Parallel processing of excavation in soils with randomly generated material properties

L. Margetts, I.M. Smith, L.M. Lever
University of Manchester, United Kingdom

D.V. Griffiths
Colorado School of Mines, USA

ABSTRACT: Today, affordable high performance clusters enable engineers to carry out large nonlinear 3D analyses very quickly using parallel processing. It has been estimated that by 2018, a 1 Petaflop cluster (with 100,000 cores) will cost around $150,000; a price affordable by a reasonably sized engineering firm or University department. With these significant improvements in technology, it is becoming increasingly cost-effective to incorporate uncertainty into analyses, for example by undertaking Monte Carlo simulations with randomly generated soil properties. As each realisation is independent, it can be executed at the same time. For large models that require move memory than is available on a single core, each realisation can be solved by subdividing the problem over multiple cores. The authors will present results for an excavation problem using this two-level parallelisation strategy. The parallel software used, ParaFEM, has been recently updated to interface with a number of external tools including the RFEM library, the visualisation tool ParaView and the graph partitioner METIS. For a single realisation, ParaFEM can make good use of 32,000 cores and solve problems with 1 billion degrees of freedom.

1 INTRODUCTION

1.1 *Overview*

Over the past few years, the authors have been working towards a generalised method of parallel finite element analysis based on an element-by-element solution strategy using iterative solvers. Further details are given in the 5th edition of “Programming the Finite Element Method (Smith et al. 2014).

The resulting software has been released as the freely downloadable open source “ParaFEM” project (http://parafem.org.uk). The software has benefited research in a broad range of disciplines including geomechanics (Smith and Margetts 2003), palaeontology (Falkingham et al. 2009, 2011), nuclear materials (Evans et al. 2013, Evans 2013) and geodynamos (Chan et al. 2006, 2007).

This previous work has focused on "fixed mesh" solutions. In some problems, geometry changes radically as the solution progresses. In geotechnical engineering, the construction process of excavation is a typical example.

Change in geometry occurs when material is removed from the ground in either open excavations or in enclosed tunnels. In many real engineering situations, the natural geometry of geological units necessitates three-dimensional analysis, leading to a finite element representation with many millions of degrees of freedom. Algorithms for non-linear elasto-plasticity are needed to adequately capture the physical response of the soil each time material is removed. Large problem size and complex material behaviour lead to heavy computational demands, both in terms of storage (memory) and solution time. For spatially random soil, analyses may have to be repeated many times to develop statistically meaningful results. These difficulties can be overcome by parallel computation. Once an analysis is complete, interpretation of the results presents a further challenge, benefiting from high performance visualisation techniques. In the context of solving and interpreting large three-dimensional excavation problems, this paper describes the authors' work in parallel computation and virtual reality visualisation. The ultimate aim of the research is to enable the geotechnical engineer to perform “Virtual Excavation”.

1.2 *Example*

A rather geometrically simple geotechnical problem is presented, namely the excavation of a large hole in the ground. In finite element terms, the soil is represented by a cubic domain of 20-noded hexahedral bricks. The base of the domain is fixed and the sides are on rollers. In total the model has around 1.5 million equations and more than 100,000 elements.

