

Pressure Vessel Calculation for Yavin Thruster

J. Simmons

July 25, 2015

Abstract

Mach 30 is designing a compressed air powered cold gas thruster (Yavin) for educational kits and volunteer training. The Yavin project requires a number of engineering calculations. This report covers the calculations required to determine the minimum necessary wall thickness for the chamber based on the design and operating conditions. These calculations are implemented in the PressureVessel class. The PressureVessel class models the chamber as a thick walled cylinder and uses a simple incremental search to determine the minimum necessary wall thickness. Verification for the PressureVessel class included white box verification, in the form of a detailed inspection of the source code, and black box verification, in the form of nine test cases from the author's PhD dissertation. For each test case the calculated wall thickness matched the expected thickness. This combination of white and black box verification demonstrates the PressureVessel class correctly calculates the wall thickness.

1 Requirements

The Pressure Vessel Calculation module is being created to model the stresses in the chamber wall for use in a sizing loop to determine the required wall thickness in the Yavin Thruster's chamber. The resulting wall thickness will be used to drive the chamber wall thickness parameter in the CadQuery model of the Yavin Thruster. This approach will give a CAD model that can parametrically size and generate geometry from the thruster's design variables and operating conditions.

The key design variables driving the pressure vessel calculation include the chamber's inner radius, the yield strength of the selected chamber material, the desired factor of safety, and the chamber pressure. The key operating condition is the ambient pressure. The single result of running the pressure vessel calculation is the required wall thickness. Metric units will be used for all calculations and results. Table 1 lists all required variables and their units.

Table 1: Pressure Vessel Calculation Parameters

Direction	Name	Symbol	Units
Inputs			
	Chamber Inner Radius	r_i	m
	Yield Strength	σ_y	MPa
	Chamber Pressure	P_c	Pa
	Ambient Pressure	P_{amb}	Pa
	Factor of Safety	FS	
Outputs			
	Chamber Wall Thickness	t	m

2 Theory

A chamber is essentially a hollow cylinder with one end open to the propellant source (the injector face) and the other end converging to the opening where the thrust leaves the chamber to supply the thrust (the throat). There are two primary models available for analyzing the stress in the walls of a hollow cylinder under pressure: thin walled cylinders and thick walled cylinders. The thin walled cylinder model, while simpler, is only valid for cylinders with radius-to-thickness ratios greater than 10. In order to support chamber geometries regardless of radius-to-thickness ratio the chamber is being modeled as a thick walled cylinder. The sizing loop will find the minimum wall thickness which yields a maximum stress in the chamber wall within the material's yield strength, including factor of safety. The stresses calculated during the sizing loop are the tangential and radial stress. Equation (1) shows the calculation for the tangential stress in a thick walled cylinder and Equation (2) shows the calculation for the radial stress in a thick walled cylinder [1].

$$\sigma_t = \frac{p_i r_i^2 - p_o r_o^2 - r_i^2 r_o^2 (p_o - p_i) / r^2}{r_o^2 - r_i^2} \quad (1)$$

$$\sigma_r = \frac{p_i r_i^2 - p_o r_o^2 + r_i^2 r_o^2 (p_o - p_i) / r^2}{r_o^2 - r_i^2} \quad (2)$$

The equations above are in a generic form and can be applied to a number of design scenarios. Table 2 shows how the variables in Equation (1) and Equation (2) map to the parameters in Table 1. The variable r , the radius at the point of interest, bears some explanation. This variable determines where in the thick walled cylinder the stresses are being calculated. The pressure vessel calculation uses the maximum stress values to size the chamber wall thickness in a sizing loop. Therefore, the radius at the point of interest must be set to yield the maximum stress in the cylinder wall. For cylinders under tension (those with internal pressures greater than the external pressure like in the chamber wall), the point of highest stress is located at the inner radius. For cylinders under

compression (those with external pressures greater than the internal pressure like submarines), the point of highest stress is located at the outer radius [2].

