Proceedings of the ASME 2023
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC-CIE2023
August 20-23, 2023, Boston, Massachusetts

DETC2023-116761

EVALUATION OF CAE DESIGN CHANGE UPDATES – A CASE STUDY ON GAS TURBINE AIRFLOW DISRUPTORS

David Procopio
Clemson University
Clemson, SC, USA

John Morris
Clemson University
Clemson, SC, USA

John Wagner, PhD, PE
Clemson University
Clemson, SC, USA

ABSTRACT

The rapid growth of digitalization in design and advanced manufacturing, characterized by computer tools, has emphasized software tools throughout the process. To compete in the global marketplace, companies need to embrace virtual design and digital manufacturing methods. Computer-aided design and computer-aided engineering embody the workflow process and tools to bring forth products from conception to design, where detailed changes can be made to fabricate the product efficiently in the manufacturing phase. This paper examines three CAD/CAE patterning features used in Siemens NX on the design of gas turbine turbulators—internal turbine blade surface discontinuities that enhance the heat transfer between the cooling air and blades on whose outside surface are subject to hot combustion gases. Each patterning feature is used to produce a pattern with a high number of instances on a variety of surfaces. The simulations' execution speed, memory usage, versatility, and integrity of the produced model is evaluated to quantify the performance of the three features. The global pandemic and supply chain challenges has shown the need for virtual modeling and engineering design platforms to efficiently complete complex projects and patterning methods applicable when replicating elements.

Keywords: CAD, CAE, gas turbines, turbulators, computer simulations, engineering analysis

1. INTRODUCTION

Global manufacturing and engineering design companies have recognized the advantages of Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) software tools. The adoption of digital design and additive manufacturing methods can accelerate the development cycle and enable greater product functionality and reliability. Corresponding to the rising utilization of CAD/CAE applications has been the need for faster executing software given the growing complexity of design applications and the need for customization throughout the product lifecycle. This is particularly true when considering the time necessary to make updates to complex models. Lengthy model change times can disrupt workflows and prevent continuous development from technical staff members, leading to lower production rates. The technology associated with change updates tends to be rather complex due to the intricacies of distributed access and product informatics. Significant changes may result in extensive backup time due to the extended amount of execution time needed for computer-aided design tools and models to be as efficient as possible.

In the open literature, CAD/CAE updates have been investigated by corporate engineering departments and on gas turbine design projects [1-2]. Han and Chen [3] studied the usage of gas turbines and the critical role that turbulator ribs play in thermal efficiency and power output. Specifically, the turbine blades rotational motion and how the turbulators ribs may be placed for better flow along the blade walls were examined. Carter [4] explored the common failures associated with turbine blades and whether their rotational velocity could cause issues if there happened to be a build-up of air in certain areas. Tests were performed in support of failure analysis (how the system could fail safely without significant damage). Other CAD/CAE software studies were completed in Switzerland with mathematical calculations to automatically parameterize the models' dimensions that are deemed highly complex [5]. A test on the CAD/CAE system by Teslyuk et al. [6] considered design software's energy efficiency and time relation to find new upgrades for user-friendliness and complex geometries. Lastly, Brujic et al. [7] optimized gas turbines' various shapes and configurations methodologies to create higher performance automated loops. Instead of just the specific area being manipulated, these loops carry out changes to the whole model by orienting the figure in particular ways to allow the changes to be efficiently added.

The scope of this paper will compare the model update times related to gas turbine turbulator ribs that are patterned inside blades to help increase the productivity of development cycles and explore CAD/CAE execution issues. The turbulator rib patterns are simulated via three different structural rib platforms to evaluate the performance in terms of application ease, simulation error, adaptability, end product relevance, and the execution speeds of both an initial and then scaled model. The remainder of this paper is organized as follows. Section 2 examines what change updates are, and how they are beneficial to designers. Section 3 contains a gas turbine blade-inspired case study in which the turbulators ribs are patterned using three methods. Section 4 presents the results of the case study with analysis. Section 5 concludes the paper.

2. CHANGE UPDATES IN COMPUTER-AIDED DESIGN

The availability of CAD software can improve the work responsibilities of design engineers tasked with complex activities. Before the advent of the CAD tools, designers would draft blueprint sketches of parts for the manufacturing process by hand. Redrawing could take hours or even days because the detailing associated with these drawings had to be specific enough for the manufacturers to fabricate. Since a technical design must frequently pass through several stages of approval in a company before it is released to start manufacturing, many changes will come about that must each be approved. Change updates require the designers to go back and erase what they have already drawn; sometimes, a brand-new drawing for the product feature in question would be necessary. Redrawing was not only time-consuming and labor-intensive, which is where CAD software represents a paradigm shift. It allows engineers to create three-dimensional models and promptly facilitate updates compared to earlier methods, as shown in Figure 1.

