TETRAHEDRAL FINITE ELEMENTS AND GEOTHERMAL RELATED SUBSIDENCE

Justin Pogacnik[1], Peter Franz[1], and Lutfhie Azwar[1]

[1]Mercury NZ Ltd, 283 Vaughan Road, Rotorua, New Zealand 3010

justin.pogacnik@mercury.co.nz

Keywords: *Finite Element Method; Tetrahedral Elements; THM Modeling; Subsidence*

ABSTRACT

In recent years within the geothermal community, there has been an increase in interest to use numerical methods to represent rock mechanics phenomena such as production-related subsidence, thermal cracking, hydraulic fracturing/shearing, and stress analysis for induced seismicity. These interests require the coupling of the differential equations for heat and mass transport to the differential equation for linear momentum balance of the rock matrix. Heat and mass transport software packages are traditionally based on the finite volume method (FVM), while rock mechanics packages are traditionally based on the finite element method (FEM). Many volumes (cells) used in FVM techniques are inadmissible as finite elements as they do not satisfy the isoparametric element criterion required to guarantee solution convergence. However, any mesh structure can be broken down into a mesh of tetrahedral cells without increasing the number of nodes (vertices). The 4-noded tetrahedron is an admissible finite element type, although it is a notoriously poor performer in stress analyses. In this work, we investigate two techniques (nodal integration and addition of nodal rotation degrees of freedom) in order to improve the performance of standard linear tetrahedral elements.

We compare those results to the results of traditional tetrahedral and hexahedral element meshes for a variety of benchmark problems as well as a hypothetical geothermal production related subsidence problem. While tetrahedra are convenient from a mesh construction viewpoint, that convenience comes at the expense of computational time, effort, and accuracy. The finite element user should endeavor to use hexahedral elements when possible.

1. INTRODUCTION

Numerical modeling has played an important role in understanding the behavior of geothermal systems. There is a growing interest in the investigation of THM(C) (thermo-hydro-mechanical (chemical)) coupled simulations for physical processes such as chemical species transport, rock alteration (dissolution and precipitation), injectivity/productivity changes, thermal cracking, hydrofracking and/or hydroshearing, subsidence, etc. Few codes are equipped to handle all the coupled THM(C) processes simultaneously. Therefore, it is common to couple different codes (and thus, different numerical methods) to achieve this purpose. Most heat and mass (and chemical species) transport codes are based on the finite volume method (e.g., see Pruess, *et al* (1999)), while many solid mechanics codes are based on the finite element method (e.g., see ABAQUS (2002)). This work is focused on the mechanical aspect of THM(C) analyses and repercussions of element choice when using finite volume domains to populate a finite element method domain.

The next section will outline convergence in the finite element method and explain why we are interested in tetrahedral elements. It will then outline the element types that were tested in this work. The following section will then detail the numerical experiments performed on these element types and compare each element's performance. We finish with concluding remarks. The take-home message is that hexahedral elements should be the type of element that a numerical modeler endeavors to use. However, given grid constraints, tetrahedra may be necessary. In that case, care should be taken as standard tetrahedra may not perform satisfactorily.

2. FINITE ELEMENT METHOD

This section will outline some of the fundamentals for finite element convergence as well as the element types tested in this work. The performance of those elements in our test problems will be presented in the Numerical Experiments section.

2.1 Numerical Convergence

The entire basis of the finite element method is indeed the basis functions (or shape/interpolation functions) used within elements. Convergence means that an element will approach the exact solution of a given differential equation with mesh refinement (smaller elements). From Hughes (1999), the basic convergence requirements for solid mechanics are that shape functions be:

- Smooth (at least C^1) on element interiors
- Continuous across element boundaries
- Complete

C^m continuity means that derivatives of order *m* of the shape functions exist where *m* is the order of the highest derivative term of the primary variable (displacement) present in the differential equation. Linear functions satisfy the C^1 continuity requirement. Regarding the second point, if the functions are discontinuous at element boundaries, then delta functions arise in their derivatives that cannot be evaluated for stiffness integrands. This condition basically means that primary variables must be continuous along element boundaries. Elements that meet the first two requirements are typically called *compatible* or *conforming* elements. The third point, *completeness*, requires that the interpolation functions (shape functions) are capable of exactly representing an arbitrary linear polynomial based on the nodal values anywhere in the interior of the element. Basically, the sum of the shape functions must be 1 at any point in the domain. This means that any possible solution can be represented inside the element domain.

