UCRL-TR-200695

As-built model generation for a cylindrical test object

D. H. Chambers, D. M. Goodman, R. R. Leach

September 30, 2003

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Introduction

The goal of the As Built Model Development (ABMD) project in ADAPT is to determine how to produce finite element (FEM) meshes from information obtained from nondestructive inspection of parts and assemblies. These meshes could then be used in computational analysis tools to predict the actual performance of the parts, as opposed to the design performance that is obtained using meshes derived from design information. Information derived from several inspection methods could be used to derive the meshes, with some methods sensitive to geometry and others to material properties. The process is summarized in the diagram in Fig. 1.

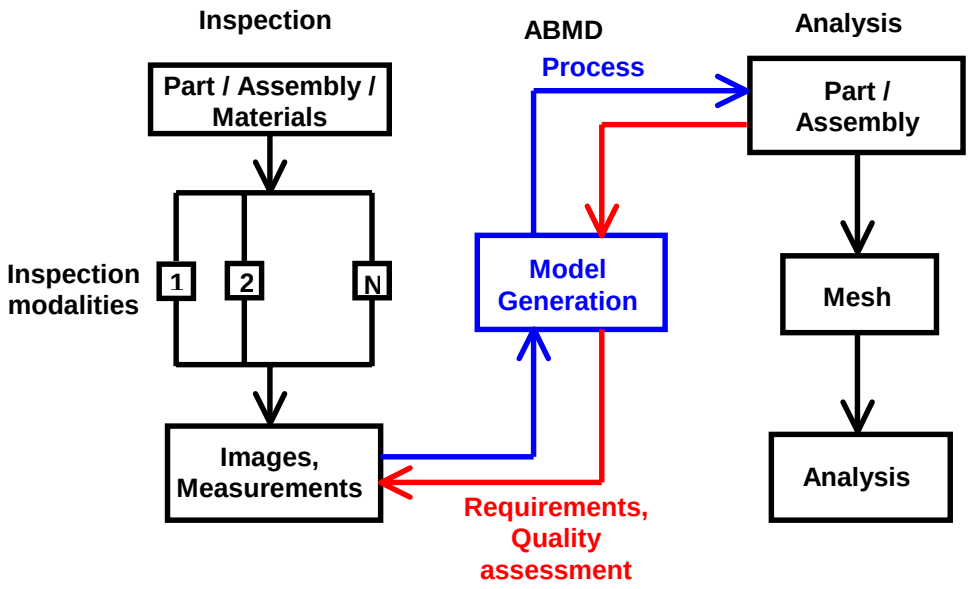


Figure 1. Flow diagram of As Built Model Development (ABMD) process.

The left side of Fig.1 represents nondestructive inspection, whose products are measurements or images of parts, assemblies, or materials. The right side represents the analysis process that begins with the production of a mesh from a part or assembly and ends with a performance analysis using a variety of computational tools. The ABMD process takes inspection results and uses them to generate a mesh and material specifications that represents a synthesis of all the inspection results. Though the main flow of information is from inspection to analysis, there are analysis requirements and quality control information that flows backward to assist in the synthesis of inspection information and possibly change the way that the inspection data is processed.

In our implementation we considered two general approaches for ABMD. The first is to directly convert the final products of inspection into a model of the part. For example, if the inspection process is x-ray CT, the output is a block of voxels representing the distribution of absorptivity or density of the part. Using image processing and segmentation tools, this could be converted into a set of surfaces representing the material boundaries of the part, which are then used to generate a finite

element mesh. The second approach to ABMD is to replace the usual processing of the inspection data with one that directly estimates the quantities required to generate a mesh. Again using the example of x-ray CT, we could decide to generate the material surfaces directly from the x-ray projection data (sinograms) without doing a reconstruction into voxels. This second approach has the advantage of eliminating the image processing and segmentation steps and the corresponding processing uncertainties. It could also be faster and use less computer memory since we would not have to work with a three-dimensional block of voxels. However, it requires a computational model of both the part and the expected deviations from the design configuration. If the actual deviations in the part are not included in the model, the resulting as-built model will be in error. For example, if the part is a sphere we could create a mathematical model of the surface using a series expansion in spherical harmonics, truncated at a predetermined order. We could estimate the coefficients of each term in the expansion directly from x-ray CT data and generate a mesh from the series. However, a crack in the sphere could not easily be represented by an expansion in spherical harmonics and would likely be missed in the as-built model. Furthermore, the effects of the crack on the x-ray CT data would cause errors in the coefficients of the series expansion, corrupting the model. In some instances, it may be possible to estimate the accuracy of the model parameters and detect when the model is insufficient to represent the part.

