DEVELOPMENT OF EFFICIENT FINITE ELEMENT SOFTWARE

D. V. Griffiths
d.v.griffiths@mines.edu
Professor, Geomechanics Research Center, Colorado School of Mines, Golden, CO 80401, USA
Jinsong Huang
jhuang@mines.edu
State Key Laboratory of Water Resources and Hydropower Engineering Science
Wuhan University, Wuhan, Hubei 430072, P. R. China

Abstract. The focus of this paper lies in the efficiency of finite element codes. Three different types of analysis are presented, 1) Elasto-plastic analysis, 2) Biot analysis of poro-elastic materials and 3) Integration of stiffness matrices for triangular elements. In each case, two or more methods are implemented to achieve the same results, but using quite different algorithms. The results from cases 1) and 2) emphasize the difficulties in making firm conclusion about which algorithm is "best". Efficiency comparisons appear problem dependent, with some algorithms being the most efficient in some cases but not others. Case 3) consistently indicates the improved efficiency of analytically derived software, while recognizing that this often represents a relatively small part of the overall CPU time expended in analyses with large FE meshes.

Keywords: Finite element method, Tangent stiffness, Constant stiffness, Direct and iterative solvers, analytical derivations, stiffness matrix,

1. INTRODUCTION

The paper presents results from three different studies relating to the efficiency of finite element codes. Firstly in the context of elasto-plastic analysis of solids, algorithms that use "constant stiffness" are compared with "tangent stiffness". The key issue here is whether the constant stiffness approach takes more time to run than the the tangent stiffness approach, which uses fewer iterations, but carries the overhead of requiring regular assembly and factorization of the global stiffness matrix. Secondly in the context of coupled Biot analyses of poro-elastic materials, several different solution algorithms are contrasted for their efficiency in solving the symmetric but non-positive definite systems of equations that occur at each time step of the process. Both direct and iterative solvers are considered. Finally, results are presented which compare the time taken to develop the stiffness matrix of triangular elements using conventional numerical integration with some novel analytical approaches using software developed with the help of computer algebra systems (CAS).

2. TANGENT STIFFNESS METHOD AND CONSTANT STIFFNESS METHOD

Nonlinear finite element analysis involves solving the following nonlinear equation,

$$[K(u)]\{u\}=\{F\} \quad \text{or} \quad \{R(u)\}=[K(u)][u]-\{F\}=0 \tag{1}$$

where u is the solution to be determined, $[K(u)]$ is a known function of $\{u\}$, $\{F\}$ is the known "force", and $\{R(u)\}$ is the "residual".

This kind of nonlinear equation needs to be solved by iterative methods. The most popular method is the Newton-Raphson Method. Suppose that we know the solution to equation (1) at the $r-1^{st}$ iteration and are interested in seeking solution at the r^{th} iteration. We expand $\{R(u)\}$ about the known solution $\{u\}_{r-1}$ by Taylor's series and assuming that the second and higher order terms are negligible. We can write equation (1) as,

$$\{\delta u\}_r=-[K(u)_{r-1}]^{-1}\{R(u)_{r-1}\} \tag{2}$$

where $\{\delta u\}$ is the displacement increment, $[K]_T$ the tangent stiffness matrix of the curve $\{R(u)\}$ at $\{u\}_{r-1}$ (e.g. Reddy [1]) . The residual or out-of-balance force $\{R(u)\}$ is gradually reduced to zero if the procedure converges.

The full Newton-Raphson method requires that the tangent matrix $[K]_T$ be assembled and factorized at each iteration. This can be very expensive when many degrees of freedom are involved. A modified Newton-Raphson technique involves, for a fixed load step, either keeping $[K]_T$ fixed (called the constant stiffness method) while updating the imbalance force at each iteration or updating $[K]_T$ periodically after a pre-selected number of iterations while updating the imbalance force at each iteration.

