Solution - Session 2 Report

Anirudh Agarwal

October 20, 2023

Abstract

First things first, the files to be uploaded to the google drive for submission are:

1. Overleaf source file in a separate folder
2. PDF report generated with overleaf
3. Mendeley references pdfs/ docs/ datasheets/ other in a separate references folder
4. Python codes, inputs, and outputs, in a separate folder

1 Objectives for this Session

The objectives of this session are:

1. Enlist typical satellite sizes for CubeSats
2. Find mechanisms for folding solar panels used on satellites
3. Find coatings used for radiation protection in space
4. Put pictures and copy paste brief explanations of the mechanisms in the report
5. Add references to mechanisms, coatings, and solar panels searched in the bibliography
6. Write code for sizing folded solar panels for given equations
7. Write code for sizing outer cover solar panels for given equations

2 Search

2.1 Typical CubeSat Sizes

Satellites come in a large range of masses as mentioned on the wiki page for satellites:

1. Femto satellite <0.1kg
2. Pico satellite 0.1 to 1 kg
3. Nano satellite 1.1 to 10 kg
4. Micro satellite 11 to 200 kg (our range)
5. Mini satellite 201 to 600 kg
6. Small satellite 601 to 1200 kg
7. Medium satellite 1201 to 2500 kg
8. Intermediate satellite 2501 to 4200 kg
9. Large satellite 4201 to 5000 kg
10. Heavy satellite 5001 to 7000 kg

11. Extra Heavy satellite > 7000 kg

We focus on the CubeSats which have masses in the Nano to Micro range. CubeSats are a special class of satellites which are made in multiples of the volume 10x10x10cm or 1U. This helps to standardize their design, size, and mass limits to streamline the entire process from design to launch. These come in specific sizes as given in the CubeSat size guide[3] for sizes below 3U and the EXO Launch manual [1] for sizes 3U to 16U.

Here are the dimensions:

Size (in U)	Dimensions (L x W x H) (mm)
1U	113.5 x 100 x 100
1.5U	170.2 x 100 x 100
2U	227 x 100 x 100
3U	340.5 x 100 x 100
6U	340.5 x 226.3 x 100
6U XL	365.9 x 226.3 x 100
8U	454 x 226.3 x 100
12U	340.5 x 226.3 x 213.5
16U	454 x 226.3 x 213.5

Table 1: Size Options for CubeSats up to 16U

There are more sizes possible as CubeSats go up to 27U but for now, Galamad is only concerned with sizes up to 16U since the launch provider Exolaunch does not give specifications for sizes above 16U. However, if you can find more sizes up to 27U, please populate the table.

2.2 Folding mechanisms for solar panels

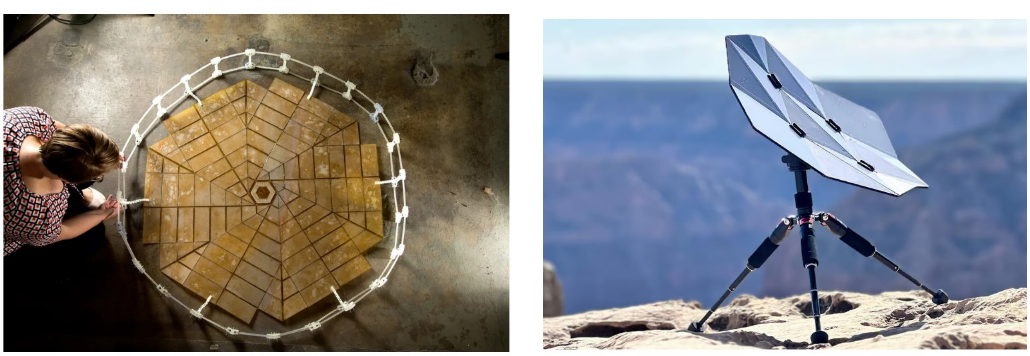
There are 3 main types:

1. Origami folding
2. Conventional hinged folding inside
3. Hinged folding outside

It is also possible to not fold at all and just use the outside surface if the power requirement is extremely low.

2.2.1 Origami folding

Origami folding for solar panels has been used in space for a while now. This space.com article shows the most commonly used circular flower design for origami panels. The company, Sego Innovations,



Shannon Zirbel, a graduate student at Brigham Young University who helped work on the project, unfolds one of the solar panel arrays. (Image credit: BYU)

Sego Innovations is looking to launch a solar panel that instantly compacts to 1/8 its expanded size Sego Innovations

Figure 1: Origami folding of panels. (a) Usage in the space industry. (b) Commercial use folding

is making origami folded solar panels for use on the ground for camping purposes. The figure shows both configurations.

