IV International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2015
E. Oñate, M. Bischoff, D.R.J. Owen, P. Wriggers & T. Zohdi (Eds)

VORTEX PARTICLE METHOD FOR AERODYNAMIC ANALYSIS: PARALLEL SCALABILITY AND EFFICIENCY

K. Ibrahim[1], G. Morgenthal[2]

[1] Bauhaus-University Weimar
Marienstrasse 13, 99423 Weimar, Germany
khaled.ibrahim.tolba@uni-weimar.de

[2] Bauhaus-University Weimar
Marienstrasse 13, 99423 Weimar, Germany
guido.morgenthal@uni-weimar.de

Key words: Vortex Particle Method, Wind Structure Interaction, Bluff Body Aerodynamics, Aero-elastic Response.

Abstract. This paper presents an analysis of the scalability and efficiency of a simulation framework based on the Vortex Particle Method (VPM). The code is applied for the numerical aerodynamic analysis of line-like structures. The numerical code runs on multi-core CPU and GPU architectures using OpenCL framework. The focus of this paper is the analysis of the parallel efficiency and scalability of the method being applied to an engineering test case, namely the aero-elastic response of a long-span bridge girder. The target is to assess the optimal configuration and the required computer architecture, such that it becomes feasible to efficiently utilize the method within the computational resources available for a regular engineering office. The simulations and the scalability analysis are performed on a regular gaming-type computer.

1 Introduction

The Vortex Particle Methods (VPM) uses a grid free formulation in order to solve the Navier-Stokes equations of an incompressible fluid flow. The VPM models vorticial flows typical for aerodynamic applications especially around bluff bodies i.e. moderate to high Reynolds number via solving the vorticity transport equations in Lagrangian form [1, 2]. Vortex sheets are modelled as particles with invariant strength which are transported downstream by convection and diffusion due to the induced velocity field and the free stream velocity. The vorticity field can be computed as superposition of the local vorticity created by each particle. Based on the Bio-Savart law the induced velocity field is computed and accordingly the particles are convected through a suitable time integration scheme.

Different algorithms are available in literature [3] for solving the Bio-Savart integral. For N particles discretisation of the domain, the naive particle-particle interaction method scales as $O(N^2)$. Alternatively, within the current implementation of the VPM method, an enhanced algorithm termed the Particle-Particle-Particle-Mesh (P3M) algorithm is used [4]. This algorithm requires an intermediate correction step that can be efficiently computed in parallel, for which Graphical Processing Units (GPUs) were used in a thread-per-particle implementation. Here, in parallel fashion each thread calculates the contributions on its local particle due to all the other particles at a given point in time [5].

2 Modelling and Simulation

The simulations of the structural aero-elastic response can be divided into two intertwined aspects. Namely, the structural dynamics problem modelled by the interaction between the wind induced forces and the dynamic characteristics of the structure. And the fluid dynamics problem representing the wind flow, modelled by the Navier-Stokes (N-S) equations. For an incompressible viscous flow N-S equations are given by,

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

$$\nabla \cdot \mathbf{u} = \mathbf{0}. \tag{2}$$

Where for time $t \in (0, \infty)$ the vector velocity field $\mathbf{u}$ in three spatial dimensions is $\mathbf{u}(\mathbf{x}, \mathbf{t}) : \mathbb{R}^{\mathbf{3}} \times [\mathbf{0}, \mathbf{t}] \rightarrow \mathbb{R}^{\mathbf{3}}$, while the scalar pressure field is $p(x, t) : \mathbb{R}^3 \times [0, t] \rightarrow \mathbb{R}$ and ν is the kinematic viscosity.
Equation [1] represents the conservation of momentum, where to the left hand side is the material derivative i.e. the rate of change of momentum and to the right hand side are the pressure and viscous forces. Due to the incompressibility condition, the conservation of mass implies that velocity u is a solenoidal field i.e divergence free.

The N-S equations in the form presented above, constitutes four equations of the primitive variables $\mathbf{u}$ and p. Analytical solution of the N-S equations exists only for rare simplified test cases [6]. Hence, simulation of flow phenomena is based primarily on numerical methods.

