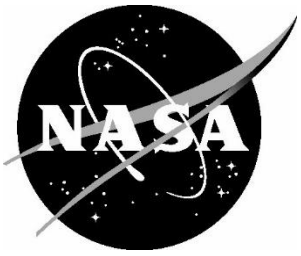
NASA/TM–2020-5003819



Computer Program to Generate Tri-Truss Structures for On- and Off-Axis Telescope Dishes

Jonah D. Vanke
Roanoke Valley Governor's School for Science and Technology, Roanoke, Virginia

Henry T. Holbrook
Roanoke Valley Governor's School for Science and Technology, Roanoke, Virginia

Matthew K. Mahlin
NASA Langley Research Center, Hampton, Virginia

Brace W. White
NASA Langley Research Center, Hampton, Virginia

June 2020

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

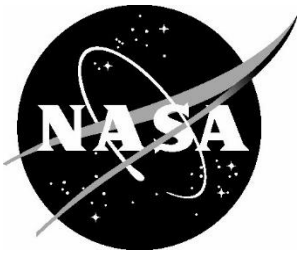
- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM–2020-5003819



Computer Program to Generate Tri-Truss Structures for On- and Off-Axis Telescope Dishes

Jonah D. Vanke
Roanoke Valley Governor's School for Science and Technology, Roanoke, Virginia

Henry T. Holbrook
Roanoke Valley Governor's School for Science and Technology, Roanoke, Virginia

Matthew K. Mahlin
NASA Langley Research Center, Hampton, Virginia

Brace W. White
NASA Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

June 2020

Acknowledgements

The authors are grateful to Monique Sims, Chauntée Pitts, and Ariel Wright for their devoted involvement in the NASA Summer Residential Governor's School Program and to Melissa Fisher and the Roanoke Valley Governor's School for Science and Technology for their commitment to advanced high school math education.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Contents

List of Figures

List of Tables

Nomenclature

ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
CAD	**C**omputer-**a**ided **D**esign
iSA	**i**n-**S**pace **A**ssembly
iSAT	**i**n-**S**pace **A**ssembled **T**elescope
STL	**St**ereo**l**ithography
VTK	The **V**isualization **T**ool**k**it

Abstract

The size of traditional space telescopes has been limited by the size of the launch vehicle shroud size. Designs of space telescopes with larger apertures and greater resolving power can be achieved with multiple launches incorporating modular elements and in-space assembly techniques. The modular elements of an in-space assembled telescope include segmented reflectors supported on an assembled truss structure. The modular truss structures are most commonly based on a tessellation of triangles across a reflector's surface, and in some cases utilize deployable trusses to simplify in-space assembly. One design effort, the in-Space Assembled Telescope (iSAT), proposes the use of deployable "tri-truss" modules to form an off-set parabolic reflector. In support of the iSAT, a computer program was developed to tesselate tri-truss modules over a parabolic surface. Based on several user-defined variables, the program renders a tri-truss structure and generates a stereolithography file of the generated truss structure for three-dimensional (3D) printing. In addition, due to the importance of packing efficiency for space launch, the program attempts to minimize unnecessary differences in strut lengths using a grade parameter based on previous work.

Introduction

The size of traditional space telescopes has been limited by the size of the launch vehicle shroud size. Designs of space telescopes with larger apertures and greater resolving power can be achieved with multiple launches incorporating modular elements and in-space assembly techniques [1]. The modular elements of an in-space assembled telescope include segmented reflectors supported on an assembled truss structure. The modular truss structures are most commonly based on a tessellation of triangles across a reflector's surface, and in some cases utilize deployable trusses to simplify in-space assembly. One design effort, the in-Space Assembled Telescope (iSAT), as shown in Figure 1, proposes the use of deployable "tri-truss" modules to form an offset parabolic reflector [2, 3, 4].

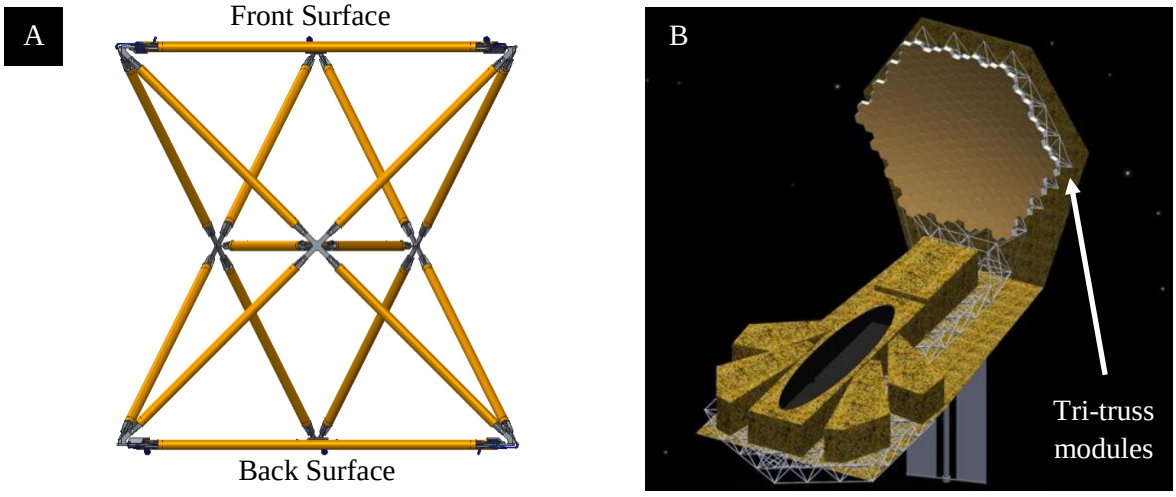


Figure 1. A) A tri-truss module from Ref [2] and B) Conceptual design of the in-Space Assembled Telescope assembled with tri-truss modules from Ref [4]

However, the use of truss structures in the construction of doubly curved surfaces inherently requires struts of various lengths. The tri-truss modules must have unique strut lengths on both their front and back faces, relative to the axes of symmetry in the structure, to achieve the desired surface contours. A wide range of strut lengths can be used to achieve the same surface shape but minimizing the difference in strut length is desirable. Designing strut lengths to be as close to equal as mathematically possible optimizes the packing of tri-trusses within a launch vehicle fairing and simplifies the supporting in-space assembly tools and procedures. Additionally, manufacture of modules can also be facilitated by having each tri-truss module be more dimensionally similar.