The isoparametric element concept, attributed to Irons (1966), results in the C^1 convergence criterion being generally achieved. It simply states that the same shape functions can represent both geometric interpolations and primary variable interpolations, i.e.,

$$x = \sum_{a=1}^{\mathrm{Nen}} N_a \, x_a \qquad (1)$$

and

$$u = \sum_{a=1}^{\mathrm{Nen}} N_a \, d_a \qquad (2)$$

where boldface is used to represent vectors, N_a are the finite element shape functions, *a* represent the element nodes, *x* are coordinate positions, and *d/u* are displacements.

2.2 Why Tetrahedral Finite Elements?

Triangular (2D) and tetrahedral (3D) elements are bound to be the first element that a reader of virtually any finite element text will encounter. They are relatively easy to understand and do not require a local coordinate system definition. When standard piecewise linear shape functions are used, the spatial derivatives of responses are constant within the element. This makes them easy to implement and very efficient in finite element programs. Moreover, many automatic meshing programs are based on triangular or tetrahedral element creation (Dohrmann, *et al* (2000)).

In our work, we are seeking to couple two different numerical methods for multiphysics thermo-hydro-mechanical (THM) analyses. The details of this coupling are presented in plenty of other literature (e.g., Pogacnik, *et al* (2015)), but the important point for this work is that Voronoi cells may be used in finite volume method simulations of heat and mass transport. Voronoi cells are not natively isoparametric finite elements and, thus, not guaranteed to converge. However, we would like to use the same numerical grid structure and not rely on interpolation of results between grids types to have a conforming finite element mesh. A possible solution to this dilemma is to tetrahedralize a given Voronoi grid. Simply put, any structure of 3D points can be tetrahedralized.

Finally, tetrahedral elements satisfy all of the convergence requirements outlined above. Therefore, as a mesh is refined, these elements are guaranteed to approach the exact solution of the differential equation. However, in some stress analyses (especially bending scenarios), these elements are known to be notoriously poor performers and require prohibitively small elements to converge to reasonable solutions. This shortcoming is due to the constant derivatives of shape functions (Filippa, 2017). However, all is not lost. The following sections will outline the element types tested in this work, including three methods of improving tetrahedral element performance.

2.3 Standard Hexahedron (Hex8)

The general workhorse element for solid mechanics simulations is the linear hexahedron with 8 nodes (often called Hex8 in literature) and trilinear interpolation functions. The shape functions are given by the relationship:

$$N_a = \frac{1}{8}(1 + \xi_a \xi)(1 + \eta_a \eta)(1 + \zeta_a \zeta) \qquad (3)$$

Where *ξ, η,* and *ζ* are local Cartesian coordinates for the element that range from [-1,1] in each direction. The shape functions and their derivatives are mapped from the local coordinate system to the global Cartesian system during element integration (see Hughes (1999)).

2.4 Standard Tetrahedron (Tet4)

The 4-noded tetrahedron (often called Tet4) is the case described earlier in this section. The shape functions can be constructed by collapsing nodes of the Hex8 element (see Hughes (1999)) or in “Tetrahedral Natural Coordinates” (Zienkiewicz, *et al* (2013)). In the natural coordinates representation, the shape functions vary linearly from [1,0] along a line from node *a* to the face opposite the node as seen in Figure 1. At the geometric centroid, the natural coordinates take the values of {¼, ¼, ¼, ¼}.

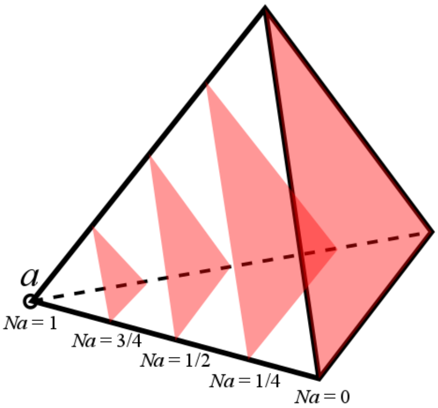

Figure 1: Tetrahedral natural coordinates for node *a*.

