SANDIA REPORT

SAND2019-14619

Printed September 2019

Sandia National Laboratories

Implementation and Verification of Isolated Defects for GTS Lifecycle Analyses

Guy L. Bergel, Lauren L. Beghini

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

ABSTRACT

Often, the presence of cracks in manufactured components are detrimental to their overall performance. We develop a workflow and tools in this report using CUBIT and Sierra/SM for generating and modeling crack defects to better understand their impact on such components. To this end, we provide a CUBIT library of various prototypical crack defects embedded in pipes and plates that can be readily used in a wide range of simulations, with specific application to those used in Gas Transfer Systems (GTS). We verify the accuracy of the J-integral post-processing capability in Sierra against solutions available in existing literature for the cracks and geometries of interest within the context of linear elastic fracture mechanics, and describe ongoing efforts to quantify and assess numerical errors. Through this process, we outline overall suggestions and recommendations to the user based on the proposed workflow.

ACKNOWLEDGMENT

We are indebted to members of our full-circle team for their constant engagement and support in this work, specifically Jay Foulk, Mike Veilleux, Yuki Ohashi, Dorian Balch and Andy Stershic. They have provided context for applications of the defect library developed herein, in addition to support brainstorming, troubleshooting and implementing suggestions for improvement of the work. We also acknowledge the funding provided by ASC V&V (portfolio led by Jeff Payne and now Jill Blecke) that has allowed us to complete the work described throughout the report.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In Gas Transfer Systems components, defects and flaws introduced during the manufacturing process can be detrimental to the overall performance and reliability of the part. To better understand and qualify such components, GTS analysts typically assume the presence of defects under a certain size and perform subsequent analyses accordingly. To this end, the capability to compute the J-integral to predict driving forces for such defects using Sierra was previously introduced in [10], but lacked generality and robustness since its initial onset.

In the work proposed here, we attempt to simplify and streamline the workflow needed to run a J-integral simulation for GTS analysts by leveraging recent code development efforts to test, verify and improve L_2-projections and interpolations (frequently termed "L_2 transfer") between meshes in Sierra. With the workflow proposed here, the previous computationally expensive models can also be simplified by performing a mapping procedure, wherein a full model is mapped using an L_2 transfer in Sierra to a smaller subset mesh with significant speedup. Additionally, by performing these mappings, a greater degree of flexibility in choice of element formulation can be attained for other pieces of the analysis (since the J-integral calculations have traditionally only been supported for the uniform gradient hexahedral elements).

We also aim to introduce more generality in the previous approach by developing a templated defect library that can readily be inserted into a variety of common geometries for GTS components, such as plates with elliptical flaws and pipes with penny-shaped, longitudinal, or circumferential cracks to assess performance. This library will enable us to create an additional step (as part of our Full Circle Lifecycle modeling approach) where a flaw might be added along the way, and subsequent analysis performed to assess the performance down the road. To this end, we provide verification of the defects under investigation here assuming linear elastic fracture mechanics to enhance the credibility associated with our Full Circle approach. We note that additional work will be conducted in FY20 to reformulate the J-integral to accurately incorporate the manufacturing history accumulated through a series of steps to obtain the full benefits of the Full Circle approach, which is outside the scope of this report at this point in time.

The remainder of this work is organized as follows: In chapter 2, we provide illustrations and templates of geometries and meshes for the defects under study using CUBIT, including parameters to be edited for generality, which are included in full in appendix A. The results for the studies are described in chapter 3, including verification of the results with analytical solutions or other closed-form polynomial expressions available in prior works. Finally, we conclude with some discussion of the main contributions of our approach and suggestions for future work.

2. GENERATION OF GEOMETRIES AND MESHES

This section contains images of the geometries and meshes produced by the CUBIT journal files in appendix A. Each section focuses on a single geometry/mesh, which will be used to compute the fields and post-processed J-integral in chapter 3. Note that all geometries presented here have certain symmetries (*e.g.* planar symmetry or axi-symmetry) which are assumed when generating the meshes in order to minimize the overall degrees of freedom.

2.1. PLATE WITH EMBEDDED ELLIPTICAL FLAW

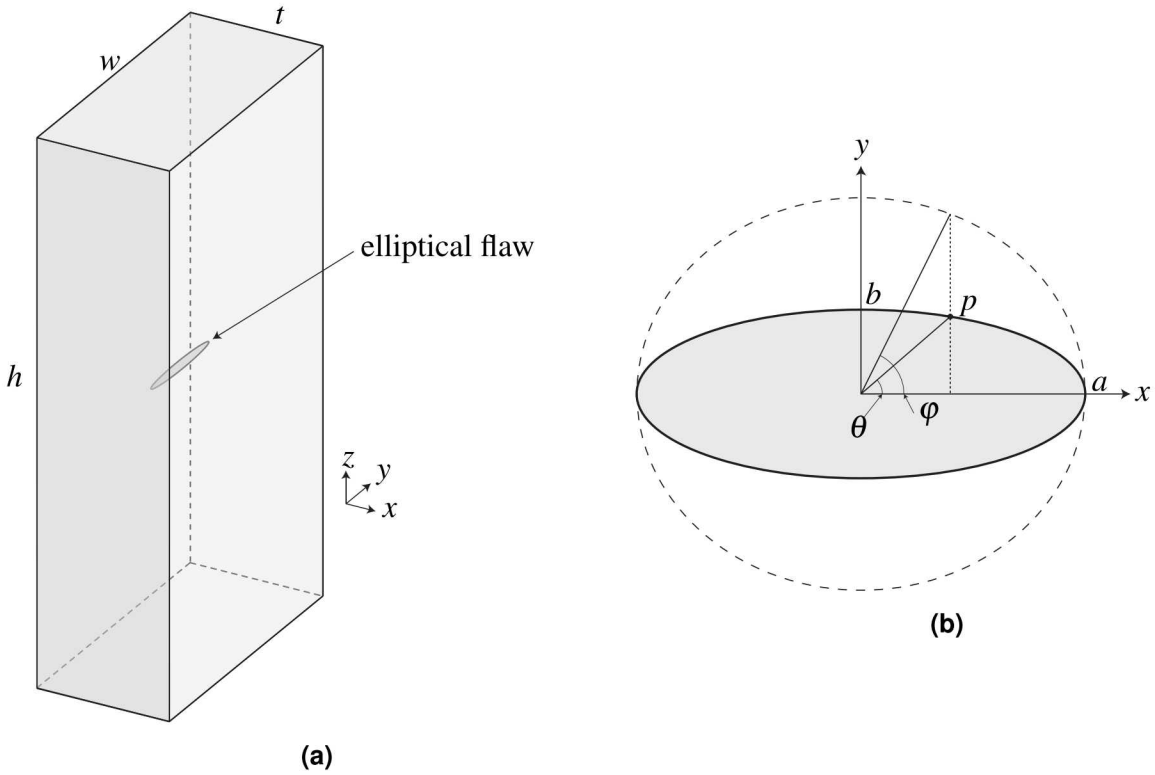

Figure 2-1. Plate with embedded elliptical flaw: Schematic of embedded elliptical flaw with (a) specimen dimensions and (b) flaw dimensions

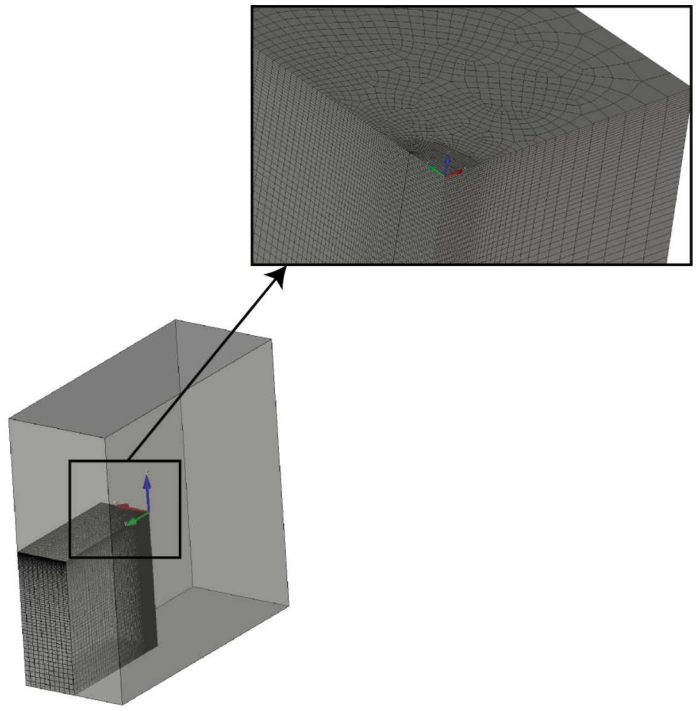

Figure 2-2. Plate with embedded elliptical flaw: Reduced mesh of embedded flaw employing symmetry along the x-, y-, and z-axes

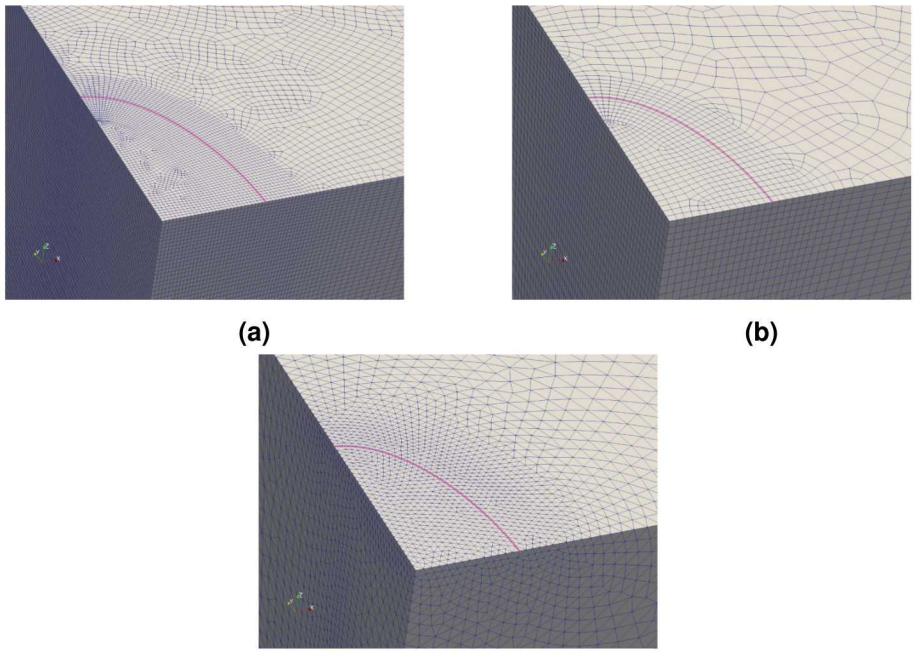

Figure 2-3. Plate with embedded elliptical flaw: Mesh along the crack front for (a) fine hexahedral mesh, (b) coarse hexahedral mesh, and (c) tetrahedral mesh. Nodes and curve on crack front are shown in purple.

2.2. PIPE WITH LONGITUDINAL PENNY-SHAPED CRACK

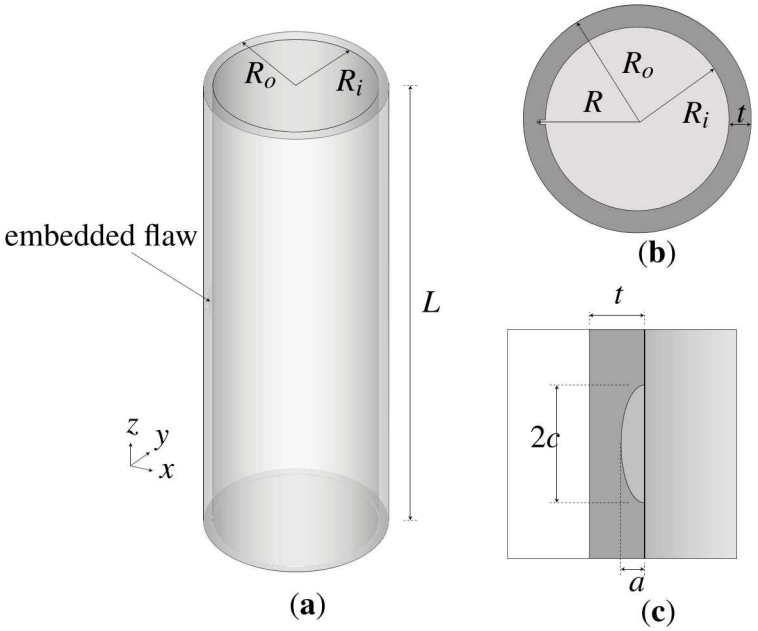

Figure 2-4. Penny-shaped longitudinal crack: Dimension parameters shown for (a) entire cylinder, (b) region near crack, and (c) top view of cylinder

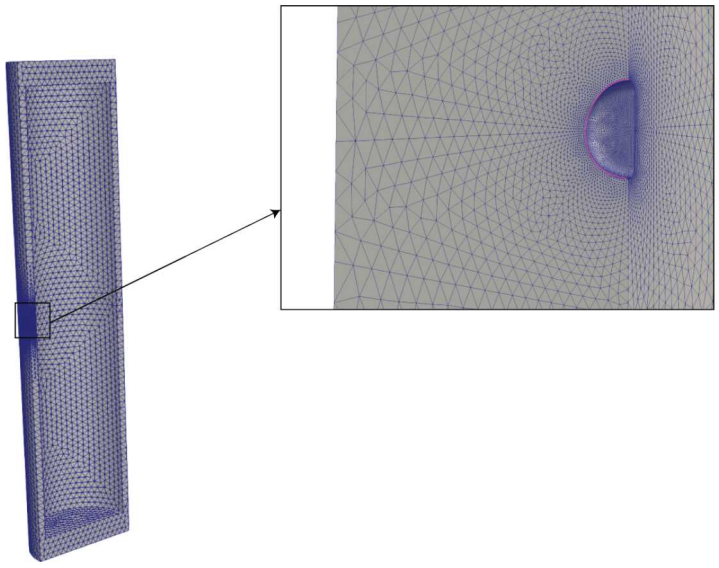

Figure 2-5. Pipe with longitudinal penny-shaped crack: Mesh with crack front curve and nodes shown in purple

2.3. PIPE WITH LONGITUDINAL THROUGH CRACK

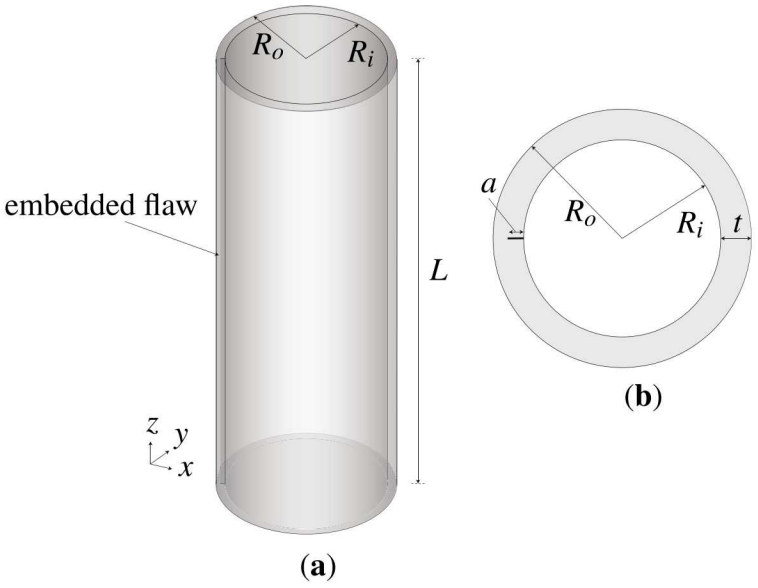

Figure 2-6. Pipe with longitudinal through crack: Dimension parameters shown for (a) entire cylinder, and (b) region near crack