In support of the iSAT, a computer program was written to generate a network of surface struts of tri-truss modules on an elliptical paraboloid on- or off-center, and, using this network, to generate a set of tri-trusses laying on the paraboloid's curved surface. Based on several user-defined variables, the program renders a tri-truss structure and generates a stereolithography (STL) file of the generated tri-truss structure. In addition, due to the importance of packing efficiency for space deployment, the program attempts to minimize unnecessary differences in strut lengths using a grade parameter based on previous work by Bush, Herstrom and Stein [5]. This paper describes the functions of the truss tessellation program in detail.

Integration of Prior Work

The approach outlined by Bush, Herstrom, and Stein [5] was intended for tetrahedral trusses, but it does lay out an excellent framework for nodal distribution by arc division. The tetrahedral truss program generated a network of trusses similar to the assembled tri-truss modules as seen in Figure 1 on the surface of the dish in part based on a circumferential strut rotation parameter, K. The approach in this work was used as a guide while making accommodations for off-axis dishes. The method used divided arcs AE and DE shown in Figure 2, into equal length segments, and used those segments as guides when creating the rest of the structure's struts. The way the parameter K affected the length of arc DE was found to be helpful in minimizing the differences in segment lengths.

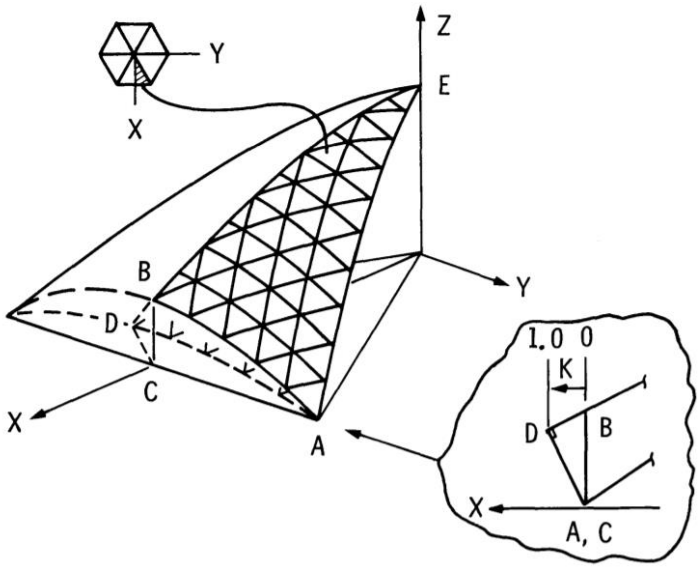


Figure 2. K Parameter Diagram from Ref [5]

The majority of the calculations performed by the code involve dividing the surface of the dish into a network of triangles with as little difference between edge lengths as possible. The first approach taken was to make it possible to, given any three points (two as endpoints and a third to fully define the plane intersecting the paraboloid), subtend the paraboloid along that path into an arbitrary number of equal-length arcs. Optimization functions and line integrals were used to make the function as accurate as possible, and although this function worked in the majority of cases, it was not consistently successful enough to be included in the final program.

Planar sections of a paraboloid are parabolic when the plane is parallel to the z-axis, as shown in Figure 2, and elliptical in all other nontrivial cases. Attempts to subtend arcs along the paraboloid on nearly flat sections of the paraboloid, where the plane being used was very close to but not parallel to the z-axis, led to problems where the use of equations based on a parabola would be unacceptably inaccurate. The elliptical planar section necessary to provide useful accuracy would be so large and stretched that optimization functions using angle increments of a polar function to draw the ellipse would often fail. Additionally, even when working properly the arc division approach focuses on the curved arc lengths rather than the lengths of the lines across the arcs. Focusing on arc lengths means that any significant curvature will lead to unnecessary differences between final line lengths.

The Truss Tessellation Program

The methods and functions of the Truss Tessellation Program are described in this section. First, how the program divides the paraboloid into a truss network by applying several functions: tessellation, grid generation, and a triangle generation. Second, the procedures to populate the divided paraboloid with tri-trusses in the node centered “Type A” structure and the tri-truss centered “Type B” structure, to be defined later in this report, is described. Then, how the tri-truss geometry is generated, and how the ideal grade analysis optimizes the truss lengths are presented. Finally, a description of the generation of the STL file output compatible with Computer-Aided Design (CAD) software is provided.

A list of user input variables is available in Table 1 with descriptions of each variable. These variables allow the user to generate a variety of parabolic truss structures. In particular, the variables allow the paraboloid to be redefined with different focal length to depth ratios, numbers of tri-trusses across the edge of the dish, and depths of the back-lattice structure. Under varying circumstances, the desired truss structure may either have a nearly identical lattice of struts between the front (touching the dish) and back sides of the truss lattice or have an expanded lattice on the back side. Objectives such as optimally packing trusses in a fairing or creating the stiffest possible structure will affect the extent of back-lattice expansion. The program allows the user to specify the back-lattice expansion by having tri-trusses expand from front to back along a series of lines cast by a point a given distance away from the dish, explained further in the tri-truss generation section.

Table 1. Table of Adjustable Variables

Adjustable Variables	Values	Description
typeB	Boolean	Generate either a type A or a type B structure
casterPoint	Float	How far away from the center of the dish (along its normal axis) lines are "cast" through the surface layer of trusses to create the rest of the tritrusses
triangles	Integer	Number of triangles that lie on any given edge of the dish
offset	Float	Offset of the center of the dish from the origin
grade	Float	Slope of the lines cast to find surface points on each sixth of the parabola (optimizable, current iSat uses 0.0358625)
focalLength	Float	Focal length
diam	Float	Diameter
axisLength	Float	Length of the axes
genSTL	Boolean	Either generate an STL file of the structure, or visualize it via VTK
filename	String	Name of generated STL
resolution	Integer	Resolution (in number of fractions of the circle to use) for generation of spheres and cylinders regardless of type of rendering
radius	Float	Radius relative to total length of unit member (recommended 0.05)
fastLines	Boolean	If visualizing data, visualize tritruss members as lines instead of as cylinders (speeds up generation, but harder to make out individual members)
outputWhileWriting	Boolean	Give output when making an STL
stlScale	Float	Scale factor for making the STL. File will be 2 millimeters in diameter times whatever value is put here. Recommended factor of 100 for a 20cm width.
printRatios	Boolean	Whether or not to print the ratio between the longest and shortest struts
castDistance	Float	Distance to cast down for each tritruss

Tessellation Function

The purpose of the tessellation function is to divide the paraboloid surface into equal line segments. The function was written using many of the same ideas as the arc division function described in the first approach [5] but focusing on line lengths. Given two points and a vector locally normal to the paraboloid surface, the function divides an arc between those two points into a given number of equal-length lines. The arc lies on a plane defined by the two given points and parallel to the given normal vector.