In material nonlinear such as plasticity analysis, the tangent $[K]_T$ takes the form

$$[K]_T=\sum_e\int_{\Omega^e}[B]^T\left[\hat{D}\right][B]\,d\Omega^e \tag{3}$$

where $\left[\hat{D}\right]$ is the stress-strain matrix. Note that the stress-strain matrix $\left[\hat{D}\right]$ is dependent on the choice of stress update procedure. In this paper, the radial return algorithm, which is a special case of "closest point projection method" of Simo and Taylor[2] for a Von-Mises material is used. A consistent elasto-plastic modulus is also employed to achieve a quadratic convergence rate as $\{R(u)\}\to 0$.

As mentioned before, the full Newton Raphson tangent stiffness method has a quadratic convergence rate but needs to form the global tangent stiffness at every iteration The constant stiffness method forms the global stiffness matrix once only, but only has a linear convergence rate.

In order to investigate the efficiency of these two methods. A footing problem with different meshes has been analyzed. Figure 1 shows a mesh involving 8×4 elements with a flexible strip footing at the surface of a layer of uniform "undrained clay". In order to investigate the efficiency of constant stiffness method and tangent stiffness methods, another two bigger mesh for the same problem are also analyzed. They are 20×10 elements and 200×10 elements. The footing supports a uniform stress, q, which is increased incrementally to failure. The elasto-plastic soil is described by three parameters , namely the undrained "cohesion" c_u, followed by the elastic properties, E and v. Theoretically, bearing failure in this problem occurs when q reaches the "Prandtl" load given by

$$q_{ult} = (2+\pi)c_u \tag{4}$$

For 8×4 elements mesh, the form of a dimensionless bearing capacity factor q/c_u versus centerline displacement is plotted in Figure 2. It is seen that the displacements are increasing rapidly when the load reaches of 520, indicating that bearing failure is taking place at a value very close to the "Prandtl" load of 5.14.

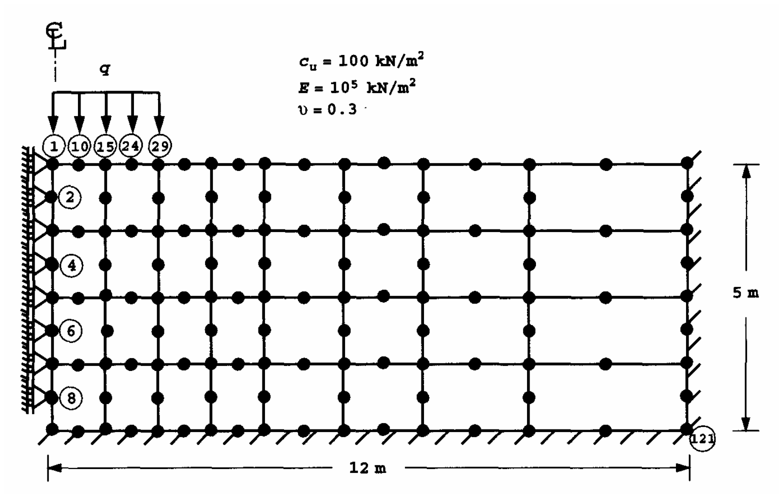


Figure 1- Mesh of a strip footing

The number of iterations to achieve convergence for each load increment is shown in Table 1. In the constant stiffness method, the iteration count is relatively low to start with but increases significantly as failure is approached. In the tangent stiffness method however, the algorithm converges within 10 iterations for all load increments. The CPU time taken by the different methods is plotted against the number of freedom in Figure 3. All analysis were run on a 3.0MHz PC with 1G ram memory. For a small mesh, the tangent stiffness approach ran faster, but it can be seen that with increasing mesh size, the tangent stiffness performed less well due to the time taken to assemble and factorize the global matrix at each iteration. For the 200×10 mesh the constant stiffness approach ran faster. A compromise approach might be to reform $[K]_T$

periodically, but in any case the relative efficiency of the algorithms is problem dependent .

