OASIS Challenge Documentation

(Open-source Additive Scanning Implementation Strategy)

Prepared by

The National Center for Defense Manufacturing and Machining (NCDMM)

Shawn Nesmith, Senior Project Engineer

486 Cornell Rd.

Blairsville, PA 15717

shawn.nesmith@ncdmm.org

3 September 2020

America Makes OASIS Challenge

Table of Contents

Table of Tables

Table of Figures

1.0 Executive Summary

Over the past five years, America Makes has sponsored research programs to develop open-source programs for scan path generation and machine control for Laser Powder Bed Fusion Additive Manufacturing (PBFAM). These codes are available to all America Makes members and are ideally suited for research into the process. This OASIS Challenge is an experiment in crowd-sourcing: We ask all interested parties to contribute by improving these open-source programs, and we are encouraging participation with prizes totaling $68,000.00.

America Makes, working in collaboration with the Air Force Research Laboratory (AFRL) and General Electric – Global Research (GEGR), has created an innovative approach to advance the additive manufacturing industry by challenging the brightest minds from industry and academia to utilize open-source software developed in America Makes' project 4039, "*Development and Demonstration of Open-Source Protocols for Powder Bed Fusion Additive Manufacturing (PBFAM)*", to advance the technology of open-source software and machine platforms. While America Makes has made tremendous progress in advancing open framework software, controls, and equipment with multi-laser metal additive manufacturing (AM), substantial development is necessary to realize improvement in microstructure homogeneity, surface roughness, geometric fidelity, and strength via optimization of laser scan strategies. This challenge, directed at the AM industry, looks to advance the state-of-the-art in temporal and spatial thermal management through submission of innovative scan strategy codes, algorithms, or methods.

A code submission and down-selection process will be conducted according to a series of quantitative metrics for ranking the submissions and identifying the best codes. America Makes will work with GEGR in utilizing the most promising scan strategies to produce physical Ti-6Al-4V samples on the Concept Laser M2-Open Direct Metal Laser Melting (DMLM) machine that was developed under America Makes project 3024, "*Acceleration of Large-Scale Additive Manufacturing (ALSAM)*", and evaluating the samples using a set of relevant evaluation/inspection criteria.

Once the samples have been evaluated, America Makes will host a virtual workshop allowing top performing participants to present their approaches to their peers and government. America Makes will coordinate and present monetary awards for the top three submissions and three honorable mention submissions. Cash prizes up to $25,500.00 will be awarded. The results and associated data will be integrated into the America Makes National AM Roadmap and the America Makes Digital Storefront.

2.0 The Challenge

The OASIS Challenge is a direct challenge to the additive manufacturing industry to advance laser-based powder bed additive manufacturing through the submission of open-source computer codes that output laser scan paths with innovative strategies, algorithms, and methods. This is a *scan strategy challenge*, as opposed to a *process parameter challenge*. All bulk parameters are constrained, and participants are challenged to develop the best methods for scanning each layer of the challenge geometries.

3.0 Eligibility

Domestic and foreign members who are in good standing with America Makes are eligible to participate. Only the lead proposer must be a member of America Makes. Participants of the lead proposer's team may be made up of America Makes members and non-members. If the lead proposing organization is not currently an America Makes member, the organization must become one by **December 1st, 2020**. Information on how to join America Makes is available on the membership page. (Note: The foreign membership approval process requires a minimum of three weeks to complete.)

Members associated with the evaluation panel are not eligible. This includes participants employed by NCDMM, AFRL, and GE.

4.0 How the Challenge Works

1. Download all necessary files and file folders from: https://github.com/AmericaMakes/OASIS-Challenge-Baseline-Code-and-Models
   - **OASIS_Challenge_Parts_Release_v0.zip** (download and unzip)
   - **OASIS baseline source code** folder
   - **OASIS baseline precompiled binaries** folder
   - **OASIS Challenge Documentation.pdf**
2. Review this document and all the documentation contained within the file folders thoroughly.
3. Develop your code (see "Requirements" for submitted code specifics).
4. Complete the build parameter file (see "Requirements" for build parameter file specifics).
5. Design your own trophy/medallion (see "Requirements" for trophy/medallion design specifics).
6. Enter the contest at https://www.americamakes.us/oasis/. You will receive a link to a private GitHub repository where you will submit your content. After the deadline, the repository will be switched from private to public and all participant access will be removed until judging is complete.

