Editorial opinion: public dissemination of raw turbulence data

Juan A. Sillero and Javier Jiménez

School of Aeronautics, Universidad Politécnica de Madrid, 28040 Madrid, Spain

E-mail: `sillero@torroja.dmt.upm.es, jimenez@torroja.dmt.upm.es`

Abstract. Many of the papers in this issue deal with processing of pre-existing large-scale turbulence data. We argue here that there is a certain urgency to the discussion of whether raw data should be made publicly available within the turbulence community, and of which are the best procedures, technology and rules for possible dissemination. Besides expressing the personal opinion that such sharing would be advantageous for the field, the urgency mostly arises from the danger that funding agencies and other institutions would otherwise set standards without proper community input. The experience of the Madrid School of Aeronautics with the dissemination of numerical simulation results is briefly reviewed, including the present technological solutions and usage statistics.

1. Introduction

Perhaps because of the natural tendency of engineering research to proprietary information, the traditional working model of fluid mechanics has been that data are generated and analysed within the same group. Data bases have always existed, and professional societies have specially contributed to generate and maintain them, but they usually contain processed results from which it is difficult to extract anything beyond what had been anticipated by the data compiler. The difference between raw data and what is eventually made public has worsened with the appearance of increasingly large simulation and experimental data sets. These are usually archived, but because they are bulky and difficult to read, they are often lost without further use. In some cases in which several interesting computations or experiments have been performed within the same institution, workshops and personal contacts have provided a mechanism for the use of unprocessed data by other groups, but the democratization of computer power, with data being generated at many places around the World, is making that model increasingly cumbersome and hard to justify.

Other communities have used a different working method for some time. High-energy physics, meteorology and molecular biology come to mind as examples in which there are specialists who generate data and others who exploit them, collaborating by means of more or less public data bases to which the former contribute and from where the latter draw their raw material. The process of collaboration is not trivial, and each community has developed its own protocols for data interchange, archiving, quarantine periods and attribution. The loss is a certain 'personal' way of doing science, which should probably not be completely abandoned, but the gain is a greater freedom in what can be observed, and the greater use to which the data can be put. An instructive case is astrophysics, where the division of labour dates at least to the analysis

and publication by Kepler of Tycho Brahé's data in 1627. This eventually led to his planetary laws, and probably to Newton's dynamics, but it was not free of controversy; Kepler had to fight Tycho's heirs for ownership of the data, and eventually paid for the publication himself [1]. Astrophysical data are today mostly in the public domain under well-defined rules and formats.

Whether fluid mechanics should follow this model is open to discussion, and we believe that it should be discussed. Many groups make information public in their private websites, but there is no standard way of disseminating the availability of new data, no generally agreed way of storing or reading them and, in truth, little incentive to prepare the data for general use. An important problem is long-term curation and availability. In our experience, large data sets have a useful lifetime of around ten years, after which they are easier to recompute or remeasure than to retrieve from a database. During this time, they have to be maintained and, occasionally updated or reformatted for purposes that were not anticipated when they were first stored. Most obvious are changes due to the evolution of the computational environment, and of the balance between central and user resources. An example could be the change in graphics requirements during the past few years due to the introduction of cheap graphic processors. As a forwards-looking example, the possible impact of cloud computing on future data use is unpredictable.

Even if limited, ten years is longer than the duration of most research contracts, and it is unclear how the cost of maintaining and preparing data is to be supported after the contract expires. Moreover, large simulation or experimental data have peculiarities that are in many cases only understood by their originator (typically a student or postdoc), and these constraints may be forgotten once the originator leaves the research group. For example, if the results of a spectral code are differentiated using low-order finite differences, the resulting errors are often unacceptable. Again, the useful lifetime of a data set typically exceeds the span of a thesis or a postdoctoral stay.

Broadly speaking, there are two models. The first one is a service centre offering data and on-site postprocessing facilities for a particular kind of problems [2]. The second is a more unstructured facility where data are stored and indexed with minimal on-site processing capabilities [3, 4, 5]. The former are very useful for the purpose for which they are developed but, in our personal experience, tend to be cumbersome when they are used for a very different purpose. They are also critically subject to the long-term funding issues mentioned above, especially if they are restricted to a particular problem. The unstructured services typically shift more of the required computer power to the side of the user. They also require continuing funding, but they are broader and easier to justify. The curation problem is also different in both cases. While the first one typically requires the continued attention of somebody who is familiar with the fluid mechanics of the problem, the second, if properly documented, only requires computational expertise. Thus, the first model tends to be permanently associated with the group that generated the data while, in principle, the second could be outsourced to a more generic institution.

