That **S** cell actually represents two ghost cells, the one associated to the cell F1, and the second one associated to the cell F2. The two ghost cells must in fact exist independently of each other in the NCG data structure, though they refer to the same “physical” location.

This implementation allows, for instance, the treatment of walls with zero thickness, where the fluid cells on one side seem to correspond, or to overlap, to the ghost cells of the fluid cells on the opposite side, but they do not. Ghost cells are only memory locations which do not interfere or overlap with other fluid and/or ghost cells that may share the same location.

To summarise: F1 and F2 cells force their velocity vector to be tangent to the blue object's surface (application of boundary conditions). Then, the air flows freely through the west side of the cell F2, exiting the computational domain, and re-enters the computational domain through the south side of the cell F1. The tangency condition impedes the air flow to penetrate the surface of the object. So, at convergence, the upper triangle of the encircled cell has a net mass flow rate equal to zero, and the NCG technique does not create or lose spurious mass flow rates.

The key feature that allows VENTO AEC to perform quickly on a complex model without the need of cleaning all of the "small" irrelevant objects, is the capability that we call "subgrid" objects.

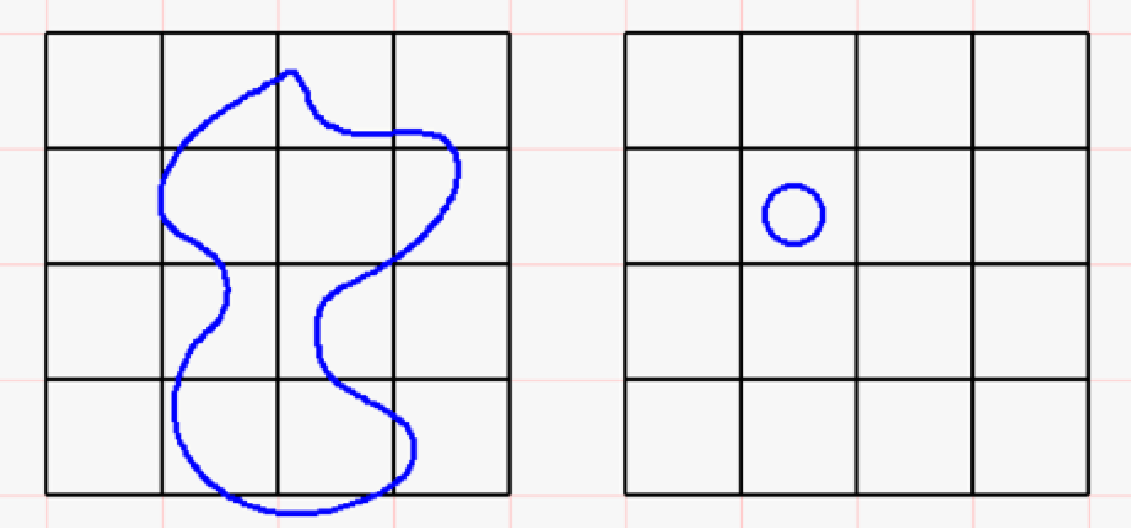


Fig 4: examples of sub-grid objects

The sub-grid objects are represented by all small geometrical details smaller that the grid size (Fig. 4 left) or by those objects whose shape cannot be reconstructed by the NCG algorithms because they are resolved (surrounded) by a number of cells which is inadequate to reconstruct the true shape (Fig.4 right). Alternatively, sub-grid objects may be defined as those objects whose shape is characterised by curvature radii smaller than the cell size.

Sub-grid objects represent irrelevant details that a user would like to clean when using a Conformal grid technique. By using the NCG technique, the user is allowed to keep them all. There is no need to spend hours and days cleaning the models, and the simulation is closer to reality and the graphical representation is more beautiful as well.

The “sub-grid” object is therefore a relative concept, not absolute. There are many examples where the smallest objects present are indeed relevant to the analysis: for instance when, in an indoor space, the air intakes are very many in number and very small in dimension, and uniformly distributed all over the space. They may even represent the smallest objects present. They have to be resolved accurately then, by using an adequate cell size, and must not be treated as “sub-grid”.

Case Study

We would like to show a good example of a real object which allows us to highlight the power of Vento’s NGC solution.

It is a high quality STL of a telecommunication antenna purchased on the web, made of 180k triangles (see Fig. 5). The complexity in terms of very many extremely small geometrical details is clearly visible.