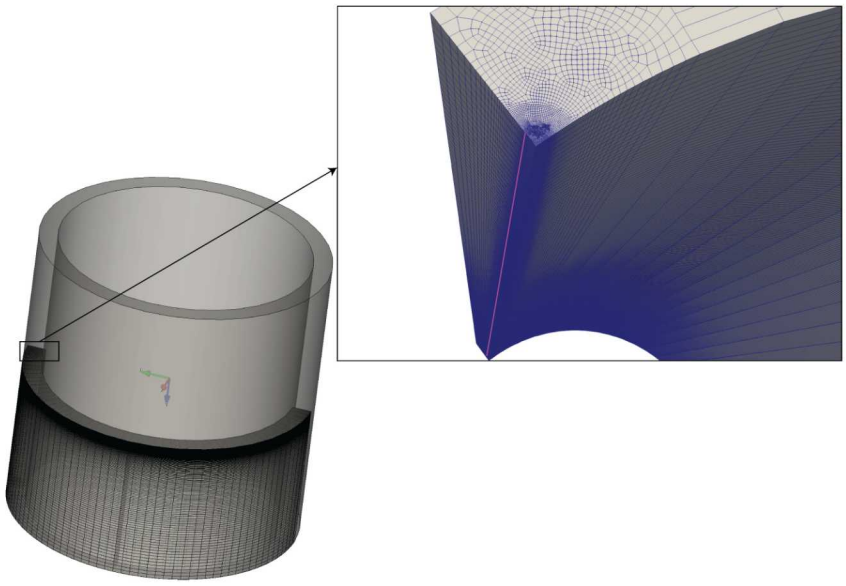

Figure 2-7. Pipe with longitudinal through crack: Mesh with crack front curve and nodes shown in purple

2.4. PIPE WITH CIRCUMFERENTIAL PENNY-SHAPED CRACK

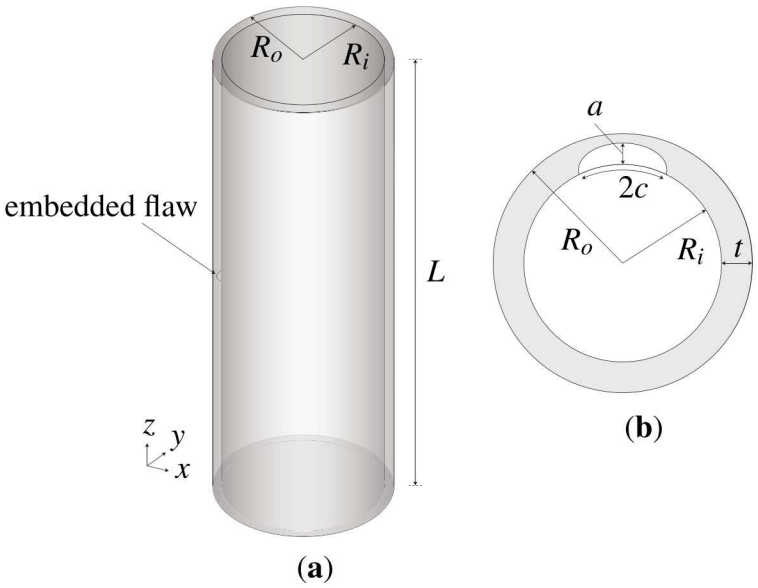

Figure 2-8. Pipe with penny-shaped circumferential crack: Dimension parameters shown for (a) entire cylinder, and (b) region near crack

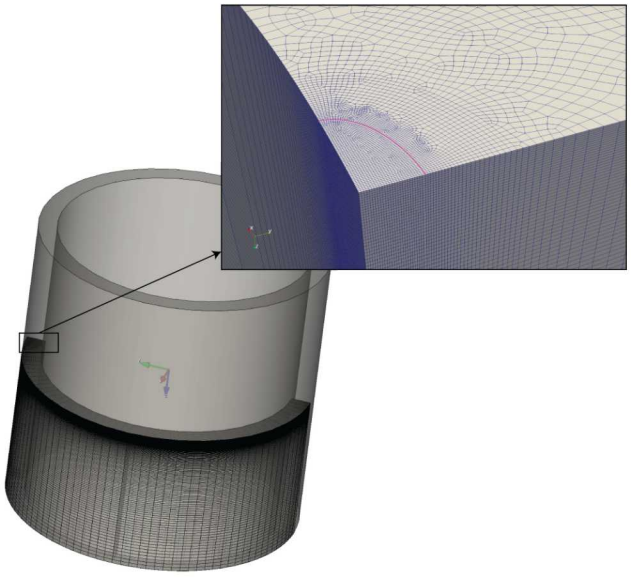

Figure 2-9. Pipe with penny-shaped circumferential crack: Mesh with crack front curve and nodes shown in purple

2.5. PIPE WITH CIRCUMFERENTIAL CRACK

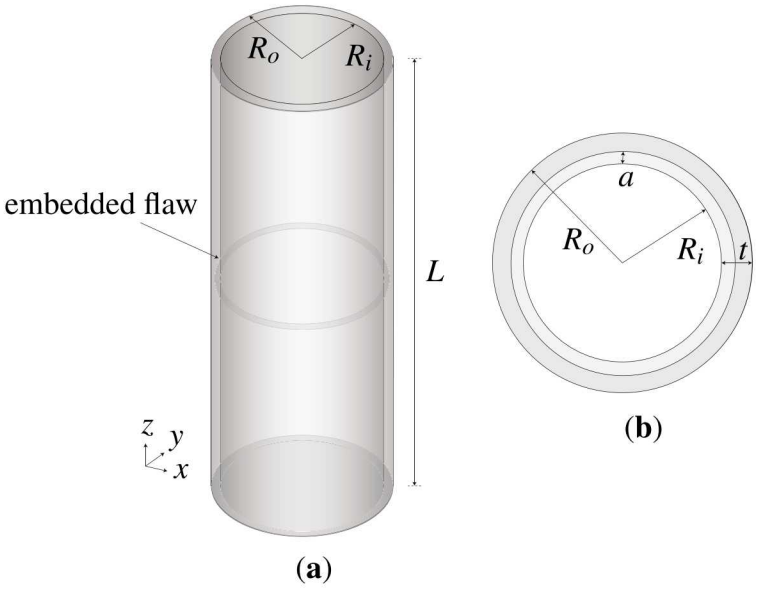

Figure 2-10. Full circumferential crack: Dimension parameters shown for (a) entire cylinder, and (b) region near crack

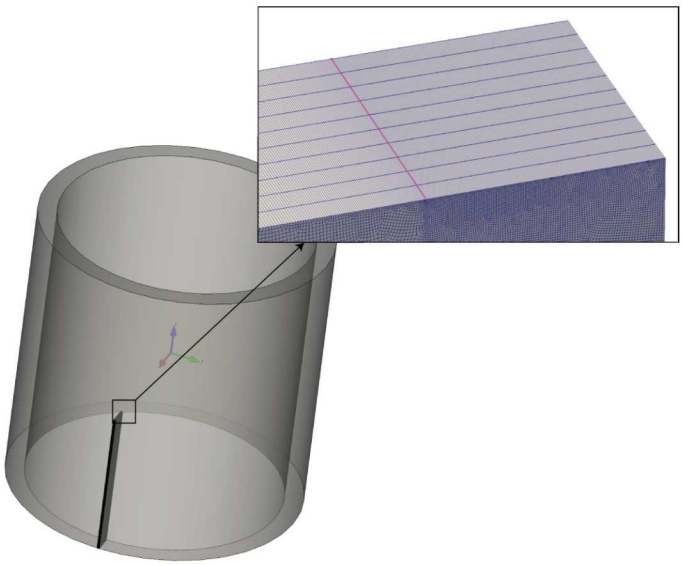

Figure 2-11. Full circumferential crack: Mesh with crack front curve and nodes shown in purple

3. VERIFICATION OF J-INTEGRAL

In this chapter, the distribution of the J-integral computed in Sierra along the crack front is presented for the geometries and configurations from chapter 2. This model problem primarily serves to illustrate the error in the numerically computed J-integral compared to the exact values. For the purpose of verification, it is assumed that the strains remain small for all cases, and that the loads are applied in a quasi-static manner. With this assumption, the material response is characterized within the context of linear elasticity. The elastic constants are assigned as $E = 200GPa$ and $\nu = 0.25$, which are values typical for stainless steel. To serve as a contrast, the example in section 3.2.1 assumes a viscoplastic constitutive law, though strains are assumed to be small enough to represent a linear elastic material response. For simplicity, the configuration and defect geometries remain fixed for all cases. It is assumed that the cracks are "small" (approximately 10% of the wall thickness for thick-walled pipes).

3.1. PLATE WITH EMBEDDED ELLIPTICAL FLAW UNDERGOING INTERNAL PRESSURE

3.1.1. Model Description

The schematic of the plate and embedded elliptical crack is shown in fig. 2-1. The selected parameters for the geometry of the plate and embedded elliptical crack is shown in table 3-2. The elliptical flaw is exposed to a outward-oriented (towards the crack faces) pressure of $100kPa$.

The analytical stress intensity factor is presented in [12] along the crack front of an elliptical flaw embedded in an infinite medium and subjected to a far-field stress along the z-axis, which is

name	symbol	value
width	w	5.0 m
thickness	t	2.0 m
height	h	5.0 m
major radius	a	0.2 m
minor radius	b	0.1 m

Table 3-1. Plate with embedded elliptical flaw: Dimensions of plate and embedded flaw

expressed for a given point p as:

$$K_I^p = \frac{\sigma\sqrt{\pi b}}{E(k)}\left(sin^2(\varphi) + \frac{b^2}{a^2}cos^2(\varphi)\right)^{1/4}, \tag{3.1}$$

where σ is the far-field stress, a and b are the major and minor radii (respectively) such that $a > b$, and φ is the angle between a and the projection of the point p in the direction orthogonal to a onto a circle which is circumscribed around the ellipse, as is illustrated in fig. 2-1b. Moreover, $E(k)$ is the complete elliptical integral of the second kind defined as follows:

$$E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 sin^2(\phi)}\, d\phi, \quad k = 1 - \frac{b^2}{a^2}\,. \tag{3.2}$$

To attribute relevance of eq. (3.1) to the problem-at-hand, the infinite medium is approximated by a plate of finite (albeit sufficiently large) dimensions. As examined in [11] (of particular importance to the current setting are figs. 6c-d), the placement of physical boundaries plays a negligible role in the deviation of the stress intensity factor along the elliptical crack front from its analytically exact counterpart for cases where the ellipse is not very flat (*i.e.* $b/a > 0.4$), and the minor radius dimension is roughly less than 40% of the thickness. In addition, superimposing a far-field stress equal and opposite to the assumed value as well as an inward-oriented pressure on the open crack faces generates a closed-form shown which identical to eq. (3.1) due to the fact that the case where the inward-oriented pressure is applied on the crack faces leads to a stress intensity factor of $K_I = 0$, and the assumption that the material response is linear elastic [1, Chapter 2]. Thus, with the appropriate assumptions, the exact solution in eq. (3.1) can be consistent with the stress intensity factor expected for the geometry and loading shown in fig. 2-1.

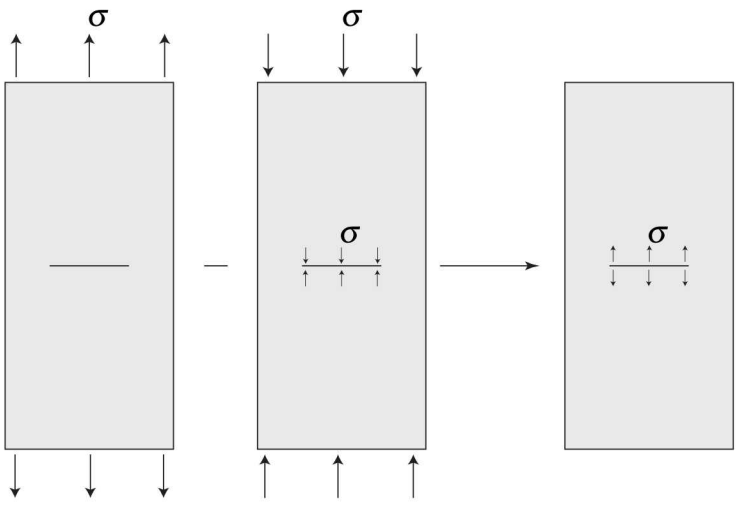

Figure 3-1. Plate with embedded elliptical flaw: Superposition principle illustrating equivalence of eq. (3.1) for far-field and pressure loads

3.1.2. Error in J-integral Due to Mapping

In the first model problem, a meshed elliptical flaw and plate is used to produce the elastic fields in Sierra based on the loading and boundary conditions presented in section 3.1.1. Symmetry is assumed along the x-, y-, and z-axes (as shown in fig. 2-2), hence reducing the total number of elements by a factor of 8. The base simulation is performed for: (a) 1.0e+06 hexahedral uniform gradient elements (baseline case), (b) 1.2e+05 hexahedral selective deviatoric elements, and (c) 6.5e+05 composite tetrahedron elements. The relative element sizes for the three cases are shown along the cracked plane in fig. 2-3.

The relevant fields on the base mesh in cases (a)-(c) are mapped onto a template mesh composed of uniform gradient hexahedra due to the limitation that the J-integral capability in Sierra can only be computed on this element type. The level of element refinement along the crack front is predetermined for all cases to ensure that the density of integration points is roughly preserved between base and template mesh. A single selective deviatoric element (which by definition, exactly integrates bilinear deviatoric fields) consists of 8 integration points; the composite tetrahedron element consists of 4 integration points; and the under-integrated uniform gradient element consists of 1 integration point. To maintain the same density of integration points in the base mesh composed of hexahedra and composite tetrahedra, the element size along the crack front is selected to fit approximately 4 and 8 uniform gradient elements, respectively. Note that the base and template mesh in case (a) coincide, hence the error in the mapped fields are directly a product of the global L_2-projection. This scenario serves as a useful baseline in differentiating projection errors from interpolation errors.

The distribution of the numerically computed J-integral for integration radii ranging from 1.0e-3 m to 6.0e-3 m and its exact counterpart is shown in fig. 3-2 along the crack circumference. The numerically computed J-integral approaches the exact value as the integration radius increases in all cases, which is largely due to the decreasing influence of the plastic zone near the crack tip on the fields enclosed by the semi-circular integration domain. The J-integral in cases (b) and (c) generally exhibits higher errors with a small integration radius due to the increasing effect of the errors in the mapped fields. This effect is more pronounced in case (c) since the base mesh consists of tetrahedra which are unstructured near the crack tip. In this case, the greater misalignment between the mesh composed of composite tetrahedra and the mesh composed of uniform gradients leads to oscillations and a decrease of regularity in the distribution of the fields projected onto the latter mesh, as is illustrated in the comparison of the crack-face normal component of the projected Cauchy stress for both cases (b) and (c) in fig. 3-3. Lastly, the J-integral deviates from the exact value near the surface cutting through the major radius of the ellipse due to the fact that the semi-circular integration domain does not fit within the bounds of the base mesh domain, which emanates from the combined presence of a boundary placed to enforce symmetry as well as the high relative curvature of the crack front near this boundary.