Product changes that occur during the design phase are known as change updates. Change updates are an essential aspect of product engineering as they enable designers to track design modifications and convey them to subsequent members of the technical staff [8]. These changes occur more frequently during the early design phases, before the product definition becomes more rigid. Changes can cause significant disruption in the development process, though this disruption can be mitigated through effective change management and clear communication between involved departments. Changes can also propagate throughout the product system, for instance, changes to a critical dimension may require updates to the interfacing parts, revised testing standards, cost analyses, and manufacturing processes.

Frequently, product changes occur for reoccurring reasons, leading to the notion of predicting changes. Designing a product for change reduces the cascading effect of change updates, as well as guiding the development process more generally. Though predicting the impact of a change update is imprecise, simulation tools espoused in CAD/CAE software can provide product managers more information about the effects of arbitrary changes. These new software features could also prohibit inexperienced users from updating a design when they make an unrecommended change.

Product Data Management (PDM) software extends these benefits when used in tandem with CAD/CAE programs. Composing modifications through by a PDM system prevents unauthorized model manipulations as well as providing methods for change reversal through version control. Another advantage of using PDM software is the distribution of changes to engaged users (such as through cloud-based engineering design [9], so that all stakeholders can examine the modifications and access the accompanying justifying information. Users are no longer responsible for saving and updating any changes that they, or someone in a separate department, create. Instead, the automatic saver can trace changes and add them to the same server so that each change is uploaded for the availability of the entire company. Change examinations should not create issues with the final working product since it is saved in the workspace. PDM systems can also typically handle merge and branching operations, allowing parallel development. Though heavily utilized in the design stage, PDM brings value to an organization throughout a product's lifecycle, especially when employed in the management of engineering changes [10]

A collaborative environment exists with the automatic update feature to accommodate simultaneous contributions from multiple local/distant engineers on design projects. For example, two engineers collaborate to design a compact mobile robot transmission. One person creates the planetary gear set while the other designs the support structure with holes and protrusions. Each individual can immediately view the design progress, allowing for interference and compatibility issues to be resolved. All workers must adopt the same software toolset and follow established guidelines and company practices for maximum effectiveness and efficiency. Manual overrides of the software protocols should be discouraged to prevent the loss of information due to missing data/time stamps accompanying ownership attributes [2]. As many parts have a long service time, future engineers will likely appreciate the details and contact information to inquire about their applications.

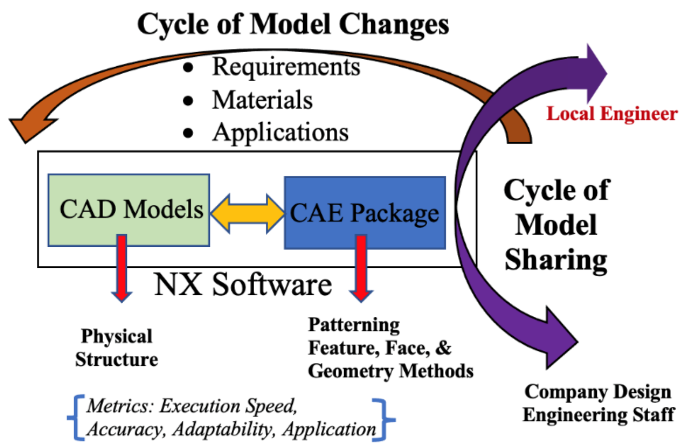


FIGURE 1: LANDSCAPE OF DEPOSITORY MODEL CHANGES AND UPDATES

3. CASE STUDY – APPLICATION OF CHANGE UPDATES ON GAS TURBINE TURBULATORS

The selected case study investigates design changes associated with automatic features implemented in CAD software for turbulator ribs on gas turbine blades. The case study specifically addresses how design changes and subsequent engineering time cascade through a complex component. The NX modeling software is used to model the gas turbine blade features and explore updates for additional reductions. The model of the gas turbine blade is shown in Figure 2, which also shows the thousands of turbulator tabs inside a serpentine channel located within the blade to promote cooling. In a gas turbine, the inlet air is compressed in the compressor stage, mixed with (natural gas) fuel, and ignited in the combustion chamber [11]. Power is then harvested from the exhaust gas to generate electricity. Portions of compressed air are routed through the inside of the blades and out surface holes to cool the blades. The turbulator ribs are flow disruptors to help promote air mixture and convective heat transfer before exiting through the holes into the turbine chamber. The serpentine surface turbulators are modeled through CAD software using a patterning feature option. Any changes made to the drawings take a significant amount of time, decreasing the change update time. The software offers several different patterning options, which are explored in detail for linear and curved surfaces to find the most suitable choices for implementing changes.

