EFFICIENT SIMULATION OF BUILT ENVIRONMENTS USING OPEN SOURCE CFD APPLICATIONS

Francisco Campos
Andrew Jackson
Eugene de Villiers

ICON, Rofel House, Colet Gardens, London, W14 9DH, UK
www.iconCFD.com

THEME

Computational Fluid Dynamics

KEYWORDS

CFD, Open Source, Parallel Meshing, DES, Built Environments, Aerodynamics, Wind Loads.

ABSTRACT

In this paper a set of Open Source tools are presented for performing aerodynamic CFD calculations for the built environment. These tools allow all stages of the CFD process, from mesh generation to post-processing, to be performed in a distributed fashion. A new parallel hex/split-hex automatic mesh generator has been developed to provide fast, robust volume meshing capability. The mesh generator is complimented by a GUI for setting up the meshing and initialisation of solution parameters. Coupling these components with a state-of-the-art Detached Eddy Simulation (DES) solver allows exceptional flow accuracy to be achieved within acceptable time frames.

The code transparency and “peer review” nature of Open Source software gives clear advantages over proprietary software. Not only does it provide the user with confidence in the implementation, but it also provides significant cost benefits. In recent years Open Source CFD software has also reached a level of maturity which allows the whole CFD process to be completed in a fast, robust, and accurate way. This combination of factors, and the potentially unlimited scalability of these tools, enables the use of DES as a possible substitute of traditional wind tunnel measurements for assessing wind loads on large structures and buildings. The CFD workflow presented herein has been tested using a complex model of the Sears Tower, located in the city of Chicago.

EFFICIENT SIMULATION OF BUILT ENVIRONMENTS USING OPEN SOURCE CFD APPLICATIONS

1: INTRODUCTION

The ability to evaluate wind loads accurately is becoming increasingly important given the growing demand for unusual buildings and structures and the proliferation of non conventional materials for construction. The changing weather patterns in recent years means more frequent and severe wind storms are expected. The result is that wind induced property losses now annually exceed the sum of all other losses from natural hazards [1]. CFD can offer a faster and more cost-effective solution for evaluating these loads at design level than traditional wind tunnel methods.

A set of tools for carrying out a complete CFD simulation have been developed around the Open Source computational library OpenFOAM, an object oriented C++ toolbox for solving systems of partial differential equations in a finite volume framework.

The main components of the CFD process consist of:

- a user interface for setting-up meshing and solution parameters
- an automated parallel hex/split-hex mesh generator
- a boundary condition and field setup and initialisation utility
- a DES flow solver
- and analysis and post-processing tools.

Due to the parallel nature of the mesh generator, setup utility and solver, the entire process can be executed in a distributed fashion. This allows for the simulation of much larger computational meshes than would be the case if the model had to be aggregated into a single domain.

Efficient solution algorithms provide exceptional accuracy when compared to Reynolds averaged simulation (RAS) approaches without sacrificing stability even when the flow exhibits high Courant numbers. Typically, standard DES can cost an order or more of magnitude more than traditional methods, but optimisation and automation of mesh generation, setup and solution algorithms ensures quick turn-around times.

The outlined methodology is highly customisable due to its Open Source nature, which allows for targeted developments to suit an individual user's needs. The scalability of the approach makes optimisation based product design and other demanding applications a real possibility.

In Section 2, the CFD process for performing an external aerodynamics simulation is discussed. This includes a user interface for controlling the process, a new parallel mesh generation capability and flow solution and post-processing options in OpenFOAM. The versatility and potential of these tools for complex built environments applications are highlighted in Section 3, where DES is employed to evaluate the wind loads on the Sears Tower (Chicago, Illinois) under extreme wind conditions.

Figure 1: Sears Tower, Chicago - Illinois (from www.wikipedia.com).