Division of the arc into equal line segments begins by first generating an unequal series of line segments on the parabola based on equal line segments projected from a plane tangent to the center of the paraboloid, referred to as the base, dividing a line between the two points into equal segments, and then travelling along its given normal vector until it intersects the paraboloid. The initial division of equal line segments on the base, producing unequal line segments on the paraboloid, is represented in Figure 3. The program then takes these unequal parabola line segments and shortens or extends the base line segments used. Modification of the base line segments is iterated until the parabola line segment lengths converge to equal length segments within a very small tolerance, as shown in Figure 4.

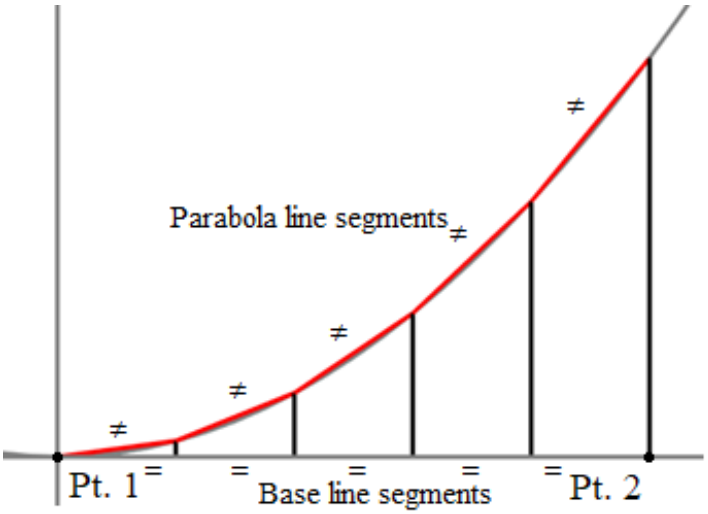


Figure 3. Initial unequal line segments produced from equal base line projections

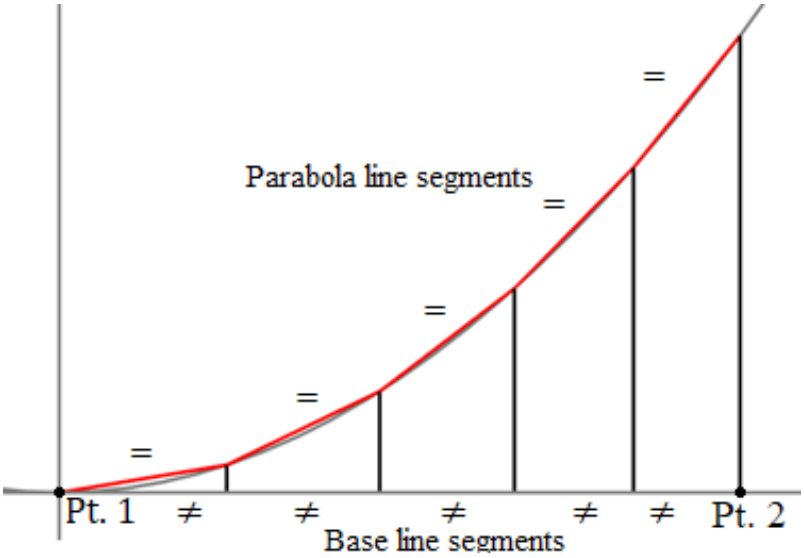


Figure 4. Final equal line segments produced from unequal base line projections

Grid Generation

After the tessellation function was created, the tesselation function was repeated six times from the center of the dish using linear transformations and coordinate transforms to generate a "star" of six sets of line segments, corresponding to the number of trusses meeting at a node, radiating outward from the center of the dish, as shown in Figure 5. The center of the dish was calculated using the user-defined dish offset, and the star was oriented to align a flat edge of the resulting hexagon with the origin defined in the program (the bottom of the paraboloid). The directionality seen in Figure 5, as opposed to generating a line coincident with the origin, was chosen because the iSAT's design involves mounting a flat edge of the dish to the body of the telescope.

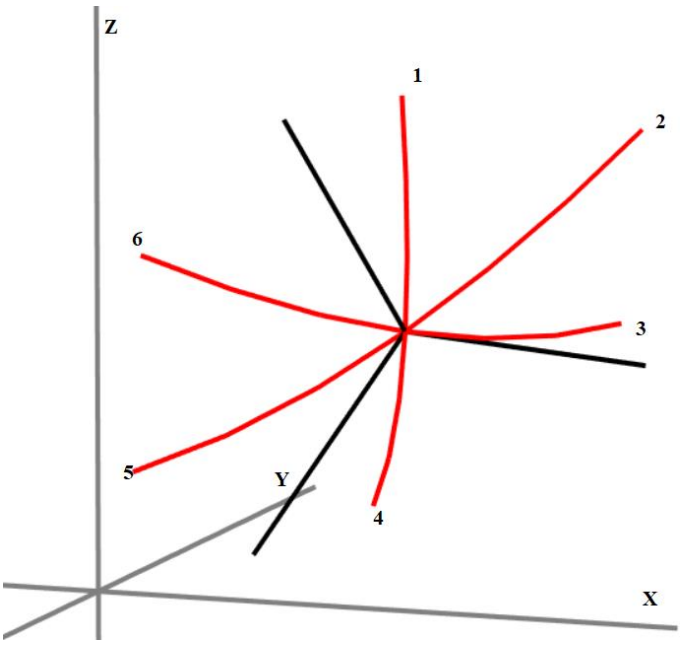


Figure 5. Transformed axes (black) and first generated "star" of lines (red)

An array was created containing six elements, each element itself containing each point along one branch of the star. The same tessellation function was used to connect these points in rings as many times as necessary for the user-specified triangle count. A three ring corresponds to the three edge-triangle count as shown in Figure 6.

