

---

# Validation of Grid-Based Hex Meshes with Computational Solid Mechanics

Steven J. Owen and Tim R. Shelton

\*Sandia National Laboratories, Albuquerque, New Mexico, U.S.A.  
sjowen@sandia.gov

**Abstract.** Grid-based methods for generating all-hex meshes show tremendous promise in automating and speeding up turnaround for computational simulations for solid mechanics. Recognizing some of its inherent weaknesses, there has been hesitancy in accepting this technology as a viable option for critical FEA. This study attempts to compare meshes generated with traditional manual pave-and-sweep technologies with those generated with an automatic overlay grid method. We use a simple torsion pin analysis to understand both linear-elastic and non-linear elastic-plastic responses with grid-based meshes. This study demonstrates that in the cases tested, equivalent or superior results were achieved with grid-based meshes when compared to pave-and-sweep meshes.

**Keywords:** grid-based, overlay grid, hexahedral mesh generation, parallel meshing, solid mechanics, pillowowing.

## 1 Introduction

For computational simulation in solid mechanics, the tri-linear 8-node hex element has long been favored over its tetrahedron counterparts. In spite of the often enormous overhead required to generate an all-hex mesh versus generation of a tet mesh of similar geometry, hex meshing remains an important requirement for many analysts. The ability to automatically generate a quality all-hex mesh for an arbitrary solid model has long been a major research challenge.

Most methods for all-hex meshing can be classified as either *geometry-first* or *mesh-first* approaches. The geometry-first approaches, which may include

---

\* Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

algorithms such as mapping[3], sweeping[17], plastering[2][18], whisker weaving[20] and medial axis[19], involve developing a mesh using the CAD boundary representation as a framework from which to build the nodes and elements of a mesh to fill the geometric domain. Mesh-first approaches, such as overlay and octree grid-based methods[14][22][8] first construct a space-filling grid or mesh of nodes and elements. They then employ methods to locally modify the mesh to capture features of the geometry and topology of the CAD model.

Where geometry-first methods can successfully be employed, they will usually result in high quality hexahedral elements. Since these methods are generally sensitive to boundary topology, mesh quality at boundaries is normally high. On the other hand, since mesh-first methods operate on node locations near the boundary to deform elements to fit a given geometry, the element quality near boundaries can often suffer. Current state of meshing research and development has not yet yielded a fully automatic geometry-first approach that will work for general geometries. Instead, manual tools[4][6][7][1][21] that provide some automation for decomposing the model into sweepable or mappable topologies have become the staple of all-hex mesh generation practice. Meshes produced in this manner can be heroic efforts requiring the work of many analysts weeks and months of work to produce meshes of complex assembly models.

Meanwhile, requirements for rapid design turnaround are not being met by current hex meshing practice. For example, at U.S. national labs where critical decisions of national security often hinge on information gained from computational tools, the inability to turn around a high quality hex mesh in minutes or hours has increasingly become the bottleneck. In addition, science-based engineering questions requiring vast computational resources are increasingly becoming the norm. With exascale computing soon to become a reality, current hex meshing practice will not scale to meet these needs adequately.

Because of the attractive features of geometry-first approaches the desire to develop all-hex solutions using these methods have not waned. However, the practicality and tractability of developing such solutions is not high. In recent years, the authors have invested significant resources developing a mesh-first, overlay grid method called *sculpt*[10][9]. Sculpt has the dual objectives of meeting both the rapid design turnaround and scalability objectives of the U.S. national labs. The current path to full automation of a sculpt-based mesh for complex systems of parts appears to be within reach, whereas geometry-first approaches do not currently appear to hold similar promise. Likewise, sculpt's parallel implementation demonstrates that scalability to massively parallel regimes may be tractable.

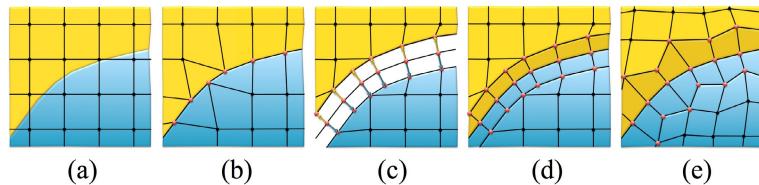
An important aspect of deploying such a tool in practice is its validation in comparison to existing meshing tools. There is a clear advantage in user overhead required to generate a sculpt mesh versus a user-crafted pave and sweep mesh. However, the resulting mesh quality inherent with overlay grid

methods should not be overlooked. In this study we attempt to weigh both speed and accuracy issues using the sculpt methodology. Considering the speed and scalability advantage for sculpt, should the meshes developed using the sculpt procedure produce meshes that are sufficiently accurate to answer important engineering design questions, it would be an enormous win. It would indicate that sculpt is a viable technology that could have immediate impact on rapid-turnaround design problems that may have critical national security implications.

Because hex meshing is particularly important for both linear and non-linear solid mechanics simulation, we focus our attention on this field. In this study we will attempt to assess the accuracy of solutions derived from a sculpted mesh compared to a mesh derived from a more traditional pave and sweep approach using the Cubit[4] Geometry and Meshing Toolkit. While not intended to be comprehensive in nature, it should illuminate some of the speed-versus accuracy trade-offs when employing a tool such as sculpt.