   ****NOTE: Multiple entries per organization is permitted, however only one (1) entry per individual. A new private GitHub repository is required for each entry****
7. Update your GitHub repository with your code, trophy/medallion design, and completed build parameter files.
   - One (1) submission per entry, *must include*:
     - "**Submitted Code**"
     - "**Code License**"
     - "**Script Files**" (there are three (3) script files required per entry)
     - "**README File**"
     - "**Build Parameter Files**" (there are two (2) build files required per entry)
     - "**Trophy/Medallion File**"

8. The OASIS technical team will evaluate the submitted codes against the "*Submitted Code Evaluation*" criteria provided within this document (see Table #2 under "Review/Scoring Criteria").
9. The top selected codes will be sent to GEGR in order for the build plate (Figure #1) to be printed using Ti-6Al-4V on a laser metal powder bed system.
    - Prints will include:
        - Four (4) tensile bars
        - Two (2) angled nuts
        - Two (2) lattice cones
        - Participant designed trophy/medallion
        - "Mystery Part"

    ****NOTE: Not all submitted codes will be sent for printing, only the top few****
10. The OASIS technical team will evaluate the printed parts against the "*Ti-6Al-4V Print Evaluation*" criteria provided within this document (see Table #4 under "Review/Scoring Criteria").

    ****NOTE: Build plate will be stress relieved and parts cut from plate using wire EDM. No further processing will be done except as specified in Table #4 under "Review/Scoring Criteria"****
11. America Makes will announce the winners.

    ****NOTE: All submitted codes are open-source and will be publicly accessible to America Makes members****

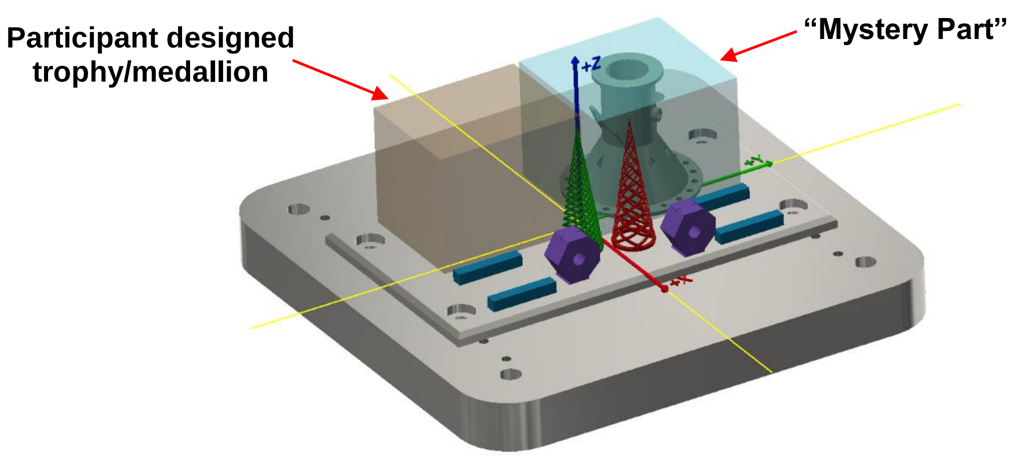

Figure #1
OASIS Challenge Build Plate
Material: Ti-6Al-4V

5.0 Requirements

"Submitted Code" Requirements:

- Source code that is submitted by the participant.
- Source code file types: C#, C++, Java or Python
- Naming convention: **Source_Code_*<Organization Name>.<file extension>***
  - Example: Source_Code_America Makes.py
- Must be open source (public) code, subject to "Code License" Requirements below.
  - All private code will be disqualified.
- All submitted code must be one (1) of the following code languages:
  - C# / C++ / Java / Python
- Virtual target machine and environment on which the code will be (compiled and) run.
  - 2 virtual CPU's Intel® Xeon® CPU E5-2673 v4
  - 2.30GHz
  - 64-bit OS
  - X64-based processor
  - 4GB RAM
  - Microsoft Server 2019 Operating System using Docker containers

  ****NOTE: Only one (1) CPU will be dedicated to the program execution, single thread****
- Submitted code that relies on custom libraries for performing the function that is the intent of the OASIS Challenge will be disqualified. Third-party libraries that provide general-purpose utilities (e.g. xml parsers, I/O, general-purpose math functions) are allowed. Examples of third-party library that would disqualify an entry: slicer, scan generation.
  - If library code is added to the entrant GitHub repository so that it can be addressed, then it is allowed.

****NOTE: All submitted codes are open-source and will be publicly accessible****

"Code License" Requirements:

- The participant submitted code must be clearly labeled and licensed under one (1) of the following three (3) open-source licenses:
  - BSD 2-Clause
  - Apache 2.0
  - MIT
- All libraries required to compile and run the code must also be licensed under one (1) of the above licenses.
- If required, subdirectories of the code may use a different license selected from the three (3) above. Include the license file in that directory. Code directories without a license file inherit the license file from the parent directory.
- System libraries that are part of the Microsoft Windows/10 or Microsoft Windows Server normal installation are allowed for code that is built on Microsoft Windows.

“Script Files” Requirements:

- The submitted code must be able to be compiled via a batch file, named “*build.bat*” or “*build.sh*”, which must be submitted with the source code. The resulting executable should go into the top-level directory. The executable shall reference the input configuration file and geometry(-ies) from the same directory.
- Two (2) additional batch files must be included to execute the compiled code on the two (2) different sets of geometries: “*run_NIST.bat*” for the NIST artifact (i.e. specified by the input configuration file “OASIS_Input_Config_NIST_Plate_Baseline.xls”) and “*run_Build_Plate.bat*” for the challenge geometry (i.e. specified by the input configuration file “OASIS_Input_Config_Build_Plate_Baseline.xls”). The code should execute as a Windows command-line program, not as an interactive UI. The argument to the executable shall be the input configuration file.
- The output from the program shall be placed into a subdirectory as specified by the input configuration file (e.g. “OASIS_NIST_Plate_Baseline”), and shall contain two (2) subdirectories: one (1) for layer files and one (1) for scan path files. The layer files subdirectory, at a minimum, will contain the SVG file output from the program. These will be used for visual inspection of the code output and shall be output at a rate of 1 per 10 layers (fixed for OASIS Challenge). The scan files subdirectory will contain two folders: one for the SVG output (fixed at 1 per 5 layers), and one (1) for XML output (1 per layer).
- The submitted code must create valid XML and SVG files as defined in the ALSAM3024 multiLaser XML schema 2020323.docx file located in the https://github.com/AmericaMakes/OASIS-Challenge-Baseline-Code-and-Models/OASIS baseline source code/Documentation directory.