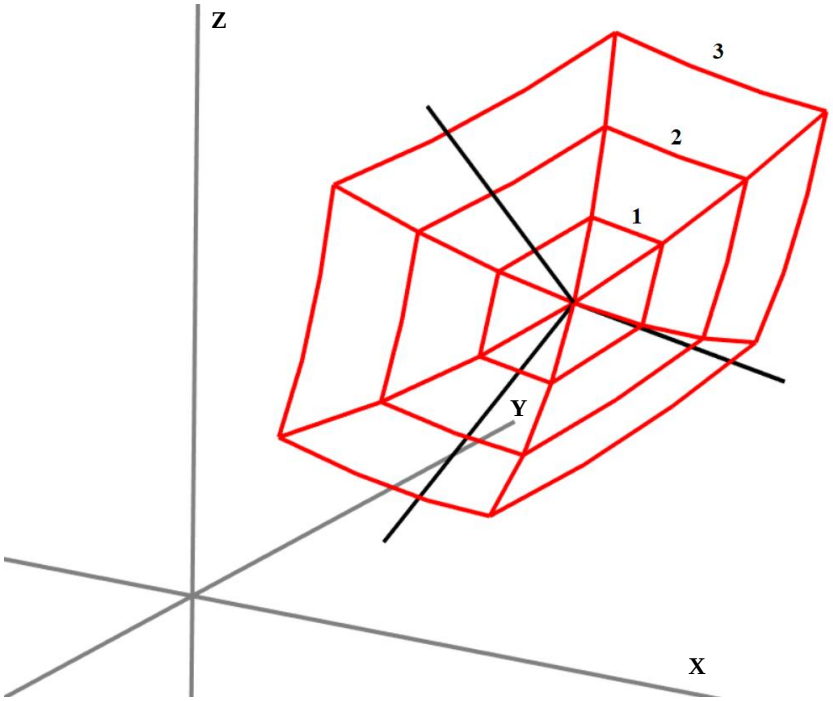


Figure 6. All generated lines with an edge triangle count of three

To emulate the "K parameter" put forward by Bush, Herstrom, and Stein [5], a "grade parameter" was created in the program to modify the tesselation function when used on ring generation. The grade

parameter, usually between 0 and 0.1, changes the normal vector, used by the arc division function, to be slightly tilted toward the center of the dish, subdividing a steeply inclined ellipse instead of a vertical parabola. The grade parameter is the absolute value of the change in the slope of the normal vector, and the sign is positive or negative depending on the relative position of the normal vector to the central axis of the structure. The change in the normal vector with increasing grade and how this affects the tesselation function can be seen in Figure 7A where arrows indicate the expansion of the sub-divided parabola. Tilting the normal vectors toward the center with the grade parameter has been observed to decrease the maximum difference between strut lengths significantly. The effect of the grade parameter correlates with the conventional use of the word "grade" to describe gradients; for each unit of vertical increase of one in the given normal vector, it represents a corresponding horizontal increase. The difference between a zero grade and a grade of one for the three-ring grid generation is shown in Figure 7B.

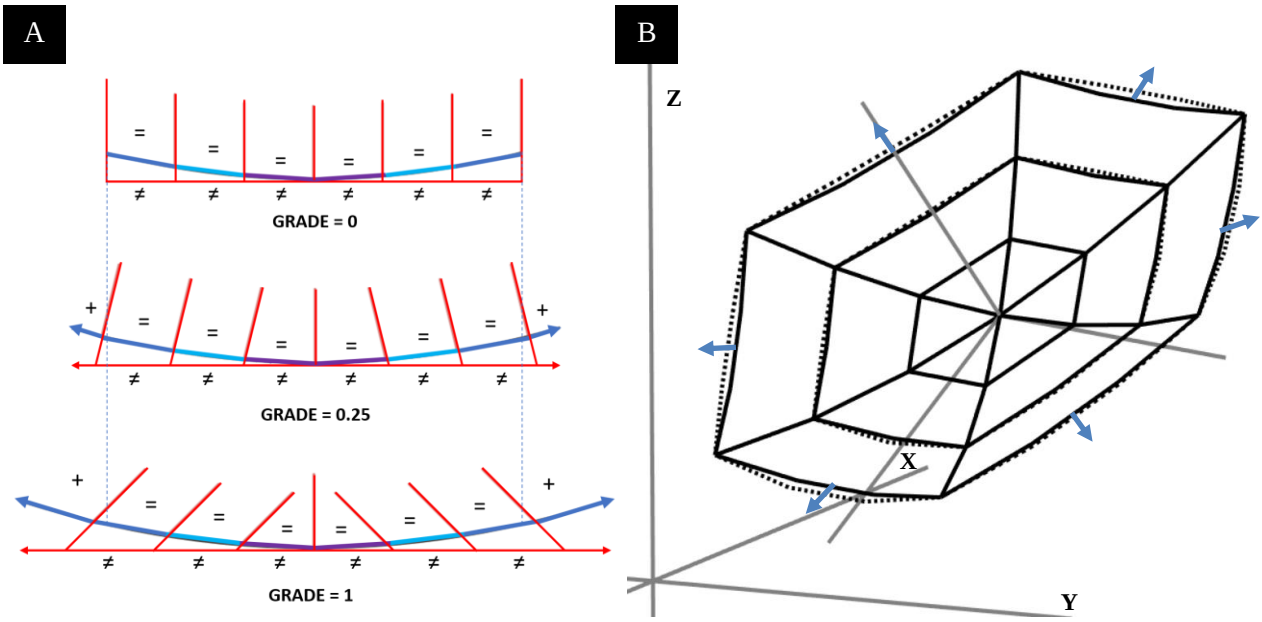


Figure 7. A) Schematic of how setting the "grade" to 0, 0.25, and 1 affects the normal vector in the tesselation function and widens the segmented parabola, B) the same base structure with "grade" set to 0 (solid lines) and 1 (dotted lines)

Triangle Generation

As each ring of lines as seen in the Grid Generation section was calculated with a given grade value, the X-, Y- and Z-values of each line endpoint were appended to an array. The coordinates of the transformed origin (the center of the dish) were also appended to the array. The result was an array containing each potential vertex of a triangle, or the nodes of a tri-truss, in the structure. The array containing the nodal points is represented in Figure 8.