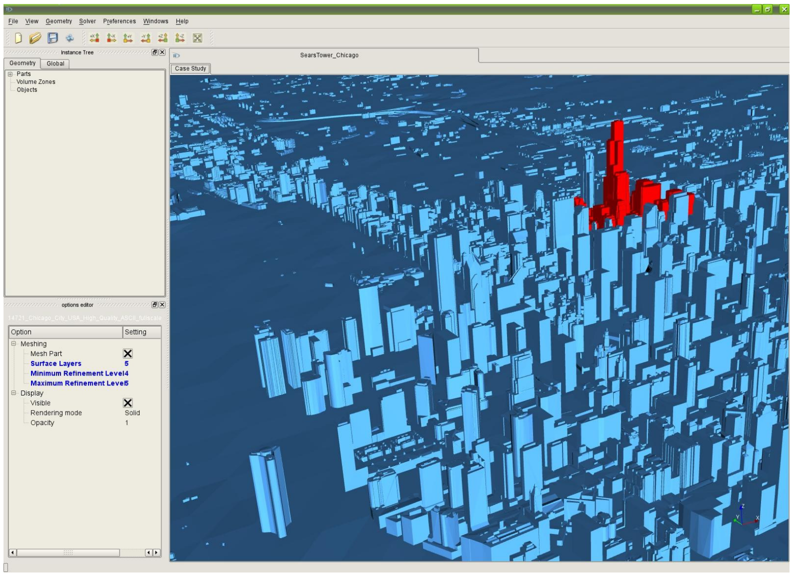
2: AN OPEN SOURCE CFD PROCESS FOR SIMULATING EXTERNAL AERODYNAMICS

The following subsections give details of the key elements in the Open Source stack for performing simulations on realistic external aerodynamic problems.

2.1: Graphical User Interface

Closed source proprietary CFD software has often been seen to have an advantage over Open Source in the interface that is provided between the user and the software. An interface provides a display of dynamically modifiable abstract information for setting up and running the problem and has clear advantages over command line in terms of ease of use and accessibility for non-experts. For this reason a Graphical User Interface (GUI) has been developed to allow easy and fast control of the setup and simulation process.

The GUI allows the geometry to be imported, viewed and manipulated. The setting of meshing parameters, boundary conditions and flow initialisation methods can also be performed from within the GUI, allowing fast and more importantly consistent error free set-up of cases. A screen-shot of the GUI is shown in Figure 2.



Option	Setting
Meshing	
Mesh Part	☒
Surface Layers	5
Minimum Refinement Level	4
Maximum Refinement Level	5
Display	
Visible	☒
Rendering mode	Solid
Opacity	1

Figure 2: Screen-shot of the GUI used for setting up the meshing parameters, setting boundary conditions and initialisation of the flow-field.

2.2: Mesh Generation

With the need to model ever more complex geometries, the rapid generation of high quality meshes has become a critical stage in the CFD design process. A new automatic parallel mesh generator has been developed inside the OpenFOAM framework. The meshing has been tailored to produce high quality meshes for viscous flow modelling. The key features of the new mesh generation tool are the following:

- Mesh generated in parallel.
- Fully automatic. Only requires a closed geometry. It is not necessary to perform any cleaning up of the CAD.
- Boundary Layer mesh generation on a patch-wise basis.
- Local patch-wise surface (including curvature) refinement and volume refinement away from the surface.
- Surface features in the geometry automatically preserved in the mesh.
- A guarantee that the generated mesh satisfies user defined mesh quality criteria.
- Code transparency.

The mesh generation stage has been designed to keep any user interaction in the process to a minimum through the use of sensible defaults and intelligent algorithms. Using the newly developed tools, it is possible to go from CAD to a mesh with tens of millions of cells in hours rather than days.

In order to achieve accurate flow solutions, it is essential that the new mesh can satisfy a set of strict quality checks. At all stages in the mesh generation process face and cell based quality is monitored to ensure that the final grid satisfies a set of user defined cell and face based quality criteria. Currently the mesh quality checks performed are for: orthogonality, face pyramid volumes, face areas, face skewness, face interpolation weights, cell volume ratio, face twist and cell determinant. If at any stage quality cannot be guaranteed, the mesh reverts back locally to a previously valid state. The various stages of the mesh generation process are outlined in the following sub-sections.