There is a certain urgency to the discussion of whether and how our data should be shared. An increasing number of funding agencies and scientific journals are beginning to insist on open public access of research results (while others specifically forbid it, and the effect of intellectual property laws is uncertain). There are increasingly common warnings from the agencies that, if we do not decide on a viable dissemination policy and format, they will do it for us without regard to any issues that may differentiate our application from others. On the other hand, several supranational initiatives are starting to form for the long-term storage and standardised access of large scientific data [6, 7], and the time to start exerting our influence over them as a community is probably now.

2. The data base of the Universidad Politécnica de Madrid

To illustrate these concepts, we will briefly describe our experience with the public databases hosted by our research group, which mostly contain results of large-scale fluid-mechanical simulations. There are two types of data sets. The first ones are large collections (~ 100 TB) of raw flow snapshots. They may include as many as 10^5 moderately large binary files (~ 1 GB), fewer ($\sim 10^3$) larger ones (~ 100 GB), or a mixture of both. The files can often only be used with the original numerical code that generated them, which is often unique to each set. These data are routinely mined by local users in a 'write once, read many' fashion, over a characteristic period of about ten years. Once checked, cleaned, and after initial publication from the data originator, most of them are made public without any particular access control [5]. The necessary codes are also made public under an open-source 'as is' license.
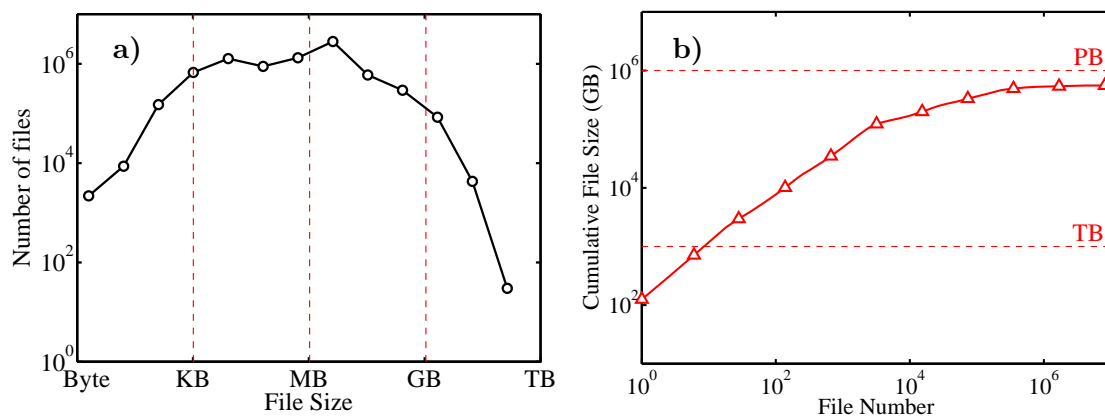

Figure 1. File size distribution in our current data centre storage. (a) Histogram of the file size distribution. (b) Sorted cumulative file size (GBytes) as a function of the total number of files.

The second type of files are obtained by postprocessing the original ones. They include flow fields expressed in primitive variables in physical space, rather than in the typically reduced set of variables and spectral representations of the original files. Another common type of postprocessed results is the temporal evolution of 'objects' defined in particular ways, some of them obtained on-the-fly during the simulations. The accumulated size of these data is typically several times larger than the original ones, and they are not intended for unrestricted public use. Although they are occasionally provided to outside researchers on the basis of personal collaborations, they are not carefully curated, and often not even well documented.

Some statistics of the file size are shown in figure 1. Figure 1(a) is a histogram of the size distribution of our current files, which we believe to be typical of those used in turbulence research. Figure 1(b) presents a compilation of the sorted cumulative file size versus the number of files. The figures show that there are $O(10^5)$ files larger than 1 GB, accounting for 65% of the total used storage (≈ 550 TB). Traditional data bases are of limited use for this application. The size of a typical transaction is larger than those in most commercial services, such as financial ones, personal records, etc., whose characteristic entry size is a few KB, or even than those in scientific data bases, such as particle physics or genomics, where the characteristic entry size is a few MB.

Our postprocessing infrastructure is optimised for price/performance, but also, most importantly, for easy of maintenance. Besides larger computational clusters dedicated to simulations, the core of the system is a collection of Linux servers with relatively large memory (≥ 128 GB). They are connected to a dedicated NFS (Network File System) server attached

to a centralized NAS (Network Attached Storage) that requires a relatively powerful front-end with substantial processing capabilities. The hardware reserved for the open-access database is a high-density commodity JBOD disk enclosure with SAS interface, accepting sixty 3.5" drives in a 4U rack space. Because of the relatively large size of the files, they are mostly accessed sequentially rather than randomly, and our solution uses exclusively traditional mechanical 4 TB hard drives. Moreover, to test the long-term behaviour of a 'low-cost' storage system with decent performance, we have limited ourselves to SATA drives instead of the more expensive SAS ones.