## 2 Sculpt

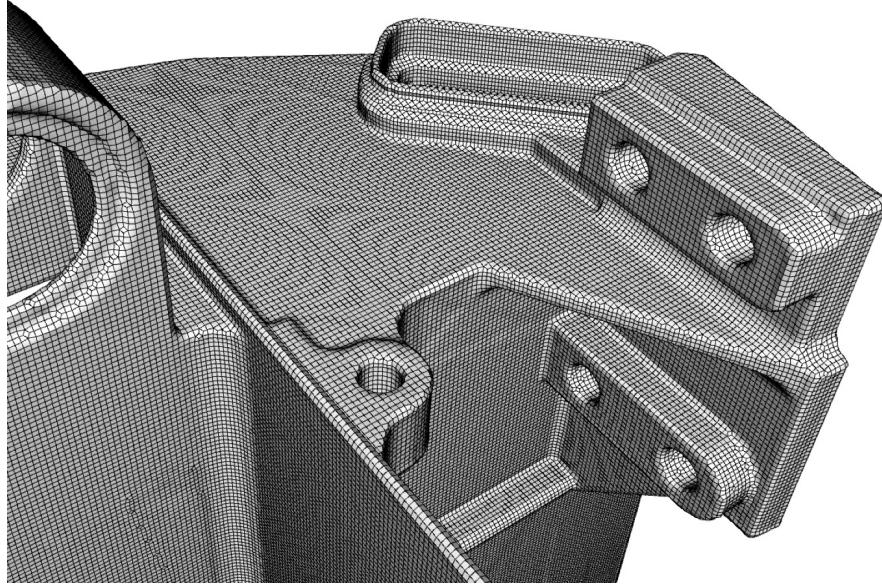
The basic sculpt procedure is outlined in figure 1. Beginning with a Cartesian grid as the base mesh, shown in figure 1(a) a geometric description is imposed. Nodes from the base grid that are near interfaces (curves and surfaces) of the geometry are projected to the geometry, locally distorting the nearby hex cells (figure 1(b)). A pillow layer of hexes is then inserted at the surfaces by duplicating the interface nodes on either side of the interface and inserting hexes (figures 1(c) and (d)). While constraining node locations to remain on the interfaces, smoothing procedures can now be employed to improve mesh quality of nearby hexes (figure 1(e)).



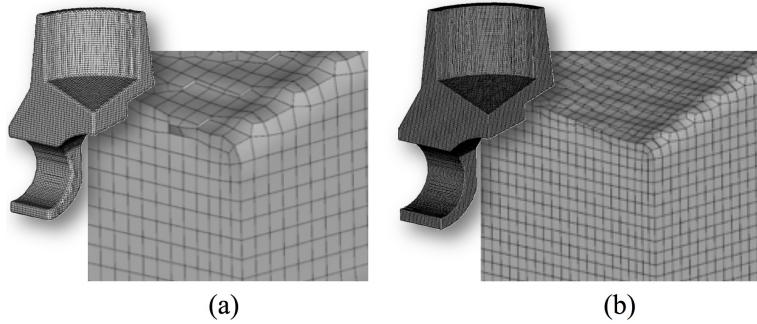
**Fig. 1** The procedure for generating a hex mesh using the Sculpt overlay grid method

Mesh-first methods are limited to capturing geometric features with the available resolution of the selected base mesh. For this reason, some methods [8][22] have employed refinement techniques to locally refine the base mesh to better capture smaller features. However accurate and robust capture of CAD features from a base mesh in practice remains a difficult process [11]. In this study we limit the scope to analysis of overlay grid meshes with

constant mesh size where no local refinement has been performed. In addition, no attempt is made to accurately capture sharp exterior features. Figure 2 shows an example of a sculpt mesh of a CAD model. Note that exterior corner features are rounded, however the effect of sharp feature capture becomes less pronounced as resolution increases as demonstrated in figures 3(a) and (b).



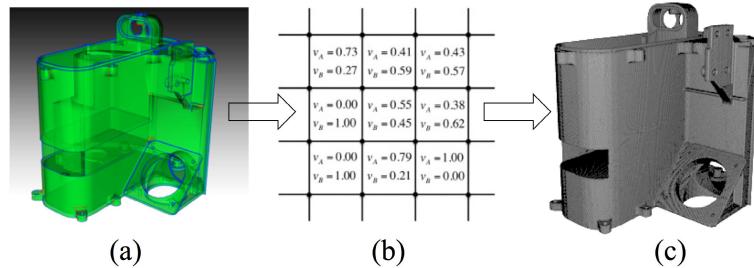
**Fig. 2** Hex mesh generated using the Sculpt overlay grid procedure



**Fig. 3** Example of same model meshed at two different resolutions

Another aspect of model preparation for computational simulation involves geometry cleanup and simplification. Traditional geometry-first meshing methods require an accurate non-manifold boundary representation before mesh

generation can begin. Small, sometimes unseen gaps, overlaps and misalignments can result in sliver elements or mesh failure. Tedious manual geometry simplification and manipulation is often required before meshing can commence. Sculpt, however employs a solution that avoids much of the geometry inaccuracy issues inherent in CAD design models. Using a faceted representation of the solid model, a voxel-based volume fraction representation is generated. Figure 4 illustrates the procedure where a CAD model serving as input (figure 4(a)) is processed by a procedure that will generate volume fraction scalar data for each cell of an overlay Cartesian grid (figure 4(b)). One value per material per cell is computed that represents the volume fraction of material filling the cell. A secondary geometry representation is then extracted using an interface tracking technique from which the final hex mesh is generated (figure 4(c)). While similar to its initial facet-based representation, the new secondary geometry description developed from the volume fraction data results in a simplified model that tends to wash over small features and inaccuracies that are smaller than the resolution of the base cell size.



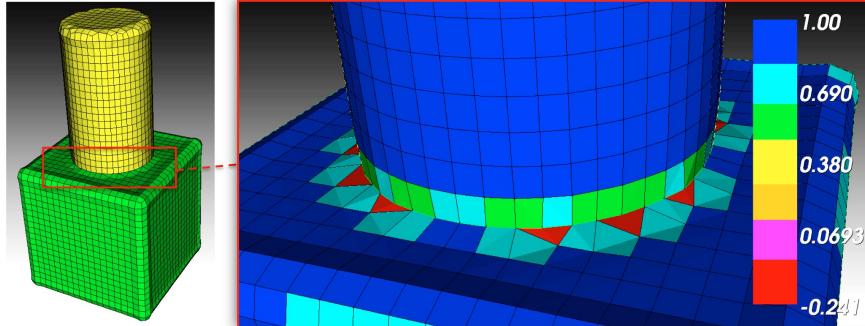
**Fig. 4** A representation of the procedure used to generate a hex mesh with Sculpt

While acknowledging some loss in model fidelity in this new volume-fraction based geometric model, the advantage and time-savings to the analyst of being able to ignore troublesome geometry issues is enormous. At the same time we must quantify what additional errors we may be introducing into the finite element solution when choosing to use this simplified volume-fraction based model. With the current study we attempt to better understand its effect on solution accuracy.