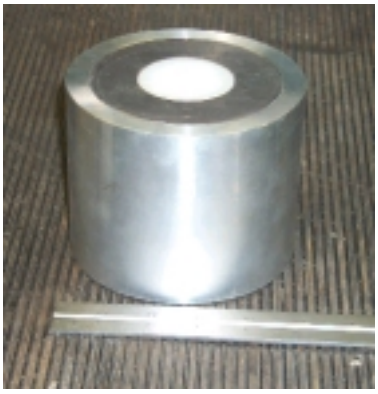


Figure 2. Cylindrical test object

Segmentation approach

To test these two approaches to ABMD, we constructed meshes from x-ray CT data of a cylindrical test object (Fig.2). The test object consisted of a central core of plastic 25.4 mm in diameter, surrounded by a 25.4 mm thick shell of lead, and encased in a 12.7 mm thick aluminum outer jacket, with an overall height of 101.6 mm. A set of 600 cone-beam radiographs was obtained, with each radiograph representing a projection of the object in a direction perpendicular to the cylinder axis (one projection every 0.6°) (Fig. 3). These were reconstructed using a cone-beam CT algorithm into 40 cross-sections, each 840 x 840 pixels, of x-ray absorptivity. The pixel size was 167.3 μm and the distance between each cross-section was also 167.3 μm, representing a 6.7 mm tall portion of the object. Figure 4 shows one of the 40 reconstruction cross-sections of the test object. The material interfaces between cylinders are easily seen in the image but are more difficult to discern in the plot of the reconstruction across a diameter of the slice.

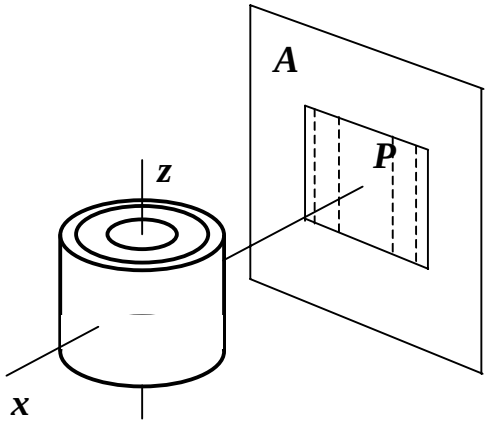


Figure 3. CT radiography geometry for test object.
Projection ***P*** onto detector plane ***A*** is taken along the *x* axis.
Object is rotated around *z* axis and projection recorded every 0.6°.