The path independence of the J-integral is achieved to varying degrees when integrating the projected fields in cases (a)-(c), as highlighted by the nearly flat curves in fig. 3-4. The J-integral deviates from the expected path-independent value integrating the projected fields for cases (a)-(c) throughout the crack circumferences, though the differences remain visibly small. The characteristic "kinks" in the J-integral distribution occurs due to the geometric constraints posed

by the curvature of the crack circumference, as elaborated in the prior paragraph. Based on the elliptical geometry of the flaw and the fact that the integration radius at a given point varies normal to the crack front, the distance at which the integration radius intersects with the boundary representing a symmetry plane is expressed as

$$\sqrt{\left[\left(\frac{b^2}{a}\right)cos\theta\right]^2 + \left(b\ sin\theta\right)^2} \tag{3.3}$$

for a given set of major/minor radius a and b, respectively, and angle from the major radius θ. To ensure that the domain of integration used to compute the J-integral does not intersect with the symmetry boundary for all angles , it is sufficient to set that maximum integration radius as

$$min\left(\frac{b^2}{a}, b\right) . \tag{3.4}$$

In scenarios where defects and cracks exist in irregular geometries as is typical for numerous engineered components, it is convenient to compute the J-integral on a simplified sub-region (also termed template geometry) which encloses a finite region around the original defect. The fields on the geometry computed in the primary simulation are transferred via an L_2 projection and interpolated onto the relevant portions of the template geometry where the J-integral procedure takes place. This process is graphically illustrated in fig. 3-5 for the simulations of cases (b) and (c) with a toroidal template geometry. In this case, a quarter of the torus intersects the original plate geometry, and hence, the projected fields are only relevant in this region. The J-integral averaged along the portion of the torus intersecting the original flaw is path-independent for both cases (b) and (c) for integration radii greater than 20% of the flaw radius (designated as r_{max}) with consistent errors on the order of 1% relative to the exact LEFM J-integral, as shown in fig. 3-4[1].

[1] These results were computed on the current version of Sierra/Master due to recent fixes in the J-integral capability that allows for integration along a crack front greater than 180^o. These fixes will be available in the next release (version 4.54).

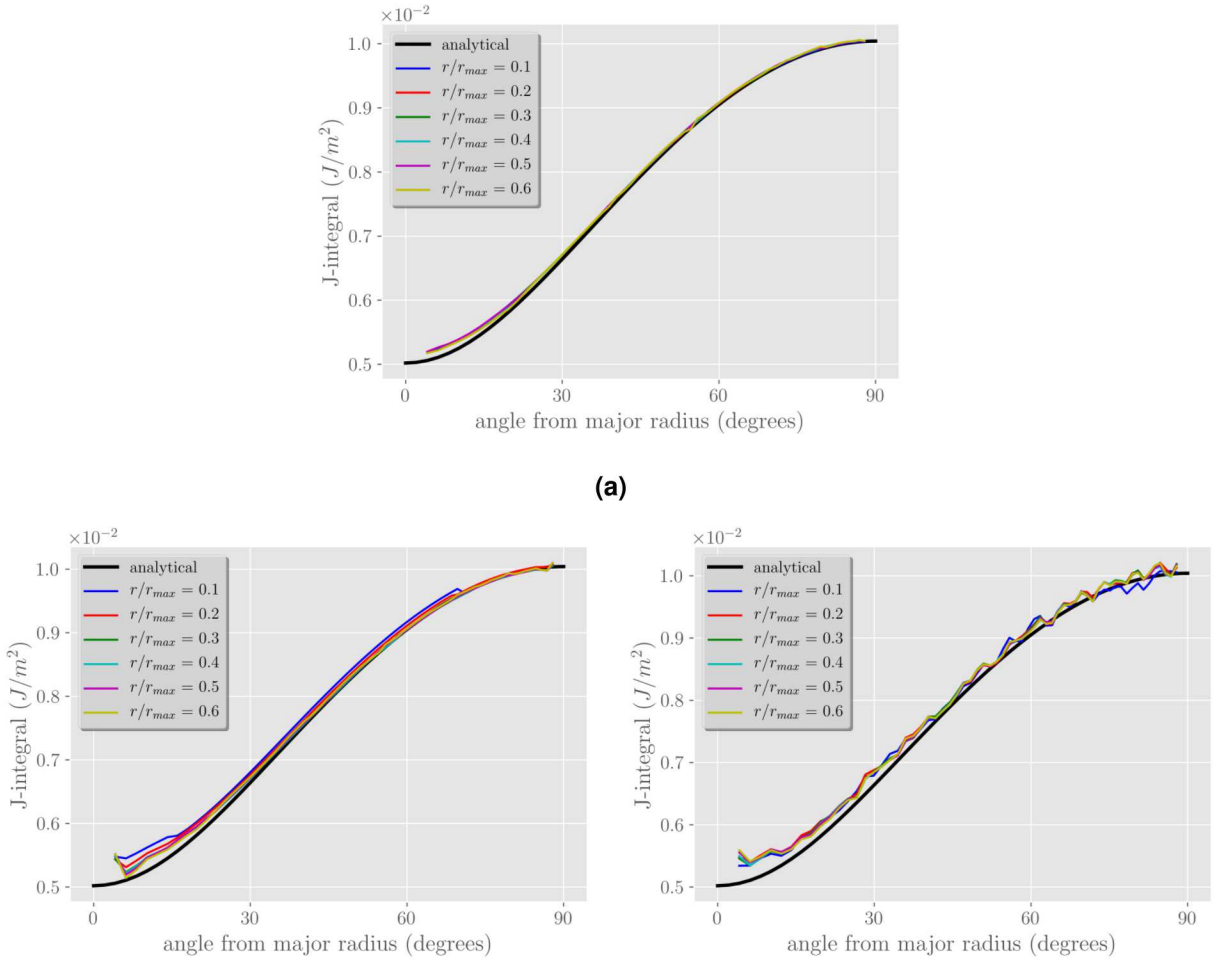

Figure 3-2. Plate with embedded elliptical flaw: J-integral distribution along crack circumference as a function of θ (as defined in fig. 2-1b) for base mesh consisting of (a) uniform gradient hexahedra, (b) selective deviatoric hexahedra, (c) composite tetrahedra. Exact (analytical) distribution is shown in black.

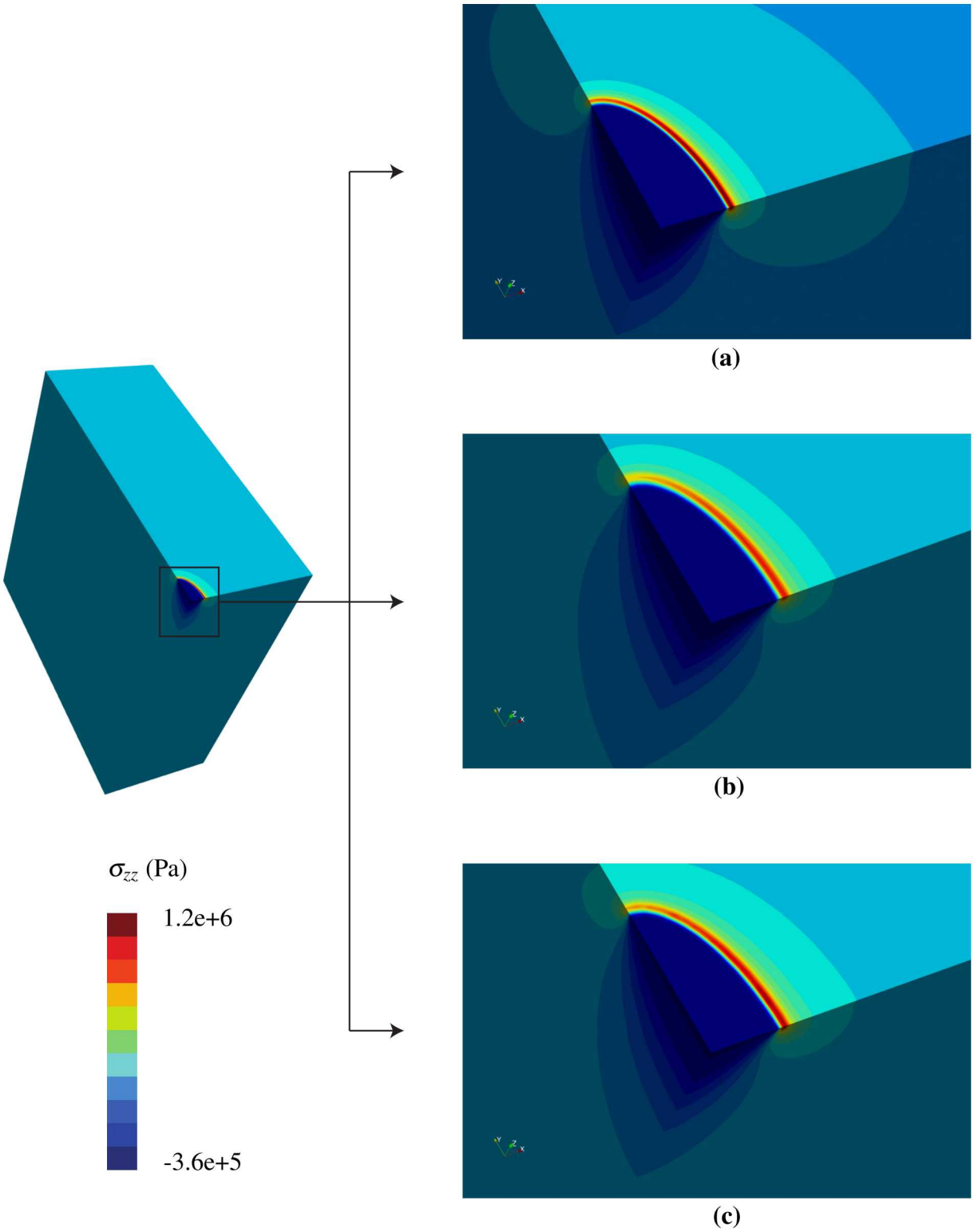

Figure 3-3. Plate with embedded elliptical flaw: ZZ-component of Cauchy stress on template mesh consisting of uniform gradient hexahedra projected from mesh consisting of (a) uniform gradient hexahedra, (b) selective deviatoric hexahedra, and (c) composite tetrahedra

name	symbol	value (m)
width	w	20.0
thickness	t	2.0
height	h	20.0
major radius	a	0.2
minor radius	b	0.1

Table 3-2. Plate with embedded elliptical flaw: Dimensions

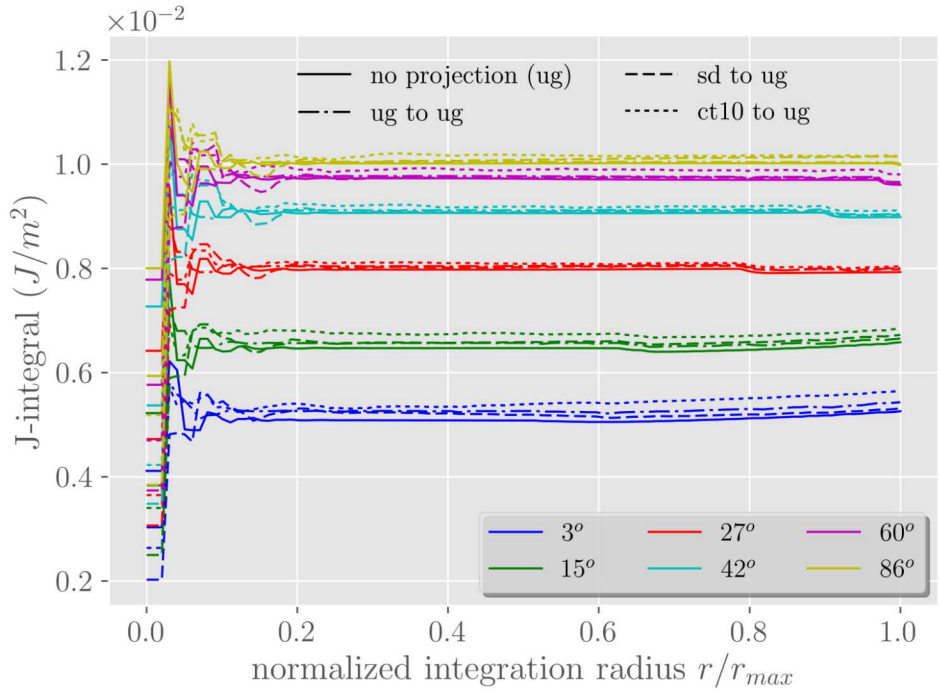

Figure 3-4. Plate with embedded elliptical flaw: J-integral value as a function of integration radius for cases (a)-(c) relative to those computed with no projection on the base mesh

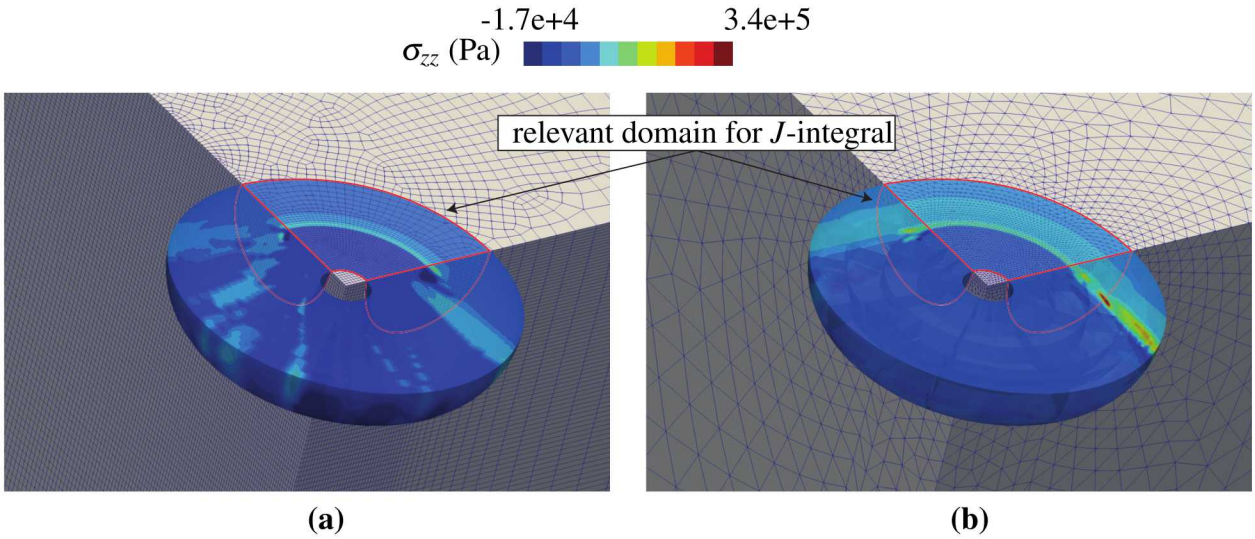

Figure 3-5. Plate with embedded elliptical flaw: ZZ-component of Cauchy stress on toroidal template mesh consisting of uniform gradient hexahedra projected from mesh consisting of (a) selective deviatoric hexahedra, and (b) composite tetrahedra.