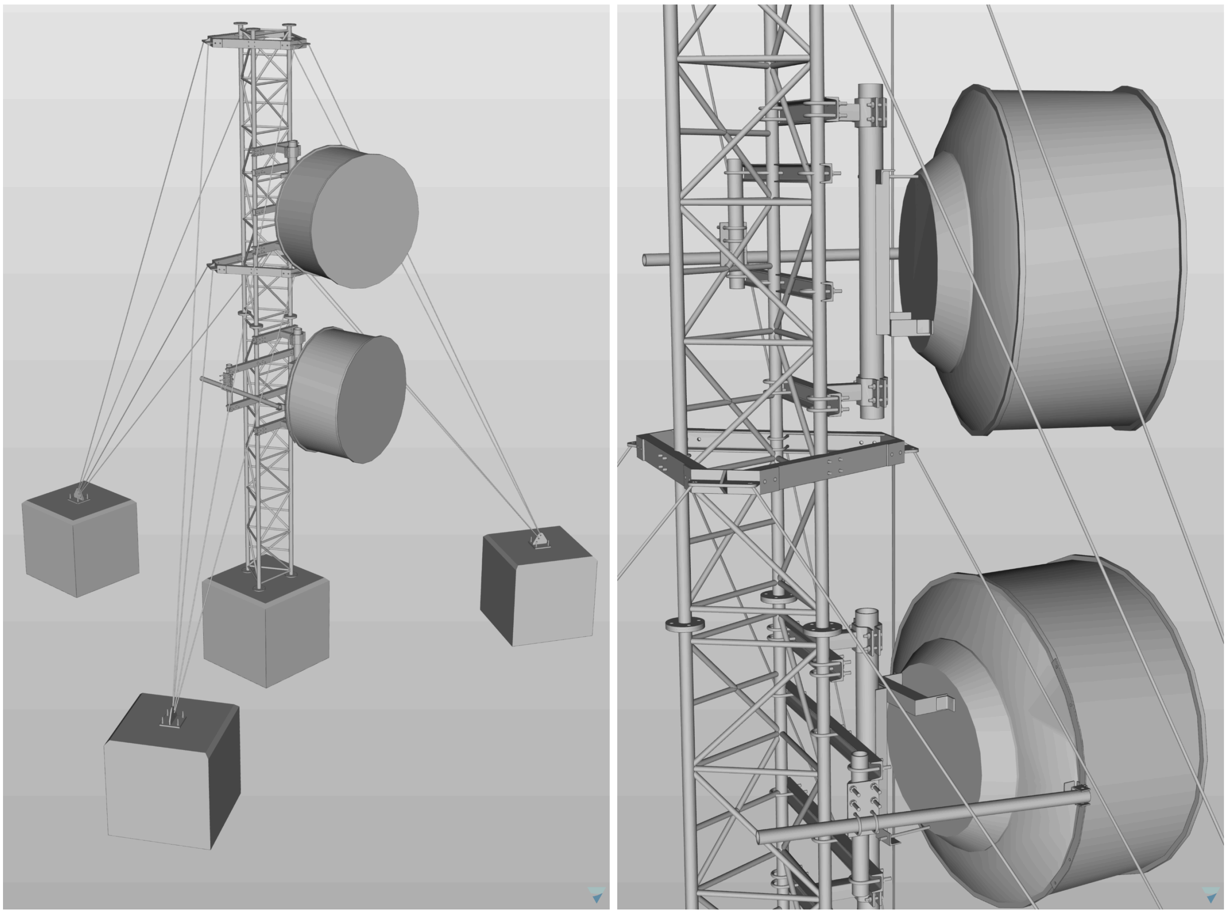
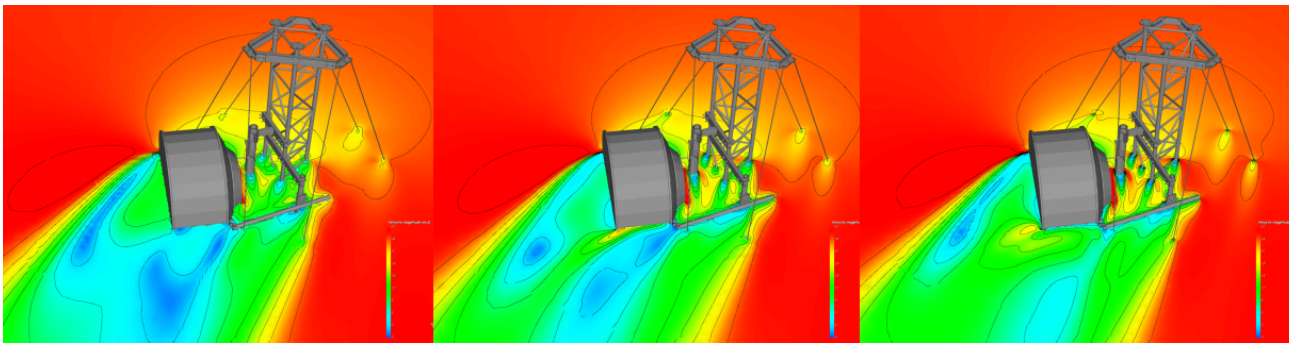