2.5 Quadratic Tetrahedron (Tet10)

The 10-noded quadratic tetrahedron (Tet10) is the first method of improving tetrahedron performance we tested. It is the second complete-polynomial member of the tetrahedron family and shape functions are quadratic functions of distance from node *a* (still varying from [1,0]). It has 6 additional nodes compared to the Tet4 case where each additional node is located at the midpoint of the edges between Tet4 nodes. Quadratic shape functions have the advantage over linear shape functions in that derivatives vary spatially resulting in the potential curvature of element faces/edges. So, bending solutions can be represented. The shape functions of Tet10 can be expressed in terms of the 4 linear shape functions of Tet4. For the corner nodes:

$$N_a = N_a^{(\mathrm{Tet4})}\left(2N_a^{(\mathrm{Tet4})} - 1\right) \qquad (4)$$

and for edge nodes:

$$N_a = 4N_i^{(\mathrm{Tet4})} N_j^{(\mathrm{Tet4})} \qquad (5)$$

where *i* and *j* represent the nodes of the line that the current midpoint node intersects.

2.6 Node-Based Uniform Strain Tetrahedron (UT4)

Dohrmann, *et al* (2000) developed a nodally integrated version of the Tet4 element to help alleviate the locking issues typically displayed by Tetrahedra. In that formulation, the code first loops over all the nodes in a mesh (instead of elements in the standard finite element method). The shape functions and their derivatives for each element that touches the current node are then computed in the same way as Tet4. Those results are volume weighted by the portion of each element associated with the current node (Voronoi volume) before being assembled in a typical finite element assembly fashion for the current node. This procedure results in a much denser stiffness matrix than traditional finite elements, however, no degrees of freedom are added to the system and the exact same grid can be used as in the Tet4 element case.

2.7 Rotational Degrees of Freedom (TETR)

Pawlak and Yunus (1991) developed a technique which adds rotational degrees of freedom to Tet4 to alleviate locking issues in tetrahedra. For conceptual purposes, we will present a 2D triangle equivalent formulation here. Consider the quadratic triangle element in Figure 6. The black nodes represent the nodes of the standard linear triangle and the red nodes represent the additional nodes of the quadratic triangle.

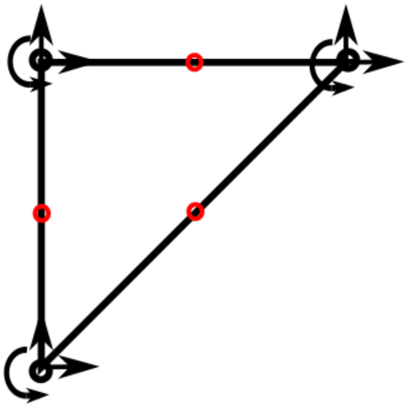

Figure 6: Rotational degree of freedom triangle with quadratic midpoint nodes to illustrate TETR.

The "x" displacement of the red mid-side nodes can be expressed in terms of displacements and rotations of the corner nodes:

$$u_m = \frac{1}{2}\left(u_i + u_j\right) + \frac{y_j - y_i}{8}\left(\omega_j - \omega_i\right) \qquad (6)$$

where ω are nodal rotations. Similar expressions can be written for y displacements and the extension to 3D is straightforward (see Yunus, et al (1989) for exact expressions). We can then use the shape functions for the quadratic triangle mid-side nodes to find those values at the current quadrature point in the interior of the element by equation (2), then transform those values via equation (6) to arrive at a matrix of shape functions that approximate shape functions for the corner node displacements and rotations based on the shape functions of Tet10 – in matrix form:

$$[N]_{\mathrm{TETR}} = [N]_{\mathrm{TET10}}[T]_{\mathrm{TETR}} \qquad (7)$$

where $[T]_{TETR}$ is a transformation matrix composed of identity tensors for corner node x/y/z displacement degrees of freedom and rotation/mid-side node displacement transformation matrices of equation (7) for the rotation degrees of freedom (see Yunus, *et al* (1989) and Pawlak and Yunus (1991)). In the TETR case, there are 4 corner nodes with 3 displacement degrees of freedom and 3 rotation degrees of freedom each for a total of 24 degrees of freedom for an element. Compare that to the Tet10 case that has 10 nodes with 3 displacement degrees of freedom for 30 total degrees of freedom. While a significant number of degrees of freedom are added to the system, most standard finite element procedures apply to TETR and the exact same grid as Tet4 can be used. Further, TETR is the standard 4 node Tetrahedral element programmed into typical finite element software packages such as ABAQUS (2002) and ANSYS (1994). The next section will summarize the performance of the element types presented in this section for a variety of test problems.