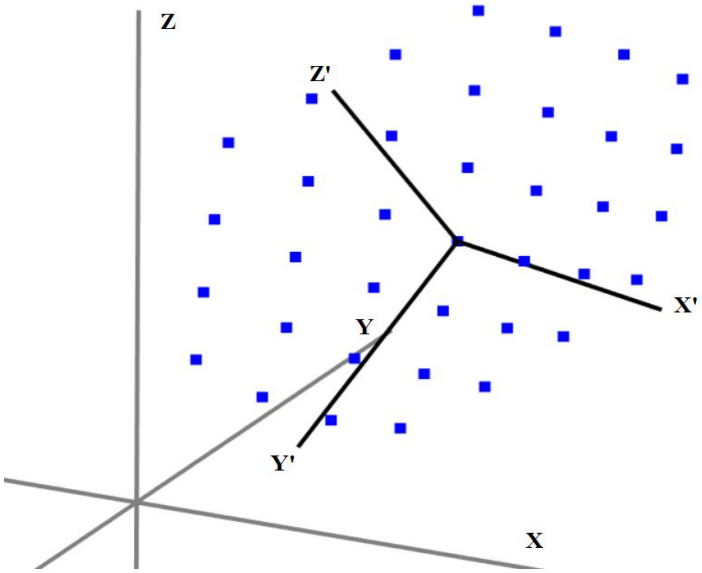


Figure 8. Array of nodal points

Two key functions are used to go from the unordered array of points to a functional list of sets of three points. The first function, given a point, finds the three closest points of all available points and appends these three points to an array. The function also returns the coordinates of a point at the average location of these same three points, or the center of the triangle. The second function follows the center point of each triangle to "walk" along the hexagonal grid of points, snapping triangle center points into place by changing point coordinates from its prediction to the actual average values of three nodal points as it traverses the curvature of the hexagonal grid of nodal points. The approach is less likely to cause misalignments or errors than the arc division method. Beginning by "walking" from the top right to bottom left of the array of nodes, from the viewpoint of Figure 9, the function then proceeds from each of these points downward and outward until every possible triangle has been reached.

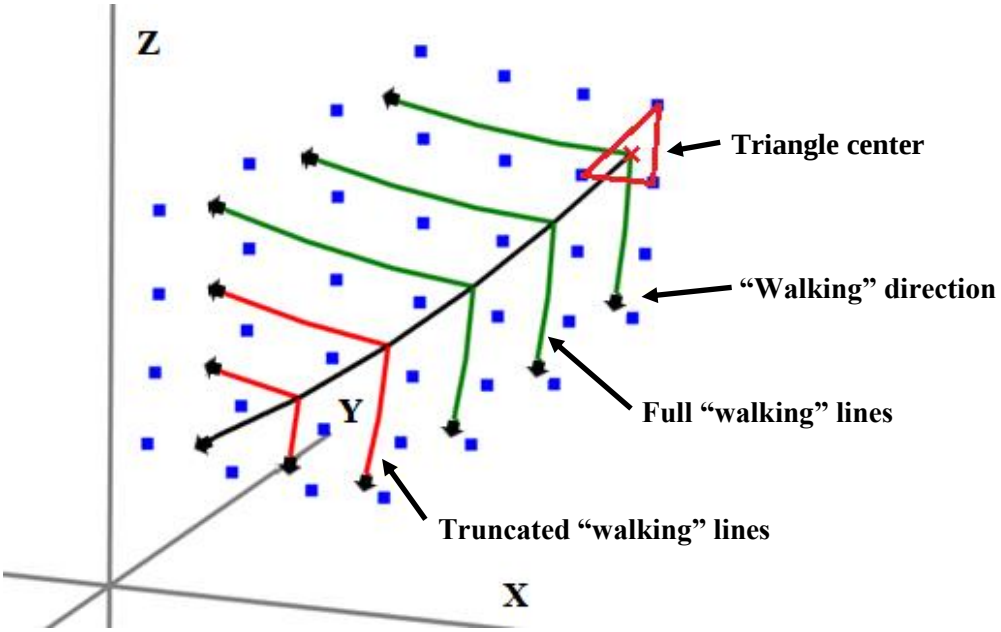


Figure 9. "Walking" along positions in the hexagonal grid of points

Type A and B Structures

The program is capable of generating both type "A" and "B" truss structures, as seen in Figure 10. The Type A structure, centered on a truss node, is the default option for the program to generate. The fact that the long lines defining their structure for Type A structures pass directly through the center makes generating them significantly easier than Type B structures. In the case of Type B structures, the same operation used to connect existing points of the dish into sets of triangles is used as its own output, using the midpoints it found during generation, as seen as the endpoints of line segments in Figure 9, as new nodes.

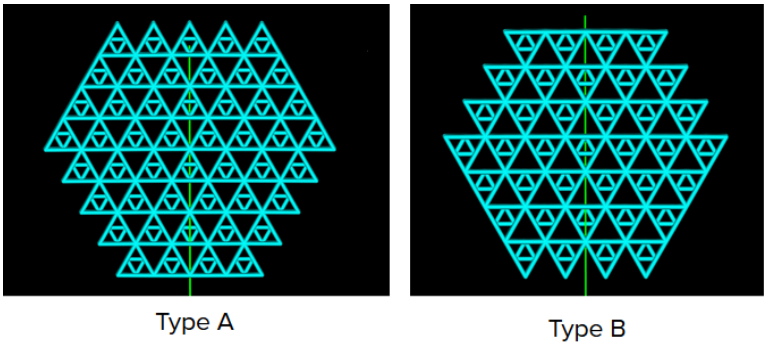


Figure 10. Type A and B structures

In the Type B structure approach, the midpoints of the Type A structure are used to create an additional set of triangles. An advantage of this approach is that by using the same triangle generation function, no additional functions are required to form the sets of three points needed for tri-truss generation. Some disadvantages include the fact that dish diameter is reduced by a nontrivial amount and requires compensation by the program, and that these midpoints are very slightly offset from the paraboloid since they lie on a secant plane rather than on the curve itself. The offset problem is fixed by the program by using the point's X- and Y- coordinates to calculate the direction the point should move to reach the paraboloid in the shortest distance, and by moving the point in that direction. The projection distance of the Type B structure nodal points back on to the paraboloid is small, but the result is more accurate.

