The calculations involved in the Channels analysis are simple averaging and calculation of the minimum and maximum values in arrays. A single JavaScript function, shown in Listing A.14 iterates over input arrays and records the minimum and maximum values while calculating the average. This function is then called for the LOX and LH2 channel thickness arrays and the LOX and LH2 channel spacing arrays.

Listing A.14: Channels Component Average Function

```
function average(values) {
        var avg = 0;
        var numVals = values.length;
        var min = values.value(0);
        var max = values.value(0);

        for (var i = 0; i < numVals; i++) {
                avg += values.value(i);

                if (values.value(i) < min)
                        min = values.value(i);

                if (values.value(i) > max)
                        max = values.value(i);
        }

        avg = avg / numVals;

        return [avg, min, max];
}
```

Inspection of the Channels component showed it properly converts the DEAN analysis' vector results to suitable scalars, with appropriate constraints, for use by the Structural Jacket sizing analysis, meeting the purpose of this component.

A.4.2 Structural Jacket Size.

The purpose of the Structural Jacket Size component is to determine the wall thicknesses for the chamber and aerospike structural jackets and the engine's plumbing. Additionally, this component calculates geometric parameters for various engine components including the plumbing cross-sectional area, the volume and weight of the structural jackets, the volume and weight of the cooling channel covers if present, and the volume and mass of the uncooled aerospike tip. The component was verified through inspection.

The DEAN engine has two structural jackets, one for the chamber and one for the aerospike. The cross-section shown in Figure A.21 shows DEAN structural jackets in relation to the cooling jackets and the ambient environment. The chamber, the black disc in the figure, is a high pressure region and the ambient environment is a low pressure region. Observe how the chamber structural jacket is containing the pressure from the chamber internally and is therefore experiencing stress in tension. Conversely, the aerospike structural jacket is containing the pressure from the chamber externally and is therefore experiencing stress in compression.

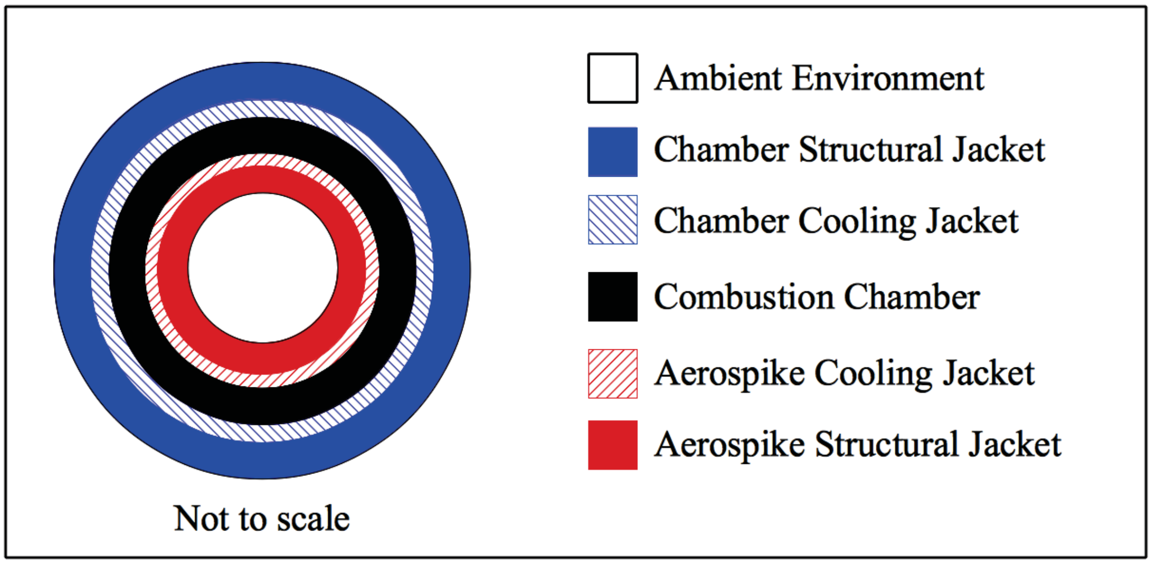

Figure A.21: Cross-Section of the DEAN Engine

The Structural Jacket Size component computes the structural jacket sizes by solving for the minimum wall thicknesses which yield a maximum stress in the structural jackets within the stress limit for the selected material, including the model's factor of safety. As part of this calculation, the wall thicknesses are increased or decreased until the stress limit is just met. The Structural Jacket Size component performs this analysis for the highest pressure differential instead of one for each station. The system level model assumes a nearly uniform temperature in the structural jacket, and in turn constant material properties along the length of the engine. The highest pressure differential is calculated from the difference between the maximum pressure in the related cooling channel and the ambient pressure.

As with the Cooling Jacket component, initial inspection of the source code for the Structural Jacket Size component showed it had a number of errors and repeated code blocks making inspection and error correction overly complicated. Before continuing with the verification process, the component's implementation was refactored to minimize duplicated code and improve readability. During the refactoring effort, the various errors were corrected and repeated code blocks were placed in user defined functions.

With the errors in the Structural Jacket Size component corrected, the inspection could continue with a focus on the implementation of the governing equations. The governing equations for the Structural Jacket Size analysis include the radial and tangential stress in a thick walled pressure vessel and the volumes for the various components of the DEAN. During the inspection, the source code for each of these calculations was compared to its governing equation and assumptions.