Fig 5: STL model of the antenna

We tested three mesh levels, coarse, medium and fine, respectively made of 2.4, 7.5 and 32.9 million cells.

The whole tower and the guy wires are treated as “sub-grid”, and the two antennas are fully resolved by the chosen cell size. The purpose of these tests is to evaluate the forces on the two antennas as well as on the whole structure, and the differences of such performance values obtained by the three simulations.

The image 6 shows the computed flow filed on an horizontal section at mid height. The image 7 summarises the performances.



Coarse grid 2.4 M | **Medium grid 7.5 M** | **Fine grid 32.9 M**

Fig 6: velocity field on an horizontal section at mid height

Case	# of cells [M]	X force [tons]	Y force [tons]	Resulting force [tons]	Direction of force [degrees]	Grid generation [minutes]
Coarse	2.4	113.4	121.8	166.4	312.9	2
Medium	7.5	121.3	120.3	170.8	315.2	11
Fine	32.9	123.8	118.8	171.5	316.2	66

Wind Direction 315° – Wind Speed 28 m/s (100 km/h)

Fig 7: performance of the simulation on 3 grid levels

The results shown in the above table speak for themselves. The presence of the guy wires is captured also by the coarse grid. Coarse mesh results differ by about 3% from those calculated with the fine mesh, and are obtained in a tiny fraction of time: the CPU time per iteration was below 7 seconds (using six i7 cores) for the coarse mesh and approximately 60 seconds per iteration using the fine mesh (with ten i9 cores). The coarse simulation was run for 3000 iterations, the medium and fine mesh cases for 4000 iterations, all reaching steady state convergence.

We tried to mesh this case by using Snappy, the cut-cell mesh generator of OpenFOAM. We did not manage to stay below 50 million cells without cleaning the STL.