2.2.1: Stage 1 - Castellated Mesh Generation

In the first stage of the process an initial uniform Cartesian base mesh is generated extending throughout the solution domain. This Cartesian mesh is refined based on user defined patch-wise surface refinement levels. Additional surface refinement can be performed based on the local surface curvature. Furthermore, volume refinement strategies are also available to refine the mesh away from the surface (e.g. wake regions). Figure 3 shows iso-surfaces which have been created to allow refinement of the mesh away from the surface on a generic F1 car example test case.

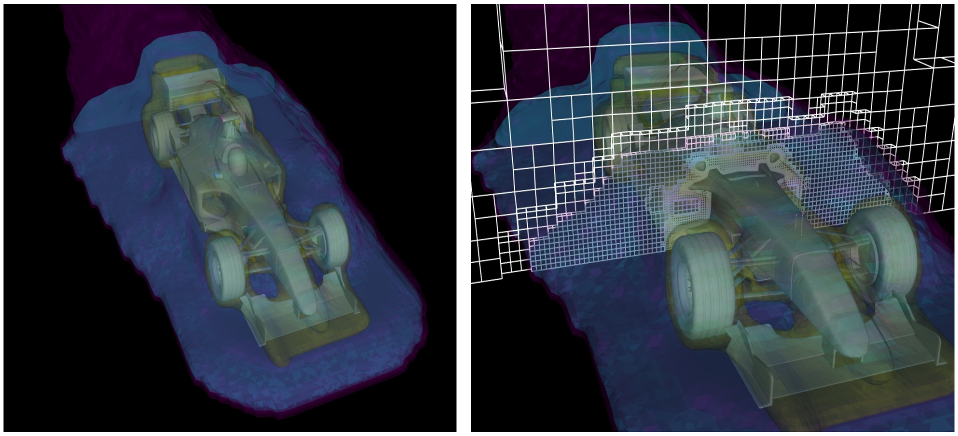


Figure 3: Iso-surfaces (left) used to control volume based refinement. Cut through the mesh (right) showing the volume refinement away from the surface.

Once the refinement phase is completed, an intersection check is performed to determine which parts of the mesh should be discarded. This is accomplished by first determining if there is a surface between neighbouring cell centres using an octree-based search. If a surface is detected, the mesh between the neighbouring cells is locally decoupled. As a final step, a topological walk is performed to agglomerate all connected elements with the remainder being deleted from the mesh. A surface representation of such an intermediate castellated mesh is shown in

Figure 4.

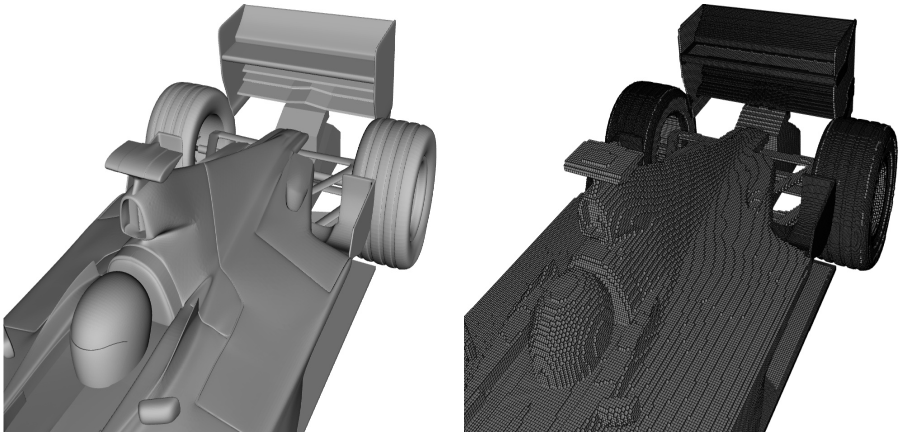


Figure 4: Surface geometry (left) and castellated surface mesh (right).