The relationship between the CAD and CAE is espoused in the estimation of the impact of rib height, pitch, angle of attack, roughness level, and orientation for the ribs and turbulators on thermal transport. The relationship between these parameters and the behavior of the internal fluid flow is highly non-linear, as well as covering a wide range of heat transfer paradigms and flow paradigms [13]. Different rib geometries must be developed in CAD and then simulated in a corresponding CAE application using numerical methods to see the impact on the heat transfer. An example of a turbulator rib is shown in Figure 3. Tests are then conducted on the ribs at different angles to determine the various placement angles of the ribs and turbulators throughout the serpentine cooling channels. Detailed distributions of the local heat transfer coefficient for developing the flow in short rectangular channels with rib turbulators are essential. The appropriate rib angles for attack must be identified in each track to establish the highest heat transfer coefficient accompanied by the highest pressure drop. The high heat transfer coefficient and pressure drop will help obtain the best heat transfer performance for a constant pumping power.

With these complex calculations, numerous change updates to original models will need to be processed, however, these model updates can be time-consuming. In some cases, due to the size of the geometric models, model updates cannot be configured automatically by the software. If the software cannot handle this update, the design engineer must go and manually change each rib and turbulator to match the change update calculated in the previous testing scenarios. Managing these computionally expensive patterns can be an laborious task for turbine designers, though some of the ardor can be mitigated through the intelligent selection of patterning features.

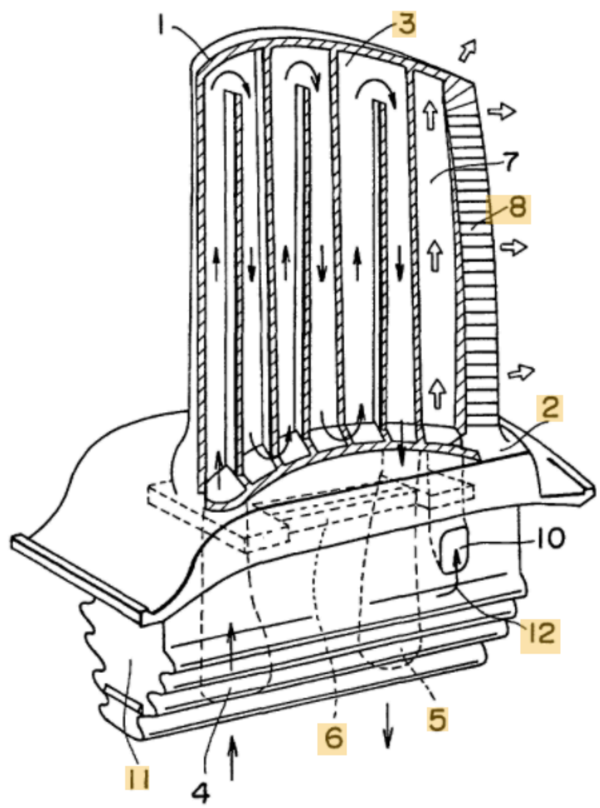


FIGURE 2: SERPENTINE CHANNEL INSIDE OF TURBINE BLADE [11]

The Siemens NX program offers three different patterning options, namely feature, face, and geometry, each of which can impact the design process in unique ways. To estimate this influence on the design process, each patterning type was utilized in various configurations to pattern a simplified turbulators. Each update option was then executed to compare performance based on the analysis parameters, including execution time. The three patterned structures with turbulator ribs can be classified as linear, curved, and cylindrical. The face of each surface will be selected for the placement and curvature of the turbulators to follow. After each initial test was run, the turbulator density was scaled by a factor of five to ten times and the metrics compared on this larger scale. To be consistent throughout the simulations, the same computer hardware capable of 1425 GFLOPS was used throughout the study. The methodology was also monitored for consistency and completed as follows: First, the platform surface is created. Second, the geometry of the simulation is identified. Third, the initial turbulator is applied to the platform surface and a 5mm spacing is entered into the patterning system for each turbulator. Fourth, the number of patterned turbulators is calculated. Fifth, the patterning method is selected. Sixth, the simulation is run and the time recorded for how long it takes the software to process the selected number of turbulators onto the platform surface.