2.1 Vortex Particle Method

For solving the N-S equations there are multitude of numerical schemes that differ in there formulations as well as there underlying computational schemes. The choice of the suitable scheme is essential for an accurate modelling of the underlying physics, while efficiently using the computational resources. The Vortex Particle Method is proposed as an efficient numerical technique for simulations of high Reynolds number, incompressible flows around bluff bodies e.g. in the context of wind engineering. This mesh-less numerical technique provides an alternative to classical mesh based Eulerian methods such as finite volume and spectral methods. The formulation and the discretisation of

the N-S equations in the VPM, reflect the natural representation in dominantly vortcial flow, which is inherent in flows around bluff bodies. Additionally, the method accounts for free-space boundary conditions, thus providing a robust approach for flows around structures such as towers, masts and bridges [5].
The VPM method is derived based on describing the fluid dynamic events in the flow in terms of the evolution of the vorticity field ω [1] such that,

$$\omega(\mathbf{x}, \mathbf{t}) = \nabla \times \mathbf{u}(\mathbf{x}, \mathbf{t}). \tag{3}$$

For three dimensional space $x \in \mathbb{R}^3 \rightarrow \mathbf{u} \in \mathbb{R}^{\mathbf{3}} \wedge \omega \in \mathbb{R}^{\mathbf{3}}$. Whilst in two dimensional domain $x \in \mathbb{R}^2 \rightarrow \mathbf{u} \in \mathbb{R}^{\mathbf{2}} \wedge \omega \in \mathbb{R}$ where the vorticity has only one component namely the orthogonal component to the $\mathbb{R}^2$.

The motion of an incompressible fluid governed by the momentum and mass conservation equations [1] and [2] might as well be represented in terms of the vorticity transport equation. By considering equation [3] and applying the curl operator to equation [1] it becomes,

$$\frac{\partial \omega}{\partial t} + \nabla \times [(\mathbf{u} \cdot \nabla)\, \mathbf{u}] = \cancelto{0}{-\frac{1}{\rho} \nabla \times \nabla p} + \nu \nabla \times \left[\nabla^2 \mathbf{u}\right]. \tag{4}$$

The pressure term vanishes, because the curl of the gradient of any scalar field is always zero. Similarly, for the left hand side using the vector identity $(\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{\mathbf{1}}{\mathbf{2}} \nabla \mathbf{u} \cdot \mathbf{u} - \mathbf{u} \times (\nabla \times \mathbf{u})$ it follows,

$$\nabla \times [(\mathbf{u} \cdot \nabla)\, \mathbf{u}] = \nabla \times (\omega \times \mathbf{u})\,.$$

For the right hand side of equation [4], Considering the vector identity for the Laplacian of $\mathbf{u}$, where $\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \cancelto{\omega}{(\nabla \times \mathbf{u})}$ and considering the vanishing term due to the incompressibility condition in equation [2] it follows,

$$\nu \nabla \times \left[\nabla^2 \mathbf{u}\right] = -\nu \nabla \times [(\nabla \times \omega)]\,. \tag{5}$$

Thus, the continuty equation $\nabla \cdot u = 0$ is now implicitly included. Following the same vector identity for the Laplacian of ω, where $\nabla \times (\nabla \times \omega) = \nabla \cancelto{0}{(\nabla \cdot \omega)} - \nabla^2 \omega$ and since the divergence of the vorticity ω is always zero. The vorticity transport equation can be written as,

$$\frac{\partial \omega}{\partial t} + \nabla \times (\omega \times \mathbf{u}) = \nu \nabla^{\mathbf{2}} \omega. \tag{6}$$

Since $\nabla \times (\omega \times \mathbf{u}) = \omega \cancelto{0}{(\nabla \cdot \mathbf{u})} - \mathbf{u} \cancelto{0}{(\nabla \cdot \omega)} - (\omega \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\omega$, the vorticity transport equation for $x \in \mathbb{R}^3, t \in [0, t_\infty]$ can be written such that,

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega = \nu \nabla^{\mathbf{2}} \omega + (\omega \cdot \nabla)\mathbf{u}. \tag{7}$$

In the left hand side the first term is the rate of increase of vorticity by time and the second term is the convection of vorticity. In the right hand side the viscous diffusion of vorticity and the rate of vorticity deformation i.e. vortex stretching respectively [1] .

2.2 Simulation framework

The simulation framework is based on a two dimensional cross sectional analysis. In analogy to section model wind tunnel testing, such approach is particularly used in predominantly two dimensional flow simulations as in line-like structures e.g. bridges and high-rise buildings. Nevertheless, as an extension to the concept and to capture the three dimensional flow phenomenon, Pseudo Three-Dimensional simulation proposed by Morgenthal [5] is implemented in the current framework. Which is a numerical approach, where multiple two-dimensional simulations are carried out such that they are linked together via the dynamical properties of the structure.

In the current framework rewriting equation [7]. Since $x \in \mathbb{R}^2 \rightarrow u =< u_x, u_y, 0 > \wedge$ $\omega =< 0, 0, \omega_z >$, therefore the term corresponding to the vortex stretching $(\omega \cdot \nabla)\mathbf{u}$ vanishes. The vorticity transport equation becomes,

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega = \nu \nabla^{\mathbf{2}} \omega. \tag{8}$$

Two benefits of this formulation are the absence of pressure term and the automatic satisfaction of the continuity equation [2]. The equations are now only dependent on vorticity and velocity field. Thus for solving the vorticity field an analytic expression connecting vorticity and velocity at each specific point in time and space is essential.