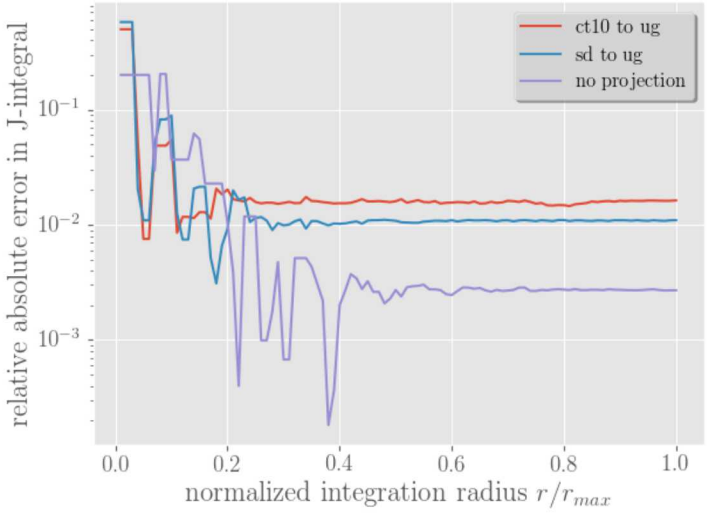

Figure 3-6. Plate with embedded elliptical flaw: Error in circumferentially-averaged J-integral relative to analytical solution based on fields projected onto torus

3.2. THICK-WALLED PIPE WITH LONGITUDINAL CRACK

The work presented in this section is motivated by burst experiments done in [6], which presented experimental data of a circular cylinder loaded internally with pressure. Both ends of the cylinder

were welded, and the top end was kept fixed. In addition, volume change during loading was measured by immersing the cylinder in a water bath. A flaw was machined longitudinally along the entire cylinder in order to control the location of crack initiation/propagation. The test was conducted both hydraulically and pneumatically. In each scenario, the relation between pressure and volume change was compared for an uncharged specimen, a Deuterium-charged specimen, and a Helium-charged specimen. The goal of the current work is to quantify the errors in the numerically computed J-integral relative to the closed-form approximations available in existing literature (namely those presented in [13, 1]) for an internally pressurized cylinder similar to the one used in [6].

3.2.1. Penny-Shaped Crack

The geometry and mesh layout is shown in figs. 2-6 and 2-7. The mesh consists of approximately $183,000$ 10-noded composite tetrahedra for a total of roughly $270,000$ degrees of freedom. The crack and cylinder dimensions are shown in table 3-3. In addition, the top and bottom boundaries of the cylinder are attached to rigid end caps that are each 7.6 *mm* thick. The crack radius was selected as 10% of the wall thickness corresponding to $a/t = 0.1$, essentially making it a "small" crack. The crack profile is semi-circular, therefore $a/c = 1$. The end cap thickness was chosen as an arbitrary large number to provide rigidity at the top and bottom cap-cylinder interfaces as well as on the end caps themselves. Note that similar results should be observed without end caps by fixing the boundaries of the edge of the cylinder.

name	symbol	value (*mm*)
Length	L	76
thickness	t	1.3
crack minor radius	a	1.3
crack major radius	c	1.3
pipe outer radius	R_o	10
pipe inner radius	R_i	8.7

Table 3-3. Pipe with longitudinal penny-shaped crack: Dimensions

The boundary conditions include pressure on all interior surfaces (including the crack face), and fixed displacements on the edges of the top and bottom caps. Additionally, symmetry boundary conditions were applied on the longitudinal surface shown in fig. 2-7. As a side note, additional symmetry could have been implemented on a horizontal plane that cuts through the crack center in fig. 2-7. This was not done intentionally since boundary conditions on the bottom cap are intended to allow longitudinal motion.

The material is assumed to be 304L stainless steel, with the material parameters fit for the dynamic strain aging (DSA) constitutive/plasticity model in Sierra [7, 3]. The selected material parameters are shown in table 3-4. Note that the results presented herein should not be affected by the material behavior due to the assumption of LEFM. As a verification of the material-independent properties of the stress intensity factor (and hence the J-integral within the

LEFM regime), the values of the J-integral were nearly identical when changing the material to neo-Hookean.

parameter name	value	units
Young's modulus	200.0E+09	Pa
Poisson ratio	0.25	-
initial REX volume fraction	1.0E-04	-
flow rule exponent temperature dependence	5.699E+03	K
rate independent yield constant	1.0528E+10	Pa
rate independent yield temperature dependence	2.688E+05	K
rate independent yield temperature dependence 2	1.87E-03	1/K
rate independent yield temperature dependence 3	8.683E+02	K
rate independent yield temperature dependence 4	3.316E+01	-
flow rule coefficient constant	9.178E-02	1/s
isotropic dynamic recovery constant	8.565E+02	-
isotropic dynamic recovery temperature dependence	5.419E+03	K
isotropic hardening shear coefficient	0.01	-
recrystallization kinetics temperature dependence	5.0E+04	K
recrystallization kinetics mobility coefficient	8.846E+16	1/Pa$^{\beta}$
recrystallization kinetics mobility exponent, β	5.431E+00	-
recrystallization kinetics boundary energy dependence	1.1E+16	K/s/Pa^2
recrystallization kinetics multiple cycle correction factor	1.0	-
recrystallization kinetics boundary area exponent 1	0.667	-
recrystallization kinetics boundary area exponent 2	1.333	-
misorientation variable hardening constant	1.67E-03	m/s/pa$^{\gamma}$
misorientation variable hardening exponent, γ	1.0	-
temperature option	0	-
plastic dissipation factor	0.95	-
density for plastic dissipation calculations	8.004E+03	kg / m^3
specific heat for plastic dissipation calculations	6.67E+02	J/kg/K
initial temperature for uncoupled adiabatic heating	294.11	K
temperature for post-processed yield strength	294.4	K
strain rate for post-processed yield strength	1.0E-03	1/s

Table 3-4. Pipe with longitudinal penny-shaped crack: DSA material properties 304L stainless steel

The pressure was ramped linearly from 0 to 1 *MPa* within the span of 10 *s* in a total of 5 time steps. Plasticity dominates the time dependence of the initial/boundary-value problem, however, it is highly localized at the crack tip due to the low applied pressure relative to the material's yield stress. Therefore, the response of the cylinder to the applied pressure is essentially quasi-static, which implies a low sensitivity of the solution to a refinement in the time step. This correlation is confirmed by observing that the L_2-norm of each component of the Cauchy stress is nearly identical (deviations less than 1%) when refining the number of time steps to 20, 50, and 100.

With these observations in mind, it suffices to conclude that the selected time step of 2 s is fine enough to temporally resolve the rate-dependent fields. Though no rigorous convergence properties are shown for mesh refinement (h-refinement) in this report, the values of the J-integral were fairly consistent to within 1% of the values obtained with the coarse mesh.

The second region consisted of an L_2-projection that was performed on the main mesh, and subsequently interpolated onto a template mesh along the crack front shown in fig. 3-7. Since the entire cylinder is difficult to mesh with hexahedral elements, a template model was created by sweeping only along the crack front. This geometry allowed for a simple meshing scheme with hexahedral elements where the J-integral can subsequently be computed. The J-integral was computed on the template mesh shown in fig. 3-7, with approximately 3 and 5-9 hexahedral elements fitting inside each tetrahedron along the crack front and end planes adjacent to the inner radius (respectively), which is roughly sufficient to maintain the density of integration points between the main and template meshes (refer to section 3.1.2 for implementation details of mesh-to-mesh transfers using L_2-projections and interpolations).

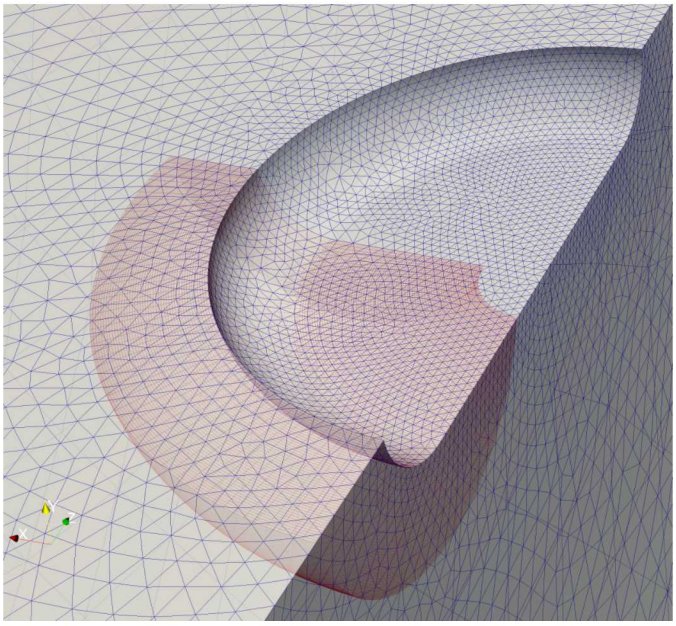

Figure 3-7. Pipe with longitudinal penny-shaped crack: Comparison of the main model mesh using tetrahedral elements (blue), and the swept hexahedral template mesh around the crack tip used for the J-integral computations (red)

The value of the J-integral for various angles from the deepest point in the crack as a function of the integration radius is shown in fig. 3-8. The values of the J-integral (regardless of angle) are steadily increasing until reaching an integration radius of approximately 3.0e-05 m. Without the pressure correction, the J-integral values reach a peak and slowly decreases with increasing integration radius. In contrast, the pressure-corrected J-integral maintains a nearly constant value for all angles after reaching a contour integration radius of approximately 3.0e-05 m. This confirms the expected path-independence of the pressure-corrected J-integral, as was asserted in

[10, 5]. The J-integral used in subsequent computations is assumed to be converged if the numerical value is obtained form the region where it is path-independent, which will be termed path-converged.

The value of the J-integral increases as the angle from the deepest point increases, thus indicating that the highest value is on the crack tip adjacent to the inner wall of the cylinder. In [2, 4, 8], the deviation of the stress intensity factor along the crack front is quantified using a non-dimensional influence function defined as

$$h = \frac{K_I}{\frac{\sigma_0\sqrt{\pi a}}{E(k)}(cos^2\phi + a^2/c^2 sin^2\phi)^{1/4}} \,. \tag{3.5}$$

In the above equation, K_I is the mode-I stress intensity factor, $\sigma_0 = p((R_o/R_i)^2+1)/((R_o/R_i)-1)$ is the effective stress, $E(k)$ is the complete elliptical integral of the second kind which for $k = 0$ (elliptic modulus of a circular crack) is approximately 1.6, and a and c are the through-thickness and longitudinal radii of the crack (as shown in fig. 2-7), respectively, which are equal in this specific case. Note that the angular variation factor that includes the sine and cosine of ϕ does not vary with the angle from the deepest point, ϕ, since the crack is circular. Hence, the only variable that affects the influence function is the change in K_I.

The variation of the influence is plotted for a path-converged J-integral in fig. 3-9 as a function of the angle from the deepest point along the crack. As expected based on the trends shown in fig. 3-8, the J-integral increases nonlinearly from the deepest point of the crack to the end adjacent to the inner wall of the cylinder, with the steepest slope at around 40°- 70°. The trend shown in fig. 3-9 is identical to those illustrated in [2, 4, 8]. However, the values obtained here are approximately 10-15% higher than those obtained in [2, 4, 8]. This can potentially be explained by the fact that the aforementioned prior works do not account for pressure on the crack face, and thus, assume that only the inner wall of the cylinder is loaded. In addition, the crack front here contains a finite notch radius, which could add a potential source of variability in the value of the stress intensity factor[2].

A closed-form expression was obtained in [13] that accounts for pressure on the crack faces, and corrects the J-integral appropriately by using the technique developed in [5]. The stress intensity factor at the deepest point of the crack tip is of the following form

[2]It is well-known that all cracks tips are blunted to some extent, and are thus inherently notch-shaped. The degree of blunting determines the shape of the notch. In the numerical modeling of crack tips, the use of small notches serves an additional purpose of preventing significant mesh distortions when the crack undergoes finite deformations/plasticity. Moreover, the presence of the notch in the simulation geometry prevents stress singularities which would otherwise be present for a linear elastic material with a sharp crack tip. Readers are referred to [9] for more details on the numerical treatment of notches and their effect on accuracy.

$$
\begin{aligned}
K_I &= \frac{pR}{t}\sqrt{\frac{\pi a}{Q}}F \ , \\
F &= 1.12 + 0.053\xi + 0.0055\xi^2 + (1 + 0.02\xi + 0.0191\xi^2)\frac{(20 - \frac{R}{t})^2}{1400} \ , \\
\xi &= \frac{a}{t}\left(\frac{a}{2c}\right) \ , \\
Q &= 1 + 1.464\left(\frac{a}{c}\right)^{1.65} \ .
\end{aligned}
\tag{3.6}
$$

In eq. (3.6), R is the radial distance from the cylinder center to the deepest point of the crack, t is the wall thickness as is defined in fig. 2-7, and p is the internal pressure. The stress intensity factor at the deepest point using the pressure-corrected J-integral in Sierra is 0.111 $MPa\sqrt{m}$ whereas the computed stress intensity factor using eq. (3.6) is $0.109MPa\sqrt{m}$. The current results thus overestimate the stress intensity factor based on eq. (3.6) by approximately 2%, which is well within the 4% bounds provided in [13].

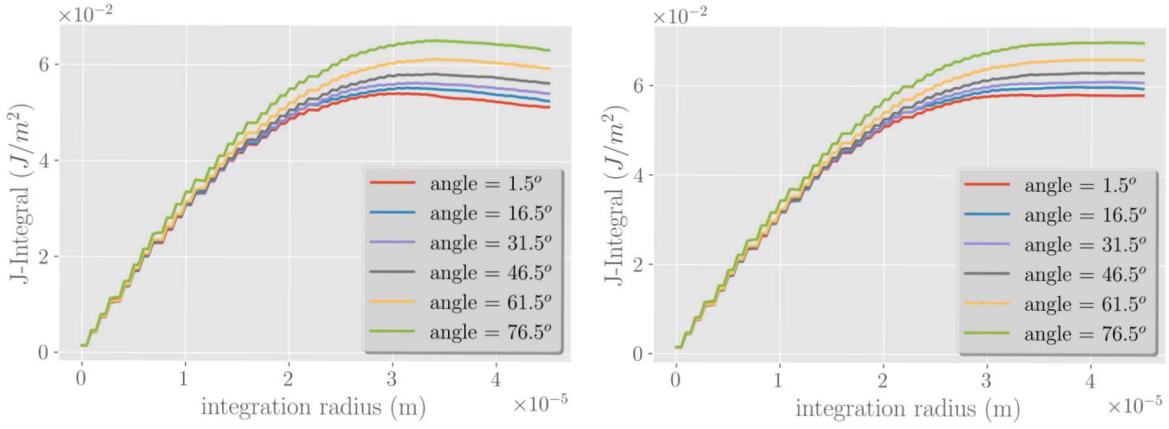

Figure 3-8. Pipe with longitudinal penny-shaped crack: Comparison of the J-integral as a function of the contour integration radius with no correction terms (left) and with the pressure correction (right)

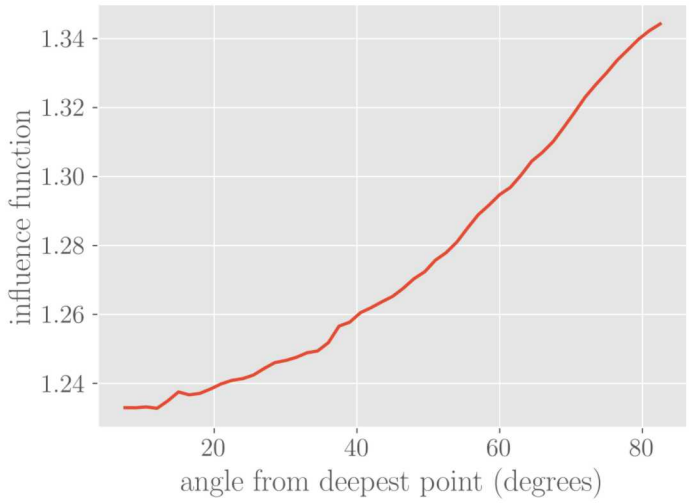

Figure 3-9. Pipe with longitudinal penny-shaped crack: The influence function based on a path-converged path-independent J-integral as a function of the angle from the deepest crack point

3.2.2. Through Crack

The relevant geometry and mesh of a pipe with a sharp crack embedded longitudinally along the entire length of the cylinder are shown in figs. 2-6 and 2-7. The selected dimensions are shown in table 3-5. The mesh consists of approximately $214k$ 4-noded hexahedral elements which leads to approximately $227k$ degrees of freedom per direction. A single integration point is used for each element, which corresponds to the uniform gradient (UG) element in Sierra. The J-integral capability is already supported for UG elements, and hence no additional projections are necessary to perform the relevant computations[3]. The inner surface (including the through crack) are exposed to a $1MPa$ tensile pressure (oriented radially outwards). The top and bottom ends are held fixed in the z-direction, and are allowed to move in the x- and y-directions. With these boundary conditions, The ends of the cylinder are free to expand and contract, thus corresponding to a long (*i.e.* "infinite") cylinder with a flaw embedded along the entire length.