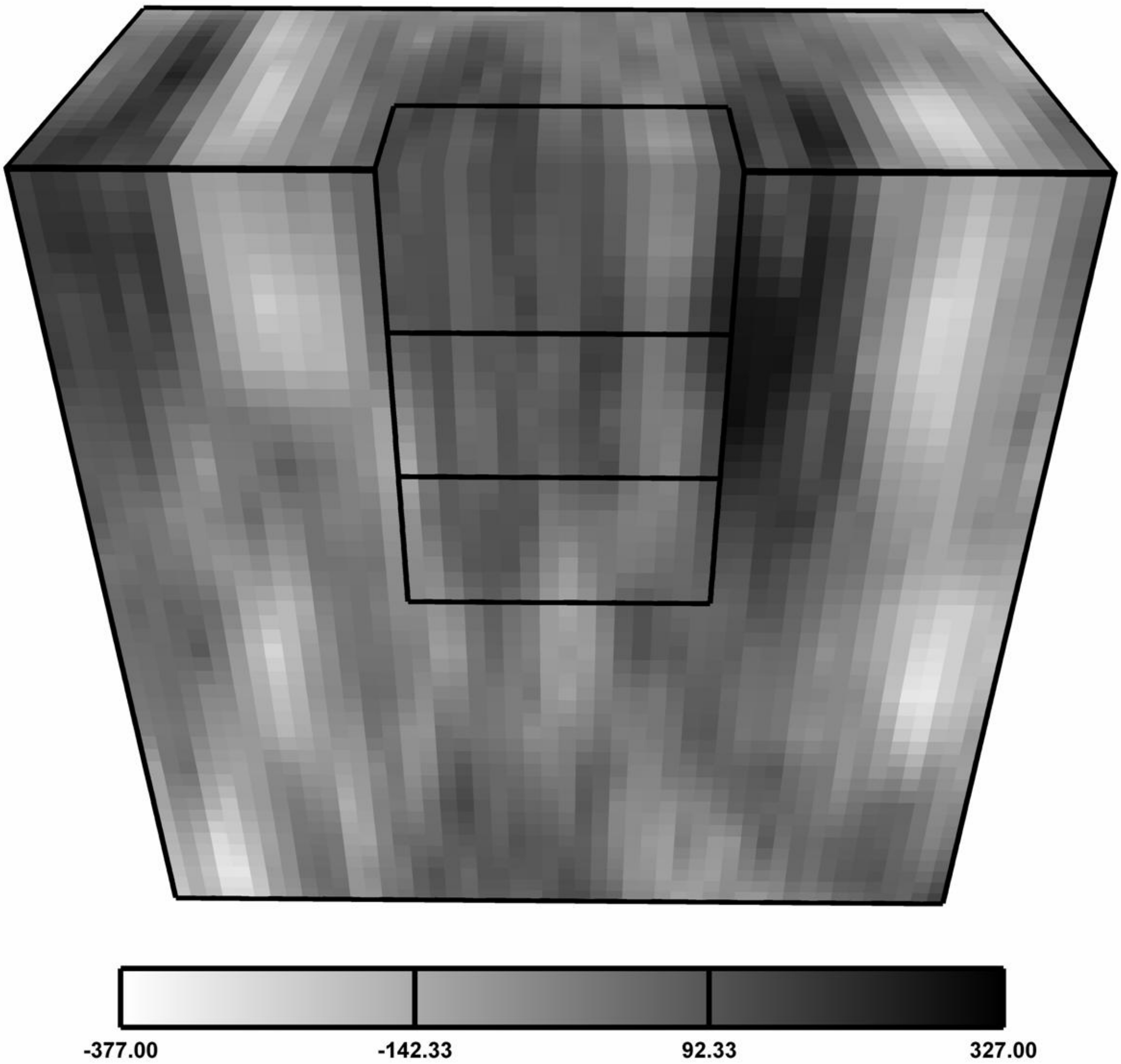

Figure 1. Excavation geometry shaded according to randomized material properties.

To make the problem more geologically realistic, the properties of the soil can be assigned statistically (Fenton and Griffiths, 2008). Figure 1 shows a typical "realisation" with the light to dark shading representing the variation over the soil property range (of stiffness and strength in this case). In this way, the natural variation of weaker and stronger areas can be captured. In Figure 1, dark and light shading represent strong and weak material respectively. The pixilation highlights the discretisation of the mesh used. Each finite element is assigned its own properties, represented visually in greyscale. In real situations, the use of stochastically generated soil properties implies that many "realisations" may be required for a single geotechnical design.

2 SOFTWARE

2.1 *Parallel Strategy*

The program used is an amalgamation of Programs 6.9 and 6.10 from Smith et al. 2014 parallelised according to the methods described in Chapter 12 of that reference. It is freely downloadable from the internet. It was written using Fortran2003 and is based on an element-by-element approach.

A preconditioned conjugate gradient (PCG) solver was used together with a diagonal preconditioner to solve the system of equations. Plasticity was dealt with using a consistent return algorithm. Parallelisation was achieved by simply inserting routines from ParaFEM, into what was essentially a serial code.