Based on Helmholtz Theorem [1, 4], the velocity field $\mathbf{u}(\mathbf{x}, \mathbf{t})$ can be resolved into the sum of an irrotational i.e curl-free vector field u_p and a solenoidal i.e divergence free vector field u_w i.e. $u = u_p + u_w$. Assuming a given potential function ϕ for the irrotational field and, a divergence free function ψ for the rotational field, u could be written as,

$$\mathbf{u} = \nabla\phi + \nabla \times \psi. \tag{9}$$

Using the definition of vorticity in equation [3], the vorticity becomes,

$$\omega = \cancelto{0}{\nabla \times \nabla\phi} + \nabla \times (\nabla \times \psi) = \cancelto{0}{\nabla(\nabla \cdot \psi)} - \nabla^2\psi. \tag{10}$$

Since, the curl of the gradient of the potential function is zero and ψ is defined to be a divergence free function, the vorticity yields to be the laplacian of the stream function i.e. Poisson equation.

Considering that, as the domain boundaries extend to infinity the solenodial stream function ψ decays to zero. Hence, the irrotational component becomes $u_p = u_\infty$. And

using the Green's function solution to the above mentioned Poisson equation [10], this yields the velocity field in terms of the so-called Biot-Savart law. Such that,

$$\mathbf{u}(\mathbf{x}) = \mathbf{U}_\infty - \frac{\mathbf{1}}{\mathbf{2}\pi} \int_{\mathcal{D}} \frac{\omega(\mathbf{x_0}) \times (\mathbf{x_0} - \mathbf{x})}{|\mathbf{x_0} - \mathbf{x}|^{\mathbf{2}}} \, \mathrm{d}\mathcal{D}_\mathbf{0}, \tag{11}$$

where $\mathcal{D} \in \mathbb{R}^2$ and the velocity $\mathbf{u}(\mathbf{x})$ at point x in space is the superposition of all the contributions from the vorticity field in the domain.

2.2.1 Discretisation

The vorticity can be discretized in numerous ways e.g. particles or segments in 2D and filaments or sheets in 3D [1]. In this implementation by Morgenthal [4], particles are used for discretisation and are convected with the flow such that the strength of each particle τ_p is the local circulation. whereby circulation is a quantity closely related to vorticity, it is defined as the line integral of the velocity around a closed contour, such that

$$\tau = \oint \mathbf{u}.\mathbf{dx}. \tag{12}$$

Due to this discretisation the vorticity field created by N_p vortex particles (circulation particles) becomes,

$$\omega(\mathbf{x}, \mathbf{t}) = \sum_{\mathbf{p=1}}^{\mathbf{N_p}} \delta\left(\mathbf{x} - \mathbf{x_p}(\mathbf{t})\right) \mathbf{\Gamma_p}, \tag{13}$$

where δ is the Dirac delta function and x_p is the position of particle τ_p contributing to the vorticity field based on the separation distance. Whereby the strength of each particle $\tau_p = \tau_p ez$ remains constant. The velocity at point x in space is thus the superposition of the contributions from all particles in the domain

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{U}_\infty - \sum_{\mathbf{p=1}}^{\mathbf{N_p}} \mathbf{K}_\sigma(\mathbf{x_p} - \mathbf{x}) \mathbf{\Gamma_p}. \tag{14}$$

Singularities arise when two particles are at very small separation to each other, those are mollified by using a smooth function $K_\sigma(\mathbf{x})$, in the current implementation Gaussian mollification kernel is applied [4].

2.2.2 Fluid Structure Interaction

Using Boundary Element Method (BEM), the structural cross section is discretised in $\mathbb{R}^2$. Generaly the BEM uses a given boundary conditions to fit boundary values into an integral equation, rather than values throughout the space defined by a partial differential equation e.g. the Poisson equation equation [10]. Accordingly, the the velocity boundary

conditions of equation [11] are satisfied as well its corresponding surface vorticity. The local vorticity at each panel is calculated and released to the domain at every time step.

For computation of the resultant integral forces on a rigid body, the pressure gradient is computed through,

$$\nabla p = \nu\rho\nabla^2 u \tag{15}$$

where ν is the kinematic viscosity, ρ is the density and u is the velocity field. By integrating the surface pressures the aerodynamic section forces can be directly calculated, based upon which the static wind coefficients are calculated. For drag coefficient,

$$C_D = \frac{F_D}{1/2\rho D U_\infty^2}, \tag{16}$$

where F_D is the drag force and D is the characteristic depth of the cross section. Similarly the cooeficents of lift and moment are calculated from the lift force and the moment respectively.