2.2.2 Stage 2 - Boundary Recovered Mesh

In the second stage of the mesh generation process the surface geometry is recovered. The surface faces of the castellated mesh, generated at the end of stage 1, are moved (snapped) to the surface. By constraining the boundary faces to be planar during the process, the feature edges present in the geometry are preserved in the surface of the mesh. As an additional step, co-planar boundary faces from the same cell are merged provided this leads to an improvement in the overall mesh quality. The resulting surface of the mesh after these snapping and merging operations is shown in

Figure 5.

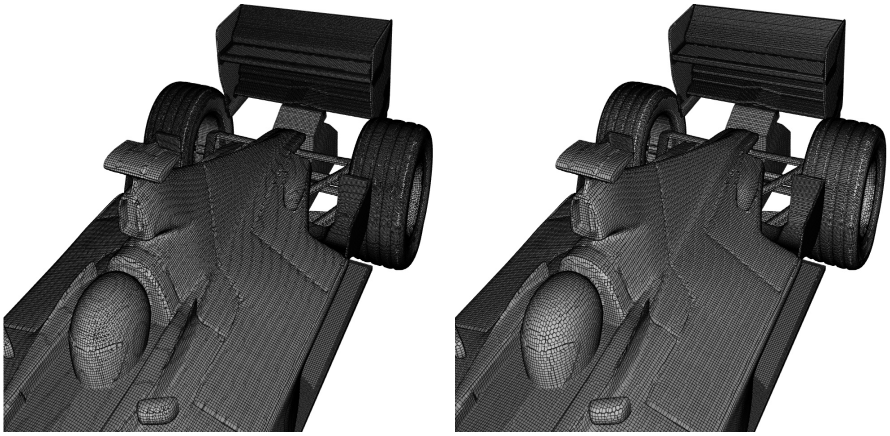


Figure 5: Boundary recovered surface mesh (left) and surface mesh after merging of faces (right).

At every stage during the mesh generation process cell and face based mesh quality are monitored; if a step leads to mesh elements that do not satisfy some user defined quality criteria the mesh is moved locally back to a previously generated valid state. This continuous forwards and backwards process guarantees the overall quality of the final mesh.

2.2.3: Stage 3 - Boundary Layer Addition

With the need for improved viscous flow modelling, an optional final stage in the mesh generation process is to add layers of cells next to the surface. This can be performed on a region-wise basis. The base mesh, generated after stage 2, is projected off the surface and a layer of cells is generated in the void that is created. As in the previous stages of the mesh generation process, if layer addition results in any of the mesh quality criteria being violated, surface layers are not created in that region.

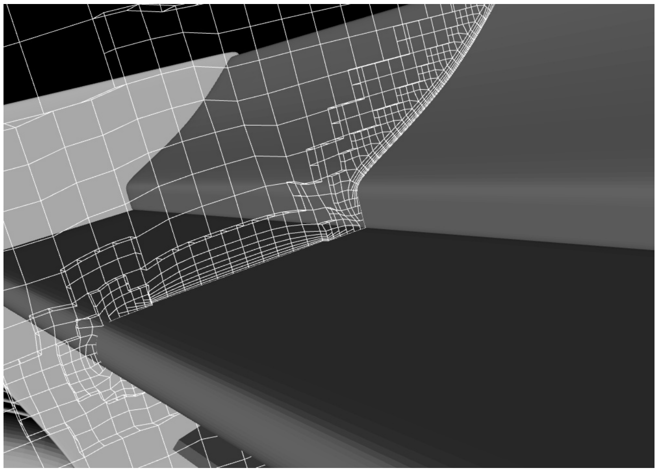


Figure 6: Layer mesh generated on rear wing of F1 car.

Figure 6 shows the surface layers created on the rear wing of the F1 test case. In this figure the layer is terminated in concave and convex regions where mesh quality in the layer mesh cannot be maintained.

Currently there are two layer termination strategies that can be employed; these two approaches are shown in Figure 7. In the first the layer is terminated with a single cell, while in the second approach the final terminated cell is merged with the cell directly above. The second approach has benefits when using wall functions by reducing jumps in first cell height in regions where the layer terminates. Termination tends to introduce errors due to the inaccuracy of linear assumptions in the logarithmic layer; therefore, it should be confined to regions where the equilibrium assumptions do not hold (e.g. sharp corners).