The closed-form expression for the stress intensity factor along the longitudinal flaw according to the work of [13] is

$$K_I = \frac{2pR_o}{R_o^2 - R_i^2}\sqrt{\pi a}F \ , \tag{3.7}$$

[3]It is well-known that UG elements are not robust for many high-deformation and/or plasticity-inducing simulations. In the case of LEFM, we assume that UG elements are sufficient in producing accurate fields despite the fact that there are consistent integration errors.

where p is the pressure and F a the geometry-dependent function defined as

$$F = 1.1 + A\left[4.951\left(\frac{a}{t}\right)^2 + 1.092\left(\frac{a}{t}\right)^4\right], \tag{3.8}$$

$$A = \begin{cases} \left(0.125\dfrac{R_i}{t} - 0.25\right) & \dfrac{R_i}{t} \in [5,10] \\ \left(0.2\dfrac{R_i}{t} - 1.0\right) & \dfrac{R_i}{t} \in (10,20] \\ \text{undefined otherwise} \end{cases}. \tag{3.9}$$

The error in the J-integral computed in Sierra relative to the one based on the the expression for K_I in eq. (3.7) is shown in fig. 3-10. With the assumed geometry/mesh, and loading/boundary conditions mentioned above, the error in the J-integral converges at approximately 3%. Additionally, the error maintains its value for all integration radii greater than 20% of r_{max} thus illustrating the path-independence of the J-integral.

name	symbol	value (mm)
Length	L	20
thickness	t	1.3
crack length	a	0.13
pipe outer radius	R_o	10
pipe inner radius	R_i	8.7

Table 3-5. Pipe with longitudinal through crack: Dimensions

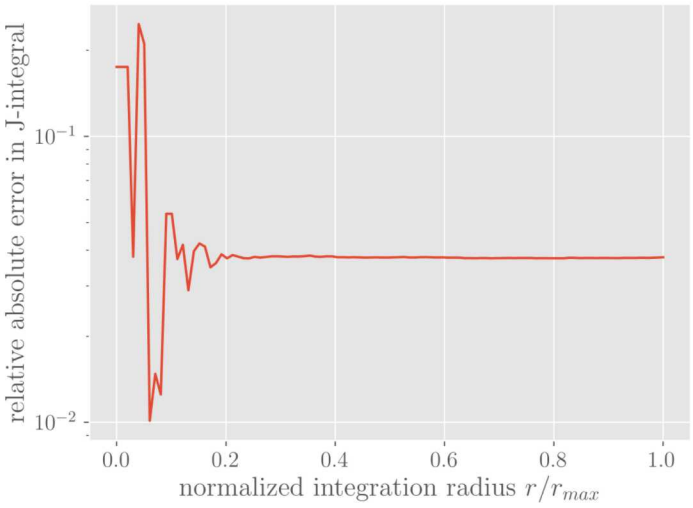

Figure 3-10. Pipe with longitudinal through crack: Error in J-integral as a function of integration radius

3.3. THICK-WALLED PIPE WITH CIRCUMFERENTIAL CRACK

In this section, the error in the J-integral relative to the closed-form values based on the stress intensity factor from [13] is evaluated for a pipe of similar dimensions to the ones presented in section 3.2 with both a penny-shaped and fully axisymmetric crack oriented circumferentially. With this crack orientation, the force that contributes to the mode-I opening of the crack (with the assumption of small deformation) comes from the pressure on the exposed crack faces as well as the pressure on the capped ends of the cylinder which generate additional tensile tractions. Due to the superposition principle described in section 3.1, the net effect of these forces can be captured by simply applying a longitudinal tensile traction at the ends of the cylinder. Therefore, the pressure acting on the interior and end caps is neglected in the following examples.

3.3.1. Penny-Shaped Circumferential Crack

The relevant geometry and mesh for a penny-shaped crack embedded circumferentially inside a pipe is shown in figs. 2-8 and 2-9. The selected dimensions are shown in table 3-6. The mesh consists of $589k$ uniform gradient hexahedral elements which corresponds to $615k$ nodes. The top boundary is loaded to $1MPa$, and symmetry is assumed along the mid-length surface where the embedded crack lies.

The stress intensity factor from [13] for the deepest point along the crack (in the midpoint) is given as

$$K_I = \sigma_t \sqrt{\frac{\pi a}{Q}} F \ . \tag{3.10}$$

In eq. (3.10), σ_t is the tensile stress defined as

$$\sigma_t = \frac{P}{2\pi R_{avg} t} \ , \tag{3.11}$$

where P is the axial load, and $R_{avg} = 1/2(R_o - R_i)$. The geometry-dependent functions Q[4] and F are defined as

$$Q = 1 + 1.464(a/c)^{1.65} \ , \tag{3.12}$$

and

$$F = 1 + \left[0.02 + \alpha\big(0.0103 + 0.00617\alpha\big) + 0.0035\big(1 + 0.7\alpha\big)\big(R_{avg}/t - 5\big)^{0.7} \right] , \tag{3.13}$$

respectively. In eq. (3.13), $\alpha = (a^2)/(2ct)$.

The J-integral error from the values computed in Sierra compared to the one based on the stress intensity factor in eq. (3.10) is shown in fig. 3-11. The error stabilizes at approximately 1% for

[4]Note that [1] defines the function Q in terms of $(a/2c)$ rather than (a/c), which is an error.

integration radii greaer than 20% of r_{max}, indicating that the value of the numerically computed J-integral eventually reaches path-independence as the integration radius increases.

name	symbol	value (mm)
Length	L	20
thickness	t	1.3
crack length	a	0.13
crack width	$2c$	0.507
pipe outer radius	R_o	10
pipe inner radius	R_i	8.7

Table 3-6. Penny-shaped circumferential crack: Dimensions

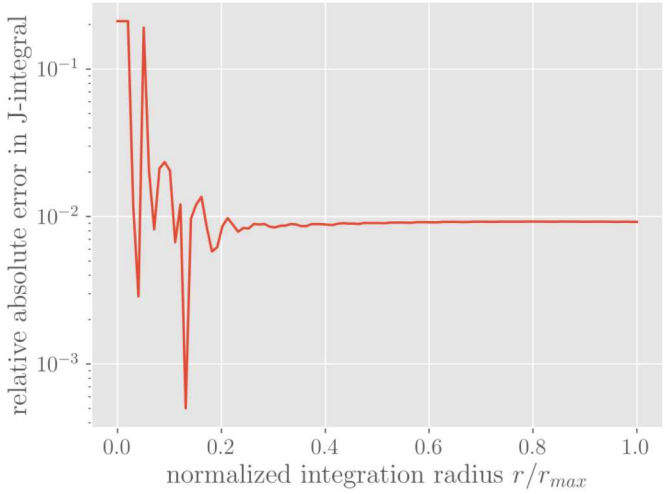

Figure 3-11. Penny-shaped circumferential crack: Error in J-integral as a function of integration radius

3.3.2. Full Circumferential Crack

The geometry and mesh of the pipe with a penny-shaped crack oriented circumferentially is shown in figs. 2-10 and 2-11. The relevant dimensions are shown in table 3-7. Axisymmetry of both flaw and geometry is assumed, this three-dimensional problem can be transformed to two dimensions, being the radial position and length. Since Sierra does not support 2D elements, axisymmetry is imposed on a 1^o cylindrical wedge. The mesh consists of approximately $316k$ hexahedral elements leading to roughly $352k$ degrees of freedom per direction. Uniform gradient elements are used, with the same assumptions elaborated in section 3.2.2. The top end is exposed to a normal tensile traction of $1MPa$, and the side edges are constrained in the normal direction.

The closed-form stress intensity factor from [13] is expressed as

$$K_I = \sigma_t \sqrt{\pi a} F \ . \tag{3.14}$$

In eq. (3.14), σ_t is defined by eq. (3.11). Additionally, the geometry-dependent factor F in eq. (3.14) is defined as

$$F = 1.1 + A\left[1.948\left(\frac{a}{t}\right)^2 1.5 + 0.3342\left(\frac{a}{t}\right)^{4.2}\right], \tag{3.15}$$

where the factor A is defined in eq. (3.9).

The error in the numerically computed J-integral relative to the one based on eq. (3.14) is shown in fig. 3-12, and the ZZ-component of the Cauchy stress is shown along the crack tip in fig. 3-13. The J-integral error converges to approximately 0.2% for integration radii larger than 10% of r_{max}. The flat profile of the error curve shown in fig. 3-12 indicates that the J-integral is path-converged (*i.e.* path independent).

name	symbol	value (mm)
Length	L	20
thickness	t	1.3
crack length	a	0.13
pipe outer radius	R_o	10
pipe inner radius	R_i	8.7

Table 3-7. Full circumferential crack: Dimensions

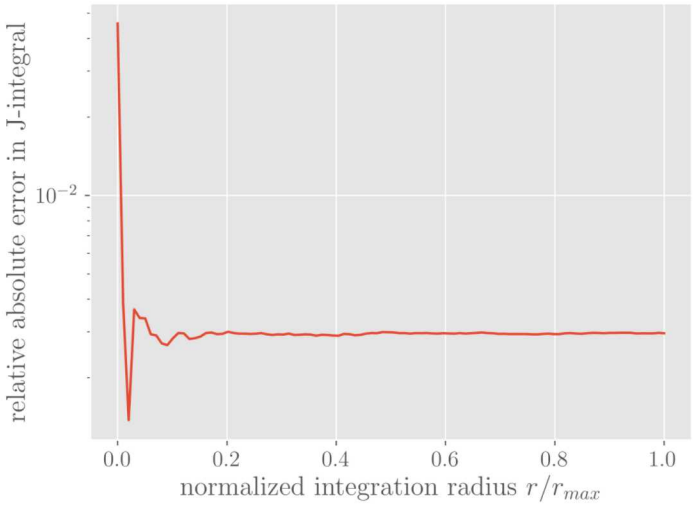

Figure 3-12. Full circumferential crack: Error in J-integral as a function of integration radius

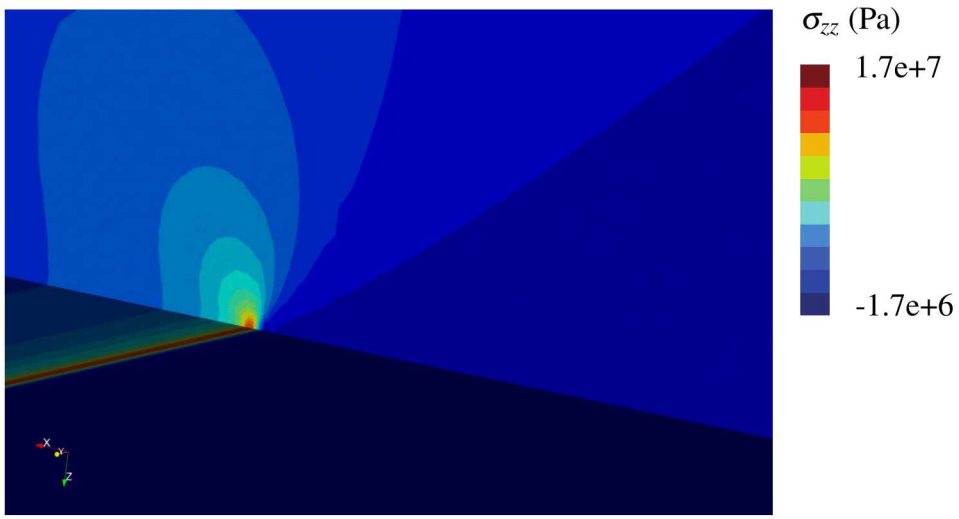

Figure 3-13. Full circumferential crack: ZZ-component of Cauchy stress along crack front

4. CONCLUSIONS

In this work, we attempt to generalize and simplify the work described in [10] by proposing a library of defects to be readily inserted our Full Circle modeling approach for GTS components. We note that in the majority of solutions, the implementation discussed here shows error within 1% of the analytical solution. The main contributions of this work can be summarized as follow:

- generation of a defect library including parameterization of meshes for plates with elliptical flaws and pipes with penny-shaped, longitudinal, or circumferential cracks
- demonstration of mapping full (base mesh) fields to a simplified subregion (template) mesh and computation of associated error
- achievement of path-independence for the projected fields
- comparison of element formulations for varying base meshes (composite tetrahedron, selective deviatoric, uniform gradient) as the integration radius increases
- verification of the driving forces with analytical solutions or closed-form polynomial expressions available in the literature
- addition of verification tests for the J-integral to the Sierra/SM test suite

While significant progress has been made towards improving our J-integral calculation workflow, work must still be conducted (and is already underway by Jay Foulk) to refactor the J-integral formulation to account for the manufacturing history before it can be fully incorporated in our Full Circle models. We plan to leverage the work presented here to verify and test improvements in the refactored formulation in FY20.

BIBLIOGRAPHY

[1] T. L. Anderson. *Fracture Mechanics*. CRC Press, 1991.

[2] S. N. Atluri and K. Kathiresan. Influence of flaw shapes on stress intensity factors for pressure vessel surface flaws and nozzle corner cracks. *Journal of Pressure Vessel Technology*, 102(3):278–286, 08 1980.

[3] Arthur A. Brown and Douglas J. Bammann. Validation of a model for static and dynamic recrystallization in metals. *International Journal of Plasticity*, 32-33:17–35, 2012.

[4] J. Heliot, R. C. Labbens, and A. Pellissier-Tanon. Semi-elliptical cracks in the meridional plane of a cylinder subjected to stress gradients- calculation of stress intensity factors by the boundary integral equations method. *Proceedings of the Eleventh National Symposium on Fracture Mechanics*, 1978.

[5] A. Karlsson and J. Bäcklund. J-integral at loaded crack surfaces. *International Journal of Fracture*, 14(6):R311–R318, 1978.