Tri-Truss Generation

With division of the paraboloid complete, the nodal and center points are then used to populate the structure with tri-trusses. A function was written to generate a tri-truss given three points as described in the triangle generation section, a height for the tri-truss, and a fourth, user-defined, "caster" point. For the tri-truss height, a ratio of $\frac{\sqrt{3}}{3}$ between truss height and edge length is most often used, as this produces a regular hexagon when the tri-truss is flattened. The caster point provides the source of three lines passing through each of the three nodal points, as shown in Figure 11. The three back points of the tri-truss were made to lie on these lines as far back as necessary for the desired height of the tri-truss. This tri-truss function then used the three front points and three back points it had generated to draw all fifteen struts in a tri-truss, as seen in Figure 11.

The visual library used, the visualization toolkit (VTK) [6], includes functions to generate cylinders and spheres. To aid in visualization, a feature was added to the program where instead of rendering tri-

trusses as networks of fifteen lines, it instead renders them as networks of fifteen cylinders. It also places spheres at one end of each uppermost and lowermost cylinder to prevent unsightly and unrealistic gaps where cylinders meet, rendering a rounded corner at the six top and bottom vertices.

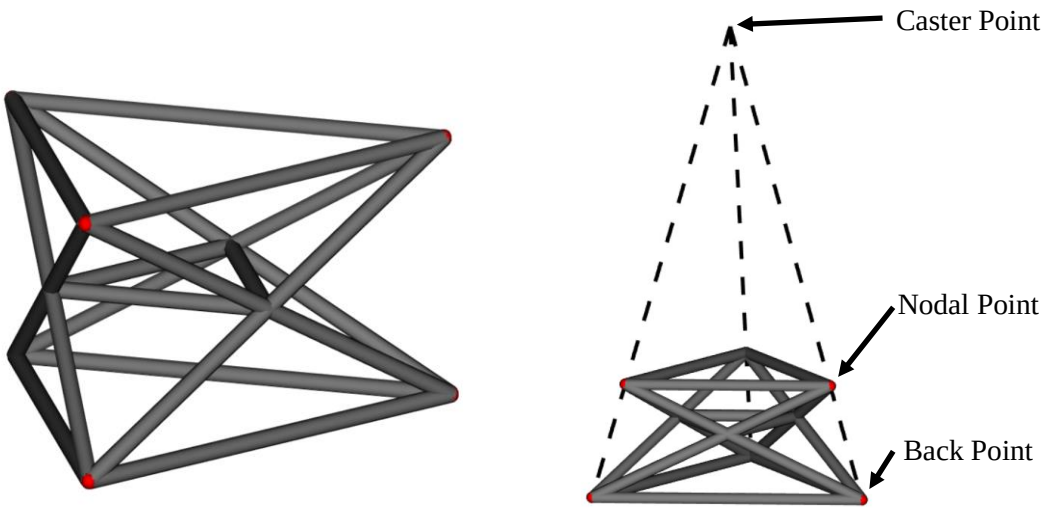


Figure 11. A single tri-truss generated by the program, and dotted lines indicating the generation method

After being written, the tri-truss function was manipulated to generate tri-trusses for every given center point on the dish. By using the same height and same casting point throughout structure generation, the back surface of each tri-truss naturally lined up with each other without gaps or overlaps. The specific casting point used was generated by moving a user-defined distance away from the center of the dish along the vector normal to the dish passing through that center point. The tri-truss generation function was repeated until an entire tri-truss structure was generated, as shown in Figure 12.

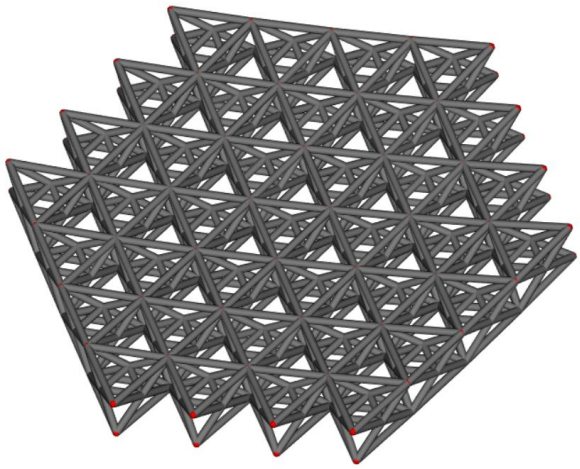


Figure 12. A fully generated tri-truss structure for the iSAT

Ideal Grade Analysis

Determination of the ideal grade parameter value for generating the truss structure was performed with a function. The ideal grade is used to minimize the strut length difference in the truss tessellation program. The ideal grade analysis function searches for the ideal value down to step sizes of 2^{-32} units. The assumption that the ideal value lies between 0 and 1 is used to narrow down the range. The ideal grade analysis function takes about two minutes to run on an average computer and is not meant to be executed for every use of the program, but only when determining the ideal parameter for a new telescope design. For reference, the ideal value for the iSAT is around 0.036.

The output of the ideal grade analysis between 0 and 0.1 is shown in Figure 13 with grade plotted against the strut length ratio for a 20-meter offset parabolic reflector with a focal length to diameter ratio of two. The strut length ratio is calculated as the longest strut length divided by the shortest strut length on the surface of the dish. Note that the strut length ratio is different from the normalized strut length difference ratio used by Bush, Herstrom, and Stein [5] and should not be directly compared. Minimizing the strut length ratio results in strut lengths that are as equal as possible. One notable feature of the trend is discontinuities in slope. Through the testing of various display methods, these abrupt linear changes were demonstrated not to be flaws in the plots generated by the plotly Python library used [7]. Although it is possible that they are the result of errors in the calculations the program performs, this is unlikely since each data point was calculated entirely independently from each other data point.