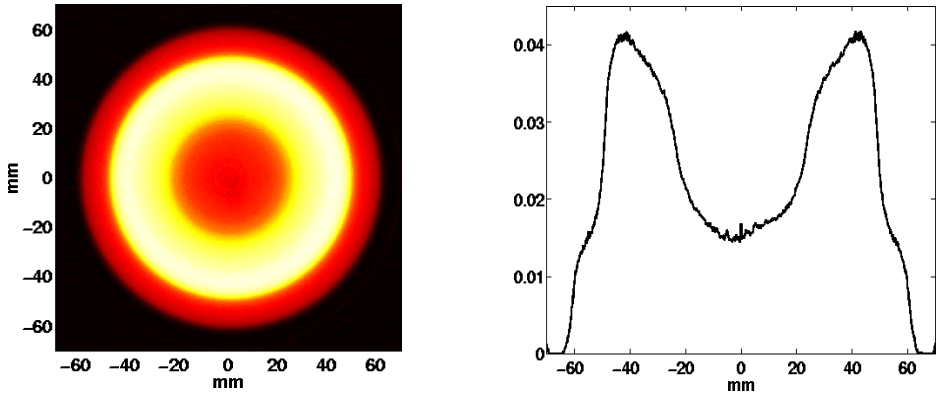
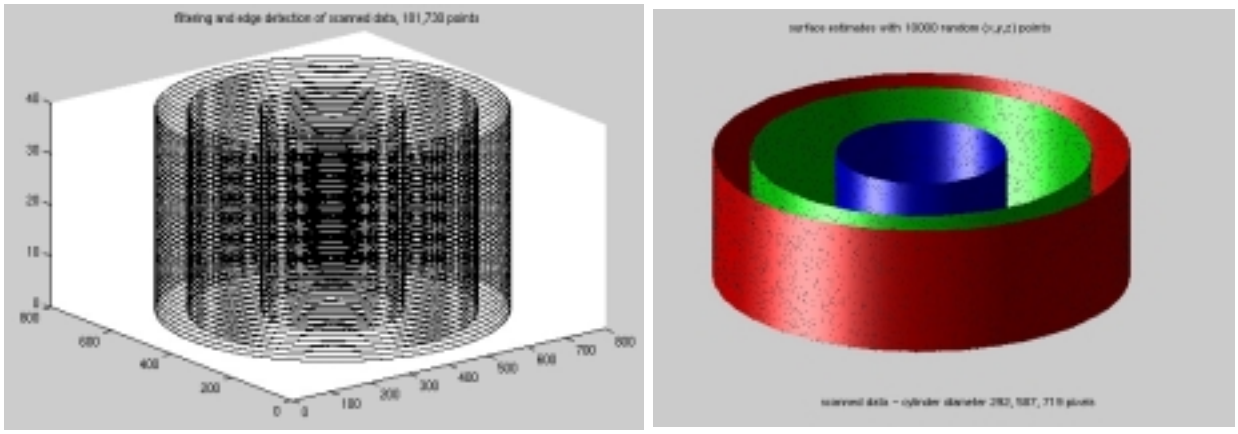


Figure 4. Cross-section (left) and profile along the diameter (right)
of the CT reconstruction.

The interfaces are expected to be the points of steepest slope (maximum absolution gradient). Using Matlab, the absolute value of the gradient in the radial direction is calculated and a smoothing filter applied for each slice. The points on the interfaces are determined by selecting every point where the smoothed absolute gradient exceeds a given threshold. The same threshold was chosen for all interfaces and slices. The collection of points (Fig. 5a) defines a point cloud of the interfaces, which is then used to create a cylindrical surface and calculate the mean radius for each interface (Fig. 5b). This process of extracting geometrical information from volumetric data is commonly called segmentation. The slice-to-slice variation in the mean radius was minimal (Fig. 6), implying that the part radius was constant to within the reconstruction resolution (~170 μm). Finally the radii and cylindrical surface information is imported into Truegrid to create the surface mesh shown in Figure 7. Table 1 shows the estimated mean radius for

each interface and the actual radius. These show that the gradient approach to estimating radius consistently underestimates the actual radius.



(a) (b)

Figure 5. Point cloud (a) and resulting surfaces (b).

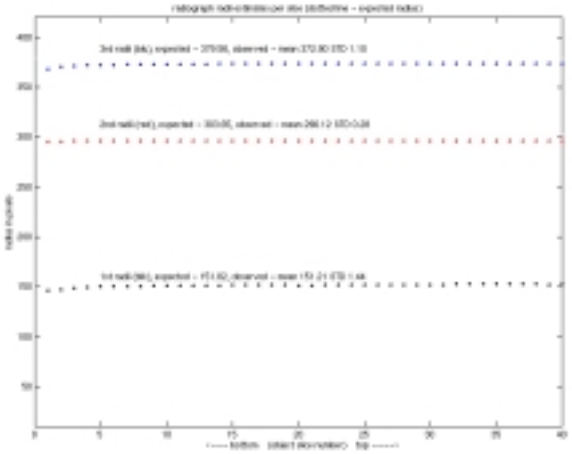


Figure 6. Interface radii estimated from each CT cross-section. Variation between slices is minimal.

The process of CT reconstruction, calculation of gradients, thresholding to produce a point cloud, defining a surface, and finally generating the mesh requires multiple inputs from the user at each step. The resulting mesh is a product not only of the original CT data but also of all the user inputs required in the process. In this example, the data was fairly clean (low noise) and the choices for the inputs were relatively straightforward. For noisier data, the choices would be more problematic and the final result could depend strongly on user input. It is not obvious how to quantify the accuracy of the final mesh. One might try a Monte Carlo approach that would involve generating a statistical ensemble of meshes using a range of user inputs. However, such an approach

would likely be impractical because of the large number of parameters used in the production of the mesh.