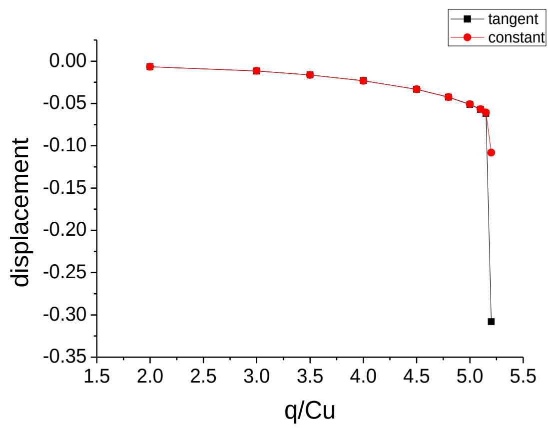


Figure 2- Pressure vs centerline displacement

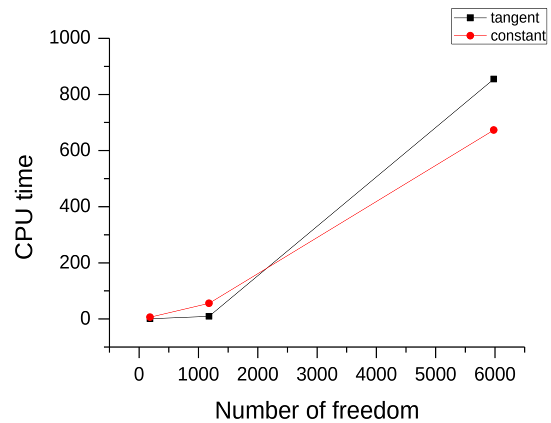


Figure 3- CPU time vs. degrees of freedom

Table 1 Number of iterations and CPU time

		8×4		20×10		200×10	
Step	Loads	Constant	Tangent	Constant	Tangent	Constant	Tangent
1	0.2000E+03	2	2	2	2	2	2
2	0.3000E+03	22	5	21	5	21	5
3	0.3500E+03	38	5	48	7	50	7
4	0.4000E+03	66	5	97	8	103	8
5	0.4500E+03	94	8	143	10	154	10
6	0.4800E+03	137	8	204	8	223	10
7	0.5000E+03	165	7	259	10	286	10
8	0.5100E+03	178	8	285	8	315	9
9	0.5150E+03	213	6	409	7	436	6
10	0.5200E+03	3000	8	3000	6	3000	6

3. BIOT ANALYSIS OF 2-D PORO-ELASTIC MATERIALS

Another important factor that influences the efficiency of a FEM codes lies in the strategy used to solve large systems of linear equations such as equation (2). The global matrices in discretization methods such as FE are banded. When using direct solvers such as Gaussian elimination only elements inside the band need to be stored. Zero elements inside the band must be stored because during the factorization phase they subsequently become non-zero due to “fill in”. Even with efficient storage strategies such as “skyline”, as the problem size grows the storage requirement can becomes a burden. For this reason, alternative solution strategies to Gaussian elimination have been sought, and there has been a resurgence of interest in iterative techniques for the solution of large systems. Griffiths and Smith [3] describe a number of algorithms of this type, the most popular for symmetric positive definite systems being based on the method of “conjugate gradients” (Jennings and McKeown, [4]). The method can also work for non-positive definite systems with improved performance often achieved via a “preconditioner”, hence the term “preconditioned conjugate gradient” (PCG) methods. These methods consist of simple vector operations which can be carried out "element-by-element" (EBE) without ever assembling global matrices at all. This idea proves to be very attractive when solving large problems, because the matrix-vector multiplication can be done in parallel on a series of processors. A possible disadvantage of the element-by-element data structure is the limited number of preconditioners that can be formulated from the unassembled matrices. Chen *et al.* [5] have proposed a modified Symmetric Successive Overrelaxation (SSOR) preconditioned symmetric Quasi-Minimal Residual (QMR) solver. This method requires global matrix assembly, but the authors report good convergence properties.

In this part of the paper we have compared the efficiency of several solvers in a 2-d FE analysis of poro-elasticity using Biot's theory. This type of analysis leads to systems of equations which are symmetric but non-positive definite. The method compared are Gaussian elimination, PCG and QMR. In the case of the PCG, both assembly and EBE approaches were attempted, and in each of those cases two different initial guesses were investigated, a "zero" initial guess in which each time step was treated as an entirely new problem, and a "previous" initial guess in which the converged solution from the previous time step was used as the initial guess to the next.The Gaussian elimination and assembled PCG both used a skyline storage strategy, while the QMR assembly used what the authors referred to as a Compressed Sparse Column(CSC) method.