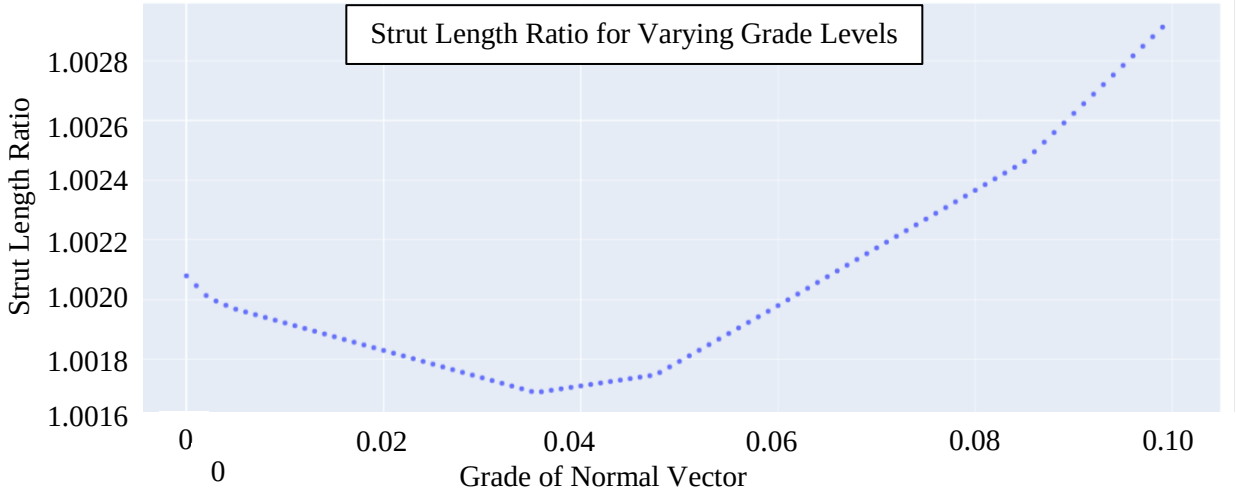


Figure 13. Graph of Maximum ratio between struts for different grade values for the iSAT

The discontinuities in slope of the strut length ratios are not fully understood. However, the equivalent graph produced by Bush, Herstrom, and Stein [5] for the K parameter provided in Figure 14 shows similar behavior. A notable difference is the trend follows a smooth curve after the minimum point rather than a series of straight lines. One possible explanation of the difference in results is that the asymmetry of an off-axis dish leads to different struts being the longest or shortest at different times in a manner more complex than with an on-axis dish, but this explanation has not been fully explored. Another possible explanation is that similar linear trends arose in the K parameter, but the size and resolution of the graph available from the reference make it difficult to discern whether it represents a similar series of straight lines or a smooth curve.

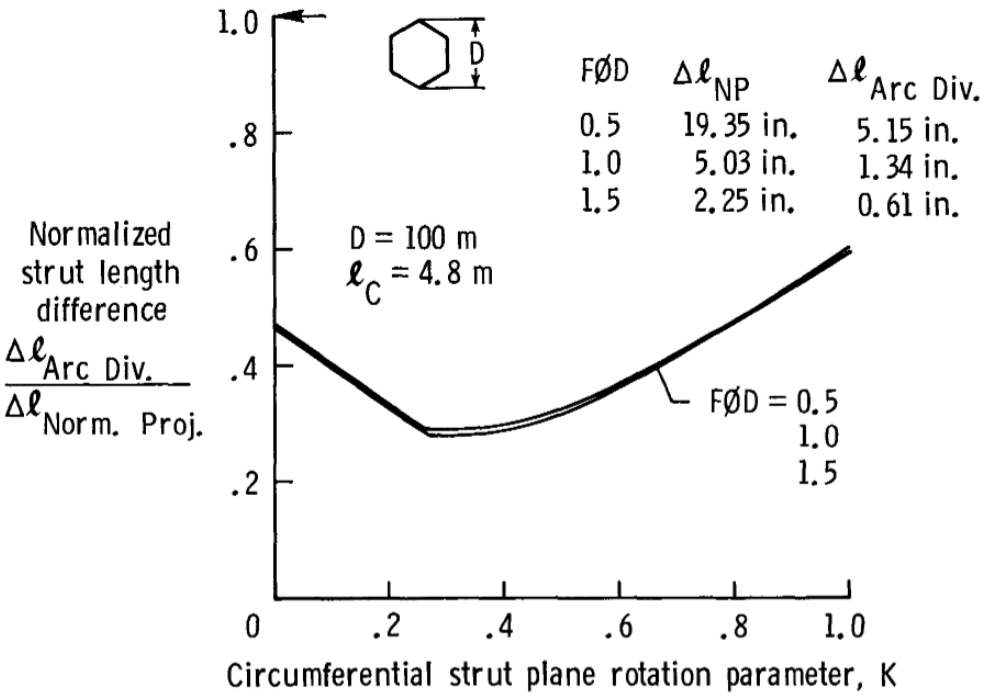


Figure 14. Optimization graph for the K parameter from Ref [5] for on-axis dishes

The ultimate result of applying the ideal grade in the truss tessellation program is a minimized strut length difference. Previous work achieved a maximum strut length difference of 2.25 in. for a 100-meter diameter tetrahedral truss structure with 4.8-meter core trusses. However, the program presented here can produce a strut length difference of 0.89 in. for the same parameters. The approach used in this program provides a significant improvement over the arc division method previously used.

STL Generation

Triangular mesh files, most often stored as STL files, are commonly used in CAD software and 3D printing, making the generation of one necessary for many practical uses of the program. Once an STL of a truss structure is generated by the program the STL can be imported into a CAD file and be more easily incorporated into a large project or model. With points in space calculated for a tri-truss structure, the next step is to generate an STL file of the structure. However, the generation of this STL proved more difficult than anticipated. The STL export function of the library being used for visualization, VTK, could not easily accommodate transformations of objects, instead generating hundreds of copies of the same unit cylinder, making it unusable for the purposes of this program. Other python STL libraries were found to be similarly unwieldy and difficult to implement. For this reason, a function was written to directly write an STL file.

STL files come in two formats, American Standard Code for Information Interchange (ASCII) and binary. An STL file written in the ASCII format takes up a significant amount of storage space when dealing with large and complicated structures, but has the advantage of being human-readable, making it relatively easy to troubleshoot errors. Binary STL files, although not human-readable, are much smaller in size. This aspect of binary STL files make them the preferred choice for the end version of this program, because as

truss count or resolution increases significantly, ASCII file sizes can become very large and inconvenient. Exemplifying this distinction is Figure 15, which displays the differences between the two STL file formats.