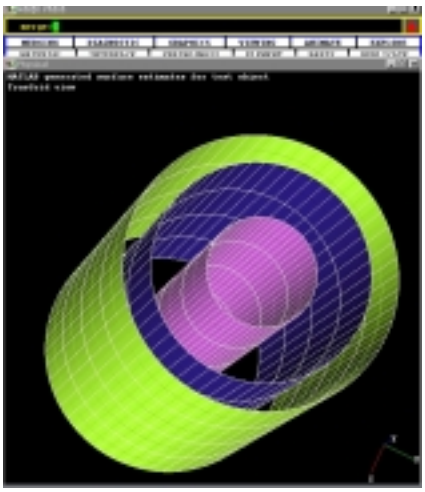


Figure 7. TrueGrid surface meshes of material interfaces of test object.

Table 1. Comparison of interface radii

Interface	Design radius (mm)	Estimated from CT reconstruction (mm) (0.1673 mm voxels)
Inner	25.4	25.30
Middle	50.8	49.54
Outer	63.5	62.39

In summary, the process of generating a mesh from CT reconstructions was straightforward in our example. Relatively simple tools that are available in Matlab and other commercially available image processing packages can be used to extract the information required by mesh generation software (e.g. Truegrid). This approach requires no prior information about the object and could, in principle, generate meshes that contain highly irregular defects (cracks, voids, etc.). However, the required processing is object dependent and requires the interaction of the user at each step. It may also require substantial computational resources (e.g. memory, processing speed) for more complicated objects at high resolution. Finally, the accuracy of the resulting mesh may be difficult to quantify. These results will now be compared with those from the model-based approach.

Model-based approach

As described in the introduction, the model-based approach attempts to circumvent the processing required for the segmentation approach by using a model of both the object and its expected deviations. The model parameters are then estimated directly from the raw inspection data (CT projections in our example) and used to generate the mesh. As expected, the effectiveness of this approach depends critically on the choice of model and parameterization. Bad models lead to bad meshes regardless of the quality of the inspection data. However, given a good model, this approach can be significantly faster, less complicated, require less user interaction, and provide error estimates that quantify the accuracy of both the model and the resulting mesh.

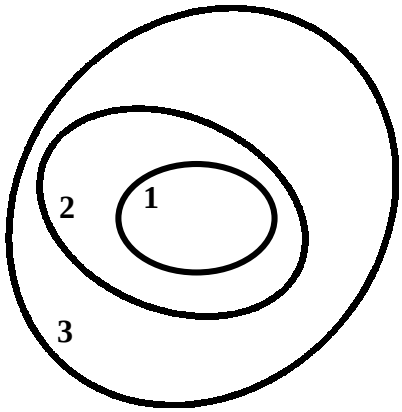


Figure 8. Three ellipse model of test object cross-section.

For the cylindrical test object, we attempted to extract the radii of the material interfaces for a single cross-section near the middle of the cylinder directly from the x-ray projections of the cross-section. As designed, the material interfaces in a cross-section would form three concentric circles. In our model, we chose to allow for errors in concentricity and eccentricity. These errors could be accommodated by constructing an interface model of three nested ellipses, each with their own major and minor radii, orientations, and centers (Fig. 8). The ellipses were labeled 1, 2, and 3 beginning with the innermost interface, and each described by the following equation:

$$\left(\frac{x-x_i}{\sigma_{xi}}\right)^2+\left(\frac{y-y_i}{\sigma_{yi}}\right)^2+r_i\,(x-x_i)(y-y_i)=1 \qquad (1)$$

where σ_{xi} and σ_{yi} are scaling parameters, and r_i determines the orientation (tilt) of the ellipse. The tilt parameter satisfies $|r| \le 2/\sigma_{xi}\sigma_{yi}$ for a true ellipse. The index i takes the values 1, 2, or 3, depending on which ellipse is being referenced. The angle θ_i between the x axis and the minor axis of the ellipse is given by

$$\theta_i = -\frac{1}{2}\arctan\left(\frac{r\sigma_{xi}^2\sigma_{yi}^2}{\sigma_{xi}^2-\sigma_{yi}^2}\right). \tag{2}$$