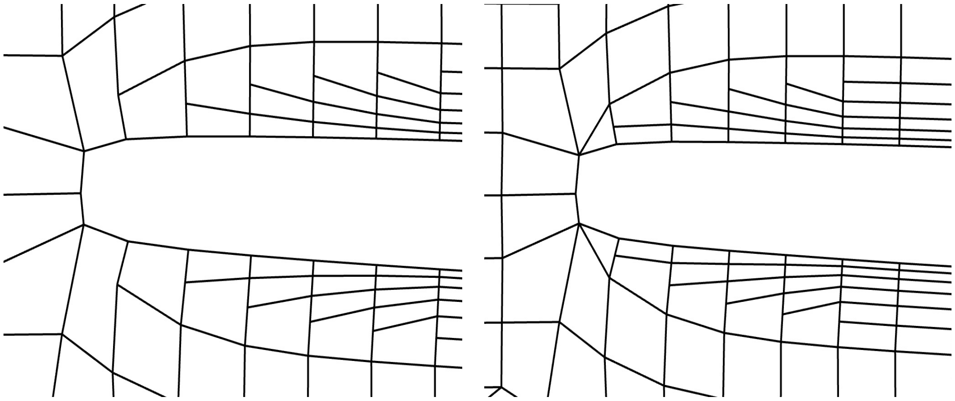


Figure 7: Details of two layer termination strategies employed.

2.3: Flow Initialisation and Flow Modelling Using DES

Both setting up the boundary condition and initialisation of the flow field can be performed using the GUI described previously. Most commonly used boundary conditions are supported and flow variables can be initialised using boundary smoothed potential flow, fixed values or length scale and turbulent energy factors in the case of turbulent quantities.

OpenFOAM contains a number of flow solvers and turbulence models for performing CFD on different flow regimes. The emphasis of the methodology described in this paper is on incompressible Detached Eddy Simulation, as proposed by Spalart [2], employing a modified hybrid formulation of the Spalart-Allmaras SGS model [3].

Although DES is considerably more computationally demanding than equivalent Reynolds Averaged methods, the approach is generally more accurate due to the fact that it directly calculates the larger scales of turbulent motion. This improved accuracy is especially important in the wake of bluff body flow as are commonly encountered in aerodynamic applications.

The standard DES solver in OpenFOAM was further modified to improve the stability and range of applicability of operation. Deferred correction and the implementation of non-oscillatory low diffusivity convection schemes have greatly enhanced the quality and stability of the solver. The addition of CFL-adaptive numerical schemes, which maintain overall accuracy despite high local Courant numbers, and improved parallel communications scheduling and compression have significantly reduced the computational cost of a typical DES simulation.

2.4: Post Processing

Within OpenFOAM there is support for a number of third party post-processing packages. Currently the flow solutions can be exported to Fieldview, Ensight and Paraview formats. The latter is an Open Source visualisation package which is capable of being run in parallel on distributed and shared memory machines, allowing all CPU intensive stages in the CFD process to be performed in a distributed manner.

In addition to these tools, dedicated post-processing functions have been developed within the OpenFOAM framework to record statistics of important quantities, such as: drag, lift, pressure coefficients, etc. This data is often employed for monitoring solution convergence and extracting valuable engineering information.

3: DES SIMULATION ON A BUILT ENVIRONMENT: THE SEARS TOWER, CHICAGO

A realistic CFD model of the Sears Towers and surrounding buildings was prepared in order to demonstrate the Open Source based workflow described above. The STL geometry employed is illustrated in Figure 8.