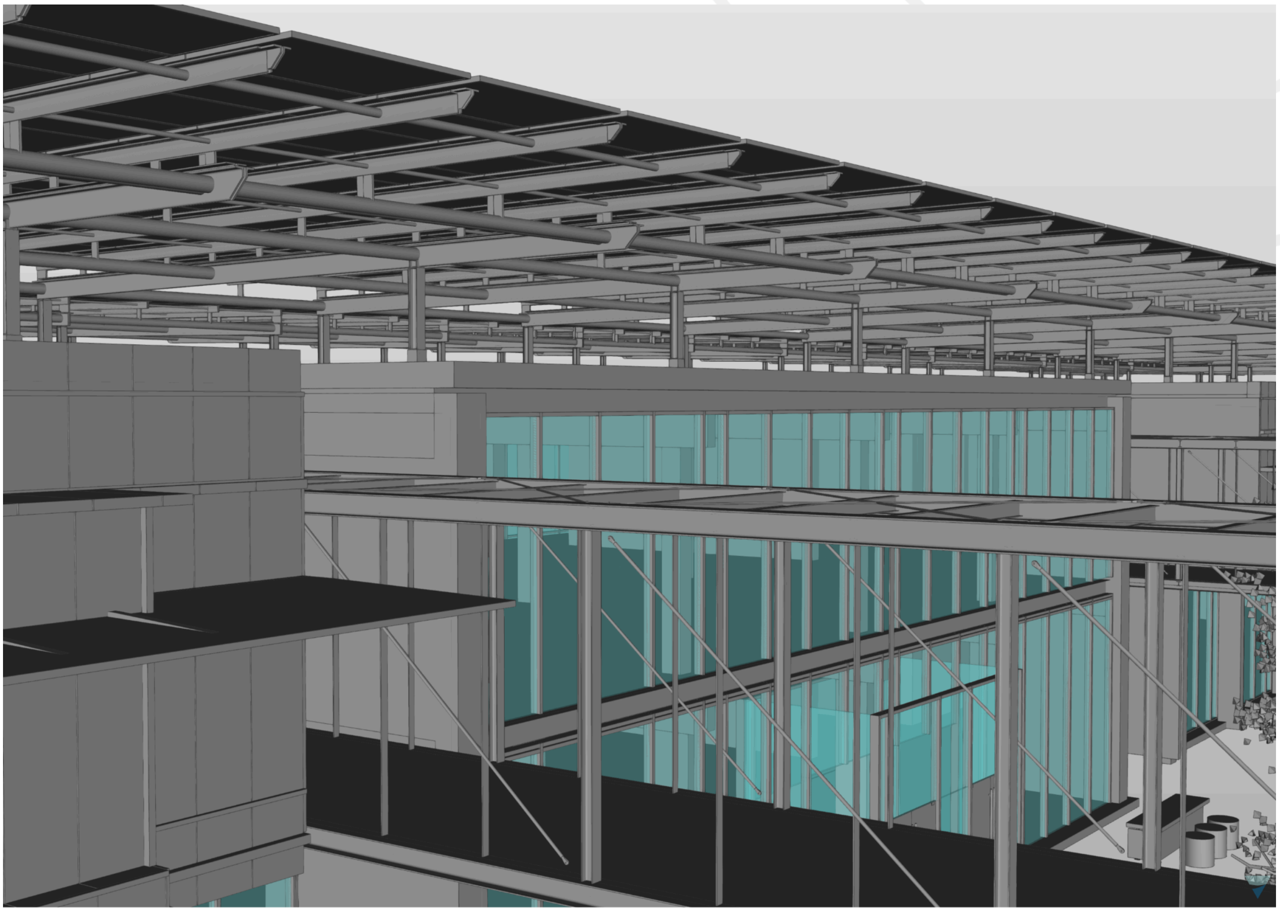
Conclusions

Vento brings a disruptive innovation in the world of CFD by making it:

1) Easy to use: as the mesh construction is automatic, CFD analysis can be done by non-experts.
2) Faster: up to 90% of pre-processing time is saved, enabling quicker design iterations and increased productivity
3) BIM ready: Vento allows direct interaction with the BIM environment and can import extremely complex and realistic models in IFC format.