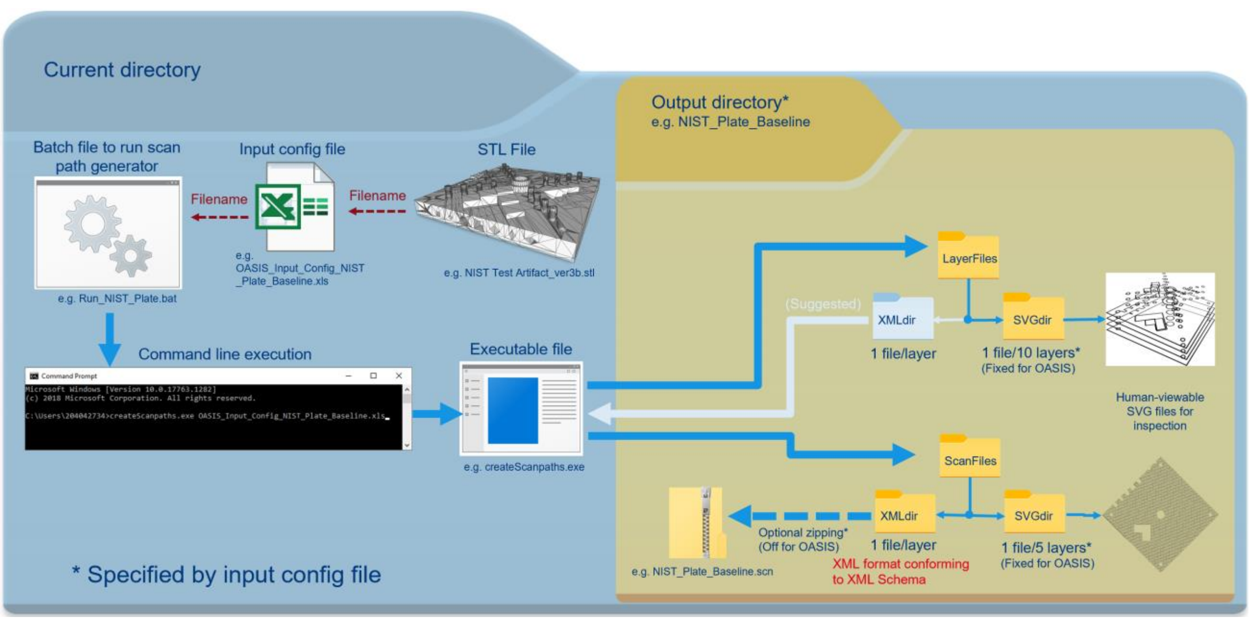

Figure #2
Batch-file execution of scan path generator

- Code will be compiled using Docker containers. Entrants may provide a Dockerfile to use for building their executable and the Docker build and run commands in their script. If a Dockerfile is not provided, then the following images in Table #1 will be used to build the entries:

Docker Base Image **Table #1**		
Build File	**Language**	**Docker base image**
build.bat	C/C++	Probably based on mcr.microsoft.com/dotnet/core/sdk: 3.1.401-nanoserver-1809 with mingw and gcc installed.
build.bat	Python	python:3.7-windowsservercore-ltsc2016
build.bat	Java	openjdk:16-windowsservercore-ltsc2016
build.bat	C#	mcr.microsoft.com/dotnet/core/sdk: 3.1.401-nanoserver-1809
build.sh	C/C++	gcc:10.2
build.sh	Python	python:3.8.5-alpine3.12
build.sh	Java	openjdk:11
build.sh	C#	mcr.microsoft.com/dotnet/core/sdk: 3.1.401-alpine3.12

“README File” Requirements:

- Code submission must include a README file describing how to build and run the code.
- Must also state any custom libraries that the submitted code uses.

“Build Parameter Files” Requirements:

- Scan strategy parameters to use for building the part. The participant will specify the values to be used, if the submitted code is selected for printing.
- Two (2) pre-filled configuration files are provided as part of the baseline source code for the (up to) two code evaluations.
  1. OASIS_Input_Config_NIST_Plate_Baseline.xls
     - Configured to run the scan path generator on the NIST test artifact geometry (NIST Test Artifact_ver3b.stl)
     - File is located in the https://github.com/AmericaMakes/OASIS-Challenge-Baseline-Code-and-Models/OASIS_baseline_precompiled_binaries/Project examples/C OASIS parts directory.
  2. OASIS_Input_Config_Build_Plate_Baseline.xls
     - Configured to run the rest of the challenge geometry, as laid out on the build plate.