2.2 *Excavation Pseudo-code*

In the parallel program, each core in a multi-core processor executes instructions according to the pseudo-code below:

```
Initialise
READ input data on one core and distribute
DO for local elements
   Calculate starting stresses
END DO
DO Excavate Layer
```

```
    DO for local elements
       Calculate excavation loads
    END DO
    DO for local elements
       Calculate stiffness & preconditioner
    END DO
    DO Apply Load Increment
       DO Apply Plasticity Increment
          Solve using Element-by-Element PCG
          DO for local elements
             Check yield surface & Update gauss point stress
             Compute bodyloads vector
          END DO
       END DO
    END DO
    WRITE collect results and write on one core
END DO
```

The random fields are generated externally to this program using the driver program RFEMFIELD (freely available with ParaFEM) that calls subroutines distributed with the Fenton and Griffiths (2008) text book. The fields generated are for cubic domains and the RFEMFIELD software uses Boolean operations and interpolation to map the fields onto irregular geometries if required.

The random field generation software could be integrated into this program as a further outer loop. However, in the current implementation, we have distinct programs that can be joined together in a workflow. This could be considered inefficient computationally as it uses files to exchange data between programs. However, it has benefits for those wishing to understand or modify source code. Each program is short and self contained. Furthermore, this strategy makes it easier to distribute the computation on Cloud Computing platforms such as Window Azure.

2.3 *Performance*

As the problem is non-linear, requiring the execution of many load increments, most of the computation time is spent in the PCG solver. This is executed in parallel and each processing core works on its allocated set of elements. Table 1 presents some performance figures for a single load step using the PCG solver for different sizes of problem. The largest problem shown in the table has more than one billion degrees of freedom. As indicated in the table, this corresponds to a 400x400x400 cube of 20-node bricks. In 2D, 400x400 is considered a small problem. In 3D, this resolution is beyond the capability of commercial software packages.

The system used is HECToR, a Cray XE6 which is provided by the UK National HPC Service. The system comprises a total of 704 compute blades. Each blade contains four compute nodes giving a total of 2816 compute nodes, each with two 16-core AMD Opteron 2.3GHz Interlagos processors. This amounts to a total of 90,112 cores. Each 16-core socket is coupled with a Cray Gemini routing and communications chip. Each 16-core processor shares 16GB of memory, giving a system total of around 90 TB. The theoretical peak performance of the system is over 800 Teraflops. For some problems, ParaFEM has been shown to make good use of 32,000 cores.

Table 1. Performance statistics

Mesh	Equations	Cores	Time(secs)
40x40x40	777,520	8	96
		16	48
		32	25
		64	14
		128	8
100x100x100	12,059,800	16	486
		32	265
		64	140
		128	83
400x400x400	1,023,368,720	1024	2721
		2048	1213
		4096	662

The success of iterative methods depends on the number of iterations for convergence as a proportion of problem size (Smith and Wang, 1998). This is illustrated in Table 2 for a single load increment. For some types of problem, running on different numbers of cores may lead to different iteration counts. This is because of the effect of roundoff, particularly when values are summed across cores. The iterative solvers used compensate for this effect and always give the same engineering answer to the required tolerance (Smith and Margetts, 2006).

Table 2. Iterations to convergence vs problem size

Problem Size	Iterations To convergence	Iters/size
12,000	156	1.30E-2
98,000	297	3.03E-3
777,000	568	7.31E-4
1,514,000	704	4.65E-4
2,613,000	838	3.21E-4
6,812,000	1049	1.54E-4
12,059,000	1297	1.07E-4
768,959,200	3963	5.15E-6
1,023,368,720	4152	4.06E-6

2.4 *Cloud Computing*

Here, the Monte-carlo stochastic finite element analyses proceed by running essentially the same problem many times with different initial randomly generated material properties. Computer scientists refer to Monte-carlo simulations as “embarrassingly parallel” as each analysis is completely independent of the others. It each individual analysis fits within the available memory, the most efficient strategy is to run many single core jobs.

A number of companies, such as Amazon, Google and Microsoft own vast server farms which have a capacity designed to cope with surges in demand, for online shopping, search and software development respectively. They hire out their excess capacity through Cloud Computing services. If an engineer wished to run 1,000 realizations of a 3D excavation problem, they could purchase 1,000 “instances” of a virtual machine, each with 8 cores say, for a very short period of time. This is particularly attractive for small engineering firms or independent consultants as they do not need to invest in their own infrastructure. Typical costs are shown for Microsoft Azure in Table 3.

Table 3. Typical costs: Microsoft Azure

Size	Cores	Cost (month)	Cost (hour)
Extra small	Shared	$9.36	$0.013
Small	1	$57.60	$0.08
Medium	2	$115.20	$0.16
Large	4	$230.40	$0.32
Extra large	8	$460.80	$0.64

It should be noted that Cloud Computing services typically provide “virtualized” hardware. Virtualization incurs a performance penalty and the author’s tests show that individual jobs can take ten times longer than on the “native” hardware (Christias 2013). Table 4 compares the time for a single load step of one realization using a single core on Microsoft Azure and a single core on a Cray XE6.