The same meshes as in Section 2 (e.g. Figure 1) were used for Biot consolidation analyses of a flexible strip footing resting on a homogeneous soil subjected to the ramped vertical pressure shown in Figure 4.

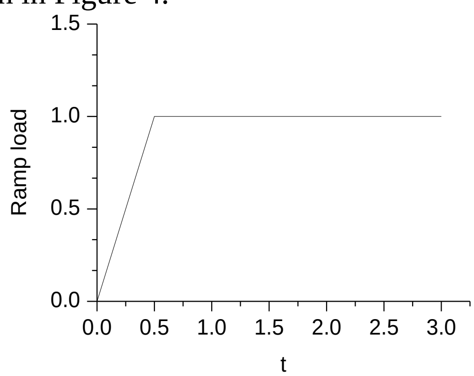


Figure 4- Ramp Loading

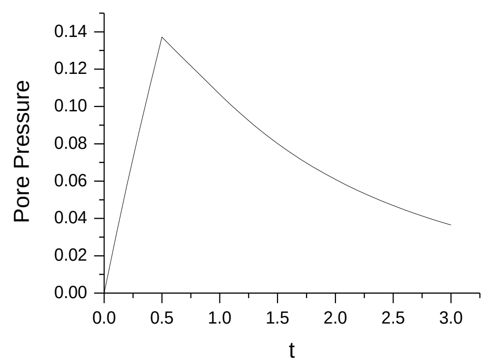


Figure 5- Pore pressure at a typical node

The example and properties are similar to that describe in the text by Smith and Griffiths [3] (pp.416-434). The analyses use an 8-node quadrilaterals solid element coupled with a four-node quadrilateral fluid elements were used. The ground water table is assumed to be at the ground surface and is in hydrostatic condition at the initial stage. The ground surface is free-draining with all other sides undrained. The soil material is assumed to be isotropic and linear elastic with $\upsilon' = 0$, $E' = 1\text{kN/m}^2$ and an isotropic permeability $k_x = k_y = 9.81\,\text{kN/m}^3$. Subsequent dissipation of the pore water pressure and settlement beneath the footing are studied by using a Crank-Nicolson time discretization scheme. A typical result is shown in Figure 5. The analysis is taken to a time of $t = 3$ secs by a) 30 steps with $\Delta t = 0.1$ secs, b) 300 steps with $\Delta t = 0.01$ secs, c) 3000 steps with $\Delta t = 0.001$ secs and d) 30000 steps with $\Delta t = 0.0001$ secs. The CPU time for different methods are list in Table 2 and the amount of memory required by skyline vs. CSC in Table 3.

Table 2 CPU time by different methods

Mesh	Time Steps	QMR	Gauss	PCG Previous	PCG Zero	PCG_EBE Zero	PCG_EBE Previous
8×4	300	3.22	0.64	16.33	29.03	24.19	13.73
	3000	32.17	5.80	124.50	392.95	330.14	117.08
	30000	323.63	60.86	973.78	5010.30	4193.14	1538.69
20×10	300	51.09	20.52	429.69	744.38	215.45	147.81
	3000	505.78	180.44	3797.14	11285.53	3606.27	1458.98
	30000	4936.52	1737.67	28350.70	171099.69	57937.13	17787.55
200×10	30	61.84	301.31	1846.61	2800.75	204.48	145.97
	300	691.06	668.44	18471.00	42928.83	2939.20	1245.92
	3000	7401.50	4338.22	184750.01	657996.77	41067.16	10733.97

Conclusions relating to efficiency are different depending on which mesh is being considered and how many time steps are computed. Assuming that efficiency gains for large meshes are the priority, consider the results corresponding to the 200×10 mesh. The most efficient of all the solution methods was the conventional Gaussian elimination based on assembled global equations. For linear problems the method is able to benefit from a single factorization followed by repetitive forward and back-substitutions at each time step. The more time steps taken, the better the Gaussian elimination efficiency becomes compared with the other methods. Regarding PCG, the most efficient was the EBE approach with the previous solution as a first guess. The least efficient was the assembled version with a zero initial guess. The QMR approach fell between Gaussian elimination and PCG but was clearly the most efficient of the iterative methods considered. The QMR algorithm looks promising, but needs an EBE version to reach its full potential for very large problems that cannot fit in the core of the computer. This is particularly true since it can be noted from Table 3 that CSC required significantly more memory than skyline.