- Contains placeholders for the Mystery Challenge part (Mystery_challenge_part_bounding_box.stl) and the user's designed trophy/medallion part (Users_choice_part_bounding_box.stl).
- File is located in the https://github.com/AmericaMakes/OASIS-Challenge-Baseline-Code-and-Models/OASIS_baseline_precompiled_binaries/Project examples/C OASIS parts directory.

- The two (2) configuration files are designed to be run using the precompiled code that is released with the baseline source code.
    - These two (2) files are intended to be modified and used as inputs to your source code. Save the files for submission using the following convention:
        - The file type must remain ".xls" (Excel 1997-2003 compatible)
        - Change the word "Baseline" in the file name to your organization's name
        - Example: "OASIS_Input_Config_NIST_Plate_America Makes.xls" and "OASIS_Input_Config_Build_Plate_America Makes.xls"
- Thoroughly review the OASIS allowable build parameter ranges.pdf file located in the https://github.com/AmericaMakes/OASIS-Challenge-Baseline-Code-and-Models/OASIS_baseline_precompiled_binaries/Project_examples/C_OASIS_parts directory.
    - This file contains the scan strategy parameters that are ***adjustable and can be changed*** by the participant along with the process parameters that are ***fixed and cannot be changed*** by the participant for this Challenge. Cells in the configuration spreadsheets with greyed out text are fixed for the purpose of this challenge.
    - Your scan strategy may require more parameters than are provided by the baseline configuration file. You may add additional parameters to the configuration file, but do not remove any of the existing parameters to maintain backward compatibility with the provided hatching schemes.
        - Hint: You may add a new tab/worksheet to the configuration spreadsheet to contain the newly added parameters.
        - Hint: You can add the enumeration to your hatching scheme to the existing list by adding it to the "Hatch scheme" on tab "5.Regions", column L.
- Be sure to replace the "*Users_choice_part_bounding_box.stl*" (on Tab **6.Parts**) with your own trophy/medallion stl file (including position offsets) in your "*OASIS_Input_Config_Build_Plate_<Organization Name>.xls*" file.

- "Trophy/Medallion" Requirements:
    - Designed trophy/medallion submitted by the participant that will be printed, if the submitted code is selected for printing.
    - File type: Binary or ASCII *.stl (Please dimension your part in millimeters.)
    - Max file size: 100 MB
    - Naming convention: **Designed_Part_<*Organization Name*>.stl**
        - Example: Designed_Part_America Makes.stl
    - Volume bounding size: 60mm x 75mm x 60mm (X, Y, Z).
        - The part must be located inside of the highlighted volume shown in the drawing Users_choice_part_bounding_box.pdf (note the center point), which is included with the challenge geometry. This can be achieved by:

1. exporting the STL file with the same coordinates as the build plate, or
2. applying the X, Y and Z offsets to the STL file in the input configuration file.

- Consolidated volume (including supports) is limited to 30,000mm^3 (30cm^3).

  ****NOTE: This is just over 10% of the bounding volume****
- The designed trophy/medallion must have features (i.e. holes) to allow unfused powder to drain out and support, if required.
- There will be no post processing beyond stress relieving and cutting the part from the plate (no support removal, no polishing, no heat treatment, etc.)
- No additional modifications (e.g. scaling, positioning, support geometries, repair, etc.) to the stl file will be performed by the OASIS team. It is strongly recommended that participants confirm their stl files are well formatted (water-tight, with no inverted normals), and that their code will produce valid outputs for the submitted part.

**** NOTE: If the designed trophy/medallion does not fit within the volume bounding size, or the consolidated volume is larger than specified above, it will not be printed****

6.0 "Mystery Part"

The "Mystery Part" is a representative aerospace component, defined by the OASIS Team, composed of several geometric features including:

- Thin walls
- Thick to thin transitions
- Various radii
- Small overhanging features
- Channels/holes of various geometries and in multiple orientations

Volume bounding size: 70mm x 75mm x 60mm (X, Y, Z).

"Mystery Part" information:

- Height: 44.45 mm
- Volume: 40,328 mm^3
- Surface Area: 60,878 mm^2

The performance of all code entries will be evaluated using the same Mystery Part along with the other geometries of the Build Plate. No further information regarding the Mystery Part will be provided until after judging is complete.