Note that if $\sigma_{xi}=\sigma_{yi}$ and r is non-zero, the ellipse is oriented with its major and minor axes 45° from the reference axes. Alternatively, if $\sigma_{xi}=\sigma_{yi}$ and $r=0$, the ellipse becomes a circle and the orientation is ambiguous. The major and minor radii (a_i and b_i respectively) are given by

$$a_i^2 = \frac{2}{4-r^2\sigma_{xi}^2\sigma_{yi}^2}\left[\sigma_{xi}^2+\sigma_{yi}^2+\sqrt{(\sigma_{xi}^2-\sigma_{yi}^2)^2+r^2\sigma_{xi}^4\sigma_{yi}^4}\right],$$

$$b_i^2 = \frac{2}{4-r^2\sigma_{xi}^2\sigma_{yi}^2}\left[\sigma_{xi}^2+\sigma_{yi}^2-\sqrt{(\sigma_{xi}^2-\sigma_{yi}^2)^2+r^2\sigma_{xi}^4\sigma_{yi}^4}\right]. \tag{3}$$

We will also use the geometric mean R_i of the radii for comparison to the nominal value:

$$R_i = \sqrt{a_i b_i} = \frac{\sqrt{\sigma_{xi}\sigma_{yi}}}{\left(1-r^2\sigma_{xi}^2\sigma_{yi}^2/4\right)^{1/4}}. \tag{4}$$

In addition to the 15 parameters that describe the three ellipses we added the center of rotation for the projections (x_c,y_c), and the absorptivities of the plastic, lead, and aluminum, for a total of 20 model parameters to be estimated from the projection data. This can be compared with the nearly 400,000 pixel values for absorptivity represented by a single cross-section from the CT reconstruction. Given this large difference we might naturally expect better estimates of the 20 as-built model parameters than for the pixel values in the CT reconstruction using the same set of projections. This expectation is generally true (the "parsimony principle") and allows us to calculate Cramer-Rao (CR) lower bounds for the variance of each model parameter.

The Cramer-Rao lower bounds for the variances of the parameters are useful indications of the compatibility between the model and the data set. These are calculated from an estimate of the noise variance in the data. Consider a one-dimensional model $y = f(x)+n$, where x is the model parameter, y is the data, $f(x)$ is the model, and n is additive noise. Let $\hat{x}$ be an unbiased estimator of the parameter x given the data y. The Cramer-Rao lower bounds for the variance of the estimate is given by

$$Var(\hat{x}) \geq \frac{\sigma_n^2}{N}\left(\frac{\partial f}{\partial x}\right)^{-2}, \tag{5}$$

where σ_n^2 is the variance of the noise, and N is the number of data points. If the model is well matched to the data, the C-R bounds will be small. Large C-R bounds indicate either noisy data or that the data is not sensitive to the parameter. Since we expect the ellipses for our cross-section to be nearly circular, the C-R bounds for the tilt parameters r_i may be quite large compared to their values ($\partial f/\partial x$ is small). If the actual variance of the

estimate is significantly larger than the C-R bounds, then the model is not capturing all the significant features in the data. In our case this would mean that the projection data contains features caused by cross-section variations not describable by our model of nested ellipses, *e.g.* large cracks or notches in the interfaces between materials. We would then have to modify the model in order to capture these features. Calculating the C-R bounds gives us measures of both the performance of our current estimates and the ultimate accuracy we could expect from the given data. This is a significant advantage for the model-based approach to as-built model development.

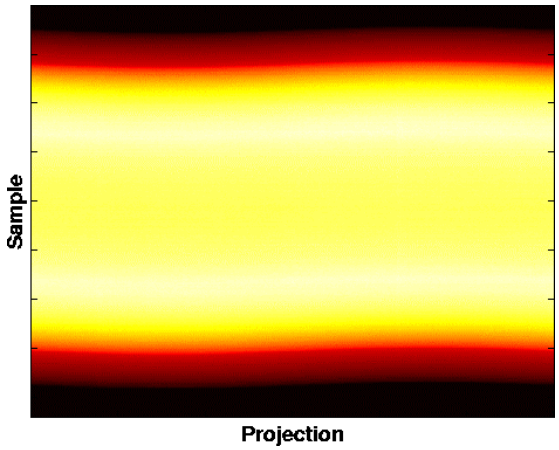


Figure 9. Sinogram of cross-section.