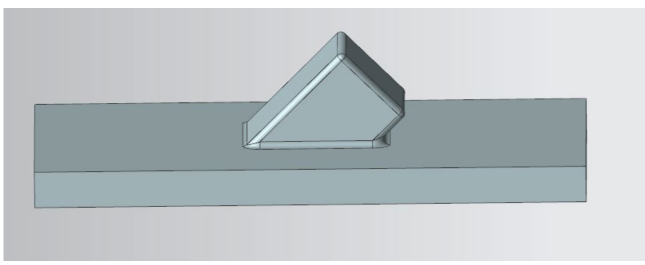
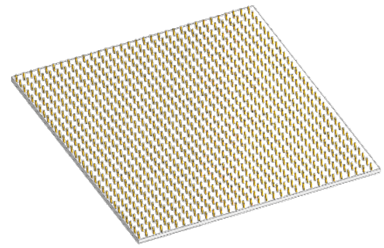


FIGURE 3: CLOSE UP VIEW OF A TURBULATOR RIB ATTACHED TO A WALL SIDE

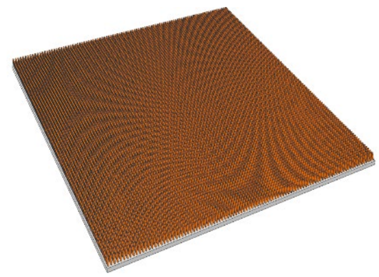
3.1 Test 1- Linear Geometry with Varied Patterning

The feature pattern method shown in Figure 4a is the most detailed patterning mechanism in NX. The structure selected seed element is resolved for each patterned instance. The initial patterning process featured 1024 turbulators and took 12 seconds to complete, finishing without any complications. The number of turbulators was then scaled by ten, as shown in Figure 4b, so that 10,000 turbulators were created. This test finished in 1345 seconds, an increase of more than two orders of magnitude. Despite its long runtime, feature patterns are required for many simulation software and manufacturing processes, since each patterned instance can be manipulated independent of the others while maintaining the patterned relationship with the original seed.

The face pattern command was also tested which copies only the outside surfaces and Booleans of the selected fixture, limiting the feature information of the newly created instances. Additionally, the turbulator ribs are not attached to the initial linear front, which means that a system would recognize them as separate entities instead of a whole body. The initial simulation patterned 1,024 turbulators with the face method. The generated execution time for those turbulators was six seconds and did not produce any errors. The test was scaled by ten so that the number of patterning instances was now 10,000. The execution time for this simulation was 748 seconds. The face patterning method was similar to the feature pattern in simulation accuracy, versatility, and model integrity.



(a)



(b)

FIGURE 4: LINEAR STRUCTURE – (a) WITH 1,000 PATTERNED TURBULATOR RIBS, AND (b) WITH 10,000 TURBULATOR RIBS PLACED USING FEATURE PATTERN METHOD PER TEST 1A

The geometry patterning command copies the coordinates of the planar faces of the seed geometry, duplicating the geometric entities at each new instance. This method does not include Boolean information, which increases processing speed and versatility but also causes the pattern to become less accurate. The initial test of the geometry pattern was with 1,024 turbulators. The execution time was 11 seconds with no reported errors. The second test, scaled by a factor of ten, produced 10,000 turbulators with geometry patterning. The execution time was 369 seconds. This patterning strategy was the least promising as it showed the lowest performance numbers for the ease of application, simulation accuracy, adaptability, and relevance of the end product. The only comparable performance was the execution time, consistent with the two other patterning tests.

3.2 Test 2- Curved Geometry with Varied Patterning

A curved geometry configuration was considered next for virtual testing, as shown in Figure 5, with the three different patterning options. The pattern's initial test was on one-half of the curved surface. For the second test, the entire inside surface of the curved surface was patterned with the selected turbulators. However, the number of turbulators may vary slightly due to the differing constraints of the specific pattern command chosen. Note that the memory used by Test 2 was not collected due to these feature differences.

The first curved geometry simulation used the feature patterning command. As shown in Figure 6, this command recreates the individual features of the original seed entity. The initial simulation was feature patterned with 124 turbulators. This execution took 45 seconds, however, issues arose as the created turbulators did not maintain the correct surface orientation throughout, limiting the models viability. The scaled test was applied to 1240 turbulators on the entire front surface. The execution time was 2 hours which was close to the timeout limit of the application.