3. NUMERICAL EXPERIMENTS

3.1 Linear System Size/Density

As in most numerical methods, the finite element method requires one to solve a matrix system of equations (**Ax**=**B**), where **A** is the stiffness matrix, **x** is an array of unknowns (displacement) and **B** is the external force vector. Some of the most important factors regarding the size and speed in which the problem be solved are related to the length (or size) of **x** (or **A**) and the density of **A**. The matrix **A** is symmetric, sparse, and positive definite matrix in isotropic linear elastic finite elements. To test the matrix density and size we set up a cube problem of dimension 1m x 1m x 1m with two elements in each direction (27 nodes and 8 elements for the hexahedral case – linear tetrahedra have 6 elements per hexahedral elements, so 48 total elements and 27 nodes). We then computed the finite element stiffness matrix **A** for each of the methods outline above. Table 1 displays the results of this study.

Table 1: Unknowns/Matrix Density of Each Element Tested.

Ele Type	Elements	Nodes	Unknowns	size(A)	nonzeros	% density
Hex8	8	27	81	6561	3087	47.1
Tet4	48	27	81	6561	2007	30.5
Tet10	48	125	375	140625	21969	15.6
UT4	48	27	81	6561	4689	71.5
TETR	48	27	162	26244	8028	30.6

The **Unknowns** column gives the number of unknowns to be solved for each method and **size(A)** is simply **Unkowns**^{2}. The **nonzeros** columns gives the number of nonzero terms in the matrix and **% density** is **nonzeros/size(A)*100**. Tet10 requires the largest number of unknowns and has the most nonzero terms, but is the least dense of the matrices. UT4 was the highest density matrix as the connectivity of node interaction is higher under that formulation. Figures 7-9 display the matrix density plots to display Table 1's results graphically. The black areas represent zeros in the matrix and the white areas represent values. These results can give some indications of solve time, i.e., we would expect Tet4 to be the fastest, followed by Hex8, UT4, TETR, then Tet10 based on number of unknowns and density. However, construction of the larger density matrices does come at a cost as we will show in total computational time plots in the next section.

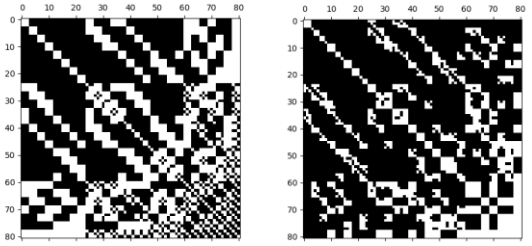

Figure 7: Matrix heat map for Hex8 (L) and Tet4 (R) element types.

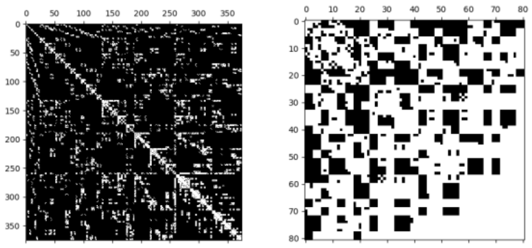

Figure 8: Matrix heat map for Tet10 (L) and UT4 (R) element types.

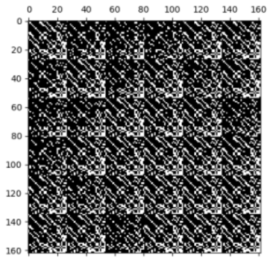

Figure 9: Matrix heat map for TETR element type.

3.2 Cantilever Beam under Gravity Load

Tetrahedral elements are known to perform very poorly in pure bending problems. To test bending deformation, we tested each element type in a cantilever bending scenario under gravity load only. A beam was constructed that was 1m x 1m x 50m. Based on small strain 1D beam theory, the maximum tip displacement of a cantilever beam under uniform body forcing is given by:

$$d_{\max} = \frac{\mathrm{wL}^4}{8\mathrm{EI}} = \frac{2500 \times 9.81 \times 1 \times 50^4}{8 \times 1e10 \times (112) \times 1 \times 1^3} = 22.99m$$

where $w = \rho \mathrm{gA}$ is the distributed load where A is the cross-sectional area of the beam, L is the length, E is Young's Modulus, and I=bh^3/12 is the bending moment of inertia of the beam. For completeness, Poisson's ratio was set to be 0 for these simulations to compare exactly to 1D analytic beam behavior. Figure 10 displays the results of the beam bending analysis for each element type with an element refinement level of one element through the thickness/height direction.