[6] E.J. Majzlik. Savannah river national laboratory fy2008 annual report. 2009.

[7] K. L. Manktelow and L. L. Behini. Development of a multi-physics capability for predicting residual stress in a gts reservoir. 2016.

[8] J. J. McGowan and M. Raymund. Stress intensity factor solutions for internal longitudinal semi-elliptical surface flaws in a cylinder under arbitrary loadings. *Proceedings of the Eleventh National Symposium on Fracture Mechanics*, 1978.

[9] R. M. McMeeking. Finite deformation analysis of crack-tip opening in elastic-plastic materials and implications for fracture. *Journal of the Mechanics and Physics of Solids*, 25(5):357–381, 1977.

[10] Y. Ohashi, J. W. Foulk III, and A. J. Lindblad. Verification of j-integral capability in sierra mechanics. 2012.

[11] R. C. Shah and A. S. Kobayashi. Stress intensity factors for an elliptical crack approaching the surface of a semi-infinite solid. *International Journal of Fracture*, 9(2):133–146, 1973.

[12] Hiroshi Tada, Paul C. Paris, and George R. Irwin. *The Stress Analysis of Cracks Handbook*. Del Research Corporation, Hellertown, PA, 1973.

[13] A. Zahoor. Closed form expressions for fracture mechanics analysis of cracked pipes. *Journal of Pressure Vessel Technology*, 107(2):203–205, 05 1985.

APPENDIX A. CUBIT SCRIPTS

A.1. PLATE WITH EMBEDDED ELLIPTICAL FLAW

```
#{height = 5.0}
#{length = 5.0}
#{thickness = 2.0}
#{major_radius = 0.1}
#{minor_radius = 0.2}
#{inner_offset_factor = 0.007*6.0}
#{outer_offset_factor = 0.007*6.0}
#{tet_mesh = 1}

# create geometry
create brick x {thickness} y {length} z {height}

create cylinder height {height} major radius {major_radius} minor radius {minor_radius}
create cylinder height {height} major radius {major_radius + outer_offset_factor} minor radius {
    minor_radius + outer_offset_factor}
create cylinder height {height} major radius {major_radius - inner_offset_factor} minor radius {
    minor_radius - inner_offset_factor}

# symmetry
webcut volume all with plane xplane
webcut volume all with plane yplane
webcut volume all with plane zplane

delete volume all except volume with x_coord > 0 and y_coord > 0 and z_coord > 0

chop volume 9 with volume 10
chop volume 34 with volume 11
chop volume 33 with volume 12

########################
# meshing done below
########################

# refinement factor: 1 -- coarse, 2 -- fine
#{refinement_factor = 1.0}
#{crack_mesh_size = 0.007 / refinement_factor }
#{transition_mesh_size = 0.03 / refinement_factor}
#{coarse_mesh_size = 0.1 / refinement_factor}

volume all redistribute nodes on

imprint all
merge all

{if(tet_mesh)}

volume all scheme tetmesh
Set Tetmesher Optimize Level 6 Overconstrained OFF Sliver OFF
Set Tetmesher Interior Points On
Set Tetmesher Boundary ReCovery OFF
Trimesher surface gradation 1.1
Trimesher volume gradation 1.1
```

```
surface 286 291 278 size {crack_mesh_size}

mesh volume all

{else}

surface 278 291 scheme map
surface 278 291 size {crack_mesh_size}
mesh surface 278 291

adjust boundary orthogonal surface 291 snap_to_normal curve 409
adjust boundary orthogonal surface 278 snap_to_normal curve 409 fixed curve 409

surface 284 size {coarse_mesh_size}
curve 443 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 228
curve 442 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 227
surface 284 scheme pave
mesh surface 284

surface 286 scheme pave
surface 286 size {crack_mesh_size}
mesh surface 286

curve 449 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 241
     factor 2.0
curve 447 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 240
     factor 2.0
curve 410 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 218
     factor 2.0
curve 427 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 228
     factor 2.0
curve 310 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 54
     factor 2.0
curve 445 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 239
     factor 2.0
curve 408 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 217
     factor 2.0
curve 425 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 227
     factor 2.0
curve 307 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 18
     factor 2.0
curve 308 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 2
      factor 2.0

volume 37 scheme sweep source surface 286 target surface 288
volume 38 scheme sweep source surface 291 target surface 293
volume 35 scheme sweep source surface 278 target surface 276
volume 36 scheme sweep source surface 284 target surface 281

mesh volume 35 36 37 38

{endif}

imprint all
merge all

####

nodeset 1 surface 284 278
nodeset 1 name "bottom_symmetry_surf"

nodeset 2 surface 285 290 279 283
nodeset 2 name "side_symmetry_surf_yplane"

nodeset 3 surface 282 287 292 275
nodeset 3 name "side_symmetry_surf_xplane"
```

```
nodeset 4 surface 199
nodeset 4 name "side_surf_yplane"

nodeset 5 surface 202
nodeset 5 name "side_surf_xplane"

nodeset 6 surface 281 276 293 288
nodeset 6 name "top_surf"

nodeset 7 curve 409
nodeset 7 name "crack_front"

nodeset 8 vertex 218
nodeset 8 name "crack_front_major_vertex"

nodeset 9 vertex 217
nodeset 9 name "crack_front_minor_vertex"

sideset 1 surface 281 276 293 288
sideset 1 name "top_surf_ss"

sideset 2 surface 278
sideset 2 name "crack_surf"

sideset 3 surface 286 291
sideset 3 name "pressure_surf"

block 1 volume all
block 1 name "plate_embedded_flaw"

{if(tet_mesh)}
block 1 element type tetra10
{endif}
```

A.2. PIPE WITH LONGITUDINAL PENNY-SHAPED CRACK

```
# units: meters
#{height = 0.076}
#{outer_diameter = 0.02}
#{inner_diameter = 0.0174}
#{crack_diameter = 0.025 * (outer_diameter - inner_diameter)}
#{crack_offset_factor = 0.1}
#{cap_thickness = 0.01 * height}
#{crack_longitudinal_radius = 0.8 * height}
#{cap_width = 0.05 * height}
#{scale_factor = 1.0e3}

# create hollow cylinder
create cylinder height {height} radius {outer_diameter/2.0}
webcut volume 1 with cylinder radius {inner_diameter/2.0} axis z

delete volume 2

webcut volume all with plane zplane
delete volume with z_coord > 0

# create crack
#{crack_center = inner_diameter/2.0 + crack_offset_factor * (outer_diameter - inner_diameter)/2.0
     - crack_diameter/2.0}
#{penny_radius = crack_center - inner_diameter/2.0}
create cylinder height {crack_diameter} radius {penny_radius}
rotate volume 4 about x angle 90
move volume 4 x {crack_center - penny_radius} y 0 z 0
```

```
chop volume 3 with volume 4
delete volume 5

create torus major radius {penny_radius} minor radius {crack_diameter*0.5}
rotate volume 7 about x angle 90
move volume 7 x {crack_center - penny_radius} y 0 z 0
chop volume 6 with volume 7
delete volume 8

# create caps
create cylinder height {cap_width} radius {outer_diameter/2.0}
move volume 10 z {- height/2.0 - cap_width/2.0}

webcut volume all with plane yplane
delete volume with y_coord > 0

webcut volume all with plane zplane offset {-penny_radius * 5}
webcut volume 9 with plane yplane rotate -1 about z

unite volume 9 13

########################
# meshing done below
########################

#{crack_mesh_size = crack_diameter/16.0 * scale_factor}
#{transition_mesh_size = (outer_diameter-inner_diameter)/100.0 * scale_factor}
#{coarse_mesh_size = (outer_diameter-inner_diameter)/20.0 * scale_factor}

# scaling up the body helps with merge tolerances
body all scale {scale_factor} {scale_factor} {scale_factor}

imprint all
merge all

volume all scheme tetmesh
Set Tetmesher Optimize Level 6 Overconstrained OFF Sliver OFF
Set Tetmesher Interior Points On
Set Tetmesher Boundary Recovery OFF
Trimesher surface gradation 1.1
Trimesher volume gradation 1.1

surface 81 size {coarse_mesh_size}
surface 25 44 size {crack_mesh_size}
curve 25 56 size {crack_mesh_size}
curve 51 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size} start vertex 40
mesh volume 14

mesh volume 9 10

volume all copy reflect z

imprint all
merge all

body all scale {1.0 / scale_factor} {1.0 / scale_factor} {1.0 / scale_factor}

sideset 1 surface 90 74 84 94 80 99 25 101 44 100

sideset 1 name "inner_radius_ss"

sideset 3 surface 44 100
sideset 3 name "crack_face_notch"

sideset 4 surface 25 101
sideset 4 name "crack_face"

sideset 2 surface 65 98
```

```
sideset 2 name "crack_surf"

nodeset 1 surface 65 98 69 86 91 63 54 93
nodeset 1 name "y_ns_symmetry"

nodeset 2 surface 96
nodeset 2 name "z_ns_top_cap"

nodeset 3 curve 50 149
nodeset 3 name "crack_front"

nodeset 4 surface 57
nodeset 4 name "z_ns_bottom_cap"

block 1 volume all
block 1 name "pipe_notched_flaw"

block 1 element type tetra10
```

A.3. PIPE WITH LONGITUDINAL THROUGH CRACK

```
# units: meters
#{height = 0.07935}
#{outer_diameter = 0.052}
#{inner_diameter = 0.0445}
#{height = 0.02}
#{outer_diameter = 0.02}
#{inner_diameter = 0.0174}
#{thickness = 0.5 * (outer_diameter - inner_diameter)}
#{crack_offset_factor = 0.1}
#{crack_mesh_inner_offset_factor = 0.005}
#{crack_mesh_outer_offset_factor = 0.005 * 30}
#{cap_thickness = 0.01 * height}
#{crack_ellipticity_factor = 0.0}
#{crack_depth_radius = crack_offset_factor * thickness}
#{crack_circumferential_radius = crack_ellipticity_factor + crack_depth_radius}
#{inner_crack_depth_radius = (crack_offset_factor - crack_mesh_inner_offset_factor) * thickness}
#{outer_crack_depth_radius = (crack_offset_factor + crack_mesh_outer_offset_factor) * thickness}
#{inner_crack_circumferential_radius = - (crack_mesh_inner_offset_factor) * thickness +
    crack_circumferential_radius}
#{outer_crack_circumferential_radius = (crack_mesh_outer_offset_factor) * thickness +
    crack_circumferential_radius}
#{cap_width = 0.05 * height}
#{scale_factor = 1.0e3}
#{delta = 0.5 * inner_diameter - sqrt( (0.5*inner_diameter)^2 - (crack_circumferential_radius)^2)
    }
#{delta_inner = 0.5 * inner_diameter - sqrt( (0.5*inner_diameter)^2 - (
    inner_crack_circumferential_radius)^2)}
#{delta_outer = 0.5 * outer_diameter - sqrt( (0.5*outer_diameter)^2 - (
    outer_crack_circumferential_radius)^2)}

# create hollow cylinder
create cylinder height {height} radius {outer_diameter/2.0}
webcut volume 1 with cylinder radius {inner_diameter/2.0} axis z

delete volume 2

webcut volume all with plane zplane
delete volume with z_coord < 0

# create crack
#{crack_center = inner_diameter/2.0 + crack_depth_radius}
```

```
create cylinder height {height} major radius {crack_circumferential_radius} minor radius {
    crack_depth_radius + delta}
create cylinder height {height} major radius {inner_crack_circumferential_radius} minor radius {
    inner_crack_depth_radius + delta_inner}
create cylinder height {height} major radius {outer_crack_circumferential_radius} minor radius {
    outer_crack_depth_radius + delta_outer}
move volume 4 x 0 y {inner_diameter/2.0 - delta} z 0
move volume 5 x 0 y {inner_diameter/2.0 - delta_inner} z 0
move volume 6 x 0 y {inner_diameter/2.0 - delta_outer} z 0
chop volume 1 with volume 5
chop volume 8 with volume 4
chop volume 10 with volume 6

webcut volume all with plane xplane
delete volume with x_coord < 0
webcut volume all with plane xplane rotate -10 about z
webcut volume all with plane yplane

########################
# meshing done below
########################

#{crack_mesh_size = (outer_diameter-inner_diameter)/1600.0 * 3 * scale_factor}
#{transition_mesh_size = (outer_diameter-inner_diameter)/200.0 * scale_factor}
#{coarse_mesh_size = (outer_diameter-inner_diameter)/20.0 * scale_factor}

volume all redistribute nodes on

# scaling up the body helps with merge tolerances
body all scale {scale_factor} {scale_factor} {scale_factor}

imprint all
merge all

curve 88 size {crack_mesh_size}
curve 100 scheme bias coarse size {transition_mesh_size} fine size {crack_mesh_size} start vertex
     58
curve 54 scheme bias coarse size {transition_mesh_size} fine size {crack_mesh_size} start vertex
    25
surface 77 size {transition_mesh_size}
curve 104 size {transition_mesh_size}
mesh surface 77

surface 67 scheme map
surface 67 size {crack_mesh_size}
mesh surface 67

adjust boundary surface 67 snap_to_normal curve 88

surface 55 scheme pave
surface 55 size {crack_mesh_size}

mesh surface 55

surface 106 size {coarse_mesh_size}
curve 120 scheme bias fine size {transition_mesh_size} coarse size {coarse_mesh_size} factor 1.5
    start vertex 66
curve 156 scheme bias fine size {transition_mesh_size} coarse size {coarse_mesh_size} factor 1.5
    start vertex 37
mesh surface 106
mesh surface 114 110

curve 117 size {coarse_mesh_size}
curve 97  size {coarse_mesh_size}
curve 81  size {coarse_mesh_size}
curve 67  size {coarse_mesh_size}
curve 65  size {coarse_mesh_size}
curve 144 size {coarse_mesh_size}
```

```
curve 19  size {coarse_mesh_size}
curve 29  size {coarse_mesh_size}
curve 47  size {coarse_mesh_size}
curve 142 size {coarse_mesh_size}
curve 113 size {coarse_mesh_size}
curve 115 size {coarse_mesh_size}
curve 158 size {coarse_mesh_size}
curve 160 size {coarse_mesh_size}

volume 7 scheme sweep source surface 55 target surface 57
volume 17 scheme sweep source surface 106 target surface 104
volume 9 scheme sweep source surface 67 target surface 65
volume 11 scheme sweep source surface 77 target surface 75

volume 18 12 scheme map

mesh volume 7
mesh volume 17
mesh volume 9
mesh volume 11
mesh volume 12
mesh volume 18

nodeset 1 surface 106 77 67 55 114 110
nodeset 1 name "top_surf"

nodeset 2 surface 84 73 83
nodeset 2 name "side_symmetry_surf"

nodeset 3 surface 104 115 108 75 57
nodeset 3 name "bottom_surf"

nodeset 4 curve 81
nodeset 4 name "crack_front"

nodeset 5 curve 159
nodeset 5 name "line_boundary"

sideset 1 surface 111 113 105 45 56 35 53 63
sideset 1 name "pressure_surf"

sideset 2 surface 106 77 67 55 114 110
sideset 2 name "top_surf_ss"

sideset 4 surface 63
sideset 4 name "crack_surf"

block 1 volume all
block 1 name "pipe_longitudinal_through_crack"

body all scale {1.0/scale_factor} {1.0/scale_factor} {1.0/scale_factor}
```