For additional details on the parallel sculpt procedure, see the authors previous work [10][9].

## 2.1 Sculpt Enhancements

One of the principal objectives of this work is the ability to produce a meshing technique that will produce elements of computable quality every time. This is a vital component of both scalability and fast-turnaround for design

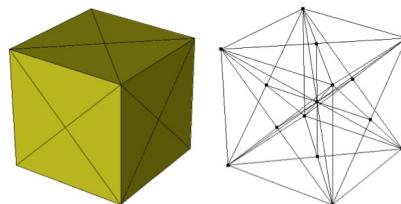


**Fig. 5** Sculpt mesh showing case of material interfaces where element quality may suffer. Scaled Jacobian metric is plotted

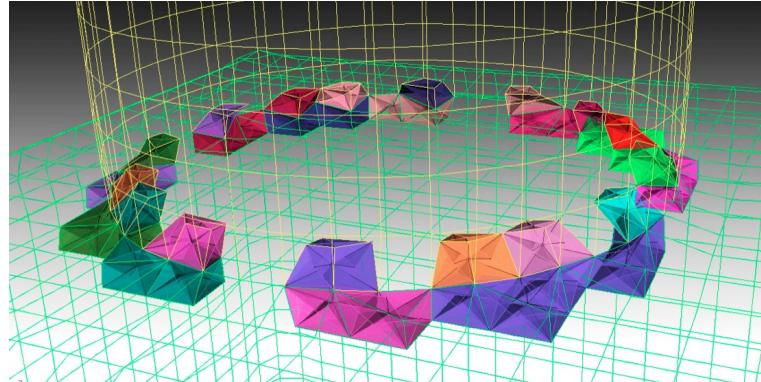
objectives. Figure 5 shows one example using the unmodified sculpt approach where element quality may suffer. In this figure two materials meet, sharing a common surface interface. For this case a curve definition must be extracted from the volume fraction representation at the boundary of the surface. Hex elements formed in this region may contain faces where 3 nodes on a single face are constrained to the same curve. Figure 5 shows the resulting mesh quality in this region where elements are colored based on the scaled Jacobian metric. Recognizing that overlay grid procedures will sometimes produce poor element quality, additional mesh modification solutions were also considered and implemented. Two such solutions, hex-dominant and pillowing are discussed here.

### 2.1.1 Hex-Dominant Meshes

For this study we looked at optionally incorporating a mixed element mesh. For simplicity, we limited the tetrahedron definition to splitting a single hex into 24 tetrahedra as shown in figure 6. For implementation purposes, the sculpt code was modified to accept an optional threshold value,  $J_s$ , where  $J_s$  is the minimum scaled Jacobian at any node of a given hex element. Any hex element falling below  $J_s$  would be converted into 24 tetrahedra. To handle



**Fig. 6** Subdivision of a single hex element into 24 tetrahedron



**Fig. 7** Tet elements generated near material interface replacing poor quality hexes

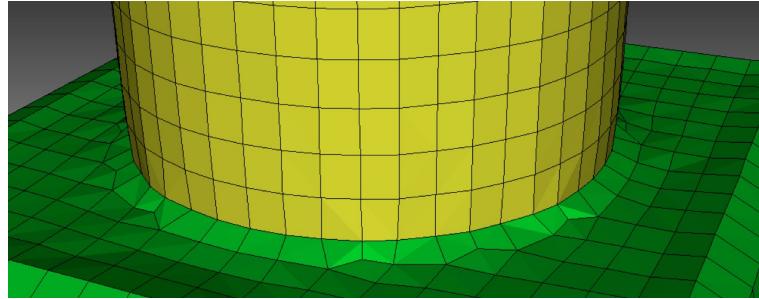
the non-conforming interface between tet and hex faces, a face-node, master-slave, tied contact pair definition was set up. The geometric description of the tied contact pairs was automatically generated with the sculpt code and used in the analysis. Figure 7 show examples where tet elements were generated near a material interface where poor quality hexes have been identified. We will evaluate the effect of incorporating tetrahedra into a sculpt mesh using tied contacts as the hex-tet interfaces.

### 2.1.2 Pillowing to Capture Curve Interfaces

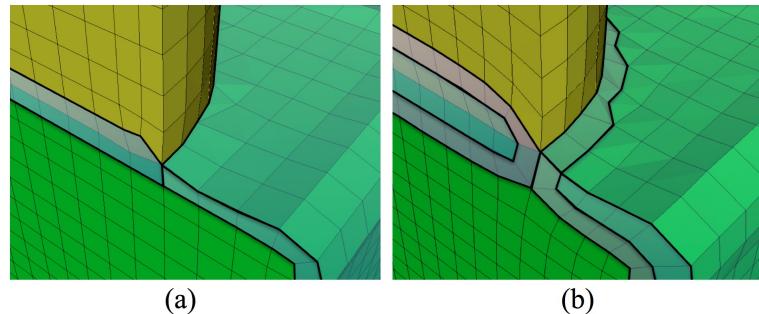
The sculpt procedure generally does a good job improving element quality for hexes immediately adjacent surfaces by inserting a boundary layer of hexes. For the case illustrated in figure 5, where curves must be captured, an additional layer of hexes may be inserted. Figures 8 and 9 illustrate the pillow insertion at a curve feature. All hexes immediately adjacent the surface attached to the curve in question are identified as shown in figure 9(a). These hexes are then shrunken and a layer or pillow of hexes is inserted to fill the space (figure 9(b)). This permits further smoothing on the surface to improve the local element quality. In this case the element quality is improved sufficient to be used in computational simulation. Sculpt provides an option to insert pillows to capture curves where element quality is lacking. We evaluate and compare the effect of pillow insertion on solid mechanics computations.