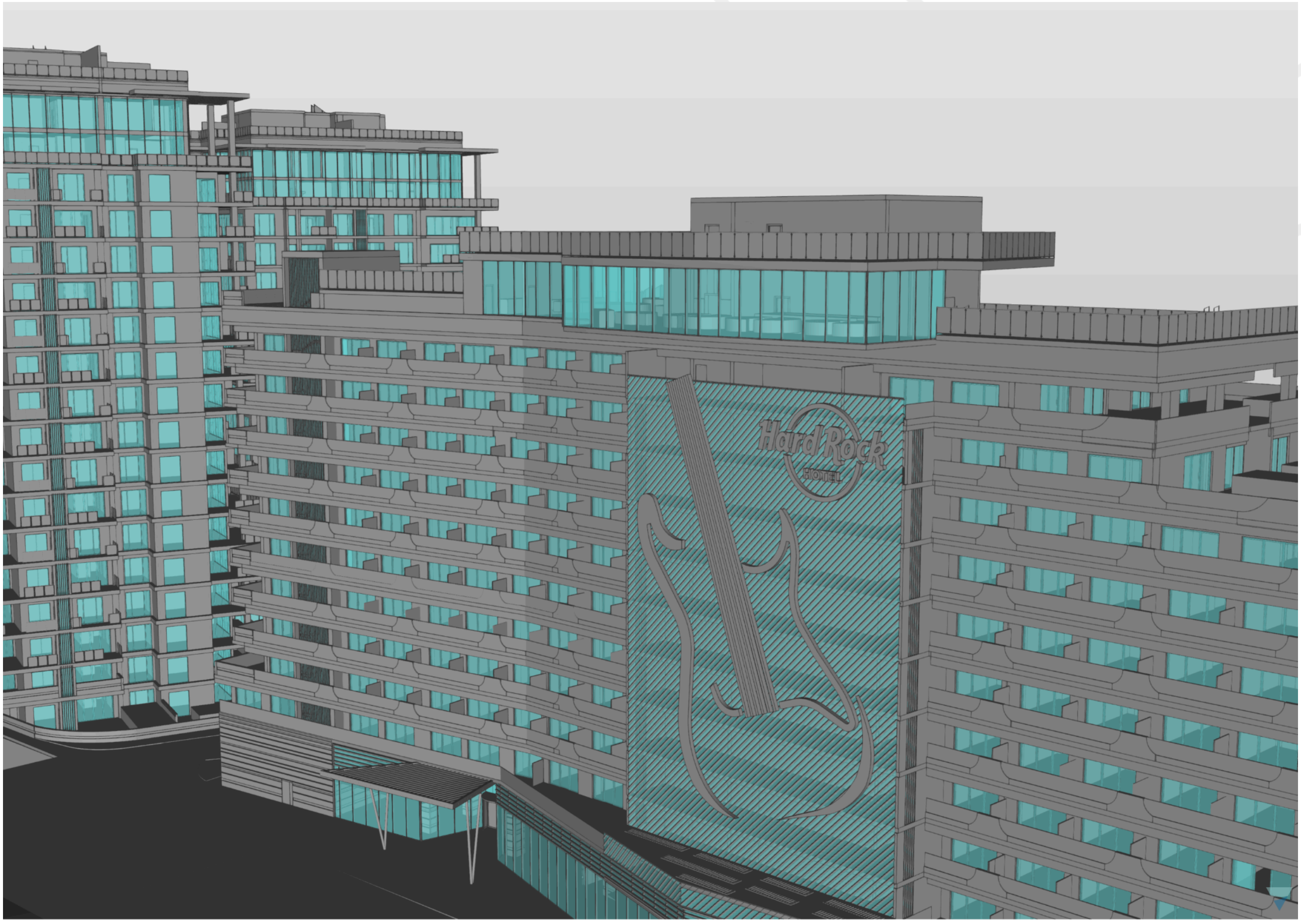
Thanks to its proprietary Non-Conformal Grid technique, Vento allows the use of models which are otherwise almost impossible to use.

The following gallery shows a few architectural 3D models of great complexity, which are easily and quickly treated by Vento’s NCG technology. Some of them are very detailed STL models purchased on the web, together with a few models generated by Revit (BIM) software and imported in IFC format.