2.2.2 Folding inside

We search "retractable or extendable solar arrays" and find some interesting results for various folded configurations. The NASA Website has an article on folded panels that shows how intensively the solar cells for the ISS were packed and stored in a small volume. When deployed, these cells are quite long.

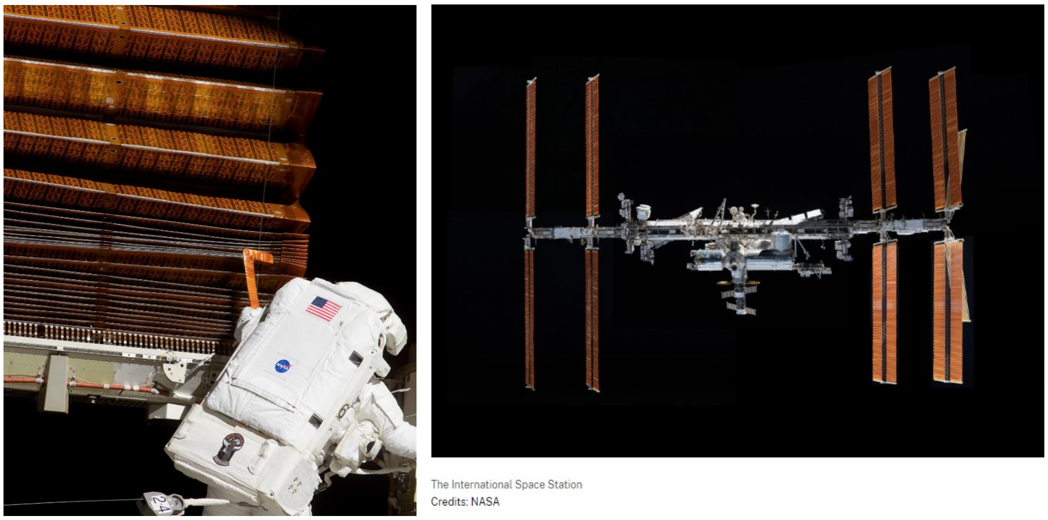


Figure 2: Regular hinge folding done by NASA for the ISS. (a) Packed state (b) Opened state

There are some commercial ventures making containerized folded solar panels for large scale usage. We can probably learn something from looking at those designs. This article shows how entire shipping containers can be filled with solar cells. This could certainly come in handy with medium (over 1200kg) satellites and bigger.

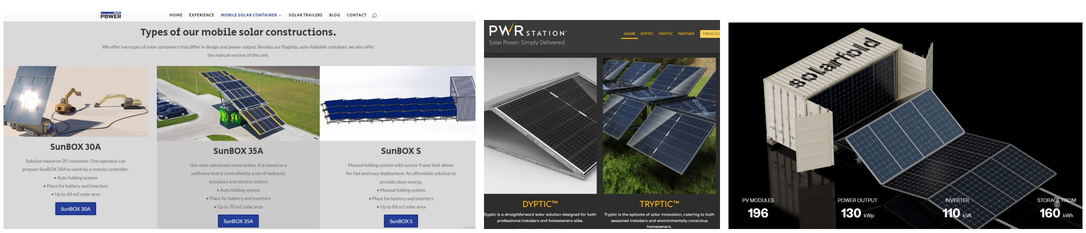


Figure 3: Commercial hinge folded panels (a) Powermoveit (b) PWRstation (c) Solar Fold

Some companies doing folded panels are:

1. Solar Fold
2. Power Moveit
3. PWR Station

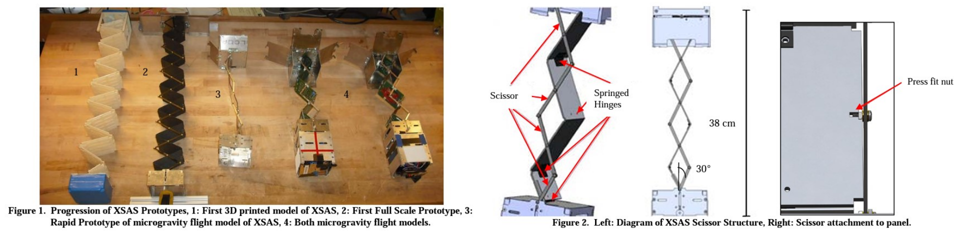


Figure 4: Research paper on scissor based folding mechanism

These can give us an idea on how to design the folding mechanism to store on the inside. However, to get more specific information, we must seek research papers or detailed drawing. This paper [2] by the University of Michigan Ann Arbor presented in the Astrodynamics Specialist Conference, goes into the details of desiging a structure like this.