## 3 Computational Simulation Using Sculpt Meshes

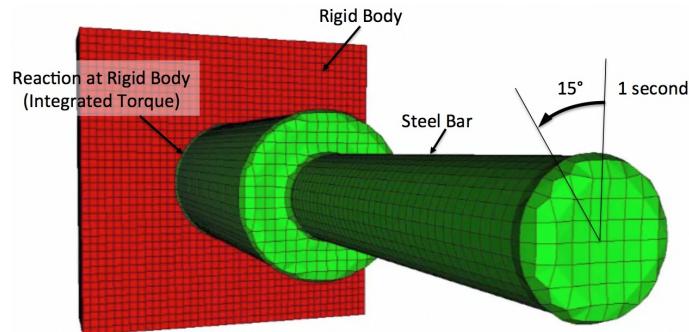
To evaluate performance of sculpt meshes we identify two typical solid mechanics scenarios and compare results generated with current finite element analysis tools. For this study we limit our scope to linear elastic and non-linear elastic plastic torsional analysis. Recognizing that many additional scenarios



**Fig. 8** Pillow inserted to capture curve feature between materials



**Fig. 9** Cut-away view of figure 8 before and after pillow insertion to capture curve

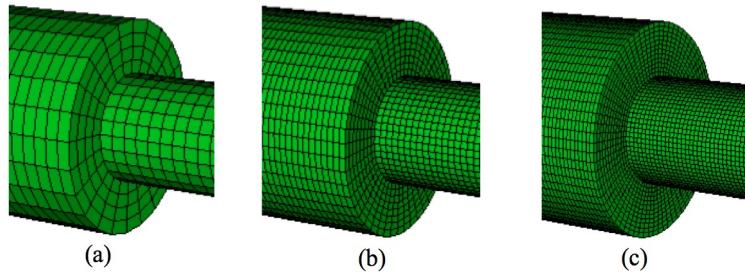


**Fig. 10** Sculpt mesh of torsion pin used in linear elastic simulation

may be tested, we purposely limit scope to cases critical to meet the authors' current industrial project design objectives.

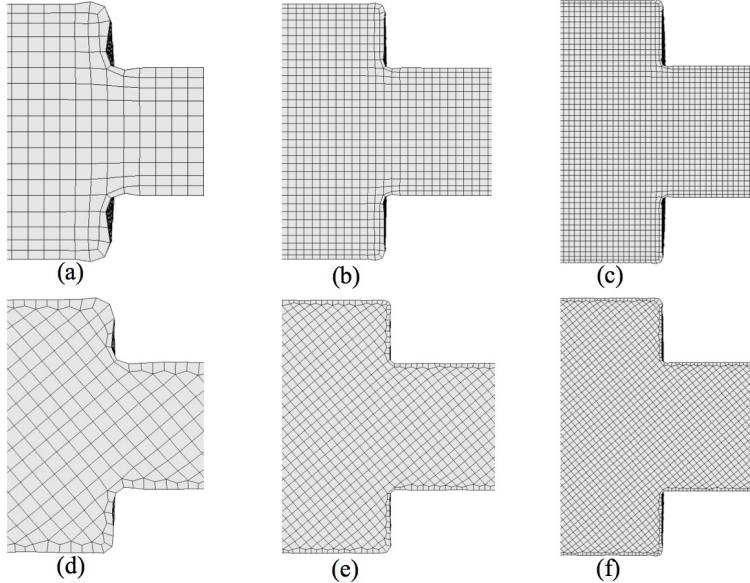
For this initial study we choose a simple torsion pin shown in figure 10. In this case we model a pin fixed to a rigid body. We apply a rotational displacement to the end of the pin and measure the integrated torque reaction at the rigid body. Analysis is performed using the explicit quasi-static code, Sierra Solid Mechanics [16] using a linear elastic material model. Rotational

displacement is applied over a 1 second time period up to a 15 degree rotation. The torsion pin itself has a step down in radius at its center recognizing that stress concentrations will develop at the re-entrant corner that must be handled by both sculpt and cubit-sweep meshes. Meshes used for this study include a series of refined hex meshes generated with cubit-sweep compared with meshes generated with equivalent cell sizes using sculpt. The cubit-sweep mesh of the torsion pin, shown in figure 11, was generated using a decomposition plus pave and sweep approach. While there is obvious simplicity in this model, the cubit-sweep mesh is representative of the highly user interactive process that is characteristic of current hex meshing practice. Figure 12 shows a cross section view of the meshes generated with sculpt for this study. The sculpt meshes displayed in figure 12, show two different orientations of the base Cartesian grid. Figures 12(a) to (c) show the mesh aligned with the orientation of the geometry while 12(d) to (e) show a 50 degree rotation of the base Cartesian grid with respect to the geometry. Although presumably ideal to align the base Cartesian grid with the main orientation of the geometric model, there is no guarantee that this can be accomplished in practice. As a result, we will look at the sensitivity of the final solution to the orientation of the base Cartesian grid. In this case we choose 10 degree increments of the Cartesian grid up to 90 degrees.



**Fig. 11** Series of refined meshes generated using cubit-sweep

Often expressed as one of the driving motivations for use of hexahedral elements is their ability to perform better in non-linear analysis than tet elements on similar geometry. As a result it was important to extend our study to incorporate a non-linear material model that would incorporate both elastic and plastic strain. For this case we incorporate the same test specimen, however we add additional twist to the rotational displacement up to 45 degrees and decrease the time to a rapid 0.01 seconds. We also incorporate a non-linear elastic-plastic material model using an explicit solver using the same Sierra Solid Mechanics analysis code. We measure the integrated torque at the rigid body and compare the results at time 0.01 seconds.