Table 2: Pressure Vessel Calculation Parameters

Thick Walled Variable (units)	Symbol	Relation to Design Parameters
Tangential Stress (Pa)	σ_t	n/a
Radial Stress (Pa)	σ_r	n/a
Internal Pressure (Pa)	p_i	$= P_c$
External Pressure (Pa)	p_o	$= P_{amb}$
Inside Radius (m)	r_i	$= r_i$
Outside Radius (m)	r_o	$= r_i + t$
Radius at the Point of Interest (m)	r	$= r_i$

3 Implementation

The Pressure Vessel Calculation module is implemented as a Python class (see Listing below) and is based on the structural jacket sizing module in the author’s PhD research [3]. This section reviews the implementation line by line to provide a white box verification of the module.

The PressureVessel class has a constructor, a method to size the chamber wall thickness (calculate_wall_thickness), and three helper methods (sigma_tan, sigma_rad, and max_stress). The class opens with its variable declarations in lines 20-28. The class variables map to the six calculation parameters in Table 1. The class has two additional variables used to maintain state during the solver routine and required as inputs for the tangential and radial stress calculations: the current guess for the outer radius and the radius at the point of interest. Finally, the class has a variable to control the step size used in the solver.

The class constructor is implemented in lines 30-39. The constructor simply needs to assign values to the class variables. Note how the constructor requires a value for the thickness (t). This value is used as an initial guess for the wall thickness in the wall sizing loop. Most of the assignments in the constructor directly pass through the method arguments. However, the outer radius (line 33) and the radius at the point of interest (line 34) take on derived values based on the discussion above and shown in Table 2.

```

1 # Copyright (C) 2015 Mach 30 - http://www.mach30.org
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 #
7 #     http://www.apache.org/licenses/LICENSE-2.0
8 
```

```

9 # Unless required by applicable law or agreed to in writing,
  ↳ software
10 # distributed under the License is distributed on an "AS IS" BASIS,
11 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
  ↳ implied.
12 # See the License for the specific language governing permissions
  ↳ and
13 # limitations under the License.
14
15 import math
16
17
18 class PressureVessel:
19     # Variables used to calculate wall thickness
20     ri = None # inner radius, needs a value
  ↳ assigned
21     ro = None # outer radius, calculated from inner
  ↳ radius and thickness
22     r = None # radius of interest in the stress
  ↳ calculations, generally set to position of max stress (ri
  ↳ for tension, r0 for compression)
23     t = None # initial guess for thickness, needs
  ↳ value assigned
24     p_c = None # chamber pressure, needs value
  ↳ assigned
25     p_amb = None # ambient pressure, needs value
  ↳ assigned
26     material_strength = None # target maximum allowable strength
  ↳ (for example yield or ultimate), needs value assigned
27     fs = None # factor of safety
28     step_size = None
29
30     def __init__(self, ri, t, p_c, p_amb, material_strength, fs,
  ↳ step_size):
31         self.ri = ri
32         self.t = t # TODO: look at automatically
  ↳ selecting a guess, maybe based on thick-walled theory
  ↳ (ri/t >= 10)
33         self.ro = ri + t
34         self.r = ri
35         self.p_c = p_c
36         self.p_amb = p_amb
37         self.material_strength = material_strength
38         self.fs = fs
39         self.step_size = step_size
40
41     def calculate_wall_thickness(self):
42         """Solves for the desired wall thickness based on the
  ↳ inputs given"""
43