FIGURE 5: CURVED GEOMETRY MODEL THAT THE TURBULATORS WILL BE PLACED ONTO USING PATTERNING OPTIONS PER TEST 2

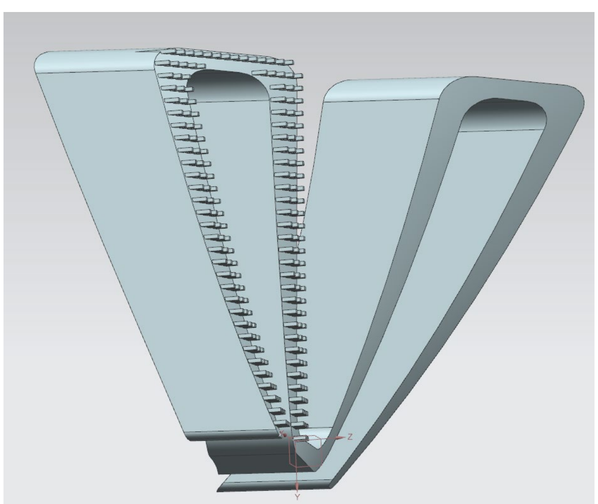


FIGURE 6: CURVED GEOMETRY WITH FEATURE PATTERN METHOD – INITIAL TEST (TEST 2A)

In Figure 7a, the face pattern command was tested on the same curved geometry. This patterning command does not require the amount of detail that the feature option needs. It only needs the surfaces of the turbulators to pattern it instead of the datum axis. However, the face pattern can only occur in one specific direction at a time. The initial simulation patterned 252 turbulators on only half of the curved surface and took 56 seconds; no errors or complications were reported. The subsequent scaled test produced 2520 turbulators on the curved surface, requiring an execution time of 12 seconds. This pattern required two independent features since the face option does not allow both halves of the surface to be patterned together. The process was then analyzed and rated. The highest performances came about in the simulation accuracy category and in the relevance of the end product. In addition, the execution times for both the initial and scaled tests were low.

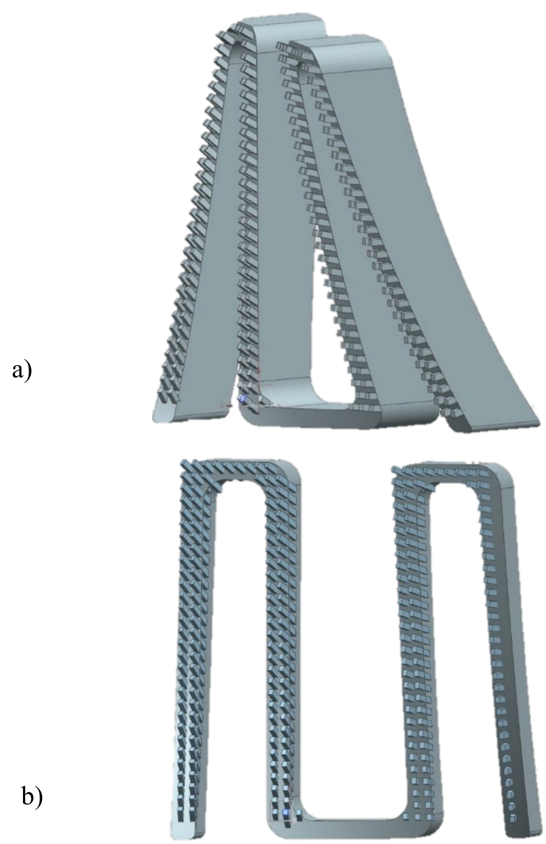


FIGURE 7: TURBULATORS APPLIED TO CURVED GEOMETRY USING (a) FACE (TEST 2B), AND (b) GEOMETRY (TEST 2C) PATTERNING COMMANDS

The geometry patterning command was used in the same initial turbulator position on the curved surface. The initial patterning test on half the curved surface created 225 turbulators. This process took 131 seconds to complete. The protruding ribs out of the different curved surface areas were recorded as an error in the pattern execution. The test was then scaled by a factor of ten so that the system now had to produce 2250 turbulators onto the entire curved surface. This process needed 25 seconds which accounted for all the turbulators with two separate patterns for each half of the surface. Errors that occurred were consistent with that of the initial test.

3.3 Test 3- Cylindrical Geometry with Varied Patterning

The last set of tests were performed upon a cylindrical shell to replicate the rounded insides of a turbine blade, as shown in Figure 8. A single turbulator ring was pre-patterned along the bottom of the cylinder for the structural elements to succeed. The initial ring was created to ensure a consistent starting point for each test.