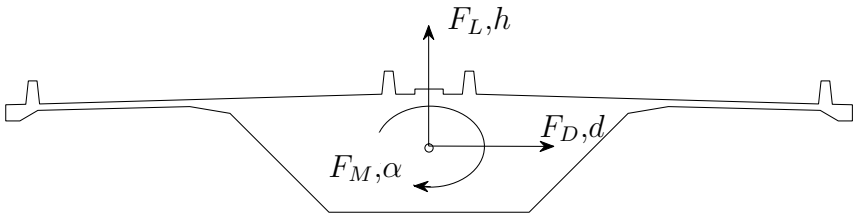

Figure 1: Schematic of a two dimensional bridge cross section with three degrees of freedom; h for vertical motion due to lift force F_l. d for lateral motion due to drag force F_D and rotation α due to moment F_M.

The two dimensional structural displacement response is achieved by solving the equation of motion for each degree of freedom. For the vertical response,

$$m\ddot{h} + 2m\xi_h\omega_h\dot{h} + m\omega_h^2 h = F_h(t), \tag{17}$$

where m is the mass, ξ_h is the damping ratio and ω_h is the natural frequency. Similarly, the equation of motion is solved for the lateral and rotational degrees of freedom with there corresponding damping, natural frequency and mass or inertia.

3 Reference Object

In the following section, the presented numerical method is compared to results from wind tunnel tests provided from an industrial test case. The dimensions of the cross section are presented in figure [2].

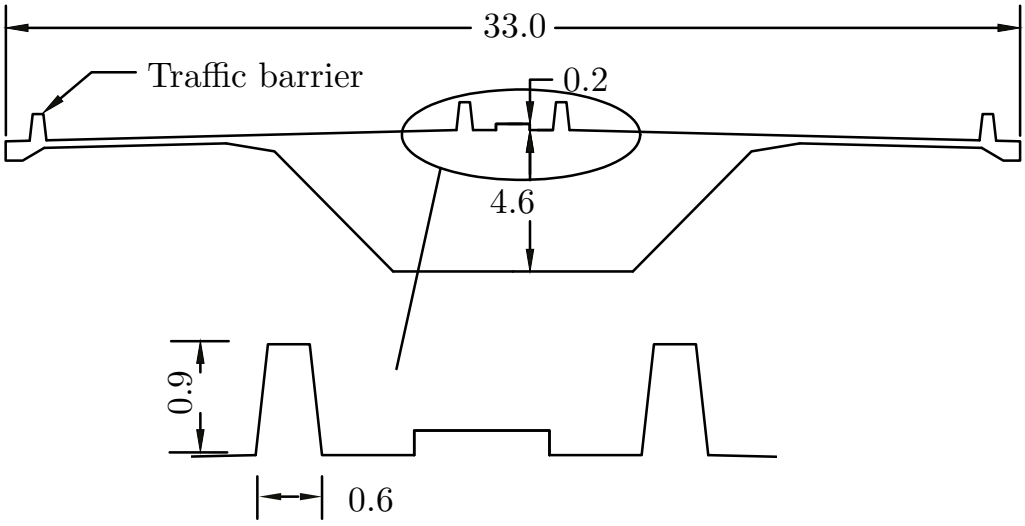

Figure 2: Bridge cross section, dimensions in meter.

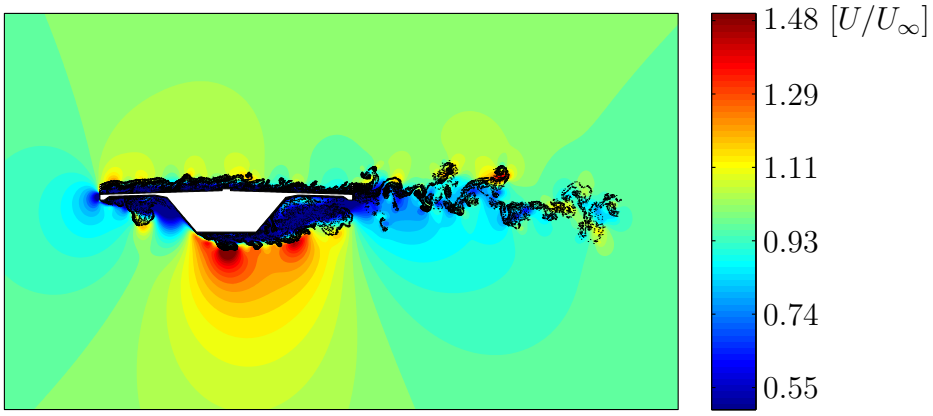

Figure 3: A snapshot of flow field around a bridge cross section using Vortex Particle Method.