```

```

44     stress_limit = self.material_strength # TODO: Are these
      ↪ terms really interchangeable like this?
45
46     # TODO: Do we want to modify copies of ro and t to keep the
      ↪ originals for reference?
47     # TODO: The advantage of modifying the class variables is
      ↪ that you end up with a PressureVessel object with
48     # TODO: optimized ro and t values that you can access
      ↪ directly, such as pv.ro and pv.t
49     # TODO: Replace this solver process with something like a
      ↪ binary search
50     if self.max_stress() < stress_limit:
51         # Our vessel can handle more stress, so gradually
          ↪ decrease the thickness
52         while self.max_stress() < stress_limit:
53             self.t -= self.step_size
54
55             # The new thickness changes our outside radius
56             self.ro = self.ri + self.t
57
58             self.t += self.step_size # the while loop exits after
          ↪ one too many steps, add one back
59
60     else:
61         # Our vessel can't handle the stress, gradually
          ↪ increase the thickness
62         while self.max_stress() > stress_limit:
63             self.t += self.step_size
64
65             # The new thickness changes our outside radius
66             self.ro = self.ri + self.t
67
68     return self.t
69
70     def sigma_tan(self):
71         """Calculate the tangential stress in thick walled
          ↪ cylinder"""
72
73         return (self.p_c * self.ri**2 - self.p_amb * self.ro**2 -
          ↪ self.ri**2 * self.ro**2 *
74             (self.p_amb - self.p_c) / self.r**2) / (self.ro**2 -
          ↪ self.ri**2)
75
76     def sigma_rad(self):
77         """Calculate the radial stress in thick walled cylinder"""
78
79         return (self.p_c * self.ri**2 - self.p_amb * self.ro**2 +
          ↪ self.ri**2 * self.ro**2 *
80             (self.p_amb - self.p_c) / self.r**2) / (self.ro**2 -
          ↪ self.ri**2)

```

```

81
82     def max_stress(self):
83         """Calculates the max stress including the factor of safety
84         ↪ """
85         return max(self.sigma_tan(), self.sigma_rad()) * self.fs

```

The wall thickness sizing loop is implemented in the calculate_wall_thickness methods (lines 41-68). The sizing method uses a simple incremental search to size the chamber wall thickness. This method tests the wall size for conformance to the objective, the minimum wall thickness which yields a maximum stress, including factor of safety, within material's yield strength, and then increments or decrements the wall thickness by a specified step size as needed until the objective is met. The method starts by defining the stress limit (equal to the material's yield strength) in line 44. The sizing loop is implemented in lines 50-66. The test in line 50 determines if the initial guess for the wall thickness is larger than needed (maximum stress is less than the stress limit indicating the wall is over sized) or less than needed (maximum stress is greater than the stress limit indicating the wall is undersized). In the first case, the while loop in lines 52-56 iteratively reduces the wall thickness (and outer radius) until the maximum stress is greater than or equal to the stress limit. Line 58 increments the wall thickness by one step increment since the loop above ran until the wall was one step too small. The second case (wall thickness is too small) is processed in lines 62-66. In this case, the while loop iteratively increases the wall thickness (and outer radius) until the maximum stress is less than or equal to the stress limit. No adjustment is required after this loop as the calculated wall size already yields a design where the maximum stress is less than or equal to the stress limit. The method concludes by returning the calculated wall thickness in line 68.

The remaining methods are the three helper methods used to calculate the tangential stress, the radial stress, and the maximum stress including the factor of safety. The sigma_tan method (lines 70-74) calculates the tangential stress using Equation (1). Visually comparing the source code in lines 73 and 74 to Equation (1) verifies this method is implemented correctly. Similarly, the sigma_rad method (lines 76-80) calculates the radial stress using Equation (2). As before, visually comparing the source code in lines 79 and 80 to Equation (2) confirms it is implemented correctly. Finally, the max_stress method (lines 82-85) calculates the maximum stress including the factor of safety by returning the product of the factor of safety and the maximum of the tangential and radial stresses for the current wall thickness.

4 Tests

This section provides additional verification of the Pressure Vessel Calculation module in the form of nine test cases. The test cases come from data collected in the author's dissertation as part of a sensitivity study investigating the impact of material selection on structural jacket wall

thickness (see Figure 1) [3]. Three test cases were selected from each of three materials (INCONEL 718, Alloy 188, Oxygen-Free Copper) to provide a range of material strengths, geometries, and chamber pressures.