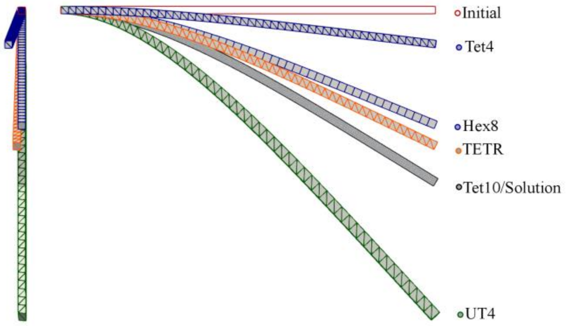

Figure 10 : Cantilever beam results for first refinement level.

The figure highlights the trouble that standard tetrahedra have in representing bending deformation. Comparing correction techniques, Tet10 performed the best, by converging to the analytical solution in the first level of refinement. The downside of Tet10 is the addition of the midside nodes requires a change in the grid structure. That is not an option in our conversion of Voronoi grids. UT4 successfully eradicated locking effects; however the response had nearly as much error as the Tet4 case only the response was softer instead of stiffer. TETR performed very well – in fact, even better than Hex8. Figure 11 shows the convergence behavior of each element type.

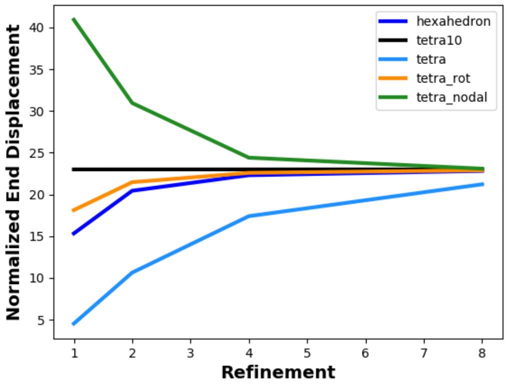

Figure 11: Convergence behavior of all element types.

Ideally, all the elements should converge to the exact solution as the mesh is refined and that is indeed the case. Tet4 experienced the slowest spatial convergence rate. UT4 converged faster than Tet4, even though the response was softer. Hex8 was the next slowest and the behavior of TETR was similar, though faster. Lastly, Tet10 was exactly equal to the solution for each level tested.

The above results indicate why Hex8 is considered such a robust workhorse type of element. It is fast, efficient, converges quickly, and easy to implement. Tet4 was the next fastest and only slower because more elements (and thus, more computations) are required to construct the stiffness matrix. TETR was significantly slower, but still comparable to Tet10 and requires less memory. This is perhaps why TETR is the standard implementation for tetrahedral elements in most finite element packages. UT4 was prohibitively slow in our implementation. The density of the matrix, does indeed slow down the simulation time. Also, there was a significant amount of computation involved in looping over nodes then the elements associated with each node. In our implementation, many elements were integrated multiple times. This could be improved at the expense of memory storage with upfront element integration. However, the best-case scenario would include a single loop over all elements, then a single loop over all nodes. This would still leave UT4 more expensive than the TETR and Tet10 cases.

3.3 Bar under Pressure or Temperature

For a second test problem, we constructed a 1m x 1m x 5m bar and subjected it to pressurization or temperature to test our implementation of the pore pressure or thermal strain effects. The 1D axial equation for change in length (dl) due to pore pressure or thermal strain effects in an elastic solid can be expressed as:

$$dl = l\frac{\beta}{3}dT + l\frac{1}{E}dP \qquad (9)$$

where β is the volumetric coefficient of thermal expansion (2.0e-5/ºC) and l is the initial length of the bar. We chose a uniform dT=100ºC and/or dP=(2/3)e7 Pa. These values were chosen so that the thermal and pressure expansion strains/displacements would be the same for either effect. Indeed, the responses were exactly the same whether pore pressure or temperature effects were considered, so we will only show the pressure scenario in this section. For the values reported here, the analytic value of $dl=0.00333$ m. Table 2 shows the axial displacements for each scenario, where we have tested an additional UT4 mesh.

Table 2: Change in length due to pressure/temperature change for each element type.