A.4. PIPE WITH CIRCUMFERENTIAL PENNY-SHAPED CRACK

```
#{height = 0.07935}
#{outer_diameter = 0.052}
#{inner_diameter = 0.0445}
#{height = 0.02}
#{outer_diameter = 0.02}
#{inner_diameter = 0.0174}
#{thickness = 0.5 * (outer_diameter - inner_diameter)}
#{crack_offset_factor = 0.1}
#{crack_mesh_inner_offset_factor = 0.005}
```

```
#{crack_mesh_outer_offset_factor = 0.005 * 30}
#{cap_thickness = 0.01 * height}
#{crack_ellipticity_factor = 0.00013}
#{crack_depth_radius = crack_offset_factor * thickness}
#{crack_circumferential_radius = crack_ellipticity_factor + crack_depth_radius}
#{inner_crack_depth_radius = (crack_offset_factor - crack_mesh_inner_offset_factor) * thickness}
#{outer_crack_depth_radius = (crack_offset_factor + crack_mesh_outer_offset_factor) * thickness}
#{inner_crack_circumferential_radius = - (crack_mesh_inner_offset_factor) * thickness +
    crack_circumferential_radius}
#{outer_crack_circumferential_radius = (crack_mesh_outer_offset_factor) * thickness +
    crack_circumferential_radius}
#{cap_width = 0.05 * height}
#{scale_factor = 1.0e3}
#{delta = 0.5 * inner_diameter - sqrt( (0.5*inner_diameter)^2 - (crack_circumferential_radius)^2)
    }
#{delta_inner = 0.5 * inner_diameter - sqrt( (0.5*inner_diameter)^2 - (
    inner_crack_circumferential_radius)^2)}
#{delta_outer = 0.5 * outer_diameter - sqrt( (0.5*outer_diameter)^2 - (
    outer_crack_circumferential_radius)^2)}

# create hollow cylinder
create cylinder height {height} radius {outer_diameter/2.0}
webcut volume 1 with cylinder radius {inner_diameter/2.0} axis z

delete volume 2

webcut volume all with plane zplane
delete volume with z_coord < 0

# create crack
#{crack_center = inner_diameter/2.0 + crack_depth_radius}
create cylinder height {height} major radius {crack_circumferential_radius} minor radius {
    crack_depth_radius + delta}
create cylinder height {height} major radius {inner_crack_circumferential_radius} minor radius {
    inner_crack_depth_radius + delta_inner}
create cylinder height {height} major radius {outer_crack_circumferential_radius} minor radius {
    outer_crack_depth_radius + delta_outer}
move volume 4 x 0 y {inner_diameter/2.0 - delta} z 0
move volume 5 x 0 y {inner_diameter/2.0 - delta_inner} z 0
move volume 6 x 0 y {inner_diameter/2.0 - delta_outer} z 0
chop volume 1 with volume 5
chop volume 8 with volume 4
chop volume 10 with volume 6

webcut volume all with plane xplane
delete volume with x_coord < 0
webcut volume all with plane xplane rotate -10 about z
webcut volume all with plane yplane

########################
# meshing done below
########################

#{crack_mesh_size = (outer_diameter-inner_diameter)/1600.0 * 3 * scale_factor}
#{transition_mesh_size = (outer_diameter-inner_diameter)/200.0 * scale_factor}
#{coarse_mesh_size = (outer_diameter-inner_diameter)/20.0 * scale_factor}

volume all redistribute nodes on

# scaling up the body helps with merge tolerances
body all scale {scale_factor} {scale_factor} {scale_factor}

imprint all
merge all

curve 88 size {crack_mesh_size}
curve 100 scheme bias coarse size {transition_mesh_size} fine size {crack_mesh_size} start vertex
     58
```

```
curve 54 scheme bias coarse size {transition_mesh_size} fine size {crack_mesh_size} start vertex
    25
surface 77 size {transition_mesh_size}
curve 104 size {transition_mesh_size}
mesh surface 77

surface 67 scheme map
surface 67 size {crack_mesh_size}
mesh surface 67

adjust boundary surface 67 snap_to_normal curve 88

surface 55 scheme pave
surface 55 size {crack_mesh_size}

mesh surface 55

surface 106 size {coarse_mesh_size}
curve 120 scheme bias fine size {transition_mesh_size} coarse size {coarse_mesh_size} factor 1.5
    start vertex 66
curve 156 scheme bias fine size {transition_mesh_size} coarse size {coarse_mesh_size} factor 1.5
    start vertex 37
mesh surface 106
mesh surface 114 110

curve 117 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 77
curve 97 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 65
curve 81 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 57
curve 67 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 52
curve 65 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 50
curve 144 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 90
curve 19 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 20
curve 29 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 26
curve 47 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 38
curve 142 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 89
curve 113 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 73
curve 115 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 75
curve 158 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 99
curve 160 scheme bias fine size {crack_mesh_size} coarse size {coarse_mesh_size*2.0} factor 2.0
    start vertex 100

volume 7 scheme sweep source surface 55 target surface 57
volume 17 scheme sweep source surface 106 target surface 104
volume 9 scheme sweep source surface 67 target surface 65
volume 11 scheme sweep source surface 77 target surface 75

volume 18 12 scheme map

mesh volume 7
mesh volume 17
mesh volume 9
mesh volume 11
mesh volume 12
mesh volume 18
```

```
nodeset 1 surface 106 77 67 55 114 110
nodeset 1 name "top_symmetry_surf"

nodeset 2 surface 84 73 63 53 83
nodeset 2 name "side_symmetry_surf"

nodeset 3 surface 104 115 108 75
nodeset 3 name "bottom_symmetry_surf"

nodeset 4 curve 87
nodeset 4 name "crack_front"

nodeset 5 curve 159
nodeset 5 name "line_boundary"

sideset 1 surface 111 113 105 45 56 35
sideset 1 name "pressure_surf"

sideset 2 surface 106 77 67 55 114 110
sideset 2 name "top_surf"

sideset 4 surface 65
sideset 4 name "crack_surf"

block 1 volume all
block 1 name "pipe_circumferential_penny_crack"

body all scale {1.0/scale_factor} {1.0/scale_factor} {1.0/scale_factor}
```

A.5. PIPE WITH CIRCUMFERENTIAL CRACK

A.5.1. Sharp Crack

```
#{height = 0.02}
#{outer_diameter = 0.02}
#{inner_diameter = 0.0174}
#{webcut_factor = 0.025 * (outer_diameter - inner_diameter)}
#{crack_offset_factor = 0.1}
#{cap_thickness = 0.01 * height}
#{crack_longitudinal_radius = 0.8 * height}
#{cap_width = 0.05 * height}
#{scale_factor = 1.0e3}
#{angle=1}

# create hollow cylinder
create cylinder height {height} radius {outer_diameter/2.0}
webcut volume 1 with cylinder radius {inner_diameter/2.0} axis z

delete volume 2

webcut volume all with plane zplane
delete volume with z_coord > 0

webcut volume all with yplane
webcut volume all with yplane rotate {angle} about z

delete volume 3 5 6

# create crack
# {crack_center = inner_diameter/2.0 + crack_offset_factor * (outer_diameter - inner_diameter)
    /2.0}
```

```
webcut volume 4 plane xplane offset {crack_center}

webcut volume all with zplane offset {-webcut_factor*10}
webcut volume all with zplane offset {-webcut_factor*40}

unite volume 8 9
unite volume 10 11

########################
# meshing done below
########################

#{crack_mesh_size = 0.00065}
#{transition_mesh_size = (outer_diameter-inner_diameter)/200.0 * scale_factor}
#{coarse_mesh_size = (outer_diameter-inner_diameter)/80.0 * scale_factor}
#{outer_mesh_size = (outer_diameter-inner_diameter)/20.0 * scale_factor}

# scaling up the body helps with merge tolerances
body all scale {scale_factor} {scale_factor} {scale_factor}

imprint all
merge all

surface 72 scheme pave
surface 60 scheme pave
surface 79 scheme pave

surface 72 size {transition_mesh_size}
surface 60 size {transition_mesh_size}
surface 79 size {coarse_mesh_size}
surface 87 size {outer_mesh_size}

curve 79 size {crack_mesh_size}
curve 89 scheme bias fine size {crack_mesh_size} coarse size {transition_mesh_size} start vertex
    40
curve 74 scheme bias fine size {crack_mesh_size} coarse size {transition_mesh_size} start vertex
    40
curve 104 scheme bias fine size {crack_mesh_size} coarse size {transition_mesh_size} start vertex
     31

mesh surface 72
mesh surface 60

surface 79 size {coarse_mesh_size}
surface 87 size {outer_mesh_size}

mesh surface 79
mesh surface 87

volume all redistribute nodes on

curve 70 interval 10
volume 4 scheme sweep source surface 60 target surface 62 autosmooth target on
volume 7 scheme sweep source surface 72 target surface 70 autosmooth target on

mesh volume 7
mesh volume 4
mesh volume 8
mesh volume 10

body all scale {1.0 / scale_factor} {1.0 / scale_factor} {1.0 / scale_factor}

nodeset 1 surface 52
nodeset 1 name "bottom_symmetry_surf"

nodeset 2 surface 87 79 72 60
nodeset 2 name "axisymmetry_surf"
```

```
nodeset 3 surface 94 92 69
nodeset 3 name "inner_wall"

nodeset 4 surface 84 82 59
nodeset 4 name "outer_wall"

nodeset 5 surface 85 81 62 70
nodeset 5 name "side_symmetry_surf"

nodeset 6 curve 70
nodeset 6 name "crack_front"

nodeset 8 vertex 40
nodeset 8 name "crack_tip_node"

nodeset 9 vertex 31
nodeset 9 name "ctod_notch_radius"

nodeset 10 vertex 31
nodeset 10 name "ctod_notch_edge"

sideset 1 surface 50
sideset 1 name "top_surf"

sideset 2 surface 52
sideset 2 name "crack_surf"

block 1 volume all
block 1 name "pipe_circumferential_crack"
```

A.5.2. Notched Crack

```
#{height = 0.076}
#{height = 0.02}
#{outer_diameter = 0.02}
#{inner_diameter = 0.0174}
#{crack_diameter = 0.001 * (outer_diameter - inner_diameter)}
#{crack_offset_factor = 0.1}
#{cap_thickness = 0.01 * height}
#{crack_longitudinal_radius = 0.8 * height}
#{cap_width = 0.05 * height}
#{scale_factor = 1.0e3}
#{crack_map_diameter = 0.002 * (outer_diameter - inner_diameter)}
#{crack_outer_map_diameter = 0.1 * (outer_diameter - inner_diameter)}
#{crack_mesh_bias_factor = 4.0}
#{angle=1}

# create hollow cylinder
create cylinder height {height} radius {outer_diameter/2.0}
webcut volume 1 with cylinder radius {inner_diameter/2.0} axis z

delete volume 2

webcut volume all with plane zplane
delete volume with z_coord > 0

webcut volume all with yplane
webcut volume all with yplane rotate {angle} about z

delete volume 3 5 6

# create crack
# {crack_center = inner_diameter/2.0 + crack_offset_factor * (outer_diameter - inner_diameter)
    /2.0 - crack_diameter/2.0}
```

```
create torus major radius {crack_center} minor radius {crack_diameter/2.0}
create cylinder height {crack_diameter} radius {crack_center}

chop volume 4 with volume 7
chop volume 10 with volume 8
delete volume 9 11

# create mapping region near crack
create torus major radius {crack_center} minor radius {crack_map_diameter/2.0}
create cylinder height {crack_map_diameter} radius {crack_center}
chop volume 12 with volume 14
chop volume 16 with volume 13

# create mapping region near crack
create torus major radius {crack_center} minor radius {crack_outer_map_diameter/2.0}
create cylinder height {crack_outer_map_diameter} radius {crack_center}
chop volume 18 with volume 20
chop volume 22 with volume 19
unite volume 21 23

webcut volume all with zplane offset {-0.0003}

########################
# meshing done below
########################

#{crack_mesh_size = 3.0e-7 * scale_factor}
#{transition_mesh_size = (outer_diameter-inner_diameter)/800.0 * scale_factor}
#{transition_mesh_size_2 = (outer_diameter-inner_diameter)/400.0 * scale_factor}
#{coarse_mesh_size = (outer_diameter-inner_diameter)/80.0 * scale_factor}

# scaling up the body helps with merge tolerances
body all scale {scale_factor} {scale_factor} {scale_factor}

imprint all
merge all

surface 88 scheme map
surface 88 size {crack_mesh_size}
curve 132 scheme bias fine size {crack_mesh_size} coarse size {crack_mesh_bias_factor *
    crack_mesh_size} start vertex 79
curve 110 scheme bias fine size {crack_mesh_size} coarse size {crack_mesh_bias_factor *
    crack_mesh_size} start vertex 66

mesh surface 88

surface 80 size {crack_mesh_size}
curve 116 scheme bias fine size {crack_mesh_size} coarse size {crack_mesh_bias_factor *
    crack_mesh_size} start vertex 65
mesh surface 80

surface 103 size {transition_mesh_size}
surface 103 scheme pave
curve 174 scheme bias fine size {crack_mesh_size} coarse size {transition_mesh_size} start vertex
     79
curve 153 scheme bias fine size {crack_mesh_size} coarse size {transition_mesh_size} start vertex
     65
mesh surface 103

surface 123 size {transition_mesh_size_2}
surface 123 scheme pave
curve 187 size {coarse_mesh_size}
curve 189 scheme bias fine size {transition_mesh_size} coarse size {coarse_mesh_size} start
    vertex 95
curve 184 scheme bias fine size {transition_mesh_size} coarse size {transition_mesh_size_2} start
     vertex 105
curve 190 scheme bias fine size {transition_mesh_size_2} coarse size {coarse_mesh_size} start
    vertex 33
```

```
mesh surface 123

surface 131 size {coarse_mesh_size}
mesh surface 131

volume all redistribute nodes on

curve 136 128 111 115 interval 20

volume 17 scheme sweep source surface 88 target surface 91 autosmooth target on
volume 15 scheme sweep source surface 80 target surface 78 autosmooth target on
volume 21 scheme sweep source surface 103 target surface 106 autosmooth target on
volume 24 scheme sweep source surface 123 target surface 125 autosmooth target on
volume 25 scheme sweep source surface 131 target surface 129 autosmooth target on

mesh volume 17 15 21 24 25

body all scale {1.0 / scale_factor} {1.0 / scale_factor} {1.0 / scale_factor}

nodeset 1 surface 120 114 92
nodeset 1 name "bottom_symmetry_surf"

nodeset 2 surface 78 91 106 125 129
nodeset 2 name "axisymmetry_surf"

nodeset 3 surface 81 104 124 130
nodeset 3 name "inner_wall"

nodeset 4 surface 128 126
nodeset 4 name "outer_wall"

nodeset 5 surface 88 80 103 123 131
nodeset 5 name "side_symmetry_surf"

nodeset 6 curve 136
nodeset 6 name "crack_front"

nodeset 7 surface 42
nodeset 7 name "top_surf_ns"

nodeset 8 vertex 82
nodeset 8 name "crack_tip_node"

nodeset 9 vertex 68
nodeset 9 name "ctod_notch_radius"

nodeset 10 vertex 70
nodeset 10 name "ctod_notch_edge"

sideset 1 surface 42
sideset 1 name "top_surf"

sideset 2 surface 92
sideset 2 name "crack_surf"

block 1 volume all
block 1 name "pipe_circumferential_crack"
```

APPENDIX B. SIERRA IMPLEMENTATION

In this chapter, the overall procedure in *Sierra* for running a physics simulation and computing the *J*-integral is discussed. In appendix B.1, the various scenarios are highlighted. A sample *Sierra* input deck is shown in appendix B.2 which was used to produce the results in section 3.1. The other simulations conducted in chapter 3 can be carried out using the same input deck, by simply replacing the input mesh.