7.0 FAQ and Challenge Clarifications

- A list of frequently asked questions and clarifications regarding the challenge will be compiled and updated periodically on the challenge website. Questions should be submitted to challenge@americamakes.us.

  ****NOTE: Due to the potential volume of requests, not every question comment may be replied to****

8.0 Submission Due Date

Participants can enter and start committing code on October 1st, 2020. The deadline for submissions is ***Tuesday, December 1st, 2020*** at Midnight. Submissions received after the deadline will not be considered.

Submit Challenge Responses to: https://www.americamakes.us/oasis/.

9.0 Winner Selection Date

OASIS Challenge award winners anticipated notification date is Friday, June 4th, 2021.

10.0 Awards

There will be six (6) awards totaling $68,000 broken down the following way:

1st Place: $25,500

2nd Place: $15,500

3rd Place: $10,500

Honorable Mention 1: $5,500

Honorable Mention 2: $5,500

Honorable Mention 3: $5,500

****NOTE: The award checks will be written to the America Makes member organization, not to the individual. No award money under this challenge can be paid to non-U.S. citizens from China, Iran, Iraq, Syria, Sudan, North Korea, Cuba, or Venezuela.****

11.0 Submission Criteria

- Submissions not consisting of all required information will be disqualified.
- Late submissions will not be considered.
- Private code will not be considered.
- One (1) submission per entry, *must include*:
  - "**Submitted Code**"
  - "**Code License**"
  - "**Script Files**" (there are three (3) script files required per entry)
  - "**README File**"
  - "**Build Parameter Files**" (there are two (2) build files required per entry)
  - "**Trophy/Medallion File**"

  ****NOTE: See "Requirements" on the specifics for each file****

12.0 Review/Scoring Criteria

Scoring is broken down into two (2) categories (**100%**):

- Submitted Code Evaluation (**20%**) – see Table #2
  - If there is a tie in the "Code Quality" score, "Code Run Speed" will be used to break the tie. The fastest "Code Run Speed" time will then be selected.
- Ti-6Al-4V Print Evaluation (**80%**) – see Table #4
  - Angled Nuts (*30%*)
  - Tensile Bars (*30%*)
  - Lattice Cones (*20%*)

Submitted Code Evaluation (20*% of Total Score*)
Table #2

Review Elements	Description	Evaluation Tools	Score Percentage	Score Weighting		How Element is Scored	Comments
Code Quality	The following will be checked: - # of vulnerabilities - # of bugs - # of code smells - # of security hot spots	SonarQube	50%	Bugs	50%	Normalized, weighted score based on evaluation from SonarCloud.io using default rules and estimate of effort to remediate.	See SonarCloud.io and Table #3 below.
				Vulnerabilities	25%		
				Security Hot Spots	15%		
				Code Smell	10%		
Code Run Speed	Total time it takes to create valid XML and SVG files for: - NIST Artifact (Figure #3) - Build Plate (Figure #4) (w/o trophy/medallion)	Run time on a virtual machine utilizing Microsoft Server 2019 Operating System using Docker containers. Single core, single threaded.	50%	1		Normalized score based on all participant results.	No run times longer than 6 hours. Run times over 6 hours will be disqualified.

<u>SonarCloud.io (Sonar)</u>

Sonar cloud service, found at http://sonarcloud.io, will be used to assess the code quality of the submissions. This tool can be run on public GitHub repositories, so will not be performed until the deadline passes and the GitHub repository is switched from private to public.