Figure 9 shows the projection data for all views through the cross-section displayed as an image (a *sinogram*). To extract the model parameters from the data we first choose an initial set of parameters, calculate the expected sinogram from the model, compare with the actual data, then update the model parameters using the differences between the calculated sinogram and the actual data. This is repeated until either the model parameters no longer change significantly, or the error between the calculated data and real data is minimized. At that point the process is judged to have converged and the final model parameters are saved. A simple line integral projection routine was used to produce sinograms from the ellipse model. This was incorporated into a MINPACK iterative optimization routine to update the model parameters. Figure 10 shows the change in the rms error between the calculated sinogram and the real sinogram as the process was iterated. The change is rms error decreases rapidly in the first 20 iterations, with little change afterward. After 35 iterations the process is judged to have converged and the final model parameters are obtained.

A comparison between model predictions for a projection and the actual data is shown in Figure 11. The first comparison uses only the 18 model parameters for the ellipses to calculate the projection. The second adds the two parameters for the center of rotation for the entire sinogram to the model. The agreement between model and data is significantly improved if the center of rotation is estimated from the data instead of assumed known from the description of the apparatus. Thus experimental parameters can be included along with the model parameters of the object. This allows corrections to be made for uncertainties in the experimental system that produces the data.

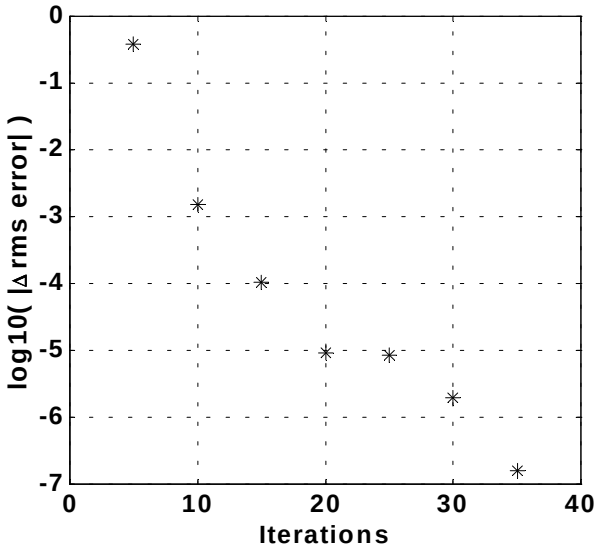


Figure 10. Change of rms error with iteration.

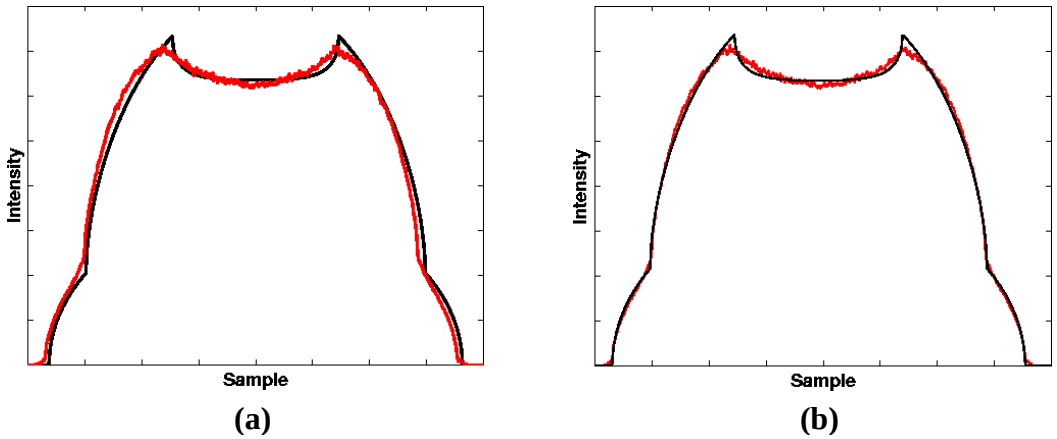


Figure 11. Comparison between model projection (red) and data (black): (a) nominal center of rotation, (b) center estimated from data.