2.2.3 Folding outside

There is another structure of unfolded solar panels by NASA mentioned in this article that shows how the panels are arranged on the walls. That is what we are looking for in the "folding outside" category. This is the NASA Psyche mission as mentioned in the magazine article. So we go to the NASA website and find the actual mission Psyche and use that.

There isn't really a commercial equivalent of this outside of satellites. The company Sparkwing makes panels folded on the outer wall that can be similar to what we intend to do with the outer cover mechanism.

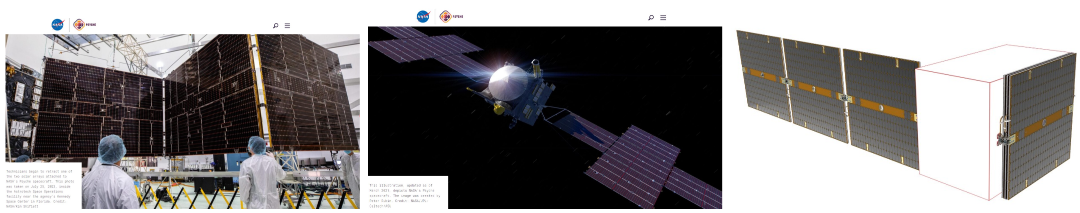


Figure 5: Outer folded mechanisms. (a) NASA Psyche mission manufacturing phase (b) NASA Psyche mission deployment rendering (c) Sparkwing commercial outer cover design

2.3 Radiation Protection Coatings

For Radiation protection, first we need to figure out what to protect against. There is too much information about all different kinds of radiation in space. We start our search from this link that details the effects of different kinds of radiation. We need to mainly protect against UV and shorter wavelengths and protons. The rest, we can ignore for now.

Raidation protection is the hardest thing with the most uncertainty for Galamad as well. We can limit our search to materials listed in this database: https://www.spacematdb.com/spacemat/m-treeview.php.

To narrow down further, we are looking for the following properties in our material:

1. Transparent to visible and infrared light. Specifically, transmissvity > 0.9 for wavelengths (300nm - 2000 nm).
2. High emissivity > 0.8 to allow for radiative cooling.
3. The material should be available in a thickness of 150 to 200 micron.
4. Stability over a range of temperatures from -60 to +120 Celcius.

We need the same properties for the adhesive to be used to stick this coating on to the solar panels. Do explore more in the coming weeks and find candidate materials. More pointers for the materials will be given in the next solution. This search will go on till the end alongside new searches. Keep exploring radiation protection materials!

3 Code and Algorithms

The code is shared in separate files in the dedicated folders along with the text files for inputs, intermediate values, and the output values.

The code is divided into separate dedicated files (supporting files) for functions that are separate from the main executable file. The 3 types of supporting files are:

1. Reader: reads the values from input/ intermediate files.
2. Writer: writes on to the intermediate and output files.
3. Calculator: Calculates specific functions that are long and cumbersome.

Other things we need to explain:

1. What the variables mean
2. What each function does
3. What are the inputs and outputs
4. A brief explanation of how the code works

3.1 Variable names

The variable names used are the same as the problem statement with the exception of nxy[] and choice. The choice variable was implied but not explicitly stated in the problem statement. It is used to determine what orientation the solar cells are arranged in so we know whether to multiply nxy[i] with lcell or wcell.

3.2 Function descriptions

3.2.1 Common functions

These are the common functions used by both types of code and stored in the same file Reader.py:

1. Read_cell_inp(): reads the inputs from the cell characteristics input file and returns the values.
2. read_frame_inp(): reads inputs from frame dimensions input file and returns the values.
3. read_body_inp(): reads inputs from body size input file and returns the values.

Having such a file is handy as it could be used for both codes without any changes making the programming process more modular and efficient.

3.2.2 Outer Cover Panel functions that could be common

Functions that could be common but aren't in this case: writer.py:

Unlike the folded panels code, for the outer cover code, some of the variables are arrays, and so require loops to write. Thus, writing here is best handled in a separate dedicated function instead of the main code to keep the main code consise and easier to understand. These functions were not as necessary in the folded panels code as the variables were simple.

1. write_int(): writes intermediate values in the corresponding text file.
2. write_out(): writes final values in the output text file.