Ele Type	*dl* (m)
Hex8	0.00333
Tet10	0.00333
uTet4	0.00336
rTet4	0.00334
uUT4	0.00456
rUT4	0.00300
uTETR	0.00334
rTETR	0.00334

In the literature, there are a series of publications working to improve the spatial stability of UT4 (e.g., Puso and Solberg (2006), Puso, *et al* (2008), and Gee, *et al* (2009)). The uniform pressure/temperature cases highlight the need for additional numerical stability in that scenario as there was significant out-of-plane warping in the uniformly oriented (uUT4) grid. However, with a random orientation of tetrahedral elements (e.g., rUT4), all of the tetrahedral element types performed well. Figure 12 displays the UT4 elements compared to the Hex8 case (in wireframe). Both images clearly show the stability issues associated with the element. The performance of the other tetrahedral elements was not an issue.

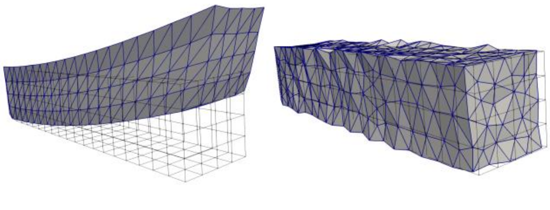

Figure 12: Displacement response (magnified 200x) of uUT4 (L) and rUT4(R) compared to Hex8 (wireframe).

3.4 Subsidence Problem

For the final test problem, we constructed a hypothetical 3D subsidence problem. The domain consisted of a 10km x 10km x 4 km mesh with a discretization of 250m element lengths in each direction. This resulted in 28577 nodes, 25600 Hex8 elements, and 153600 tetrahedral elements. After the previous tests, we were only interested in the performance of Hex8, Tet4, and TETR, so only those elements were tested. We set an initial pressure gradient of 900 kg/m^3*9.81 m/s^2. We also set temperature to vary linearly from 20ºC at the surface to 300ºC at the base. At a single node at 1500m depth in the middle of the model, we reduced the pressure to atmospheric and reduced the temperature by 100ºC to simulate a "production" change. A cross section of the model with the reduced pressure node visible is shown in Figure 13.

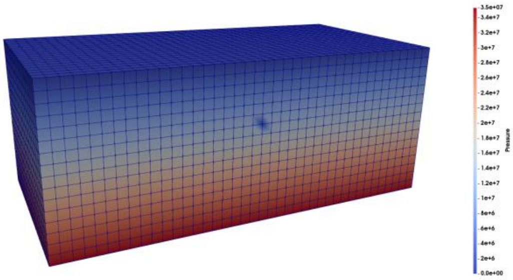

Figure 13: Pressure in subsidence model with reduced pressure in center of model at 1500 m depth due to production.

Figures 14-16 display the subsidence results for each of the element types tested. The surface subsidence values were all similar. Hex8 predicted the least amount of surface subsidence at 0.003619m. The standard Tet4 predicted the most subsidence at 0.003765m. TETR predicted a subsidence value between the two (0.003694m). Those results are consistent with Table 2 and the results of the bar under pressure/temperature load.

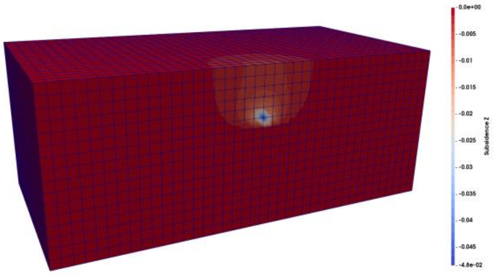

Figure 14: Subsidence response for Hex8 element type. The maximum subsidence at the surface was 0.003619m.

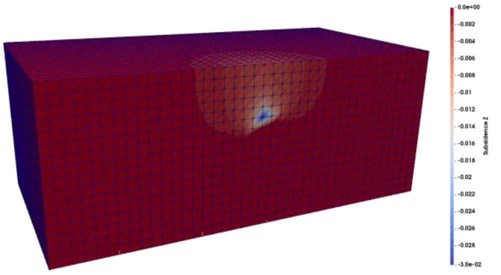

Figure 15: Subsidence response for Tet4 element type. The maximum subsidence at the surface was 0.003765m.