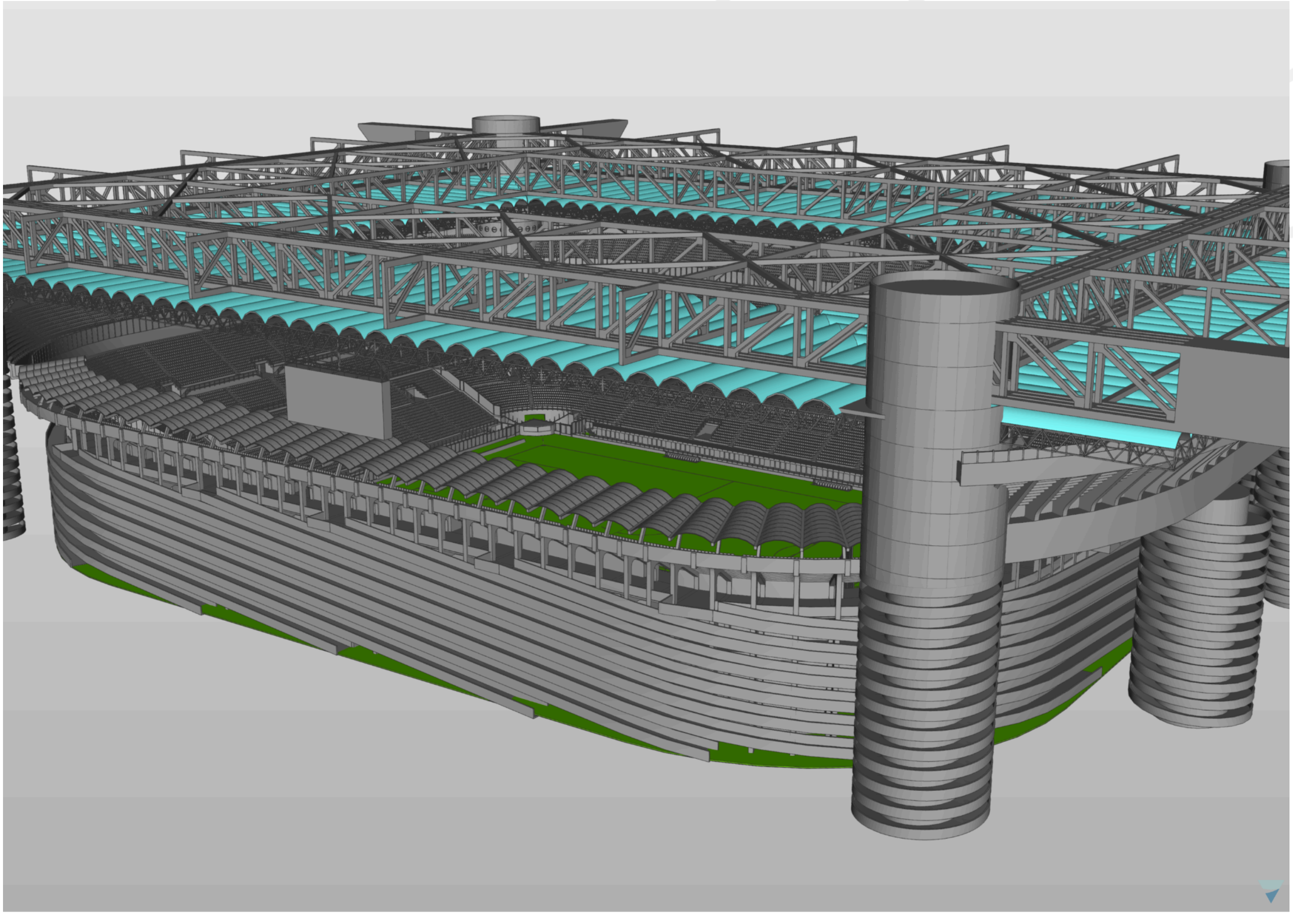


IFC model of a building. Courtesy of Milan Ingegneria, Milano, Italy.