The rules for assessing the code can be found at https://rules.sonarsource.com/. Navigate to the appropriate language.
****Note languages will have different rules which may impact the scoring of different projects****

Sonar's analysis includes an estimate of the amount of time required to remediate a given issue which is a proxy for the complexity of the remediation effort. These estimated remediation times will be used as a basis for calculating the score with a "perfect" entry having no issues. The classification of the issues will impact the weight applied to the time estimate. For instance, the time to remediate bugs will contribute to 50% of the code quality score while "code smells" will contribute 10%. Each category will be normalized by determining the worst case entered. See Table #2 below for some example scoring:

Code Quality – Example Scoring Table #3					
Project	Bugs (days)	Vulnerabilities (days)	Code Smell (days)	Security Hot Spots (days)	Total Score (%)
	50%	25%	10%	15%	
A	1.125	0.014	35.000	1.000	25.67%
B	1.208	0.001	2.083	0.001	49.12%
C	0.083	0.125	19.000	0.000	66.12%
D	0.013	0.000	9.000	0.125	95.01%
Perfection	0.000	0.000	0.000	0.000	100.00%

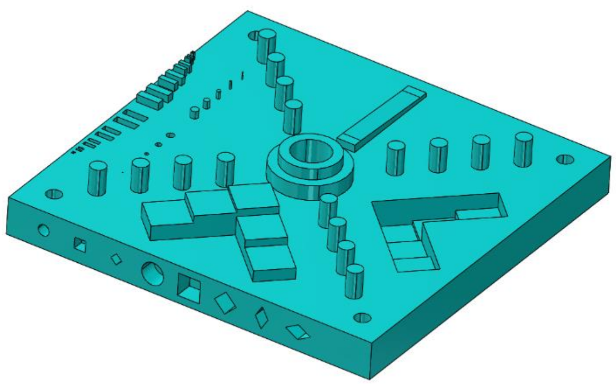

Figure #3
NIST Artifact

Participant designed trophy/medallion not included in "Code Run Speed" evaluation

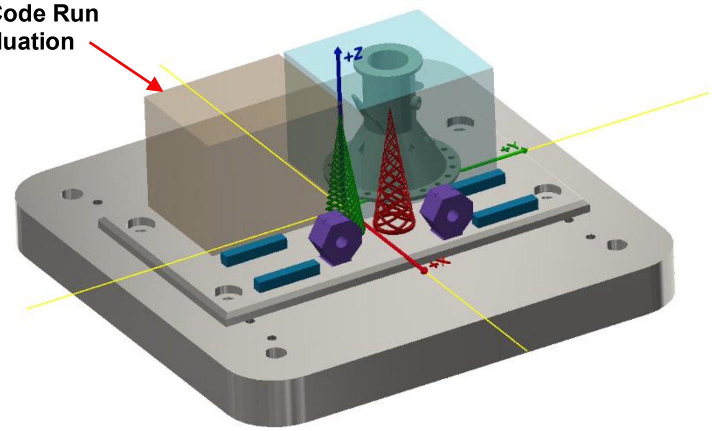

Figure #4
Build Plate

Ti-6Al-4V Print Evaluation (*80% of Total Score*) Table #4							
Angled Nuts (*30% of Total Score*) **NOTE: 1 nut will be left whole / 1 nut will be cross sectioned at notches, mounted, and polished**							
Review Elements	**Description**	**Units**	**Evaluation Tools**	**Score Percentage**	**Score Weighting**	**How Element is Scored**	**Comments**
Surface Finish	24 Measurements - 1 un-sectioned nut - 8 total surfaces - 3 patches per surface - does not include the "cut" mounted surface - does not include the hole	microns	Alicona Profilometer (S_a)	25%	1	Score (%) = 100*(1-(avgS_a-minS_a)/minS_a)	1 whole nut per build plate. Average S_a will be used in scoring.
Geometric Fidelity	Un-Sectioned Nut - All 8 surfaces - Hole size Sectioned Nut - Hole size	mm	Optical CMM (Nikon touchless point cloud scanner)	35%	1	Score (%) = 100*(1-(dev_{avg}-dev_{min})/dev_{min})	Position of EDM cut to remove the nuts from the build plate will not affect the Geometric Fidelity. Average deviation will be used in scoring.
Porosity	Cross-section one of the nuts then mount, polish, and photograph the entire cross-section. Bulk porosity, near surface porosity (250 microns from the surface), and total porosity will be recorded. Any pores smaller than 5 microns will be ignored.	microns	Optical microscopy, ImageJ	40%	1	Score (%) = 100*(1-$porosity_{avg}$-0.9)/(1-$porosity_{min}$-0.9)	1 cross-sectioned nut per build plate. Average porosity will be used in scoring. Upper porosity limit is 10%. More than 10% will receive a score of 0.