**Fig. 12** Cross sections of a series of mesh resolutions generated using sculpt (a)-(c) Sculpt-00: base mesh aligned with the geometry, (d)-(e) Sculpt-50: base mesh oriented 50 degrees to the geometry

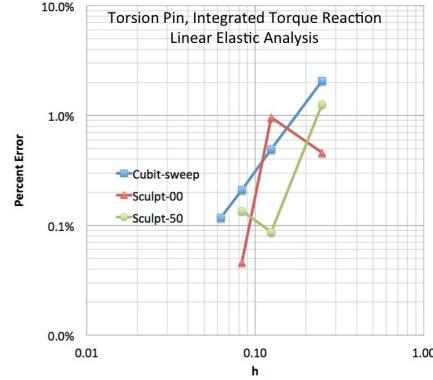
## 4 Results

### 4.1 Linear Elastic

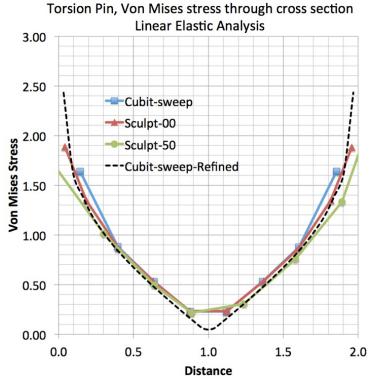
#### 4.1.1 All-Hex Solutions

The first series of tests were intended to compare results of a traditional pave-sweep mesh generated with cubit, with sculpt-generated meshes using a linear elastic material model. Figures 13 to 15 summarize the results. To determine the approximate error for our test case, a mesh convergence study was first performed on the cubit-sweep mesh. Richardson extrapolation[13] with an assumed convergence rate of 4 was then used to compute an "exact" solution which served as the basis for error analysis. Figure 13 demonstrates the convergence of the solution using the cubit-sweep mesh and compares convergence of two orientations of the sculpt meshes: the 0 and 50 degree base mesh orientations.

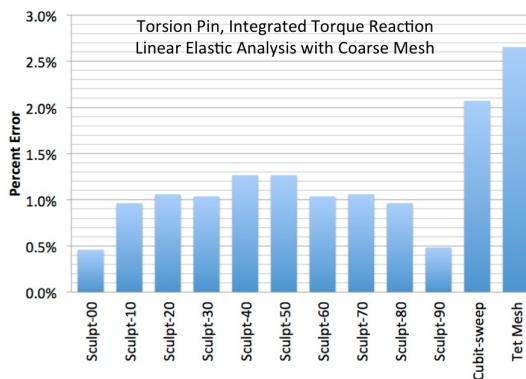
While we note the expected quadratic convergence of solutions derived from the cubit-sweep meshes, we also note some noise in the overall solution convergence from the sculpt meshes. Nevertheless results are well-within acceptable ranges to illustrate the overall convergence of the solution for sculpt. For this study we used a uniform refinement procedure for the cubit-sweep



**Fig. 13** Log of percent error vs log of mesh size ( $h$ ) comparing mesh convergence for cubit-sweep mesh and two orientations of sculpt meshes



**Fig. 14** Cross section of Von Mises stress through small diameter at reentrant corner for linear elastic analysis



**Fig. 15** Comparing percent error for integrated torque on 10 orientations of sculpt meshes with error from cubit-sweep and tet meshes

generated meshes, however used an equivalent smaller base mesh size for higher resolutions of sculpt meshes. This may contribute to the non-uniform convergence of the solution for sculpt meshes. The characteristic varying mesh size introduced by the insertion of the sculpt hex boundary layer may also be a contributing factor.

In figure 15 we record the percent error from the exact solution at time 1.0, the final time step of the simulation. Results are shown for 10 orientations of a sculpt mesh compared with cubit-sweep and tet meshes of similar resolution. For the results shown we utilize the coarsest mesh resolution for cubit-sweep and sculpt meshes as illustrated in figures 11(a) and 12(a),(d). For additional comparison, a tet mesh generated with Cubit's third party

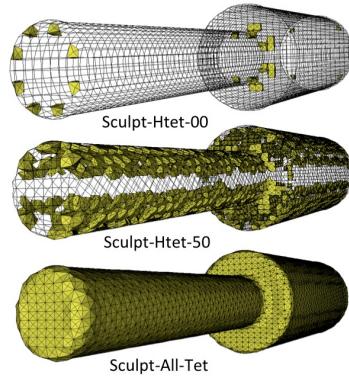
tetrahedral meshing capability[5] was also generated given an input mesh size equivalent to the cubit-sweep and sculpt meshes.

We note that sculpt meshes at this resolution and for all orientations performed with less than about 1.25 percent error while cubit-sweep and tet meshes performed with about 2.0 and 2.6 percent error respectively. Although minimal, we also note the general trend for higher error from the sculpt meshes where the base mesh was not aligned with the maximum stress plane.

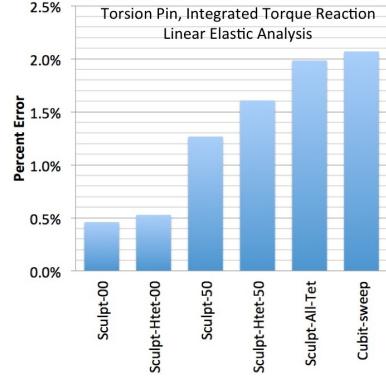
Of particular note is the superior performance of sculpt meshes for this series of simulations over a cubit-sweep mesh of similar resolution. We assert that the additional resolution provided by the insertion of the sculpt boundary layer, provides the basis for this more accurate solution. Figure 14 shows a cross section of the computed Von Mises stress through the diameter of the pin at change in radius, where the x axis represents the distance from the cylinder axis. Computed results are overlayed for cubit-sweep, sculpt-00 and sculpt-50 meshes and includes the highly refined cubit-sweep mesh for comparison. Figure 14 illustrates that for all cases the stress gradient is highest near the surface. Having more integration points (more elements) in this high gradient region produces more accurate results for the sculpt meshes of similar resolution.