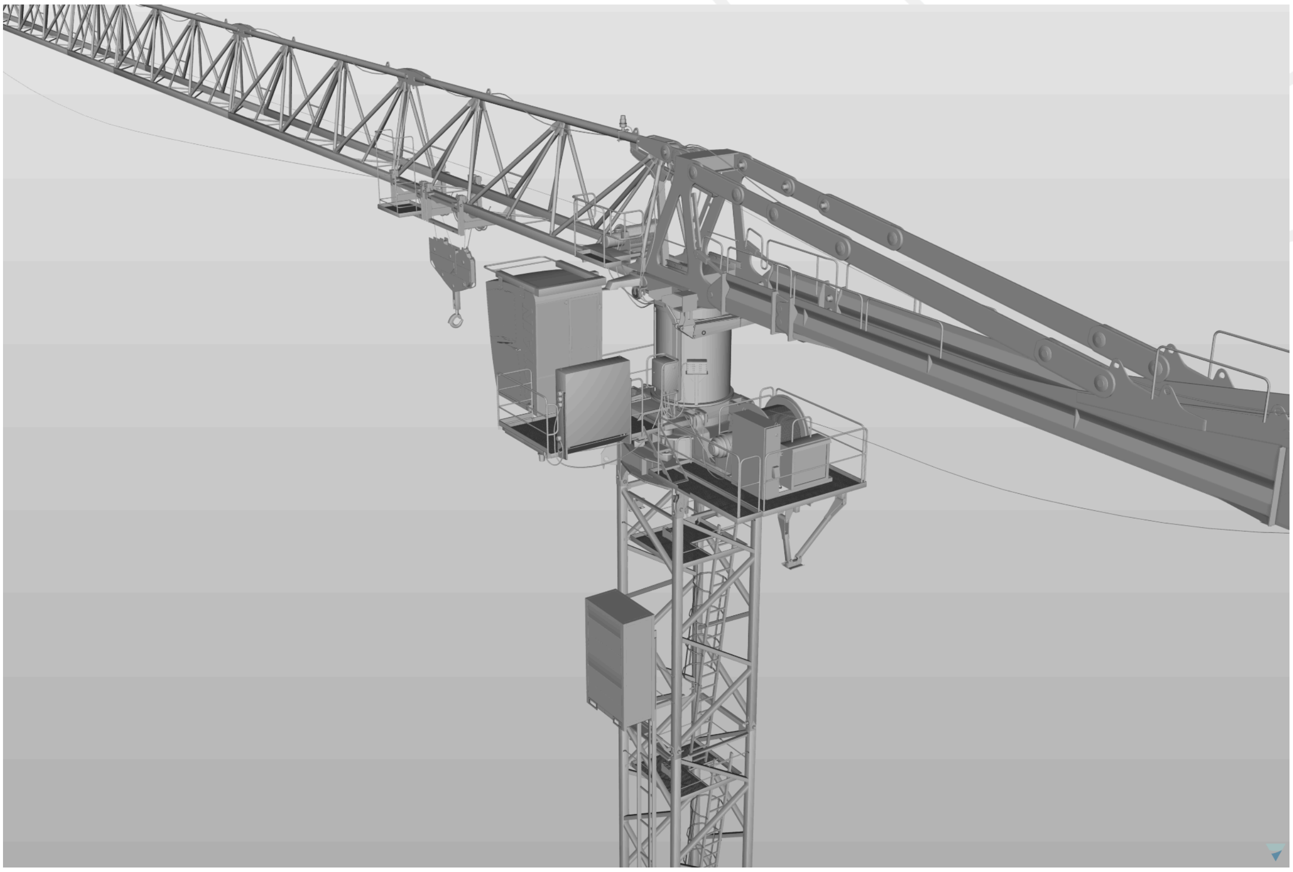
VENTO
NON CONFORMAL THINKING



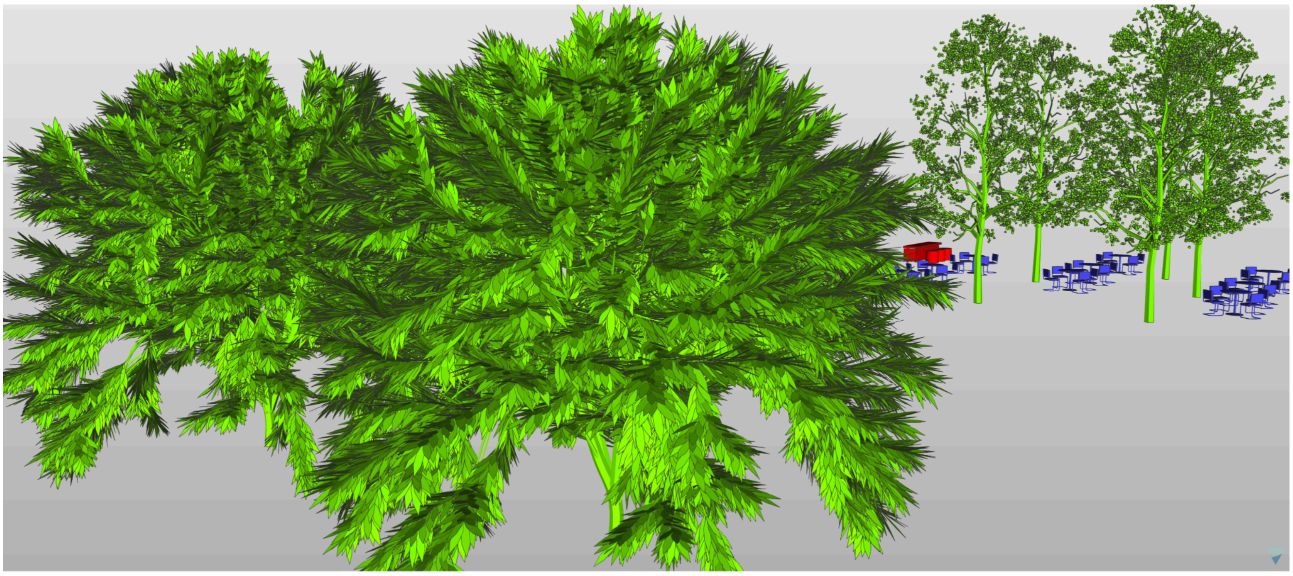
IFC model of Hard Rock Hotel in Malta. Courtesy of FM Ingegneria, Mirano (Ve), Italy



STL model of Stadio Meazza in San Siro, Milano, Italy



STL model of a Tower Crane



extremely detailed STL models of trees, together with STL models of garden chairs and tables.