3.1 Static wind coefficients

Before pursuing the dynamic response simulations, static section simulations were performed for validation of the basic aerodynamic behaviour e.g. for laminar flow see figure [3]. The static cross section model of the bridge was simulated for varying wind incidence angles α in the range of -7 to +7 degree with 1 degree steps. The static wind coefficients are compared with good agreement against the corresponding wind tunnel results, as in figure [4].

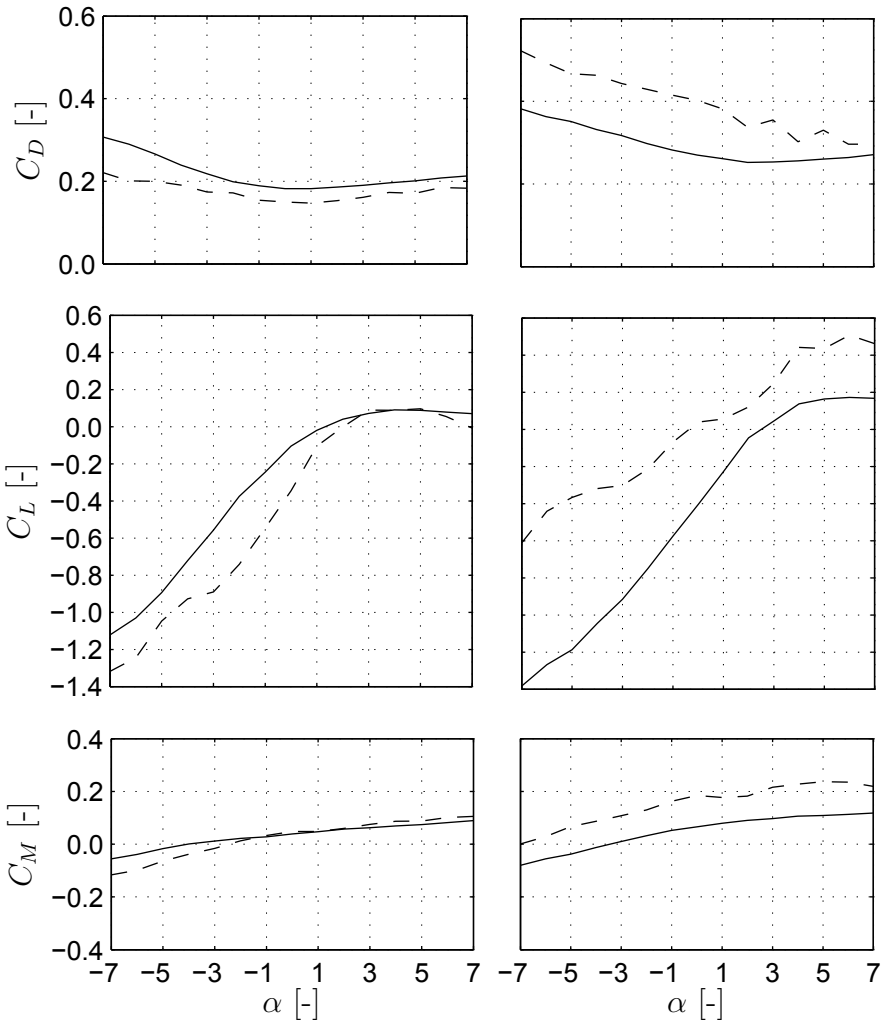

Figure 4: Comparison of static wind coefficients (C_D, C_L, C_M) for different incidence angles α; continues line for wind tunnel results and dotted line for numerical simulation results. To the left smooth wind; to the right turbulent wind with Isotropic Turbulence intensity $I = 10\%$. The simulations were performed at full scale size. The wind tunnel tests for the static cross section were done at the scale of 1:80.

3.2 Aero-elastic response

An aero-elastic model of the bridge at the construction stage of a 216 m long cantiliver i.e. 108 m from each cantilever side is modelled and simulated by a set of six two dimensional simulation slices. The vibration modes of the structure were computed using commercial Finite Element software. The four natural modes used in the analysis (two vertical and two lateral deck modes) are depicted in figure [5].