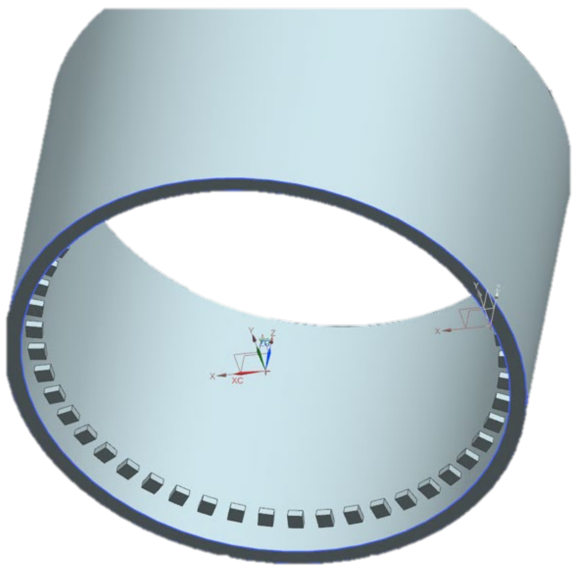
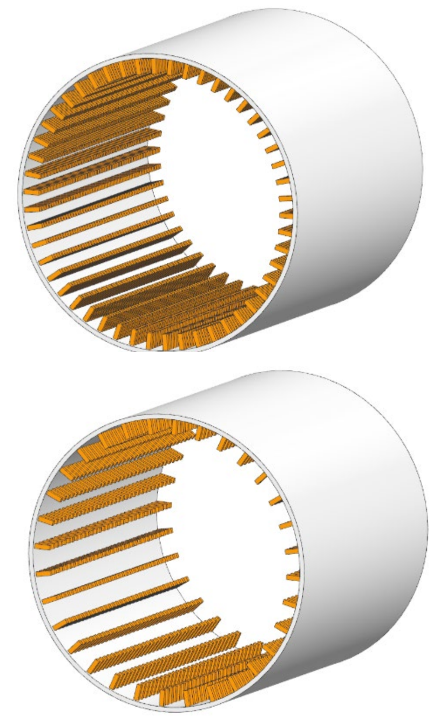


FIGURE 8: ORIGINAL CYLINDER WITH AN EXAMPLE SINGLE RING TURBULATOR RIB PATTERN

The cylindrical test first used the feature pattern command per Test 3A. The initial ring of turbulators was replicated to a final tally of 1023 turbulator ribs for the examination, as shown in Figure 9a. The total execution time required for the process to complete was 0:01:27. The test was then scaled by 2.5, and 2500 turbulators were created, per Figure 9b.

Test 3B was next for the cylindrical geometry system and used the face patterning method for the turbulator ribs. The initial test was executed using 1023 turbulators, as shown in Figure 9a. The time to complete the patterning was 8 seconds, including the pre-set initial turbulator rib ring. The test was scaled by a factor of 2.5 to create 2500 turbulators ribs inside the cylinder in Figure 9b. The scaled test final run time was 34 seconds. The quicker execution speeds was about half that of the feature pattern command. Test 3C followed the same pattern using the geometric command. The time to execution for the smaller model was 10 seconds, while the scaled-up test was executed in 21 seconds.

a)



b)

FIGURE 9: FEATURE PATTERNING METHOD ON A CYLINDRICAL SHAPE (TEST 3A) WITH (a) 1024, AND (b) 2500 TURBULATORS

4. ANALYSIS AND DISCUSSION

The value of a patterning command can be discussed along several dimensions as summarized in Table 1. This article discusses five: execution speed, memory usage, model integrity, and versatility. These quantitative measures are defined as follows:

- Execution speed: The time necessary to complete a patterning operation, rounded up to the nearest second.
- Memory usage: The increase in memory associated with a successful pattern.
- Model integrity: The usability of the model for simulation work in connection with its design intent following a patterning operation.
- Versatility: The amount of configuration necessary for a patterning command to adapt to the various surfaces used in the tests.

Both execution speed and memory usage can be reported with relative precision based on experimental procedures. While Table 1 records the results of each test, comparisons are made from a normalized scale calculated by the following:

$$\beta_j = \frac{x_i - Mdn(x)}{\frac{1}{n}\sum_{i=1}^{n}(|x_i - Mdn(x)|)} \tag{1}$$

where *Mdn* is the median of the set *x* of related test results. This metric utilizes a Mean Absolute Deviation (MAD) referencing the median of the data set *x* rather than the mean due to the scarcity of data in each set (3 values). The full meaning of the metric is to relate the residual of each data point x_i to the median of the set divided by the MAD (again, with the residual referencing the median rather than the mean of the set). Each factor, β_j, represents the ranked performance of the command on a specific test. The factors, β_j, are then mapped to a scale of 1-5, where 5 is better performance and 1 is poor performance.