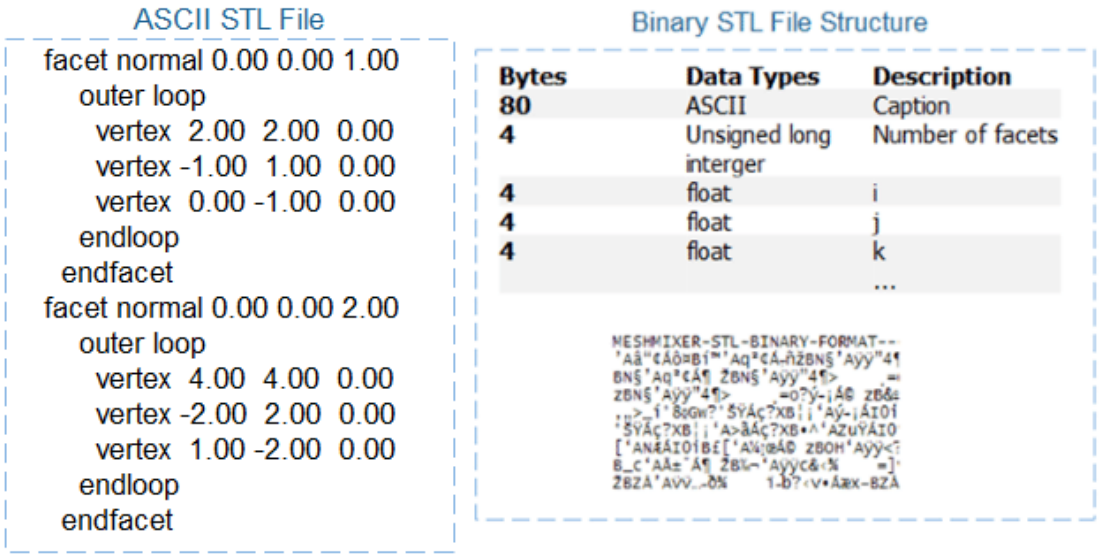


Figure 15. ASCII vs Binary from Ref [8]

A function was constructed that takes facet, a single triangle of the STL, and appends that facet to an STL file. The function receives a facet's values for its normal vector and vertices, and then records that facet to the file using Python's normal ability to write to files.

Concluding Remarks

A truss tessellation program was created that successfully generates approximate models of Type A and Type B tri-truss structures for on- and off-axis telescope dishes. Although prior work addressed minimizing the differences in strut lengths, the specific values for many test cases were not published. The current truss tessellation program further reduces the minimum strut length difference achieved in the prior work. The program's use of line intersections with a paraboloid and coordinate transforms, makes geometry generation possible for both on- and off-axis parabolic truss structures. The on- or off-axis generation capability extends the potential applications to virtually any conventional telescope dish with trusses. Additional functions created to generate tri-trusses and STL files facilitate quick modeling and 3D printing of space telescopes.

Future improvements to the program may include additional math to ensure the diameter of a Type B truss is precisely as defined by the user, as opposed to the approximation currently used, and to consistently identify and handle errors in which the user inputs a dish diameter too large to fit on the paraboloid. Additionally, when desired, tri-trusses could be generated slightly smaller and offset from the generated parabolic surface triangles to allow for the thickness of segmented reflectors.

References

1. Doggett, W.: “Robotic Assembly of Truss Structures for Space Systems and Future Research Plans” Proc. 2002 IEEE Aerospace Conf., vol. 7, pp. 3589-3598.
2. Doggett, W; Dorsey, J.; Teter, J.; Paddock, D.; Jones, T.; Komendera, E.: “Persistent Assets in Zero-G and on Planetary Surfaces: Enabled by Modular Technology and Robotic Operations”, Presented at the AIAA Space 2018 Forum, 17 – 19 September 2018, Orlando Florida. Available as AIAA-2018-5305.
3. Siegler, N.; and Mukherjee, R.: “In-Space Assembled Telescope (iSAT) Study”, California Institute of Technology, September 13, 2018.
4. Mukherjee, R.; Siegler, N.; Thronson, H.; et.al.: “When is it Worth Assembly Observatories in Space?”, California Institute of Technology, Astro2020 APC Whitepaper, 2019.
5. Bush, H. G.; Herstrom C. L.; Stein P. A.; Johnson R. R.: “Synchronously Deployable Tetrahedral Truss Reflector”, NASA CP-2368, Part 2, Large Space Antenna Systems Technology Conference, Hampton, VA, December 4-6, 1984.
6. Schroeder, W.; Martin, K.; Lorensen, B.: The Visualization Toolkit (4th ed.), Kitware, ISBN 978-1-930934-19-1, 2006.
7. Sievert, C.: Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC. ISBN 9781138331457, https://plotly-r.com, 2020.
8. Süzen, A.; Yıldız, Z.; Kayaalp, K.; and Ceylan, O.: “Customized Hallux Valgus Splint Design For 3d Printer”, July 2018.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
1-07-2020	Technical Memorandum	Jun 2019 - Jun 2020

4. TITLE AND SUBTITLE

A Computer Program to Generate Tri-Truss Structures for On- and Off-Axis Telescope Dishes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Vanke, Jonah, D.
Holbrook, Henry, T.
Mahlin, Matthew, K.
White, Brace, W.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA-TM-2020-5003819

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified
Subject Category
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The modular elements of an in-space assembled telescope include segmented reflectors supported on an assembled truss structure. The modular truss structures are most commonly based on a tessellation of triangles across a reflector's surface, and in some cases utilize deployable trusses to simplify in-space assembly. In support of the in-Space Assembled Telescope (iSAT), a computer program was developed to tesselate "tri-truss" modules over a parabolic surface. Based on several user-defined variables, the program renders a tri-truss structure and generates a stereolithography file. Due to the importance of packing efficiency for space launch, the program attempts to minimize differences in strut length.

15. SUBJECT TERMS

Structures, Parabolic, Reflector, Telescope, Dishes, in-Space Assembly, OSAM, Off-Axis, On-Axis, Truss, Tri-Truss, In-Space Assembled Telescope,

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	22	19b. TELEPHONE NUMBER *(Include area code)* (757) 864-9658