Simulations of 600s real time were performed for the bridge aero-elastic response due to turbulent inflow wind configuration. The simulations were performed based on two

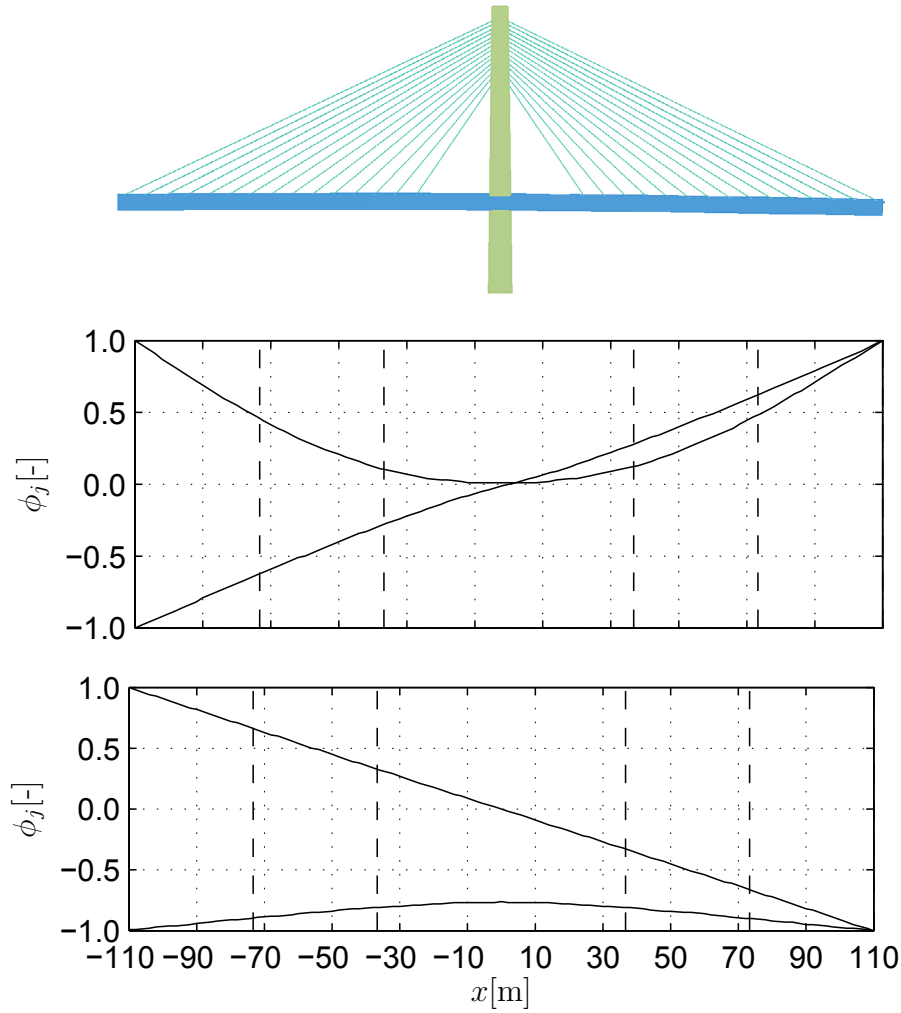

Figure 5: Finite Element model; corresponding vertical and lateral mode shapes, from top to bottom.

methods,

1. CFD using VPM as presented in section [2.2] coupled with modal superposition.
2. For validation, the structural response is computed using a commercial software based on FEM with time variant forces and admittance function to account for the turbulence wind effect on the structure, the turbulence admittance function is presented in [7, 8]. As an input to calculate the wind induced time variant forces, the wind coefficents as function of incidence angel α are provided. Those coefficents, are obtained from previous static CFD simulations e.g. as in figure [4] or from wind tunnel test results.

Model	Lateral [m]	Vertical [m]	rotation [rad]
FEM	0.10	0.39	0.0006
CFD + FSI	0.08	0.25	0.0002

Table 1: Peak displacements at the tip. Comparison between CFD simulation coupled with modal superposition for Fluid Structure Interaction (FSI) Vs. FEM model with time variant forces and admittance function to account for turbulent wind induced forces.

The maximum cantilever tip displacement in vertical, lateral and rotation for both cases is shown as a comparison with the two different methods in Table [1]. The results shows a very reasonable agreement. It is crucial to mention that the commercial FEM software based on modelled wind forces tends to be more conservative (upper estimation of displacements) in comparision to CFD simulation. Since, as a substitute to resolving the complete physics of fluid structure interaction as done in the CFD model, it uses the concept of admittance [7] of the fluctuation spectrum on the structure. A visualisation of three slices used in the CFD simulation is shown in figure [6]