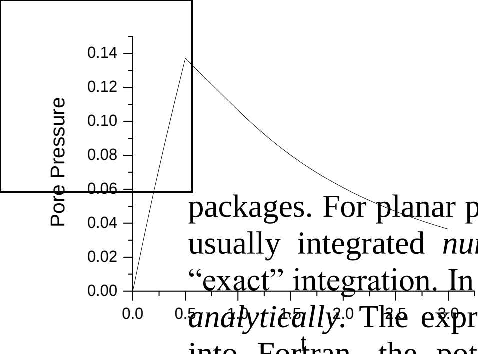
Table 3 Memory requirements for assembly methods

Mesh	Freedoms	Memory required	
		CSC	Skyline
8×4	228	10170	25317
20×10	1413	397758	165701
200×10	7010	9431907	3919382

4. ANALYTICALLY DERIVED STIFFNESS MATRICES

The last topic described in this paper involves the use of Computer Algebra Software to generate finite element software that runs faster than traditional methods. This is a project that involves collaboration between the Colorado School of Mines and the Universidad Central de Venezuela(e.g. Videla et al. [6]) with the objective of developing a library of Fortran 95 routines freely available on-line (see www.mines.edu/fs_home/vgriffit/analytical).

One of the most recent additions to the library is a routine for computing the elastic stiffness matrix of a 15-noded triangle, which is an element used quite widely in some proprietary FE



packages. For planar problems, the element stiffness matrix such as that shown in equation (3) is usually integrated *numerically*. The 15-noded triangle requires 12 integrating points for an "exact" integration. In the current approach we have used a computer algebra system to integrate *analytically*. The expressions are messy, however with the ability to output the results directly into Fortran, the potential for typographical errors is minimized. Studies by Brayton [7] comparing the efficiency of the analytical approach with the conventional numerical approach indicated a speed-up factor of greater than 20 for this element.

6. CONCLUDING REMARKS

The paper has presented three example of finite element computations in which efficiency gains can be made by a careful choice of algorithm. The first two examples considered, namely Newton-Raphson vs. modified Newton Raphson in plasticity computation and iterative vs. direct solvers for Biot analysis indicated a problem dependence. A new iterative QMR solver showed some promise although it was slower than an ordinary direct solution method. Further developments are needed to implemented QMR in the context of EBE, in which case it might be a good alternative to PCG. Finally, the benefits of analytically derived element stiffness matrix methods were presented and a web site holding freely available routines of this type provided. The example included in this paper showed that for a 15-node planar element, a routine derived analytically ran over 20 times faster than the traditional numerically integrated version, while giving exactly the same results.

7. REFERENCES

[1]. Reddy, J. N., An Introduction to Nonlinear Finite Element Analysis, Oxford University Press, 2004

[2]. Simo, J. C., Taylor, R. L., Consistent tangent operators for rate-independent elastoplasticity. Comp. Methods Appl. Mech. Eng., Vol. 48, n. 3, pp 101-118, 1985

[3]. Smith, I. M., Griffiths, D. V., Programming the Finite Element Method, John Wiley & Sons, 4^{th} ed. , 2004

[4]. Jennings A., McKeown JJ., Matrix Computation. John Wiley & Sons, Ltd, 1992

[5]. Chen X., Toh, K. C., Phoon, K. K., A modified SSOR preconditioner for sparse symmetric indefinite linear systems of equations, Int. J. Numer. Meth. Engng, Vol. 65, n. 6, pp785-807, 2006.

[6]. Videla, L., Baloa, T., Griffiths, D.V. and Cerrolaza, M., Exact integration of the stiffness matrix of an 8-node plane elastic finite element by symbolic computation, To appear in *Finite Elements in Analysis and Design*, 2006

[7]. Brayton, A., Finite Element Software Development Using Computer Algebra Systems (CAS), Independent Study Project, Civil Specialty, Colorado School of Mines, 2005