This choice implies losing some of the benefits of the SAS interface, such as better command queuing and error checking algorithms, full duplex communications allowing multipath, higher mean time between failure (MTBF), or lower Bit Error Rate (BER). To compensate for this possible degradation of data integrity, we chose a ZFS file system [8] that guarantees integrity at disk-block level. This file system differs from others in being specifically designed to protect the data from silent data corruption [9] caused by multiple factors such as bit rot, current spikes, firmware bugs, etc. ZFS was conceived to control all the layers in a modern file system, from logical volumes down to block devices. For this reason, expensive hardware RAID cards are no longer needed to control the information that goes into each block device. ZFS offers its own software RAID solution using copy-on-write semantics (COW), avoiding problems such as the 'write hole' phenomenon [8] in traditional storage appliances. To protect the integrity of the data in the event of failure of one or more disks, we have configured ZFS in Raidz2 mode, allowing the loss of up to two disks for every group of ten. On top of that, ZFS allows the use of different lossless compression algorithms to perform on-the-fly compression of the data at block level. Depending on the installed capacity, the current price of such an storage system varies from 90 to 120 €/TB, which is within the funding range of individual research groups. It delivers I/O throughputs in excess of 2.5 GB/sec between the front-end node and the JBOD, and about 700 MB/sec to the NFS clients.

Although their large size makes the use of full data set by external researchers difficult, our experience is that downloads of individual files are reasonably common. Practical data transfer rates from our university to other academic institutions are about 1–2 TB per day, although it may be hard to sustain those velocities for several concurrent requests. After about a year of continued use, we have experienced no critical loss of data.

2.1. User interface
Although data dissemination through personal contacts and visits remains important and will probably always remain so, there is an increased demand from external groups for a more impersonal interaction, which could not be satisfied until the system just described was implemented.

At present, the data open to unrestricted external use is about 150 TB, and provides the scientific community with an efficient way to download about 15 different DNS databases of incompressible channel and boundary layer flows at Reynolds numbers $Re_\tau \approx$180–4200. Information about specific databases can be found at http://torroja.dmt.upm.es/turbdata/index. The data base also provides utility tools to convert the data files between formats (e.g., from binary to HDF5), or to manipulate the original data.

Even if the requirements are simple in principle, their execution is challenging for several reasons. Firstly, there has to be a physical infrastructure able to host hundreds of terabytes, although not necessarily as fast as for a local service. This was already discussed in some detail in the previous section. Secondly, any centralized architecture in which one server interacts with many clients can strongly degrade the availability of the files to the local group, requires large bandwidth, and is expensive to run. Finally, it is important to retain the ability to resume partial file downloads when transferring large datasets, saving time and bandwidth resources. Our first attempt was to adapt a peer-to-peer protocol such as BitTorrent to distribute large

scientific files. This protocol benefits from several factors such as integrity, robustness, simplicity, and aggregate bandwidth capabilities, making it a reasonable choice to transfer large amounts of data from a local digital repository to the scientific community via the Internet network. We implemented a Torrent site for part of our databases without finding any unsurmountable technical difficulties. However, we encountered a number of practical ones that discouraged us from continuing its use: many institutions block P2P networks, the BitTorrent protocol listens from a large number of ports in the local machines (associated with the trackers), and it is not straightforward to automatise the indexation of hundreds of thousands of files. Most importantly, the essence of the P2P networks is to share a file with many peers simultaneously, and to complete a download in at most few hours. Neither has proved to be the case with our databases. The observed access pattern is most often a single user downloading a few Terabytes for a couple of days, and rarely there are several users simultaneously accessing the data. This motivated us to consider alternative solutions.