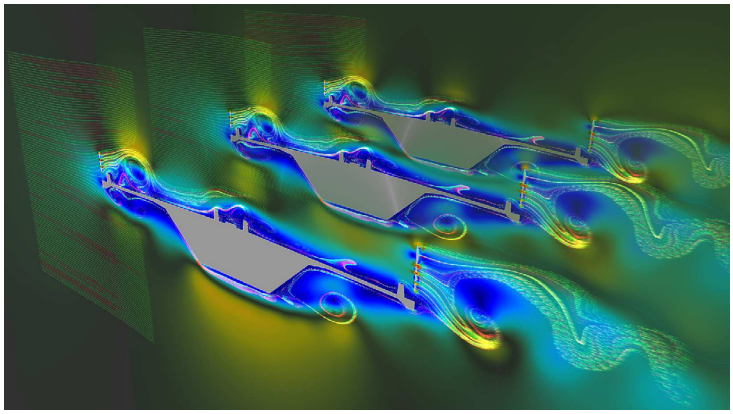

Figure 6: Three slices CFD simulation of the left cantilever side in figure [5]; Pseudo Three-dimensional simulation under turbulent wind.

4 Scaling and Parallel Efficiency

Recalling the concept of pseudo three-dimensional simulation, since each slice could be simulated independently at each time step OpenMP parallelisation is used to compute each slice in parallel. A synchronization point at the end of the time step is essential for coherence between the CFD solutions and the structural dynamics solution [5, 9]. For the local particles interaction computations, Graphical Processing Unit (GPU) is used in a thread per particle manner to compute the heavy arithmetic workload share of the algorithm. The implementation was realised by Morgental, Corriols and Bendig using Open Computing Language (OpenCL), allowing to execute the code on different

hardware architectures namely Central Processing Units (CPU) as well as GPU's. The memory fetching tasks and less arithmetic intensive operations are done on CPU, whilst the arithmetic computations are done in parallel using the GPU threads. The OpenMP strategy is designed such that each simulation slice utilizes one CPU core while the GPU kernel resources are shared between the different cores.

This section is concerned with evaluating the scalability and parallel efficiency of the simulation framework. Scalability is the ability of the algorithm to handle a growing amount of work based on a given parallelisation strategy [10]. Generaly in high performance computing two common scalablity measurements are used. Namely,

- Weak scaling, as a metric to how the solution time varies with the number of cores for a fixed problem size per CPU core.
- Strong scaling, as a metric to how the solution time varies with the number of cores for a fixed total problem size i.e. as the number of CPU cores increase, the load per core decreases.

In order to normalise the absolute run time for different problem size, MPUPS (Million Particle Updates Per Second) is defined as a measuring quantity. Such that,

$$MPUPS = \frac{nParticles \times nIterations}{Calc.Time \times 10^6} \tag{18}$$

Where $nParticles$ is the total number of particles, $nIterations$ is the number of computational iterations over the simulation kernel and $Calc.Time$ is the wall clock time for each simulation run. The number of iterations is kept constant as 1000 steps for all problem sizes and processor counts.

4.1 Weak Scaling

For a fixed problem size per CPU core, namely 2.5 $\times 10^5$ particle per core. The weak scaling is done for number of CPU cores (Computing slices) from one to four. Corresponding to 2.5 $\times 10^5$ up to 1 $\times 10^6$ particles. It is shown in figure [7] a nearly ideal weak scaling where the performance almost stay constant for a constant problem size per node. The slight drop in the performance starting from three cores, can be explained due to the serialization time that is introduced when the CPU'S share the GPU unit for the particle update computations. This idealisation time happens because after 5 $\times 10^5$ particles the GPU needs to be shared in serial manner between the different CPU cores. Using GPU with more threads and/or Multi GPU's would postpone the occurrence of the serialization instance to a higher number of cores i.e. simulation slices, because the increase of the capacity would allow sharing the GPU kernel for more number of particles.

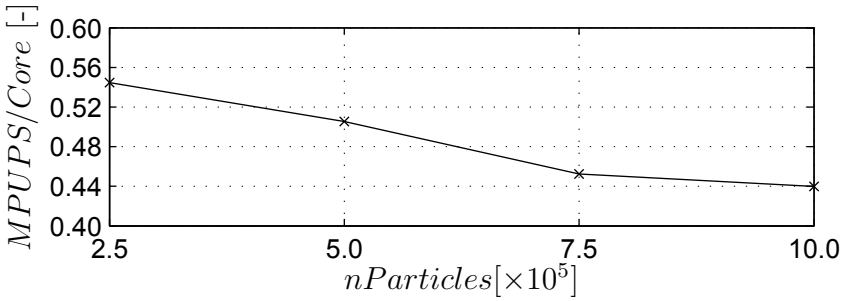

Figure 7: Weak scaling of simulation framework; representing variation in the computational efficiency per CPU core ($MPUPS/Core$) versus total number of particles ($nParticles$). As the number of particles increase the number of cores increase as well i.e. fixed problem size per CPU core.