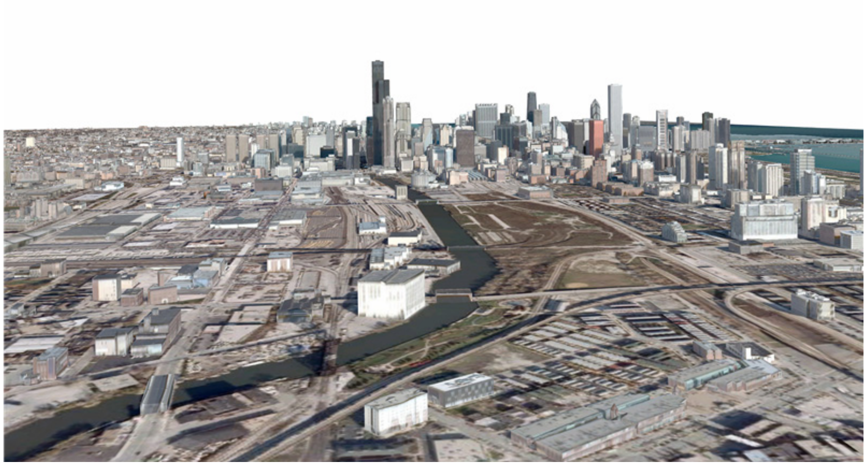


Figure 8: STL geometry of Sears Tower and Chicago (from www.3dcadbrowser.com).

Two computational grids were created in parallel over 8 CPUs following the mesh generation procedure discussed in Section 2.2: a course mesh, consisting of 1.76 million cells, and a fine mesh of 12.5 million cells. In both cases dedicated refinements were introduced around the Sears Tower area to ensure enough resolution for the selected flow solver. Details of the parallel partitioned zones are shown in

Figure 9.

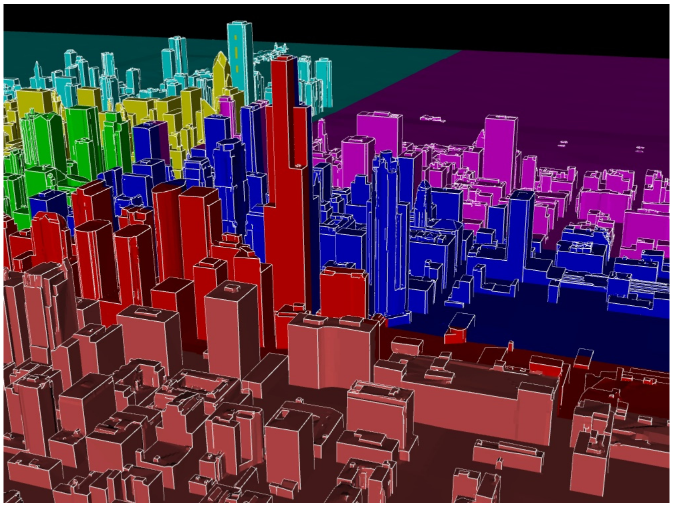


Figure 9: Details of surface mesh generated on the Sears Tower building and surroundings (colour shaded by processor id).

Following the completion of the computational grids, the DES Spalart-Allmaras solver was employed to obtain a first flow solution on the coarse mesh. This initial solution was then mapped onto the fine mesh to speed up convergence and provide a realistic starting point for the final simulation. The mesh generation, solution and mapping were all automated and performed in parallel, thus removing the need for additional geometry decompositions.

The boundary conditions were set to simulate the events of 12th March 1923, when North Easterly winds reached speeds of 60 mph, the lowest pressure system on record in Chicago. The aim was to determine the wind loads response on the Sears Tower for this type of extreme wind conditions.

The wind loads can be easily calculated from the CFD solutions by integrating the resulting pressure and shear forces over the surface of the building, as shown in Figure 10. Since DES can be combined with transient boundary conditions, real wind measurements and gusty winds could also be defined as part of the simulation. Full size models are also possible to challenge wind tunnel measurements and eliminate the Reynolds number limitations imposed by downscaling the geometry.

Furthermore, by directly resolving important turbulent scales, the DES approach not only provides superior accuracy, but also allows the study of flow dynamics and other unsteady phenomena such as flow induced buffeting and aerodynamically generated noise. A snapshot of the instantaneous flow velocity obtained during the simulation on the location of the plane in the dominant direction of wind is shown in Figure 11.