Table 4. Microsoft Azure vs Cray XE6

Number of Elements	Azure (s)	Cray XE6 (s)
82,944	65	6
196,608	160	18
384,000	374	38
663,552	722	78
1,572,864	8,906	798
5,308,416	11,985	1,558

3 RESULTS

3.1 *Visualization*

It may well be that a principal impediment to the use of three-dimensional computations in geomechanics is not so much concerned with doing the calculations, but rather with the difficulty of visualizing the results. For this reason, we have added support for the open source visualization tool ParaView to the 5th edition of the textbook.

ParaView is a GUI based tool which runs on both Windows and Linux platforms. It has parallel processing capability and can be used in client-server mode. For very large models, rendering can be performed on a remote cluster with the display exported to a local desktop system. ParaView also supports Python scripting which can help automate results processing, particularly important when executing 100s of realizations in a stochastic analysis. Furthermore, support is provided for easy output into immersive Virtual Reality facilities.

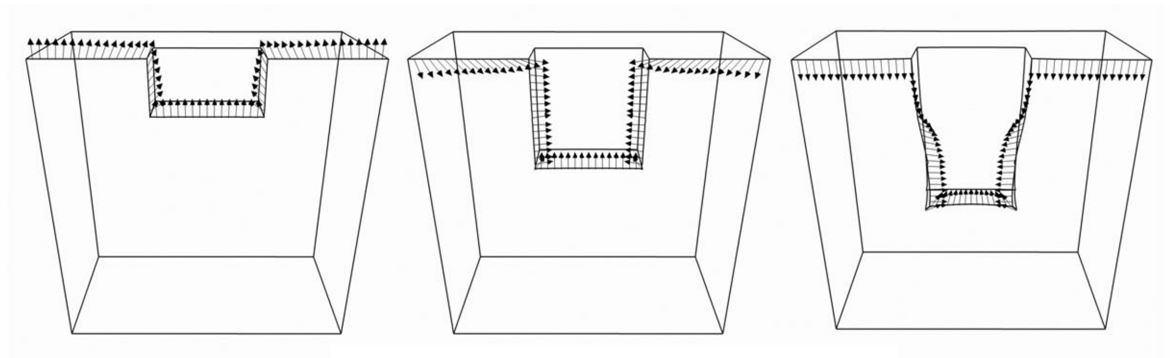

Figure 2. Excavation sequence.

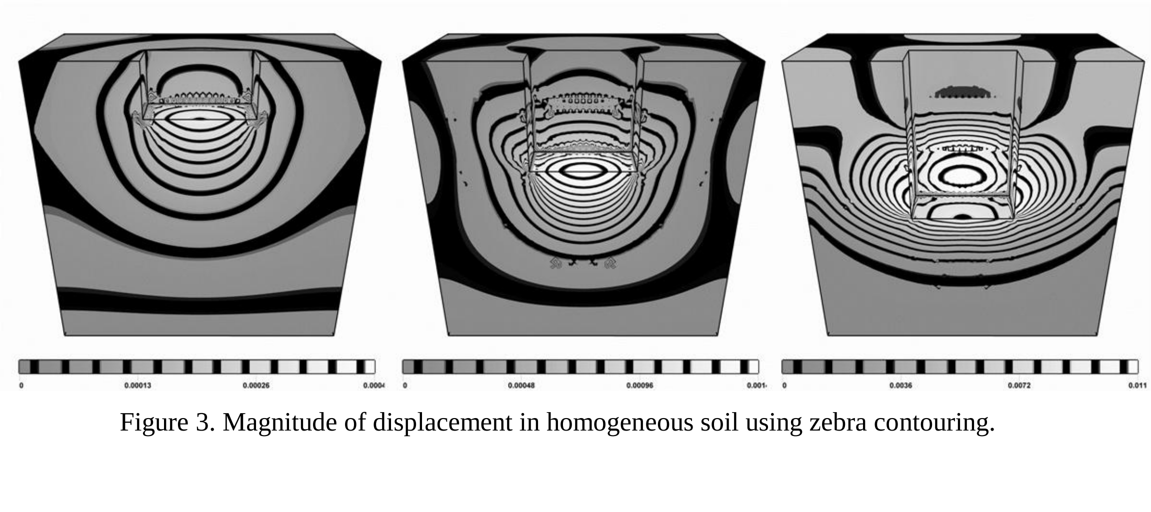

Figure 3. Magnitude of displacement in homogeneous soil using zebra contouring.