#### 4.1.2 Mixed Hex-Tet Meshes

To evaluate the performance of mixed hex-tet meshes, three different meshes were developed as illustrated in figure 16. The Sculpt-Htet-00 and Sculpt-Htet-50 meshes are the Sculpt-00 and Sculpt-50 meshes shown in figures 12(a) and (d) respectively, however a scaled Jacobian,  $J_s$  threshold of 0.35 was used where hexes falling below  $J_s$  were converted to tetrahedron. The



**Fig. 16** Combined hex-tet meshes used for linear elastic analysis



**Fig. 17** Comparing percent error for integrated torque on hex-tet and all-hex sculpt meshes

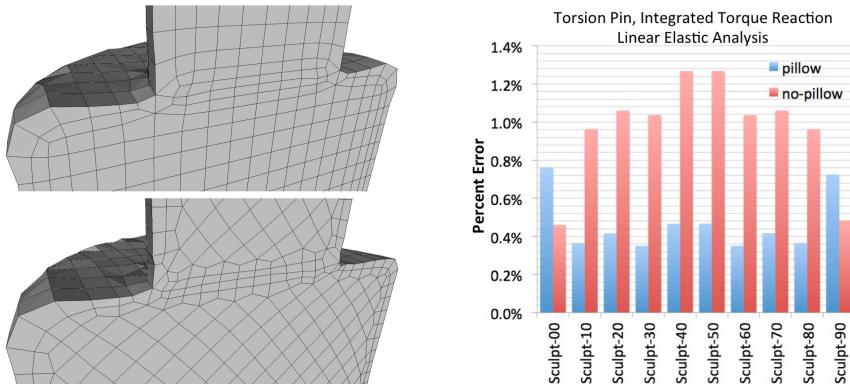
third mesh, Sculpt-All-Tet was introduced for comparison purposes where a threshold  $J_s$  of 1.0 was used, effectively converting all hexes in the mesh to tets. Figure 17 shows the percent error for the three meshes at time 1.0 for the same simulation illustrated in figure 10. Also included for comparison are the results from Sculpt-00 and Sculpt-50 meshes and the cubit-sweep mesh. Note that in all cases, percent error fell below the results from the cubit-sweep mesh. These results indicate that minimal error is introduced into the solution as a result of using a mixed element mesh with tied contacts over an all-hex mesh.

We should however note that even though the Sculpt-All-Tet mesh error came in below the cubit-sweep mesh, the mesh resolution was considerably smaller for the Sculpt-All-Tet mesh. This is a result of the 1:24 hex to tet refinement procedure illustrated in figure 6 where element edge lengths ( $h$ ) are significantly reduced. We also observe a notable increase in run-time for the mixed element meshes due to the tied contact interfaces and increased resolution from the tets (figure 20).

We also note that the additional complexity of introducing tied contacts to maintain connectiveness would certainly favor an all-hex solution over a mixed element solution when possible. Although not included in the current study, other alternatives such as insertion of pyramid elements [12] to maintain connectiveness at hex-tet interfaces should also be considered.

#### 4.1.3 Sculpt Meshes with Pillow Insertion

We also looked at the effect of introducing a pillow at an imposed surface between the two radii of the pin. A cutaway of two of the meshes used in the analysis are shown in figure 18. Results from the 10 sculpt mesh orientations



**Fig. 18** Cut-away of example pillowed sculpt meshes. Two sculpt mesh orientations are shown.

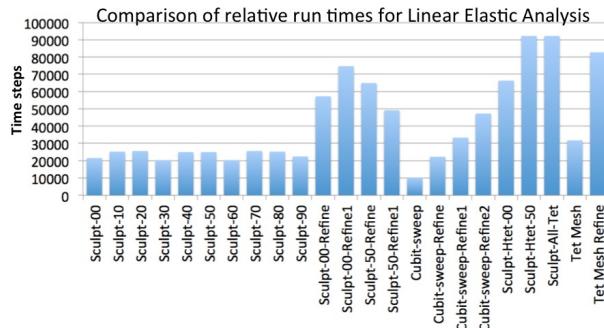
**Fig. 19** Percent error for integrated torque on 10 orientations of pillowed sculpt meshes compared with non-pillowed meshes

with pillowed surfaces are displayed in blue in figure 19. For comparison the results from the non-pillowed sculpt meshes are also displayed in red. We note that in most cases, error appears to be reduced with the introduction of the pillowed surface and that the effect of mesh orientation on solution error seems to be less pronounced than without the pillow.

The improved accuracy of the pillowed meshes can most likely be attributed to the addition of elements in the highest stress gradient region. Also, because of the case for pillowing we have selected, all orientations of the elements introduced by pillowing tend to align the mesh with the maximum stress plane at the radius reduction in the bar. This results in a lower overall sensitivity to base mesh orientation. The addition of pillowing in the sculpted meshes appears to provide some of the benefits of the geometry and mesh alignment obtained in the geometry-first techniques.