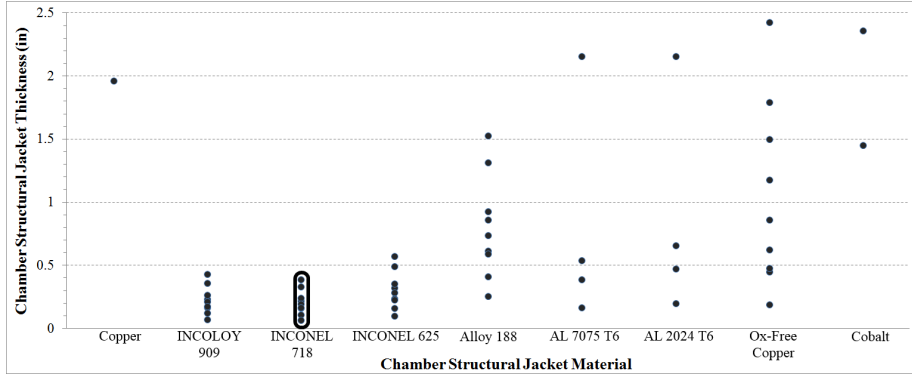


Figure 1: Simulation Results from Dissertation Used to Derive Test Cases [3]

The following code demonstrates how the tests were conducted using the design point for the demonstration chamber CAD model. For each design point, a `PressureVessel` object is created, with the class variables set to values including their units implemented using the Python Pint library. The `calculate_wall_thickness` method is then called to solve for the required wall thickness.

```

from pressure_vessel_calcs import PressureVessel

pv0 = PressureVessel(20.0e-3 * units.meter,
                    2e-3 * units.meter,
                    3.5e6 * units.pascal,
                    101325 * units.pascal,
                    40e6 * units.pascal,
                    2.0,
                    0.0001 * units.meter)
pv0.calculate_wall_thickness()

```

The results of the nine tests are shown the first nine rows of Table 3 (the final row shows the results for the demonstration design point calculation above). Each row lists the input parameters for the corresponding test case, the calculated thickness (t_{calc}), and the thickness as reported in the author's dissertation (t). Note, the original analyses were calculated in US units. The original units were used to run the test cases and converted to the units shown in Table 3 to maintain consistency with the Yavin Thruster's use of metric units and to provide compact presentation.

Table 3: Pressure Vessel Calculation Test

r_i	P_c	P_{amb}	σ_y	t_{calc}	t
133.1 mm	9192.7 kPa	0.0 kPa	1075.6 MPa	1.7 mm	1.7 mm
133.1 mm	29206.5 kPa	0.0 kPa	1075.6 MPa	5.5 mm	5.5 mm
242.4 mm	28443.4 kPa	0.0 kPa	1075.6 MPa	9.8 mm	9.8 mm
133.1 mm	9192.7 kPa	0.0 kPa	289.6 MPa	6.5 mm	6.5 mm
242.4 mm	17812.1 kPa	0.0 kPa	289.6 MPa	23.5 mm	23.5 mm
242.4 mm	28443.4 kPa	0.0 kPa	289.6 MPa	38.8 mm	38.8 mm
133.1 mm	9192.7 kPa	0.0 kPa	390.2 MPa	4.8 mm	4.8 mm
196.7 mm	30003.5 kPa	0.0 kPa	319.9 MPa	29.9 mm	29.9 mm
242.4 mm	17812.1 kPa	0.0 kPa	120.0 MPa	61.6 mm	61.6 mm
20.0 mm	3500.0 kPa	101.3 kPa	40.0 MPa	3.8 mm	

As can be seen in Table 3, the Pressure Vessel Calculation module passes all nine tests. For each test case the calculated wall thickness matches the thickness as reported in the author’s dissertation. When combined with the white box verification in the previous section, these tests demonstrates the Pressure Vessel Calculation module correctly calculates the the wall thickness.

References

- [1] Shigley, J. and Mischke, C., *Mechanical Engineering Design*, McGraw-Hill, fifth edition ed., 1989.
- [2] Kharagpur, “Thick cylinders Stresses due to internal and external pressures”, *Thin and thick cylinders*, IIT, version 2, <http://www.nptel.ac.in/courses/112105125/pdf/module-9%20lesson-2.pdf>.
- [3] Simmons, J. *Design and Evaluation of Dual-Expander Aerospike Nozzle Upper-Stage Engine*, PhD dissertation (<http://www.dtic.mil/get-tr-doc/pdf?AD=ADA609649>), Air Force Institute of Technology, 2014.