J. Wind Eng. Ind. Aerodyn. 144 (2015) 14–23

Contents lists available at ScienceDirect

Journal of Wind Engineering and Industrial Aerodynamics

journal homepage: www.elsevier.com/locate/jweia

Recent advances in computational wind engineering and fluid–structure interaction

Rainald Löhner [a,*], Eberhard Haug [b], Alexander Michalski [b], Britto Muhammad [b], Atis Drego [b], Ramakrishna Nanjundaiah [b], Raham Zarfam [b]

[a] CFD Center, M.S. 6A2 College of Science, George Mason University, Fairfax, VA 22030-4444, USA
[b] SL Rasch GmbH, Kesslerweg 22 Oberaichen, Germany

ARTICLE INFO

Keywords:
Butterfly effects
Rogue loads
Adjoint estimation of boundary conditions

ABSTRACT

Recent developments that are pertinent to the particular field of computing lightweight structures exposed to windloads are described. The topics covered include computational fluid dynamics (CFD), computational structural dynamics (CSD) and fluid–structure interaction (FSI) for wind vs. aerospace engineering, recent hardware and software trends, butterfly effects, rogue loads, and adjoint estimation of boundary conditions.

1. Introduction

The analysis and design of lightweight structures exposed to windloads requires an accurate knowledge of the ensuing flow-fields and their effect on structural loads and deformations. Time and again structures of this kind have failed catastrophically, highlighting the need to continuously improve the capabilities to calculate all relevant physical mechanisms involved.

All computational building blocks required, computational fluid dynamics (CFD), computational structural dynamics (CSD) and fluid–structure interaction (FSI) techniques, have progressed rapidly over the last three decades. A number of excellent reviews summarize the state of the art in CFD (Cochran and Derickson, 2010; Tamura, 2010; Blocken, 2014). The aim of the present paper is to give a report on recent developments that are pertinent to the particular field of computing lightweight structures exposed to windloads. The first topic is concerned with the question as to why computational wind engineering as applied to lightweight structures has lagged by almost 40 years similar developments in the aerospace and naval sectors. The second topic covers recent hardware and software developments and their consequences for future field solvers. The focus then shifts to two phenomena that have been discovered and debated recently: butterfly effects and rogue loads. Both imply much higher CPU demands than expected, highlighting the need to continuously improve field solvers. The final topic is the wide field of boundary conditions. In particular, the questions of the required accuracy as far as geometrical (how many buildings? how far out?) and wind profiles at inflow boundaries are considered.

2. Wind vs. traditional aerospace engineering

The use of computational fluid dynamics (CFD) and fluid–structure interaction (FSI) to predict the behavior of lightweight structures has lagged similar developments in aero- and hydro-dynamics. A number of mundane reasons or excuses for this may be offered:

- Large, state-funded research centers for aero- and hydro-dynamics that dwarf similar institutions for wind engineering;
- large, centralized enterprises for the production of airplanes and ships as compared to small engineering bureaus for wind engineering;
- heavy reliance on simple, rule-based norms and/or norms based on ‘this is how we always used to do it’ in civil engineering as compared to ‘we have no norms because we never did anything like this before’ in aerospace engineering.

The main reason, though, is that the requirements placed on computational fluid dynamics, computational structural dynamics and fluid–structure interaction techniques for lightweight, civil engineering structures are much higher than those for typical aerospace engineering. This may come as a surprise to many, but consider the following:

- Aerospace bodies are typically streamlined. This implies that

* Corresponding author at: CFD Center, M.S. 6A2 College of Science, George Mason University, Fairfax, VA 22030-4444, USA.

http://dx.doi.org/10.1016/j.jweia.2015.04.014
0167-6105/

one can obtain accurate engineering predictions using the Reynolds-Averaged Navier–Stokes (RANS) equations, i.e. solving a steady flow problem where the resolution of the boundary layers is only in the direction normal to walls. Lightweight structures (and many other civil engineering structures for that matter) subjected to wind loads exhibit a very unsteady incoming flowfield and, in most cases, regions of massive flow separation. This implies that one has to start with Large-Eddy Simulations (LES), i.e. unsteady simulations that require fine, isotropic grids for the lengthscales one is trying to resolve. Furthermore, the flowfield is heavily influenced by the surrounding buildings and geography, implying that one has to consider in the computational model not only the structure whose behavior is to be predicted, but also many other objects – which in turn increases geometrical complexity and leads to larger gridsizes. As the vortices shedded from all buildings interact, the region in between must also be meshed with fine, isotropic grids.