3.2.3 Functions specific to outer cover panels

calculation_funcs.py

1. get_ncp12(): Calculates nxy[][] and ncellspanel[][]. Note that the variable choice is just ncellspanel[i][1]. This function is only for the first 2 values of the variables calculated since the equations and variables needed are different for the 3rd value in the array (the front face WxH).
2. get_ncp3(): Calculates nxy[2][] and ncellspanel[2][] for the WxH face.
3. getframelw12(): calculates lframe[] and wframe[] 0 and 1.
4. getframelw3(): calculates lframe[2] and wframe[2].
5. getlwopen(): calculates lenopen[] and widopen[].
6. A brief explanation of how the code works

3.3 Inputs and Outputs

3.3.1 Inputs

All the inputs are common. We just don't use the irrelevant inputs. We have 3 types of input files:

---

cell char input.txt:
92 : (lcell) Length of cell in mm
45 : (wcell) Width of cell in mm
1.2 : (powcell) Power of cell in Watts

---

frame dimz input.txt:
3 : (rho) Density of panel material in mg/ mm^3
3 : (tf) Frame thickness in mm
4 : (clf) Clearance for cell frame in mm
6.4 : (clmside) Clearance for mechanism on either side in mm
0.2 : (clhi) Clearance for hinge on inner side in mm
0.3 : (clho) Clearance for hinge on outer side in mm
12 : (clmcen) Clearance for central actuation mechanism in mm
1 : (clcell) Clearance between adjacent cells in mm

Note that rho is put here instead of the cell characteristics as it is a property of the frame and not the cells. The original problem statement didn't do this correctly. Please refer to this solution instead.

Body size inputs.txt:
450 : (L) Length of craft in mm
240 : (W) Width of craft in mm
200 : (H) Height of craft in mm

3.3.2 Outputs

++
Folded panels:

---

intermediate vals.txt:
5.6 : (dhinge) Diameter of the hinge
59.0 : (h2hdist) Hinge to hinge distance
65.2 : (lframe) Length of frame
100.0 : (wframe) Width of frame
37 : (nstack) Number of panels per stack
4 : (numassmbs) Number of stacks / assemblies

---

output vals.txt
296 : (ncells) Total number of cells
355.2 : (ptot) Total available power in W
6493.29 : (vtot) Volume occupied by full assembly in cm^3
16.79 : (mtot) Total mass of panels in kg
4.38 : (lenopen) Length of deployed assembly in m
0.43 : (widopen) Width of deployed assembly in m

++
Outer Cover panels:

---

intermediate vals.txt:
4.5 : (tf) Frame thickness
8.6 : (dhinge) Diameter of the hinge
432.0 : 390.0 : (h2hdist[0,1]) Hinge to hinge distance
441.2 : 399.2 : 239.0 : (lframe[0,1,2]) Length of frame
195.0 : 147.0 : 195.0 : (wframe[0,1,2]) Width of frame
9 : 2 : (nxy[0][0,1]) nx and ny
4 : 3 : (nxy[1][0,1]) nx and ny
5 : 2 : (nxy[2][0,1]) nx and ny
18 : 1 : (ncp[0][0,1]) ncellspanel and choice
12 : 2 : (ncp[1][0,1]) ncellspanel and choice
10 : 1 : (ncp[2][0,1]) ncellspanel and choice

---

output vals.txt
130 : (ncells) Total number of cells
156.0 : (ptot) Total power generated in W
2911.43 : (vtot) Total volume taken up in cm^3
8.45 : (mtot) Total mass in kg
1.941 : 1.733 : (lenopen[0,1]) Length of opened assembly in m
0.201 : 0.153 : (widopen[0,1]) Width of opened assembly in m

4 Learning and Insights

I learned:

- How to implement functions and reference .py files in python
- That the nxy and choice variables needed to be implemented differently from the original perception. 2D arrays were required for ncellspanel and nxy.
- The scope and complexity of the python codes was too much for one session. The implementation of arrays, functions, imported files, reading and writing was all needed.
- The equations did not fully encapsulate what was needed in the code. Additional insight was needed to implement nxy and choice variables.
- The radiation protection coatings search is much more extensive. It needs to be continued in subsequent sessions.

References

[1] ExoLaunch. User manual, exo. exopod user manual ad-2 exopod cubesat integration guide. 2016.

[2] P. S. P. M. D. B. A. C. C. H. N. M. J. N. J. W. J. C. Rachel Trabert, Andrew Klesh and D. McKague. The extendable solar array system: A modular nanosatellite power system. *Astrodynamics Specialist Conference*, 2010.

[3] C. P. S. University. Cubesat sizes. (n.d.).