Equation (A.60) shows the calculation for the tangential stress in a thick walled cylinder [70]. Listing A.15 shows the function which calculates the tangential stress. It is a direct implementation of Equation (A.60)

$$\sigma_t = \frac{p_i r_i^2 - p_o r_o^2 - r_i^2 r_o^2 (p_o - p_i)/r^2}{r_o^2 - r_i^2} \tag{A.60}$$

Listing A.15: Tangential Stress in a Thick Walled Cylinder Function

```
function sigma_tan(ri, ro, r, Pi, Po) {
        return (Pi*Math.pow(ri,2) - Po * Math.pow(ro,2) -
               Math.pow(ri,2) * Math.pow(ro,2) * (Po-Pi) / Math.pow(r,2)) /
               (Math.pow(ro,2)-Math.pow(ri,2));
}
```

The calculation for the radial stress in a thick walled cylinder is very similar to the calculation for the tangential stress, as can be seen in Equation (A.61). In both of these equations, positive values indicate the cylinder is in tension, as would be expected for the chamber's structural jacket, and negative values indicate the cylinder is in compression, as would be expected for the aerospike's structural jacket [70].

$$\sigma_r = \frac{p_i r_i^2 - p_o r_o^2 + r_i^2 r_o^2 (p_o - p_i)/r^2}{r_o^2 - r_i^2} \tag{A.61}$$

Listing A.16 shows the function calculating the radial stress. It is a direct implementation of Equation (A.61)

Listing A.16: Radial Stress in a Thick Walled Cylinder Function

```
function sigma_tan(ri, ro, r, Pi, Po) {
        return (Pi*Math.pow(ri,2) - Po * Math.pow(ro,2) +
               Math.pow(ri,2) * Math.pow(ro,2) * (Po-Pi) / Math.pow(r,2)) /
               (Math.pow(ro,2)-Math.pow(ri,2));
}
```

With these two calculations implemented, the remaining code for calculating the structural jacket thickness is a straight forward loop over increasing or decreasing thickness values until the stress limit is just satisfied. Listing A.17 shows the function which implements this sizing loop.

Listing A.17: Calculate Wall Thickness Function

```
// tension can take on two values: 1 == in tension, -1 == in compression
function calculateWallThickness(ri, ro, t, Pi, Po, stressLimit, tension) {
  // initialize variables
  var r = 0.0;
  var stepSize = 0.001;
  if (tension == 1) {
    r = ri;
  }
  else {
    r = ro;
  }

  var maxStress = 0.0;
  if (tension == 1) {
    maxStress = fs.value * Math.max(sigma_tan(ri, ro, r, Pi, Po),
                                sigma_rad(ri, ro, r, Pi, Po)) / 1000;
  }
  else {
    maxStress = fs.value * Math.min(sigma_tan(ri, ro, r, Pi, Po),
                                sigma_rad(ri, ro, r, Pi, Po)) / 1000;
  }

  // multiply by tension to adjust for tension vs compression
  if (maxStress < stressLimit * tension) {
    while (maxStress < stressLimit * tension) {
      if (tension == 1) {
        t = t - stepSize;
        ro = ri + t;
        maxStress = fs.value * Math.max(sigma_tan(ri, ro, r, Pi, Po),
                                    sigma_rad(ri, ro, r, Pi, Po)) / 1000;
      }
      else {
        t = t + stepSize;
        ri = ro - t;
        maxStress = fs.value * Math.min(sigma_tan(ri, ro, r, Pi, Po),
                                    sigma_rad(ri, ro, r, Pi, Po)) / 1000;
      }
    }
  }
  else {
    while (maxStress > stressLimit * tension) {
      if (tension == 1) {
        t = t + stepSize;
        ro = ri + t;
        maxStress = fs.value * Math.max(sigma_tan(ri, ro, r, Pi, Po),
                                    sigma_rad(ri, ro, r, Pi, Po)) / 1000;
      }
      else {
        t = t - stepSize;
        ri = ro - t;
        maxStress = fs.value * Math.min(sigma_tan(ri, ro, r, Pi, Po),
                                    sigma_rad(ri, ro, r, Pi, Po)) / 1000;
      }
    }
  }

57    return t;
58  }
```

One important detail about this function is the purpose and use of the function argument *tension*. The function argument *tension* is a flag indicating whether the sizing loop is being run for a cylinder in tension ($tension = 1$) or compression ($tension = -1$). Naturally, the function is called with a value of 1 for the chamber's structural jacket, and a value of -1 for the aerospike's structural jacket. The argument is used in one of two ways throughout the function. First, it is used as a test condition in *if statements* such as in Lines 6, 14, 26, and 42, where it controls whether the calculation should be performed for a tension or compression case. Second, it is used as a sign adjustment for the maximum stress in the cylinder such as in Lines 24 and 41. For cases where the cylinder is in tension, the stress values will be positive, as noted above, and the *tension* argument has a value of 1. When the positive stress value is multiplied by the *tension* value of 1, the stress value remains positive. For cases where the cylinder is in compression, the stress values will be negative, as noted above, and the *tension* argument has a value of -1. When the negative stress value is multiplied by the *tension* value of -1, the stress value becomes positive.

The remainder of this function performs the iterations which increase or decrease the wall thickness from the starting value provided by the function argument t until the stress limit is just met using a pair of *while loops*. The first *while loop*, shown in lines 25-38, implements the decreasing case for cylinders in tension, and the increasing case for cylinders in compression. This reversal of implementations is required because in compression cases, the smaller stress value is a more negative number, representing a larger magnitude of stress meaning the stress limit has been exceeded. The second *while loop*, shown in Lines 41-54, similarly implements the increasing case for cylinders in tension, and the decreasing case for cylinders in compression.