B.1. PROCEDURE OVERVIEW

The general procedure for conducting simulations consists of two primary scenarios:

1. Simulations involving a single mesh
2. Simulations involving inter-procedural transfers between meshes

The differences between the basic procedures of scenarios 1 and 2 are graphically illustrated in

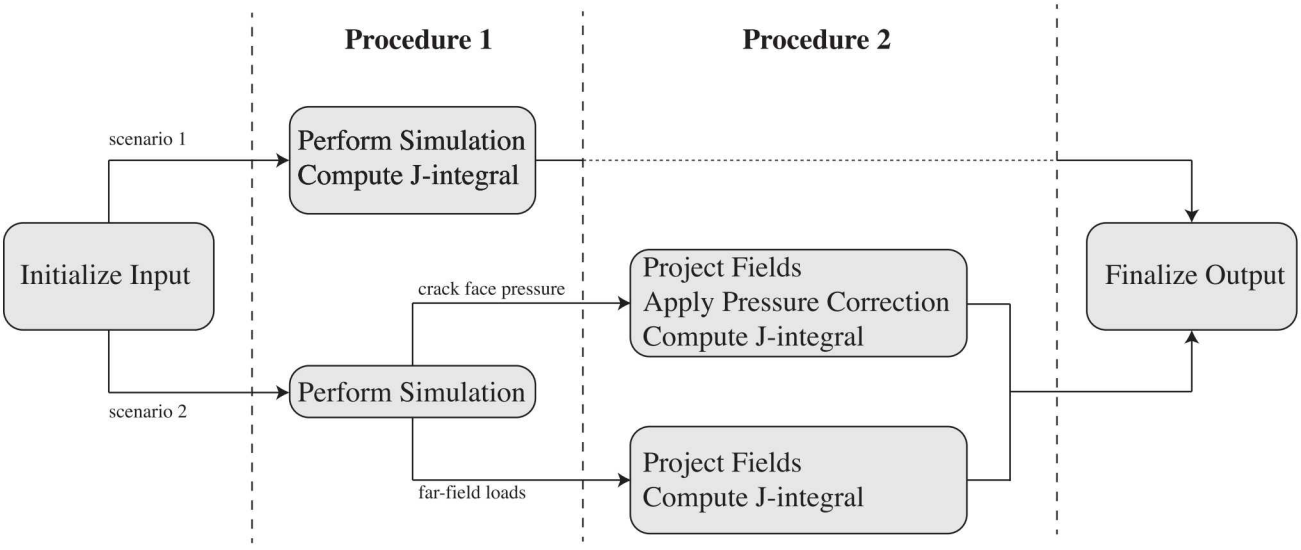

Figure B-1. Sierra Implementation: Procedural differences between scenarios 1 and 2

Scenario 1 assumes the use of a mesh consisting of uniform gradient (UG) hexahedral elements, due to the limitation that the *J*-integral capability in Sierra is currently only supported for this

element type. A single UG element contains 8 degrees of freedom at the nodes, and 1 integration point. This under-integrating leads to some error in the fields, while alleviating element locking due to over-constraints. As a rule of thumb, this element type is suitable for problems involving small deformations and highly localized crack-tip plasticity. An advantage scenario 1 is that the J-integral can be computed in a single procedure in *Sierra*, thus bypassing the need for L_2-projections and interpolations, which are error-prone. Additionally, the appropriate correction terms are automatically added onto the computed J-integral to ensure path-independence in the case of cracks faces exposed to pressure, without any required user input.

Scenario 2 is preferred in cases where deformation/plasticity is significant in the physical simulation and/or when the modeled geometry is too complex to mesh with hexahedral elements. In this scenario, the physical simulation is run using any preferred element type (*e.g.* selective deviatioric hexahedral elements, composite tetrahedral elements). The results are then projected and interpolated onto a template mesh consisting of UG elements which contains the crack and its surrounding regions. The J-integral is lastly computed on the template mesh.

Simulations using scenario 2 require two procedures, the first for the physical simulation, and the second to compute the J-integral on the template mesh. The implementation of the second procedure differs for models involving pressure on the crack face than for those with only far-field loads. The former requires a "dummy" load step to properly register the fields. This is done by selecting a very small (non-zero) time step, *i.e.*, the following line at the end of the time control block for the second procedure

```
termination time = {end_time + small_dt * end_time}
```

and assigning uniform zero velocity on all degrees of freedom, *i.e.* the following block in the boundary conditions of the second procedure

```
begin prescribed velocity
  include all blocks
  components = x y z
  function = sierra_constant_function_zero
end
```

These are both included in the sample input deck provided in appendix B.2. It has been noted that the additional load step can lead to potential sources of error. The sources of this error may be due to issues within the code. Work to quantify this error and implement fixes are currently an ongoing effort.

In summary, scenario 1 is suitable for problems involving simple geometries and/or small deformations. The fields for scenario 1 generally contain some amount of error due to under-integration. However, the J-integral can be computed as a post-processing step with minimal additional input, and no inter-procedural transfers are necessary. scenario 2 is required for most simulations involving significant deformations and/or complex geometries. However, as a consequence, an extra procedure is required to perform the J-integral, and the fields must be projected and interpolated onto a template mesh containing the crack. Additionally, cracks faces exposed to pressure must include specific user input to ensure the correction terms are properly activated for the J-integral.

B.2. INPUT DECK FOR PLATE WITH EMBEDDED ELLIPTICAL FLAW

```
############################################################
  #---------- Description ----------
############################################################
#
# This file runs an adagio simulation for a plate with a pressurized embedded flaw
#
############################################################

#{end_time = 10.0}
#{num_output=2}
#{time_steps=2}
#{initial_time_step = end_time/time_steps}
#{output_time_step = end_time/num_output}
#{peak_load = 1.0e5}
#{project = 0}
#{pressure = 0}

begin sierra  plate_embedded_flaw

  title  plate_embedded_flaw

  define direction unit_minus_z with vector 0 0 -1

#####################################################
  #---------- Function Defs ---------
#####################################################

  begin definition for function load_ramp
    type is Analytic
    expression variable: time = global time
    evaluate expression = "{peak_load}*time/{end_time}"
  end definition for function load_ramp

  begin definition for function const_load
    type is analytic
    evaluate expression = "{peak_load}"
  end definition for function const_load

$$$-------------------------------------------------------$$$

############################################################
  #---------- Materials ----------
############################################################

#-------------Adagio Materials----------

  begin material ss
    density = 8004

    begin parameters for model elastic
      youngs modulus = 200.0e9
      poissons ratio = 0.25
    end parameters for model elastic

    begin parameters for model neo_hookean
      youngs modulus = 200.0e9
      poissons ratio = 0.25
    end parameters for model neo_hookean

  end material ss
```

```
$$$-------------------------------------------------------$$$

##############################################################
  #---------- Adagio Mesh ----------
##############################################################
   #---------- Adagio Section -----------
  begin solid section ug_section
    formulation = mean_quadrature
    strain incrementation = midpoint_increment
  end solid section ug_section

  begin solid section sd_section
    Formulation = selective_deviatoric
    Deviatoric Parameter = 1
  end solid section sd_section

  begin total lagrange section section_composite_tet10
    formulation = composite_tet
  end

  #---------- Finite Element Model -----------

  begin finite element model plate
    Database name = plate_embedded_flaw.g
    Database type = exodusII

    begin parameters for block plate_embedded_flaw
      material ss
      solid mechanics use model elastic
{if (project == 0)}
      section = ug_section
{else}
      section = sd_section
{endif}
    end parameters for block plate_embedded_flaw
  end finite element model plate

{if (project == 0)}
{else}
  begin finite element model projected_plate
    Database name = template_plate_embedded_flaw.g
    Database type = exodusII

    begin parameters for block plate_embedded_flaw
      material ss
      solid mechanics use model elastic
      section = ug_section
    end parameters for block plate_embedded_flaw
  end finite element model projected_plate
{endif}

$$$-------------------------------------------------------$$$

##############################################################
  #---------- Main Procedure for Adagio ----------
##############################################################

  Begin Adagio Procedure adagio_pressure_test
   Begin time control
      Begin time stepping block time_block
         Start Time = 0.0
           Begin Parameters for Adagio Region adagio_mechanical
              Time Increment = {initial_time_step}
           End Parameters for Adagio Region adagio_mechanical
      End Time stepping block time_block
      Termination Time = {end_time}
    End time control
$$$-------------------------------------------------------$$$
```

```
##############################################################
  #---------- Adagio Region ----------
##############################################################

    begin adagio region adagio_mechanical

    # ------------------ adaptive time stepping ----------------

      begin adaptive time stepping
         target iterations = 50
         iteration window = 15
         maximum multiplier = 4.0e+00
         minimum multiplier = 1e-10
         growth factor = 1.5
         cutback factor = 0.5
         maximum failure cutbacks = 30
       end


      use finite element model plate

      #--------- Boundary Conditions ---------

      begin fixed displacement
        node set = bottom_symmetry_surf
        components = z
      end fixed displacement

      begin fixed displacement
        node set = side_symmetry_surf_yplane
        components = y
      end fixed displacement

      begin fixed displacement
        node set = side_symmetry_surf_xplane
        components = x
      end fixed displacement

      begin fixed displacement
        node set = side_surf_yplane
        components = y
      end fixed displacement

      begin fixed displacement
        node set = side_surf_xplane
        components = x
      end fixed displacement

{if (pressure == 0)}
      begin traction
        surface = top_surf_ss
        function = load_ramp
        direction = unit_minus_z
      end traction
{else}
      begin fixed displacement
        node set = top_surf
        components = z
      end fixed displacement

      begin pressure
        surface = pressure_surf
        function = load_ramp
      end pressure
{endif}

{if (project == 0)}
```

```
      begin j integral j_pl
        crack tip node set = crack_front
        integration radius = 0.1
        number of domains = 100
        crack plane side set = crack_surf
        function = plateau
      end
{endif}

      #--------- Results Output ---------

      begin results output output_adagio
         database name = plate_embedded_flaw.e
         database type = exodusII
         at time 0, increment = {output_time_step}
         Nodal Variables = displacement As disp
         Nodal Variables = temperature As temp
         Nodal Variables = coordinates
         Nodal Variables = velocity
         Nodal Variables = force_internal
         Nodal Variables = residual
         Element Variables = fluid_pressure
         Element Variables = von_mises
         Element Variables = effective_log_strain
         Element Variables = stress
         Element Variables = unrotated_stress
         element Variables = strain_energy
         element Variables = strain_energy_density
         Element Variables = left_stretch
         Element Variables = rotation
{if (project == 0)}
         element variables = j_integration_domains_j_pl as j_int_dom_pl
         global variables = j_average_j_pl as j_ave_j_pl
         nodal variables = j_j_pl  as j_pl_planar
         nodal variables = j_weight_functions_j_pl as j_pl_wtfunc
{endif}
      end results output output_adagio

      #--------- Solver ---------
   Begin Solver
     begin loadstep predictor
       type = secant
     end loadstep predictor

     Begin cg
          target relative residual     = 1e-5
          acceptable relative residual = 1e-4
          target residual              = 1e-3
          acceptable residual          = 1e-2
          Maximum Iterations = 10
          Minimum Iterations = 5
          Line Search secant
          Orthogonality measure for reset = 0.5
          Preconditioner = Elastic

          begin full tangent preconditioner
           linear solver = feti
           iteration update = 10
           minimum convergence rate = 1.0E-15
          end

        End cg
     End Solver
    end adagio region adagio_mechanical

  End Adagio Procedure adagio_pressure_test
```

```
{if (project == 0)}
{else}

##############################################################
  #---------- Projection Procedure for Adagio ----------
##############################################################

  Begin Adagio Procedure project

    Begin procedural transfer migration
      node variables = displacement
      element variables = stress unrotated_stress strain_energy_density strain_energy
      element lie group variables = rotation left_stretch
      Begin l2_projection transfer fred
        send blocks = plate_embedded_flaw
        receive blocks = plate_embedded_flaw
        transformation type = element2element
        send coordinates = original
        receive coordinates = original
        linear solver = feti_iterative
      End

    End

    Begin time control
      Begin time stepping block p0
        start time = {end_time}
           Begin Parameters for Adagio Region region_2
              Time Increment = {1.0e-5 * end_time}
           End Parameters for Adagio Region region_2
      End
{if (pressure == 0)}
    termination time = {end_time}
{else}
    termination time = {end_time + 1.0e-5 * end_time}
{endif}
    End

    Begin Adagio Region region_2

      Use Finite Element Model projected_plate

{if (pressure == 0)}
{else}
      begin prescribed velocity
        include all blocks
        components = x y z
        function = sierra_constant_function_zero
      end

      begin pressure
        surface = pressure_surf
        function = const_load
      end pressure
{endif}

      begin j integral j_pl
        crack tip node set = crack_front
        integration radius = 0.1
        number of domains = 100
        crack plane side set = crack_surf
        function = plateau
      end

      Begin Results Output output_vars
          database name = projected_plate_embedded_flaw.e
          at step 0 increment = 1
          element Variables = stress
```

```
          element Variables = unrotated_stress
          element Variables = strain_energy
          element Variables = strain_energy_density
          element Variables = rotation
          element Variables = left_stretch
          element variables = j_integration_domains_j_pl as j_int_dom_pl
          global variables = j_average_j_pl as j_ave_j_pl
          nodal variables = j_j_pl  as j_pl_planar
          nodal variables = j_weight_functions_j_pl as j_pl_wtfunc
          nodal variables = displacement
          nodal variables = coordinates
      End

      Begin solver

        Begin cg
          reference = external
          target relative residual     = 1e-5
          acceptable relative residual = 1e-4
          target residual              = 1e-3
          acceptable residual          = 1e-2
          Maximum Iterations = 10
          Minimum Iterations = 5

          Begin full tangent preconditioner
            linear solver = feti
            iteration update = 10
          End

        End cg

      End solver

    End adagio region region_2
$$$-------------------------------------------------------$$$
  End adagio procedure project
$$$-------------------------------------------------------$$$
{endif}
  begin feti equation solver feti
    Residual Norm Tolerance = 1e-6
  end

  begin feti equation solver feti_iterative
    param-string "debugMask" value "solver"
    param-string "local_rbm_tol" value 1.0e-32
    param-string "global_rbm_tol" value 1.0e-32
    residual norm tolerance = 1e-16
  end

end sierra plate_embedded_flaw
```

DISTRIBUTION

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	Dorian Balch	8443	MS 9035
1	Lauren Beghini	8752	MS 9042
1	Guy Bergel	8752	MS 9261
1	Jill Blecke	1512	MS 0828
1	Amanda Dodd	8750	MS 9159
1	James Foulk	8363	MS 9042
1	Alex Hanson	8752	MS 9042
1	Kevin Manktelow	8752	MS 9042
1	Stacy Nelson	8752	MS 9042
1	Yuki Ohashi	8752	MS 9042
1	Jake Ostien	8363	MS 9042
1	Kendall Pierson	1542	MS 0845
1	Steve Rice	8443	MS 9035
1	Chris San Marchi	8367	MS 9161
1	Andrew Stershic	8752	MS 9042
1	Michael Veilleux	1542	MS 0845

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
CA Technical Library	8551	cateclib@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.