4.2 Strong Scaling

The strong scaling is performed for a total problem size of 5 $\times 10^5$ particles. Using this approach the GPU resources have to carry the same computation once requested by one CPU core and later on requested by two, three and four cores. It is seen from figure [8] that the total computation efficiency is increased between one and two cores, later on slightly reducing. This can be explained by the time spent in moving the data between

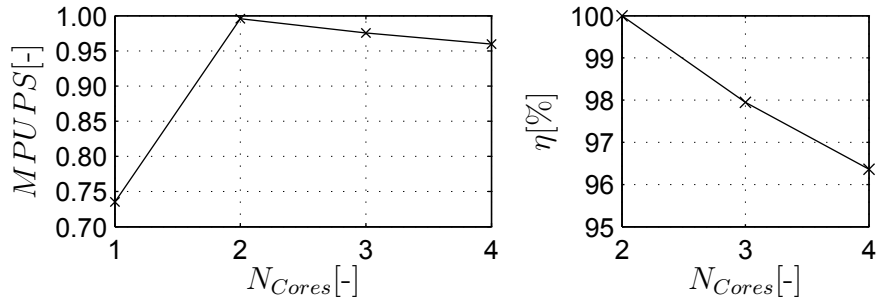

Figure 8: Strong scaling (left) and Parallel efficiency η (right) of simulation framework; Strong scaling represents the total computational efficiency ($MPUPS$) as the a problem with a fixed size is distributed on different number of cores (N_{Cores}) i.e. for increasing number of cores the load per core is decreasing. Computational efficiency is calculated relative to the optimal number of cores for a given problem size, for this case optimal N_{Cores} is 2 for $nParticles = 5 \times 10^5$.

the hosting CPU core and the GPU device memory. As the number of cores increase even for the same number of particles it is expected that the performance would drop due to the time cost of the data transfer.

4.3 Parallel Efficiency

Considering two simulation slices i.e. CPU cores as the reference point to evaluate the computation efficiency with relation to the drop down due to the data transfer. The parallel efficiency as shown in figure [8] is computed from the strong scaling, showing a very slight drop as the number of cores increase. It is crucial to remark that strong scaling as well as the parallel efficiency was done for a problem size 5×10^5 particles, the value previously concluded from the weak scaling to be optimal before the effect of the serial sharing of the GPU resources takes place. Hence, Strong scaling and Parallel efficiency measurements could be used to analyse the computation versus data transfer efficiency.

5 Summary and Conclusion

Vortex method has been presented along with its formulations and assumptions reaching to the vortex transport equation and the two dimensional Particle Vortex Method for CFD simulations. Based upon this numerical approach a simulation framework efficiently parallelised for CPU's and GPU's using OpenCL was used to simulate a test case the aeroelastic response of a bridge in the construction stage. The results were compared versus wind tunnel as well as other simulation approach.

Finaly, the scaling and the parallel efficiency of the utilised simulation framework has been presented. The simulation approach presents a computationally very efficient yet accurate method that can be used to study numerous wind engineering problems related to the behaviour of line-like structures of complex geometries exposed to natural wind conditions.

REFERENCES

[1] Cottet, G.-H., and Koumoutsakos, P. Vortex Methods: Theory and Practice. Cambridge University Press, New York, 2000.

[2] Walther, J.H., Larsen, A., 1997. Discrete vortex method for application to bluff body aerodynamics. J. Wind Engng. and Industrial Aerodynamics 6768,1831

[3] Appel, A.W., 1985. An efficient program for many-body simulation. SIAM J.Sci. Stat. Comput. 6, 85103

[4] Morgenthal, G., 2002. Aerodynamic Analysis of Structures Using High- resolution Vortex Particle Methods. Ph.D. thesis. University of Cambridge.

[5] Morgenthal, G., Corriols, A.S., Bendig, B., 2014. A gpu-accelerated pseudo-3d vortex method for aerodynamic analysis. Journal of Wind Engineering and Industrial Aerodynamics 125, 69 80

[6] Von Wendt, J.F. (Ed.). Computational Fluid Dynamics: An Introduction. New York: Springer-Verlag, 1996.

[7] Xu, You-Lin. Wind effects on cable-supported bridges. John Wiley & Sons, 2013.

[8] Rasmussen, J., Hejlesen, M., Larsen, A., Walther, J., 2010. Discrete vortex method simulations of the aerodynamic admittance in bridge aerodynamics. Journal of Wind Engineering and Industrial Aerodynamics 98, 754 766.

[9] Morgenthal, G., Kovacs, I., Saul, R., 2005. Analysis of Aeroelastic Bridge Deck Response to Natural Wind. Structural Engineering International (IABSE) 15, 232235.

[10] Hill, Mark D. (1990). "What is scalability?". ACM SIGARCH Computer Architecture News 18 (4): 18