#### 4.1.4 Solution Run Time

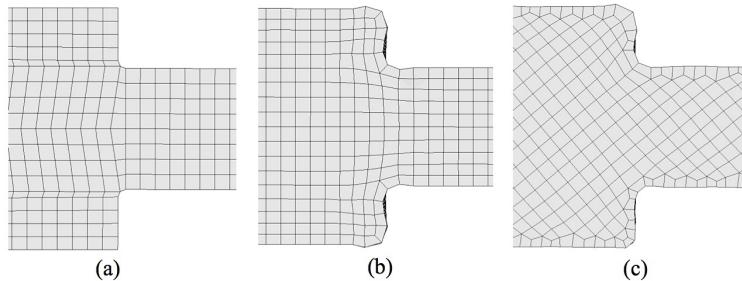
Also important to consider, is the computer run-time for the meshes we are testing. Figure 20 demonstrates the relative run-time characteristics of the different meshes in the linear elastic study. For quasi-static analysis, an explicit time-stepping procedure is employed which dictates the solution march forward below the Courant-Friedrichs-Lowy (CFL) condition. This determines the critical time step which is proportional to the characteristic element edge length of the mesh. As a result, the number of time steps used to achieve the full 1 second simulation may vary dramatically from one mesh to the next. We observed differences in the time to solution by more than an order of magnitude, with the cubit-sweep mesh running the fastest to the Sculpt-All-Tet mesh running the slowest. Sensitivity to the smallest mesh edge length appears to be a determining factor in driving computational efficiency. We should however point out that these times involve only computational resources. The human interactive time to generate the mesh is not taken into account, which for most non-trivial geometries, would significantly favor the automatic sculpt method.



**Fig. 20** Number of time steps required to achieve the 1 second simulation time for mesh cases tested

#### 4.2 Non-linear Elastic-Plastic

The next step in evaluation of sculpt meshes was the introduction of a non-linear elastic-plastic material model. In this case, recognizing the singularity of the re-entrant corner on the solution, we introduced a small fillet radius for both the cubit-sweep and sculpt meshes as illustrated in figure 21. Results from the non-linear simulations are summarized in figures 22 to 24.

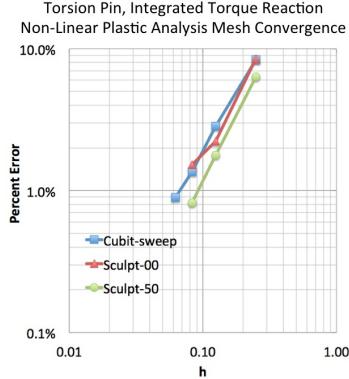


**Fig. 21** Meshes used for non-linear simulations. A fillet radius is introduced at the re-entrant corner to reduce effect of stress concentrations. (a) Cubit-sweep Mesh, (b) Sculpt-00, (c) Sculpt-50.

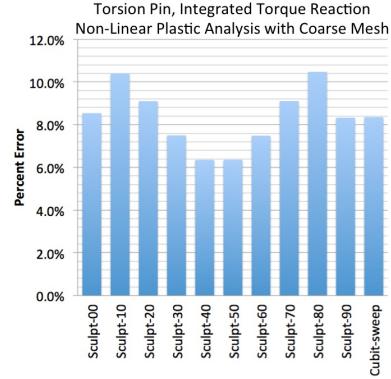
Similar to the linear-elastic case, we will define the “exact” solution using Richardson extrapolation from the solutions from four uniformly refined cubit-sweep meshes. Mesh convergence, shown in figure 22 shows reasonable behavior where both Sculpt-00 and Sculpt-50 meshes converge at a similar rate to the Cubit-sweep generated meshes.

Figure 23 shows the error at time 0.01 with respect to the extrapolated exact solution. Also shown is the error in the solution for the coarse cubit-sweep mesh. We note that the integrated torque at the final time step for the meshes of similar resolution were clustered within a few percentage points of each other. We observe that errors are considerably greater than the linear-elastic model, however seem to bracket the results from the cubit-sweep mesh.

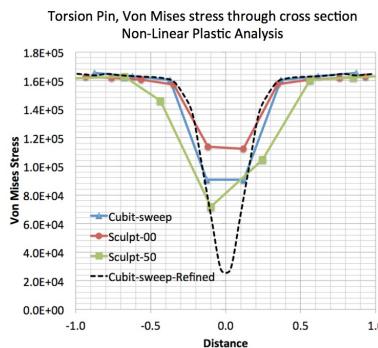
We note that close correlation of the percent error between sculpt and cubit-sweep meshes is due to the similarity in the discretization in the interior of the rod. Figure 24 illustrates the Von Mises stress through the cross section of the rod at the radius change of the bar for the coarse mesh resolutions. Von Mises stress through the same cross section from the highly refined Cubit-sweep mesh is also shown for comparison. Unlike the linear-elastic case, we observe the highest stress gradients to be concentrated on the interior where the discretization is the most regular for sculpt meshes. Since discretization for both cubit-sweep and sculpt meshes is similar on the interior, solution results also tend to be similar.



**Fig. 22** Log of percent error vs log of mesh size ( $h$ ) comparing mesh convergence for cubit-sweep mesh and two orientations of sculpt meshes for non-linear elastic-plastic analysis



**Fig. 23** Comparing percent error for integrated torque for elastic-plastic material model on 10 orientations of sculpt meshes



**Fig. 24** Cross section of Von Mises stress through small diameter at radius change in bar for non-linear elastic-plastic analysis

## 5 Conclusions

Sculpted meshes in this study have been shown to be both usable and sufficiently accurate as compared to a mesh obtained through the more user intensive cubit-sweep mesh. Indeed we note an improvement in solution accuracy for sculpt meshes for linear-elastic analysis due to the higher density of elements near the boundary where high stress gradients are observed. In a similar manner, we observe performance on-par for non-linear cases where high stress gradients tend to appear on the interior of the model. The sculpted meshes have not changed the expected convergence rate as best seen in the non-linear simulation where the discretization in the interior is virtually the same.