Performance for both of these parameters (execution speed and memory usage) increases as the values decrease, or in other words, lesser values are better. The following mapping to the five-point scale is used:

1. $\beta_j = (2, \infty)$
2. $\beta_j = (1, 2]$
3. $\beta_j = (0, 1]$
4. $\beta_j = (-1, 0]$
5. $\beta_j = (-\infty, -1]$

The other two parameters are more subjective and can't be quantified with the same metric. To compare them on the same normalized five-point scale, they are evaluated on a custom Likert scale, with the following criteria:

1. Pattern operation failed to execute
2. Pattern succeeded but with no useability
3. Pattern operation succeeded but with limited useability
4. Pattern operation succeeded with satisfactory useability
5. Pattern operation succeeded with maximum useability

The aggregated scores (the means of the individual β_j factors) are shown in Table 1 and summarized for comparison in the radar plots in Figure 10.

There are three major results made obvious in Figure 10. The first is that the feature pattern command has a much slower execution time than either the face pattern or the geometry pattern. This is due to the added operations involved in constructing independent features in a pattern set, which must be created, solved, and then stored in the design history. Juxtaposing is the speed of the geometry pattern, which benefited from far less operations, and additionally does not need to be merged with any solid bodies in the model.

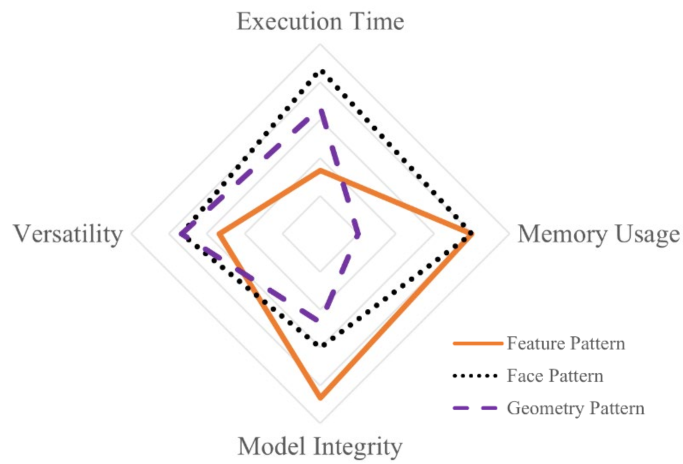


FIGURE 10: COMPARISON OF RESULTS AVERAGED FOR ALL SIX TESTS PERFORMED WITH THE THREE PATTERN COMMANDS; CORNERS OF THE RADAR PLOT REPRESENT IDEALIZED VALUES, WHILE POOR PERFORMANCE IS ASSOCIATED WITH MORE CENTRAL POINTS

The lack of Boolean operations with the patterned features is key to the second key observation, which is that the models made with the geometry pattern lack integrity. This corresponds with the intended usage of a geometry pattern, which is purposed more towards aesthetic design rather than robust production models. The third major trend is the high memory usage of geometry patterns. Though different CAD platforms may store patterning features differently, it is thought that the marked uptick in memory usage for geometric patterns is due to their lack of continuity. Feature and face pattern commands may have a systematic way for representing duplicated information, while geometric patterns have no such compression mechanism.

Much of the remaining conclusions drawn from the comparisons of the three commands reflects their varying purposes. Geometric patterns are generally lightweight, more versatile, and simple to execute, but disallow a robust solution to be found for the model. Face patterns allow for a more integral solution without much loss in speed, though instances cannot utilize feature solutions. Feature patterns are the most robust, least versatile, and most computationally expensive option. Any designer requiring modification of individual instances (created through the patterning feature) will consequently need to use a feature pattern despite the tradeoff in execution speed.

Though the actual performance of each patterning command is dependent on the workstation it is processed on, the relationships between each command help differentiate them from each other. The most critical component for execution time with a pattern is the number of features the engine is tasked with replicating. All turbulator ribs used in this study were simple bosses with five faces as the only existing features. It is added anecdotally that the time necessary for a pattern command to execute has been observed to scale linearly with the number of

Test	Surface Shape	Pattern Command	Test I No. of Instances	Test I Execution Time (s)	Test I Memory (kb)	Test II No. of Instances	Test II Execution Time (s)	Test II Memory (kb)
1A	Linear	Feature	1023	12	5573	9,999	1345	56441
2A	Linear	Face	1023	6	5582	9999	748	54672
3A	Linear	Geometry	1023	11	11274	9999	369	110132
1B	Curved	Feature	124	45		1240	7200	
2B	Curved	Face	252	57		2520	720	
3B	Curved	Geometry	225	131		2520	1500	
1C	Cylindrical	Feature	1023	14	6440	2499	68	16055
2C	Cylindrical	Face	1023	8	5928	2499	34	14212
3C	Cylindrical	Geometry	1023	10	9953	2499	21	24242

TABLE 1: RESULTS SUMMARY COMPARING THE THREE PATTERNING METHOD PERFORMANCES ON SURFACES

features—that is, a seed with ten features will take twice as long to pattern than a seed with five. Designers looking to reduce change update times should first attempt to simplify seed instances to minimize patterning times.