3.2 *Deterministic Analysis*

The response of the homogeneous ground to the removal of the soil is illustrated in Figures 2 and 3. The images show a cross-section through the domain as the interest lies in the deformation of the excavation walls. The magnitude of displacement varies greatly from one step to the next, therefore the images are not plotted to the same scale.

In Figure 2, a wireframe model is used to show the excavation sequence. Arrows are used to indicate the direction of the ground movements. As the excavation is deepened, the initial uplift at the surface and heave at the base of the excavation is followed by downward flow and eventual collapse.

Figure 3 shows the magnitude of the displacements. The colour map used consists of a number of evenly spaced black stripes, which appear as isocontours on the volume. In the gaps between the black stripes there is a continuous graduation of colour which indicates the magnitude of the isocontours. A black stripe on the volume which has a very light surround is data with a high magnitude of displacement while a black stripe on the volume with a dark surround is data with a low magnitude of displacement.

Why use black stripes? Removal of the stripes would leave a grey-scale shading that is typical of contemporary FEA visualisation. The stripes serve to highlight the variation in displacement that the eye cannot pick up with the greyscale shading alone.

3.3 *Randomised Soil*

Figures 4 and 5 show a typical realization of an excavation through randomised soil. The style of the visualizations is not the same as Figures 2 and 3. This is intentional as the authors wish to highlight some visualisation techniques.

The grey shading in Figure 4 does not represent any geomechanical property. A virtual light illuminates the scene, causing reflected highlights (light grey) and casting shadows (dark grey). The reason lighting is used is to pick out regions of deformation. Rippling occurs at the excavation boundaries and the excavation walls bulge non-symmetrically as failure is approached. Although these features are difficult to represent on paper, the visualization package can display stereoscopic images on a desktop PC (equipped with reasonable graphics card and cheap stereoscopic glasses), a 3D television or in a more expensive virtual reality centre.

The difference between deformation in a homogeneous and randomised soil is quite straightforward. In the homogeneous case, deformation is symmetric, whereas in the more natural randomised soil, deformation is influenced by the strength distribution. In this case, the soil is relatively weak on the left hand side of the excavation and relatively strong on the right hand side.

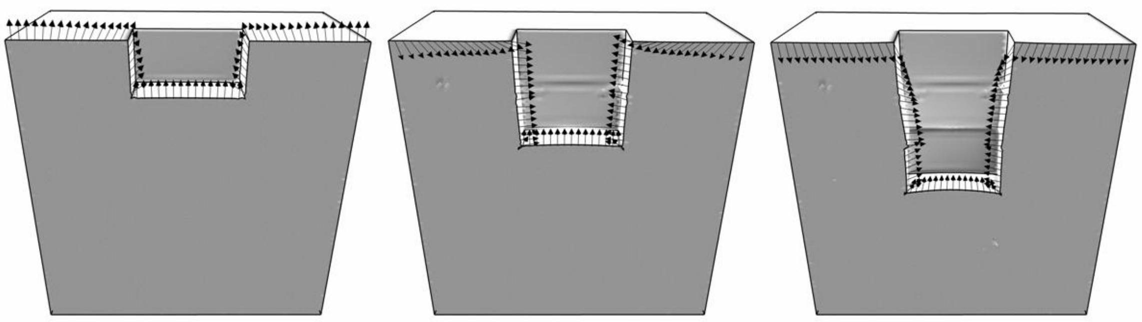

Figure 4. Virtual illumination of deformation in randomized soil.