Our presently preferred solution has been to implement a simple but practical file repository using an Apache HTTP server. The files are served to the clients through a dedicated 1 Gbps Ethernet network without impacting on the availability of the files to the local group, which is served by a private high-speed QDR Infiniband network (40 Gbps). The HTTP protocol allows resumption of partial downloads in case of failure, although some local web servers are configured to avoid header range requests, preventing resumes. Our Apache web server is configured to accept such requests, and users should be able to continue unfinished downloads at any time, either using terminal commands like wget, or through a generic web browser.
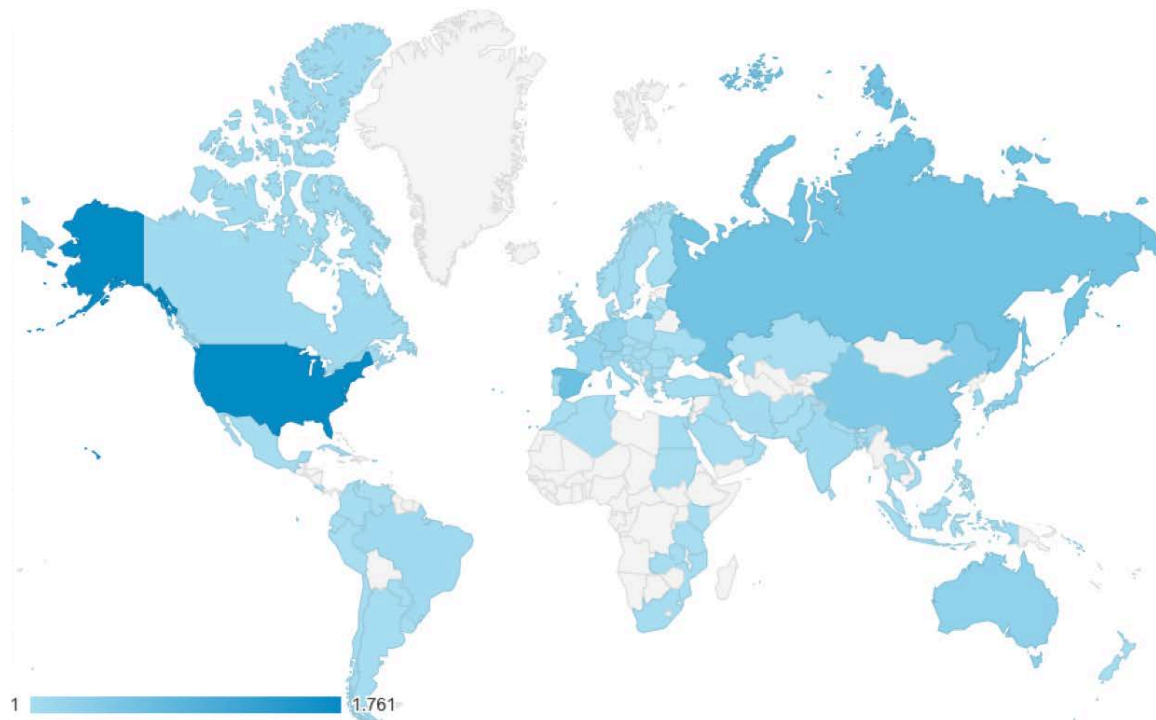

Figure 2. Statistics for the *post-processed* Boundary Layer databases based on the geographical location. Data provided by Google Analytics between October 2013 and November 2015. Source: http://torroja.dmt.upm.es/turbdata/blayers/

2.2. Statistics of outside use of the data

Besides the local data use by members of our group, data analytics collected in the open part of our website show the increasing interest of researches in raw and post-processed turbulence databases, which is also reflected in the increasing number of citations to the associated papers. Figure 2 shows a summary of the audience for the part of our website listing one- and two-point statistics for different turbulent boundary layers, based on geographical location. Since the open repository described above was brought online in February 2015 we have received about 600 visits per month: 48% correspond to direct searches, 40% to referrals, and the remainder to organic searches and social networks. Around 20% of the total hits are return visitors who check our databases often to either download new data or to check for content updates. For instance, 18% of the sessions come from a total of 251 universities from all around the world.

During the same period, the raw-data part of the web page averaged 100–150 sessions per month. About 37% of them correspond to higher education institutions, including more than 40 universities.

These statistics lead us to believe in the potential impact of a web-service that allows full access by the scientific community to raw databases, probably including applications beyond turbulence research.

Acknowledgements

References

[1] Hannan J 2009 *God's philosophers: how the medieval world laid the foundations of modern science* (Icon Books, London) ISBN 9781848310704

[2] Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A and Eyink G 2008 *J. Turbul.* **9** N31

[3] https://www.opensciencedatacloud.org/

[4] http://euhit.icm.edu.pl/services/turbase

[5] http://torroja.dmt.upm.es/turbdata/

[6] https://rd-alliance.org/

[7] http://eudat.eu/

[8] ZFS http://en.wikipedia.org/wiki/ZFS

[9] Panzer-Steindel B 2007 Data integrity Tech. rep. CERN/IT