5. CONCLUSION

To remain competitive in the global business world, companies must embrace the digital design and production methods centered around computer-aided design and computer-aided engineering. CAD and CAE encompass both processes and tools and play a critical role in today's engineering design landscape. Overall, these computer tools can digitize tasks for future data retrieval, data mining, and accelerated design cycles. This paper explores the performance of three patterning methods used on three differing surfaces to provide an even comparison of each command. When applicable to use, the face pattern command offers a good compromise on parameters of execution time, memory usage, model integrity, and viability. Lifelong learning and continuous improvements in the software will help engineers in the 21st century create more sophisticated and higher quality designs.

ACKNOWLEDGEMENTS

The authors wish to thank Betsy Black and Bryan Lewis from GE Power (Greenville, SC) for their technical guidance in this study. The periodic meetings throughout the project clarified the gas turbine turbulators and use of CAD/CAE within the power industry.

REFERENCES

[1] C. Kim, B. Kim, and H. Kim, "4D CAD Model Updating Using Image Processing-Based Construction Progress Monitoring," *Automation in Construction*, vol. 35, pp. 44-52. 2013. https://doi.org/10.1016/j.autcon.2013.03.005.

[2] R. H. Katz and E. Chang, "Managing Change in a Computer-Aided Design Database," Electrical Engineering and Computer Science Department, University of California, Berkley, pp. 455–462, January 1987.

[3] J. Han, and H. Chen, "Turbine Blade Internal Cooling Passages with Rib Turbulators," *Journal of Propulsion and Power,* vol. 22, no. 2, pp. 226-248, 2006. doi:10.2514/1.12793.

[4] T. Carter, "Common Failures in Gas Turbine Blades," *Engineering Failure Analysis,* vol. 12, no. 2, 2004. doi: 10.1016/j.engfailanal.2004.07.004.

[5] E. Petrakova, and V. Sumatokhin, "Development of Algorithm for Creating Parametric 3D Models, Controlled by Mathcad Calculations, to Study Parameters of Enclosed Gears Housing," Lecture Notes in Mechanical Engineering, Proceedings of the 5th International Industrial Engineering Conference, pp. 473-483, Sochi, Russia, 2019. doi:10.1007/978-3-030-22041-9_51.

[6] T. Teslyuk, I. Tsmots, V. Teslyuk, M. Medykovskyy, and Y. Opotyak, "Architecture and Models for System-Level Computer-Aided Design of the Management System of Energy Efficiency of Technological Processes at the Enterprise," Advances in Intelligent Systems and Computing II Advances in Intelligent Systems and Computing, International Conference on Computer Science and Information Technologies (CSIT), pp. 538-557, Lviv, Ukraine, 2017. doi:10.1007/978-3-319-70581-1_38.

[7] D. Brujic, M. Ristic, M. Mattone, P. Maggiore, and G.P. Poli, "CAD Based Shape Optimization for Gas Turbine Component Design," *Structural and Multidisciplinary Optimization,* vol. 41, no. 4, pp. 647-659, 2009. doi:10.1007/s00158-009-0442-9.

[8] K. L. Davis and R. F. Klemm, "Simplified Generation of Design Change Information on a Drawing in a Computer-

Aided Design (CAD) Environment," *U.S. Patent No. 7,086,028,* San Rafael, CA: Autodesk INC, August 2006.

[9] D. Wu, J. Terpenny, and D. Schaefer, "A Survey of Cloud-Based Design and Engineering Analysis Software Tools," in Volume 1A: 36th Computers and Information in Engineering Conference, Charlotte, NC, August 2016, pp. V01AT02A016. doi: 10.1115/DETC2016-59341.

[10] W. Liu, Y. Zeng, M. Maletz, and D. Brisson, "Product Lifecycle Management: A Survey," in Volume 2: 29th Computers and Information in Engineering Conference, Parts A and B, San Diego, CA, January 2009, pp. 1213–1225. doi: 10.1115/DETC2009-86983.

[11] M. Schobeiri, "Gas Turbine Design, Components, System Interpretations", New York, NY: Springer, 2019.

[12] K. Suenaga, and S. Aoki, "Gas Turbine Blade," *U.S. Patent No. EP0955449B1*, Washington, DC: U.S. Patent and Trademark Office, 1998.

[13] J. Han, and J. Park, "Developing Heat Transfer in Rectangular Channels with Rib Turbulators," *International Journal of Heat and Mass Transfer,* vol. 31, no. 1, pp. 183-195, 1988. doi:10.1016/0017-9310(88)90235-9.