Editorial

Michael Jeffries PEng

Editorial Advisory Panel member, Consulting Engineer, Surrey, BC, Canada

The ethos of scientific publishing is to provide sufficient information to allow another worker to duplicate the study – that is how the scientific method distinguishes advances in understanding from dogma. If one looks back, say 50 years, geotechnical papers offering methods of analysis or other design formulae were essentially complete – any reader could follow a derivation and re-derive it themselves to fill in any gaps within the printed version (and there were usually gaps, as journal length restrictions favour conciseness over communication). Likewise, test data tended to be a few points plotted on a graph following a full description of the test equipment/procedures. The ‘new knowledge’ was usable directly from the published paper by any reader.

But now consider the situation today, where numerical methods have largely superseded closed-form solutions. Numerical papers will quote equations and procedures, but these are almost invariably incremental as that is why numerics are needed. Numerical methods are the present, and the future, but pity the practical engineer trying to assimilate, let alone use, the work of a numerical paper... As for data, computer-controlled data acquisition is now the norm and the reported results in papers comprise detailed lines for a test, based on commonly hundreds to thousands of points, rather than a summary or representative value – we have gained considerable detail but are not really able to communicate that detail to the reader (okay, in principle readers can digitise a plot but that is hardly a reasonable way forward). The data situation gets worse with, for example, a cone penetration test (CPT) when we can have 3–5 data streams plotted against a common ordinate (depth in the case of a CPT); writing a discussion is severely hindered if a reader cannot access the original data as it is the interrelation between these data streams that indicates soil behaviour. We seem to have lost sight of being a learned profession and the obligation to share – for the benefit of our clients and the wider public – that goes with that. And then we see computer simulations in the news media every day (at least if you watch weather forecasts) – why should we not expect a comparable level of communication in our technical journals? What should be done?

One avenue is for authors to provide supplementary files, ideally on the journal website so as to link to the original contribution, with open-source code, other notes, data and even videos/animations for interested readers. Providing such files offer authors two benefits. First, it greatly assists dissemination (a.k.a. marketing...) and is a great way to get your paper cited – and your work used in practice. Second, the idea that some other engineer will be looking through your code or data will (or, at least, should) push you to better standards of documentation – which you will gratefully appreciate

5 years later when looking through your own work. More generally, the benefit of animations and videos should be obvious; but, if you are in doubt, just consider the impact a single video has had on mining companies’ views about dam safety: see the video on the Fundão tailings dam investigation website (FTDRP, 2016) and related articles in the business, not technical, press.

This exhortation to disclose numerical work and/or data fully is neither new nor restricted to engineering. The Free Software Foundation (FSF, 2016) has been advocating this approach for some 30 years, with the ‘free’ having the same meaning as in ‘free speech’ (as opposed to, say, ‘free beer’). The Linux operating system is based on this principle. And the principle fits perfectly with ‘Open Access’ journals, such as this one, that are doing their best to enhance dissemination of knowledge. An exemplar of the level of open code that can be provided is the listings and files that go with the Smith and Griffith text *Programming the Finite Element Method* (Smith *et al.*, 2013).

What about review of downloadable supplementary files? None. Why? Because the published paper has been reviewed for general adequacy and so on; the downloads go yet further and allow readers direct access to source data or procedures. Of course, authors should expect at least a few readers to go through the downloads in considerable detail – an incentive if there ever was one for authors to ‘dot I’s and cross t’s’.

A downside of open code, from experience, is that some readers will expect too much hand-holding. Open-source software is an invitation to use a shared resource, and that is made clear by the usual practice of releasing open-source software under the GNU License. Of course, most authors are happy to answer questions, but please put in some effort first so that you do not ask something trivial as open-source software, or provision of data files, is not an offer of free support. And, do cite the original work – it is a requirement of the GNU License and, equally important, encourages others to disclose.

This issue of the journal provides some examples of what can be done. The paper by Shuttle and Jefferies (2016) provides (i) source Fortran code, with a user-manual and post-processing spreadsheet for the numerical method used; and, (ii) an Excel application implementing the iterative forward modelling method used in the paper for calibrating soil properties, along with example data and open-source Visual Basic for Applications (VBA) code. Pretty much all of the figures and procedures in the paper can be easily replicated, used in practice and taken further. Hopefully, these examples will inspire you to enhance your contributions similarly.

REFERENCES

FSF (Free Software Foundation) (2016) <http://www.fsf.org> (accessed 24/11/2016).

FTDI (Fundão Tailings Dam Review Panel) (2016) <http://fundaoinvestigation.com/demonstrative-animation/> (accessed 24/11/2016).

Shuttle D and Jefferies M (2016) Determining silt state from CPTu. *Geotechnical Research* **3(3)**: 90–118, <http://dx.doi.org/10.1680/jgere.16.00008>.

Smith IM, Griffiths DV and Margetts L (2013) *Programming the Finite Element Method*, 5th edn. Wiley, Chichester, UK.