In this study we also observe that creation of mixed hex-tet meshes may be a viable option to avoid poor hex element quality that may result from sculpt

meshing procedures. We note that the accuracy to quantities of interest in this study showed minimal difference over a mesh using all-hex elements. We would however assert that a mixed element mesh would only be required if positive Jacobian elements were unable to be obtained without the mixed approach. While we limited this study to a simple 1:24 hex to tet replacement, a method that would more generally re-triangulate voids left where poor quality hexes are extracted would be advantageous. Collapsing edges of a limited number of otherwise poor quality hexes to form degenerate versions of hex elements may also be a fruitful area to explore as demonstrated by the author in [15] and a reasonable alternative to mixed elements.

For the case tested we noted that the addition of pillowng in the sculpted meshes improved element quality at the curve interfaces sufficient to produce a computable quality all-hex mesh without the introduction of tet elements. As an added benefit it appeared to provide some of the advantages of the geometry and mesh alignment obtained with a traditional pave and sweep approach. Further generalization and improvement of the pillowng technique to capture curve interfaces should be explored.

Given the small variance in accuracy to cubit-sweep meshes, but virtually hands off mesh creation, sculpt meshes have the potential to satisfy the needs of the design community. We recognize that the range of this study covers only a few cases for computational solid mechanics and that further work is needed to expand its scope. The current study, however indicates that the sculpt-based meshes, certainly provide a means for generating meshes that have fine enough discretization that may require large scale computing but offer the field resolution required for crucial decisions.

There remains a clear case for use of precision meshing tools such as Cubit, as they offer detailed control over geometry and mesh characteristics. Indeed we have seen highly successful and complex systems of components modeled with such tools, however at significant cost in user time. As analysts are being asked to respond more rapidly to difficult engineering design questions on larger and more complex models, these tools cannot be counted on to scale effectively to meet their needs. An approach such as sculpt, that although idealizes and may neglect some details of the model, has been shown to be an effective alternative. Enhancements such as mixed elements as well as inclusion of local pillowng operations also potentially expand the effectiveness and robustness of the method. Because sculpt, and grid-based meshing tools in general, have the potential to be more easily generalized for parallel implementation and are more easily able to mesh arbitrarily complex geometries, results from this study provide additional strong motivation to expand the development and usage of such tools.

## References

1. Ansys ICEM CFD Mesh Generation Software (2013),  
<http://www.ansys.com/Products/Other+Products/ANSYS+ICEM+CFD>

2. Blacker, T.D., Meyers, R.J.: Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm. *Eng. with Computers* 2(9), 83–93 (1993)
3. Cook, W.A., Oakes, W.R.: Mapping Methods for Generating Three-Dimensional Meshes. *Computers in Mechanical Eng.*, 67–72 (August 1982)
4. Cubit, Geometry and Meshing Toolkit (2013), <http://cubit.sandia.gov>
5. GHS3D: A powerful isotropic tet-mesher (2013),  
<http://www-roc.inria.fr/gamma/gamma/ghs3d>
6. Gridgen, Reliable CFD Meshing Software (2013),  
<http://www.pointwise.com/gridgen>
7. GridPro, Superior Engineering Solutions (2013), <http://www.gridpro.com>
8. Ito, Y., Shih, A.M., Soni, B.K.: Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. *International Journal for Numerical Methods in Engineering* 77(13), 1809–1833 (2009)
9. Owen, S.J.: Parallel Smoothing for Grid-Based Methods. In: 21st Int. Meshing Roundtable, Research Note (2012)
10. Owen, S.J., Staten, M.L., Sorensen, M.C.: Parallel Hex Meshing from Volume Fractions. In: Quadros, W.R. (ed.) *Proceedings of the 20th International Meshing Roundtable*, vol. 90, pp. 161–178. Springer, Heidelberg (2011)
11. Owen, S.J., Shepherd: Embedding Features in a Cartesian Grid. In: Proc. 18th Int. Meshing Roundtable, pp. 117–138 (2009)
12. Owen, S.J., Saigal, S.: Formation of pyramid elements for hexahedra to tetrahedra transitions. *Comp. Methods in Applied Mechanics and Eng.* 190(34), 4505–4518 (2001)
13. Richardson, L.F.: The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London, Series A* 210(459-470), 307–357 (1911)
14. Schneiders, R., Schindler, F., Weiler, F.: Octree-based Generation of Hexahedral Element Meshes. In: Proc. 5th Int. Meshing Roundtable, pp. 205–216 (1996)
15. Shelton, T.R., Crane, N.K., Cox, J.V.: An Exploration of Accuracy and Convergence of the Degenerate Uniform Strain Hexahedral Element. In: Proc. ASME 2013 Int. Mechanical Eng. Congress and Exposition (to appear, 2013)
16. Sierra Solid Mechanics Team, Adagio 4.22 User's Guide, Sandia National Laboratories, Sandia Report SAND2011-7597 (2011)
17. Staten, M.L., Canann, S.A., Owen, S.J.: BMSWEEP: Locating Interior Nodes During Sweeping. In: Proc. 7th Int. Meshing Roundtable, pp. 7–18 (1998)
18. Staten, M.L., Kerr, R.A., Kerr, O.S., Blacker, T.D.: Unconstrained Paving and Plastering: Progress Update. In: Proc., 15th Int. Meshing Roundtable, pp. 469–486 (2006)
19. Tam, T.K.H., Armstrong, C.G.: 2D Finite Element Mesh Generation by Medial Axis Subdivision. *Advances in Eng. Software* 13, 313–324 (1991)
20. Tautges, T.J., Blacker, T.D., Mitchell, S.A.: The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hex Meshes. *Int. Journal for Numerical Methods in Eng.* 39, 3327–3349 (1996)
21. TrueGrid by XYZ Scientific Applications, Inc., A Mesh Generator and Pre-processor for FEA and CFD Analysis (2013), <http://www.truegrid.com>
22. Zhang, Y., Bajaj, C.L.: Adaptive and Quality Quadrilateral/Hexahedral Meshing from Volumetric Data. *Comp. Methods in Applied Mechanics and Eng.* 195, 942–960 (2006)