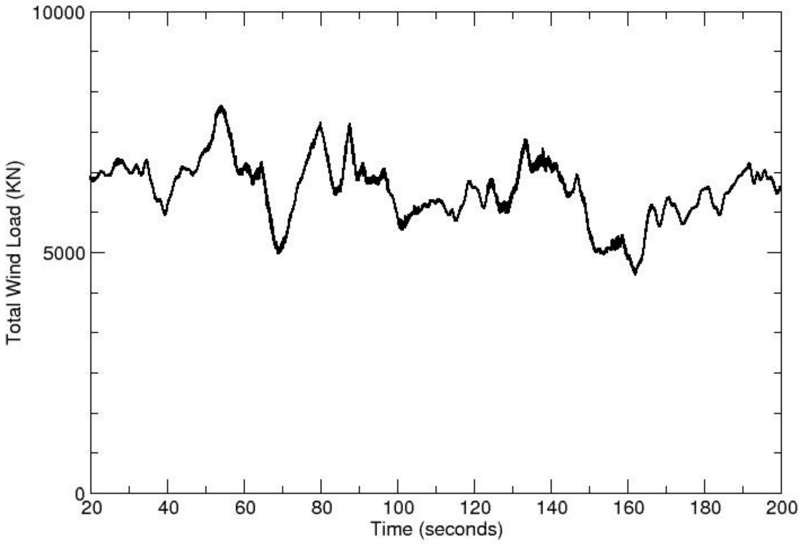


Figure 10: Wind load time history on the Sears Tower from DES solution.

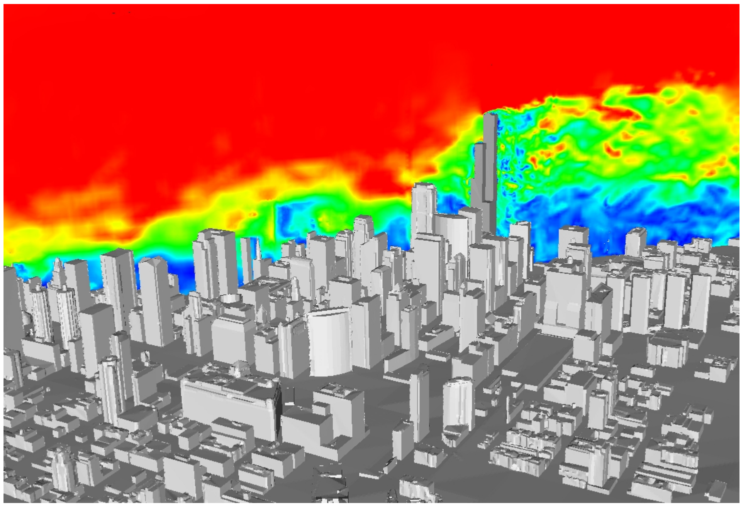


Figure 11: Instantaneous velocities from a DES simulation of the Sears Tower and surrounding buildings in Chicago.

4: CONCLUSIONS

A highly customised and vertically integrated Open Source application stack for the simulation of complex built environments has been presented. All stages of the outlined CFD process support distributed computation. Taken in combination with the optimisations performed to improve computational efficiency, this allows highly accurate DES modelling on very large data sets, limited only by hardware resource. This has the potential of substituting, or at least reducing, the dependency on wind tunnel measurements for evaluating wind loads on large exposed buildings and structures.

The Open Source nature of the tools and the generality of the OpenFOAM libraries also allows for a large degree of customisation, so that the same approach presented here could easily be transferred to different application areas like heating cooling and ventilation and computational aero-acoustics.

REFERENCES

1. ASCE. **Wind loads for buildings and other structures**. PDF document. http://www.asce.org/pdf/windloads2.pdf. Access on February 02, 2009.

2. P.R. Spalart, W-H. Jou, M. Strelet, and S.R. Allmaras. **Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach**. Advances in LES/DNS, 1998.

3. P.R. Spalart and S.R. Allmaras. **One-equation turbulence model for aerodynamic flows**. Proc. AIAA ASM, 1992.

ACKNOWLEDGEMENTS

The authors would like thank OpenCFD Ltd. for their collaboration in developing the OpenFOAM meshing and solver utilities discussed in this paper.