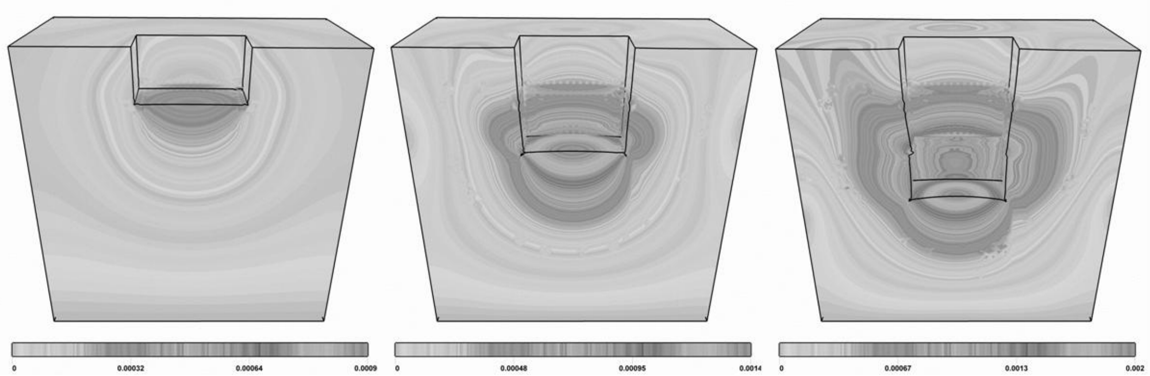

Figure 5. Magnitude of displacement in randomised soil using a randomised colour map.

Figure 5 (compare with Figure 3) highlights the influence of the non-homogeneity on the displacement profile. A randomised, alternating light and dark, colour map is used to represent the magnitude of displacement. This colour map gives added clarity compared with traditional grey-scale shading.

4 CONCLUSIONS

3D geotechnical excavation has been modelled using freely downloadable software and visualized using an open source program. In the example excavation only 1.5 million equations have been used but the authors have shown the software is capable of solving systems with a billion or more equations. The software can run on supercomputers with up to 32,000 cores.

Virtualized hardware provided by companies offering Cloud Computing services brings stochastic finite element modeling within the reach of most engineering firms, at a reasonable cost.

Current research in supercomputing is aimed at building an Exascale computer, one that is capable of performing 10^{18} operations per second, by the year 2020. It is thought that programs running on such systems will need to cope with many millions of threads of execution. In that case, we anticipate that the program used in this paper will require a new parallelization strategy, perhaps using thousands of GPUs or many-core processors such as Intel's recently launched Xeon Phi.

Programming is becoming harder and the number of engineers with programming skills fewer. Therefore, progress in geomechanics would benefit from community-based software development through the open source movement.

REFERENCES

Chan K.H., Ligang L. and Xinhao L. 2006 Modelling the core convection using finite element and finite difference met Phys Earth Planet Interior 157(1/2), 124-138

Chan K.H., Zhang K., Li L. and Liao X 2007. A new generation of convection-driven spherical dynamos using EBE finite element methods. Phys Earth Planet Interior 163(1-4), 251-265

Christias D. 2012. vProduct: Virtual prototyping on-demand using cloud computing technology. MSc Thesis. University of Manchester.

Evans Ll. M., Margetts L., Dudarev S., Young P. and Mummery P.M. 2013 Parallel processing for time dependent heat flow problems. Proceedings NAFEMS World Congress, NAFEMS, Austria

Evans Ll. M. 2013. Thermal finite element analysis of ceramic/metal joining for fusion using X-ray tomography data. PhD Thesis. University of Manchester.

Falkingham P.L., Bates K.T., Margetts L. and Manning P.L. 2011 The "Goldilocks" effect: Preservational bias in vertebrate track assemblages. Royal Society Interface 8(61), 1142-1154

Falkingham P.L. Margetts L., Smith I.M. and Manning P.L. 2009. A reinterpretation of palmate and semi-palmate (webbed) fossil tracks: Insights from finite element modeling. Palaeogeography, Palaeoclimatology and Palaeoecology 271(1/2) 69-76

Fenton G.A. and Griffiths D.V. 2008. Risk Assessment in Geotechnical Engineering. John Wiley and Sons, Chichester.

Smith I.M., Griffiths D.V. and Margetts L. 2014. Programming the Finite Element Method. 5th Edition. Wiley, London.

Smith I.M. and Margetts L. 2003. Portable parallel processing for nonlinear problems. Proceedings COMPLAS 2003, CIMNE, Barcelona.

Smith I.M. and Margetts L. 2006. The convergence variability of parallel iterative solvers. Engineering Computationas 23(2), 154-165.

Smith I.M. and Wang A. 1998. Analysis of piled rafts. International Journal for Numerical and Analytical Methods in Geomechanics 22(10), 777-790.

ACKNOWLEDGEMENTS

The authors would like to acknowledge access to the UK National High Performance Computing service, HECToR.