Ti-6Al-4V Print Evaluation (*80% of Total Score*) Table #4 continued							
Tensile Bars (*30% of Total Score*) ****Note: All 4 tensile bars will be machined into miniature tensile specimens and tested****							
Review Elements	**Description**	**Units**	**Evaluation Tools**	**Score Percentage**	**Score Weighting**	**How Element is Scored**	**Comments**
Ultimate Tensile Strength (UTS)	Miniature tensile specimens will be machined for this test. - surface finish = 32µin	MPa	Instron Tensile Tester	50%	1	Score (%) = $100*(UTS_{avg}/UTS_{max})$	Average UTS of the best 3 out of 4 results will be used in scoring.
Strain at UTS	Miniature tensile specimens will be machined for this test. - surface finish = 32 µin	mm/mm	Instron Tensile Tester	50%	1	Score (%) = $100*(EPS_{avg}/EPS_{max})$	Average EPS from the best 3 out of 4 results will be used in scoring.
Lattice Cones (*20% of Total Score*)							
Review Elements	**Description**	**Units**	**Evaluation Tools**	**Score Percentage**	**Score Weighting**	**How Element is Scored**	**Comments**
Height Lattice Cone 1	Cone height until print failure of first lattice.	mm	Calipers	50%	1	Score (%) = $100*(height/height_{max})$	Height will be measured prior to removal from build plate.
Height Lattice Cone 2	Cone height until print failure of first lattice.	mm	Calipers	50%	1	Score (%) = $100*(height/height_{max})$	Height will be measured prior to removal from build plate.

Score Key:

$avgS_a$ = Average surface roughness of all the surfaces measured on the un-sectioned (whole) nut being analyzed.

$minS_a$ = Minimum average surface roughness from the un-sectioned (whole) nuts across all entrants build plates.

dev_{avg} = Average point cloud deviation on the un-sectioned (whole) and sectioned nuts being analyzed compared to the CAD model.

dev_{min} = Minimum average point cloud deviation from the un-sectioned (whole) and sectioned nuts across all entrants build plates.

$porosity_{avg}$ = Average porosity of the sectioned nut being analyzed.

$porosity_{min}$ = Minimum average porosity from the sectioned nuts across all entrants build plates.

UTS_{avg} = Average ultimate tensile strength from the best 3 tensile bar results.

UTS_{min} = Minimum average ultimate tensile strength from the best 3 tensile bar results across all entrants build plates.

EPS_{avg} = Average strain from the best 3 tensile bar results.

EPS_{min} = Minimum average strain from the best 3 tensile bar results across all entrants build plates.

height = Measured height from the base plate surface to the first lattice print failure.

$height_{max}$ = Total height of the lattice cone.

13.0 Reasons for Disqualification

Below are reasons for disqualification from the OASIS Challenge…

1. Participant is not a member of America Makes by Tuesday, December 1st, 2020.
2. Not all the required files are submitted by the Tuesday, December 1st, 2020 deadline.
3. Submitted code is private and not open source.
4. Submitted code relies on external libraries to perform the key elements of the challenge.
5. Submitted code is not written in one (1) of the four (4) specified code languages.
6. Submitted code cannot be compiled.
7. Submitted code does not create valid XML and SVG files.
8. Submitted code takes longer than six (6) hours to run or crashes during evaluation (with and without the trophy).
9. Created files (XML/SVG) causes the print machine to “crash” (including the trophy)
10. Submitted code is identified as malicious software.
11. Proper license is not associated with the submitted code.