Table 2 shows the results for the 15 model parameters that describe the ellipses. As expected, the scaling parameters σ_x and σ_y for each ellipse are nearly identical, indicating that the interfaces are nearly circular. The estimated centers of the ellipses are all within a 0.1 mm radius circle of their mean. The last column is the geometric mean R of the major and minor radii as given by (4). The R values for the inner and middle ellipse are larger than the scaling parameters due to the effect of the tilt parameter. For the outer ellipse, the value of the tilt parameter exceeds the allowed range, giving a complex number for R (indicated by an asterisk in Table 2). This comes from the inherent

ambiguity of the tile parameter when the ellipse is nearly circular, and can be seen in the Cramer-Rao bounds given in Table 3. The bounds for the tilt parameters of the inner and middle ellipses are large compared with their estimated values. However, the bound for the outer tilt parameter is much smaller than the estimate, indicating the data should allow a fairly precise estimate of *r*. Since the actual *r* estimate is outside the allowed range, there must be another source of error that makes the real variance much greater than the lower bounds. The C-R bounds for the other parameters are much smaller than the estimate of the parameters, indicating that these parameters are well characterized by the data.

Table 2. Model parameters for each ellipse after convergence.
The asterisk in the last entry of the last column indicates no value.

Ellipse	σ_x (mm)	σ_y (mm)	r (mm)	x_0 (mm)	y_0 (mm)	R (mm)
Inner	25.59	25.70	0.0024	1.012	-0.0493	33.04
Middle	51.57	51.58	-0.00029	1.193	-0.043	53.71
Outer	63.23	63.23	-0.00075	1.147	-0.077	*

Table 3. Cramer-Rao lower bounds for the ellipse parameters

Ellipse	σ_x (mm)	σ_y (mm)	r (mm)	x_0 (mm)	y_0 (mm)
Inner	0.0033	0.0017	0.0016	0.0029	0.00013
Middle	0.000097	0.0016	0.0022	0.0013	0.0012
Outer	0.00035	0.0021	0.0000038	0.0018	0.0000041

Summary

We have constructed a finite element mesh from x-ray CT data of a cylindrical test object using two approaches. The first approach starts from a tomographic reconstruction of the radiographic data, using various signal and image processing techniques to identify, characterize, and discretize the interfaces between materials. This approach makes little or no assumptions about the regularity or geometry of the interface and could, in principle, model cracks, pits, or other irregularities in the interface. The second approach starts from a geometric model describing both the nominal interface and a limited set of possible design deviations. By comparing simulated data with measurements, the optimum values for the model coefficients are extracted directly from the radiographic data. The analytical form of the model allows us to calculate lower bounds for the variances of the model coefficients using the variance of the noise in the data. These bounds are useful for determining the accuracy of the model for describing the as-built configuration.

Table 4 compares the values for the radii of the cylindrical interfaces determined using both approaches. The radii estimated from the CT reconstruction are slightly smaller (~100 μm) than the design values. Those estimated from the model are larger than design for the inner and middle interfaces, but slightly smaller for the outer

interface. The reasons for these deviations are not clear. Unfortunately, it was not possible to measure the actual interface radii directly for the slice used in the model without destroying the object.

Table 4. Summary of radii estimates

Interface	Design radius (mm)	Estimated from CT reconstruction (mm) (0.1673 mm voxels)	Mean model parameters (mm)
Inner	25.4	25.30	25.65
Middle	50.8	49.54	51.58
Outer	63.5	62.39	63.23

From this exercise, we conclude that both the segmentation approach and the model-based approach produce reasonable results that can be used to construct a finite element mesh of the as-built configuration of the test object. The choice of which approach would be best depends on the particular object and nature of the expected deviations from the as-designed configuration. The segmentation approach requires extensive user involvement to select thresholds and judge the efficacy of various smoothing and segmentation techniques. This may make it difficult to assess the final accuracy of the grid in representing the object. However, the segmentation approach requires no restriction on the nature of the deviations from the as-design configuration and can, in principle, create meshes that include cracks, pits, and other highly irregular features that might be present in an actual object. The model-based approach may be easier and more quantifiable but at the cost of restricting the nature of the deviations between as-built and as-designed configurations. The result would be only as good as the model and would not be accurate if the as-built configuration contains features that are not included in the model. In practical applications we expect the model-based method might be useful when only particular deviations from the as-designed configuration are expected or important. It could be used to create initial as-built meshes for quick analysis of a large number of parts. The more general segmentation approach could then be used for further study of a few parts that have particular features of interest.