- Deformations of aerospace bodies are mainly linear elastic. This implies that one can formulate the structural response in terms of eigenmodes with small deformations. Indeed, aeroelastic calculations in the aerospace industry are performed mostly with specialized small deformation boundary conditions which eliminate the need to move and/or deform grids. Lightweight structures, on the other hand, can behave very nonlinearly. Tents and parasols in particular not only exhibit large jumps in stiffness when deformed, but also have fasteners, dampers, shutters, and many other mechanisms that behave nonlinearly. The computational tools required to model such behaviors are typically explicit, nonlinear structural dynamics solvers that require orders of magnitude more compute power than modal integrators.
- While the flutter behavior of aerospace structures can be obtained by coupling RANS (for CFD) and modal integration (for CSD), the response of lightweight structures subjected to wind loads requires the coupling of LES (for CFD) to nonlinear FEM codes (for CSD). Moreover, the allowable timestep in the CFD and CSD codes may be very different, prompting the need for sophisticated fluid–structure integration schemes.
- While the timescales of aerospace structures are in the range of seconds, the timescales in many lightweight structures are in the range of minutes. The wind is always unsteady, and one requires large integration times in order to obtain reliable statistics (see below the section on rogue loads). It is not uncommon to have coupled fluid–structure interaction calculations for lightweight structures that require 20–30 min of physical time.

In summary, one can see that the main reason why computational predictions in civil engineering have lagged those in the aerospace and shipbuilding industries is found in the more complex physics of the flow and the structure. These in turn lead to mesh requirements, timestep counts, and ultimately, hardware requirements that are much higher. Even after five decades of continuous improvements in computing hardware, the prediction of wind–structure interactions requires weeks on several hundred cores. It is therefore not surprising that at present it is seldomly used in practice.

3. Hardware and software trends

Scientific computing in general and supercomputing in particular have taken advantage of a remarkable combination of advances over the last three decades: on the one hand the number of transistors per area has doubled every 18 months (so-called Moore's law), and clockrates increased by two orders of magnitude. For programmers and users, this perfect combination led to an almost utopian environment: one could safely assume that without any further effort, the speed of codes would automatically increase due to clockrates and larger register/cache/memory, and larger problems could be run due to larger memory. However, due to physical limitations, these trends could not continue unabated. Given that heat generation increases with the third power of the clockrate, a clear limit is in sight here. In fact, anyone who has inspected a recent motherboard can testify that most of the mechanical engineering effort is devoted to chip cooling. Over the last five years, clockrates have stalled at 2.0–3.0 GHz, and all indications are that, if anything, they will decrease in the future. As far as packing more transistors per area, indications are that Moore's law will continue for the foreseeable future (a decade). The only way to higher CPU performance (loosely speaking: more floating point operations per second [FLOPS]) is then via massive parallelism. This can be achieved (and is pursued) at the level of the chip (either via many cores or via specialized hardware, e.g. GPUs), via a network of chips, or via a combination of both approaches. In fact, most of the Top 500 supercomputers at present use a combination of this kind to achieve outstanding performance.

The biggest problem facing supercomputer designers (and scientific programmers) is that the speed increases of subcomponents continue to advance at very different rates: peak processor speed advances much faster than memory transfer rates, which in turn advance much faster than DRAM latencies, which in turn advance much faster than the interconnect switches between processors. This so-called 'red shift' has led to a crisis for computer architects: current designs are driven by complex latency-hiding mechanisms.

Field solvers, which are commonly used for computational fluid and solid mechanics as well as electromagnetics, only perform a limited number of operations for the data that is brought in and out of memory. The gains obtainable via better compilers, prefetching, mixing floating point operations, memory fetches, etc. have already largely been exploited during the last decade. This implies that for field solvers, this 'red shift' is particularly worrisome.

3.1. Access to RAM

The speeds of CPUs have advanced to such a degree that at present field solvers are limited by the access speed to RAM. Given the number of accesses to memory per timestep, the speed of a field solver can be estimated quite accurately. This observation has been documented repeatedly (see, e.g. Löhner et al., 2014), and can also be observed in the comparison of speeds achieved between CPUs and GPUs (Corrigan et al., 2011, 2012; Löhner, 2013; Löhner et al., 2013). The aim that was pursued for several decades: obtain the highest accuracy while minimizing floating point operations has thus been supplanted by the new aim: *obtain the highest accuracy while minimizing memory access*.

3.2. Access to out-of-processor data

Fig. 1a–c, taken from Löhner et al. (2012) shows timings from runs of up to 50,000 cores. The code used is FEFLO, and a simple explicit time-marching scheme Löhner (2012) is employed.

Fig. 1b, c shows a compilation of timings obtained for different numbers of domains and cores per domain on two different machines (Intel and AMD hardware). Note that once a core deals with more than `nelem=300,000`, CPU performance asymptotes out to the values expected for the chips used in these machines ($O(10^{-6})$ (s/elem/step) or $O(0.55{\cdot}10^{-5})$ (s/pt/step)). *However, for a very small number of elements per domain, there appears a constant delay per timestep of* $T_{MPI} = O(0.1)$ (s). Given that we have $O(10^2)$ MPI messages/exchanges per timestep, it appears that every time a message is started via an MPI call, a latency of $T_l = O(10^{-3})$ (s) is incurred.