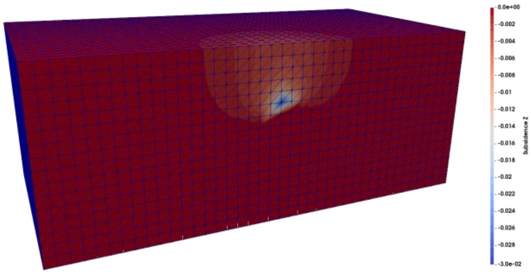

Figure 16: Subsidence response for TETR element type. The maximum subsidence at the surface was 0.003694m.

The slight asymmetry of the subsidence response in the tetrahedral cases is due to interpolation differences between grids. We specified a temperature/pressure change at a single node. In the finite element method, nodal values are interpolated within elements. By dividing an existing Hex8 element into 6 tetrahedra, some of the interpolation information was lost as not all nodes of a Hex8 are present in a Tet4.

4. CONCLUSION

We have implemented a number of tetrahedral element formulations (Tet4, Tet10, UT4, and TETR) for finite element analyses of poro-thermo-geomechanical problems and compared those to standard hexahedral elements (Hex8). In general, TETR is the best all-around tetrahedral performer. However, depending on the problem, the traditional Tet4 seems to give reasonable results. Tet10 will always be the best performer for accuracy, but has additional numerical cost and requires a different grid structure. UT4 does alleviate locking issues but is prohibitively expensive and requires additional numerical stabilization. For future geomechanical simulations, we would recommend the use of Hex8 if possible. Otherwise, the user should test a coarse mesh that can run reasonably quickly with both Tet4 and TETR to check for differences. If the differences are small, the cost of TETR is not necessary. If bending modes of deformation are present or there are large differences, then the use of TETR will be essential.

ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge Mercury NZ Ltd. for allowing the publication of this work.

REFERENCES

ABAQUS User’s Manual: *ABAQUS/Standard user’s Manual*, Volume II, Version 6.4. Hibbit and Sorrenson, Inc., Pawtucket, Rhode Island (2002).

ANSYS Users’ Manual: *Theory Reference: Release 5.6*, Edited by Peter Kohnke. ANSYS, Inc. Canonsburg, PA, USA. (1994).

Dohrmann, C., Heinstein, M., Jung, J., Key, S., and Witkowski, W.: Node-based uniform strain elements for three-node triangular and four-node tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, **47**, 1549-1568 (2000).

Fillipa, C.: *Advanced Finite Element Methods (ASEN 6367)*, Course notes retrieved July 2017 from: www.colorado.edu/engineering/CAS/courses.d/AFEM.d/, Chapters 8 and 9. (2017)

Gee, M., Dohrmann, C., Key, S., and Wall, W.: A uniform nodal strain tetrahedron with isochoric stabilization, *International Journal for Numerical Methods in Engineering*, **78**, 429-443 (2009)

Hughes, T.: *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications (2000).

Irons, B.: Engineering application of numerical integration in stiffness method, *Journal of the American Institute of Aeronautics and Astronautics*, **14**, 2035-2037, (1966).

Pawlak, T. and Yunus, S.: Solid elements with rotational degrees of freedom: Part II – tetrahedron elements, *International Journal for Numerical Methods in Engineering*, **31**, 593-610 (1991).

Pogacnik, J., O’Sullivan, M., and O’Sullivan, J.: Linking Tough2 and Abaqus to model permeability enhancement using a damage mechanics approach. *Proceedings:* World Geothermal Congress 2015, Melbourne, Australia (2015).

Pruess, K., Oldenburg, C., and Morridis, G.: *TOUGH2 User’s Guide Version 2.0,* University of California, Berkeley, California, USA (1999).

Puso, M.. and Solberg, J.: A stabilized nodally integrated tetrahedral, *International Journal for Numerical Methods in Engineering*, **67**, 841-867 (2006)

Puso, M., Chen, J., Zywicz, E. and Elmer, W.: Meshfree and finite element nodal integration methods, *International Journal for Numerical Methods in Engineering*, **74**, 416-446 (2008)

Yunus, S., Saigal, S., and Cook, R.: On improved hybrid finite elements with rotational degrees of freedom, *International Journal for Numerical Methods in Engineering*, **28**, 785-800 (1989).

Zienkiewicz, O., Taylor, R., and Zhu, J. : *The Finite Element Method : Its Basis and Fundamentals (7th Edition)*, Elsevier, New York (2013).