Development of Automated Model Generation and Analysis of Surface Cracked Plates in Bending for Interpolated Elastic-Plastic J-integral Solutions

Michael W. Renfro[1] Phillip A. Allen[2] Christopher D. Wilson[1]

19th International ASTM/ESIS Symposium on Fatigue and Fracture Mechanics

[1]Tennessee Tech University
[2]NASA Marshall Space Flight Center

Good afternoon.

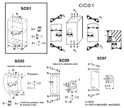
This material is part of my recently-completed PhD research, and covers the development, verification, and validation of a set of elastic-plastic models of surface cracked plates in bending.

The first two parts of the presentation cover previous work done to quantify crack driving forces for surface cracks in bending.

The next two parts briefly show the result of some automatically-generated models for existing tension cases, initial changes required to handle bending conditions, and plans for modifying the NASA TASC program.

The last two parts show some bending results and plans for the future.

- Constraint and stress states can get pretty involved.
- Semi-elliptical surface cracks (SC01) are among the simplest part-through crack cases, and are the subject of ASTM E2899.
- Handbook or curve-fit solutions exist for the other geometries, but only for linear elastic materials.
- NASA's TASC program covers elastic-plastic surface cracks in tension, but **not** in bending.



For real-world part-through cracks, the basic situation is that the geometry, applied stress, and level of constraint around the crack front gets increasingly complex.

A semi-elliptical surface crack in a flat plate is one of the simpler ones, but we only have handbook-type solutions for crack driving forces for linear elastic materials.

For elastic-plastic materials, the NASA TASC program can provide approximate solutions, but only for tension cases.

Research goals

- High quality set of elastic-plastic finite element analysis results as a basis for curve-fit or handbook calculations
- Modified TASC program for bending or tension analysis

So the fundamental goals of this part of the research are to create a quality set of finite element results for elastic-plastic materials

and to modify the TASC program to accommodate either bending or tension cases.

For a surface crack in bending, stress intensity at a given location:

$$K_\mathrm{I} = (H\sigma_\mathrm{b}F_\mathrm{b})\left(\frac{\pi a}{Z}\right)^{0.5}$$

$$\sigma_\mathrm{b} = \frac{6M}{Wt^2}$$

$$Z = 1 + 1.464\left(\frac{a}{c}\right)^{1.65}$$

$$F_\mathrm{b} = \left[M_1 + M_2\left(\frac{a}{t}\right)^2 + M_3\left(\frac{a}{t}\right)^4\right] f_\phi f_\mathrm{wb} g$$

For surface cracks in tension, we need "only" 10 equations.

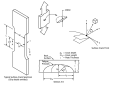
No such curve fit exists for elastic-plastic materials.

...plus another 12 equations, and that's just for linear elastic materials.

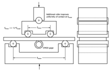
For surface cracks in bending, there's a set of 16 equations from Newman that estimate stress intensity factors around the crack front,

but for elastic-plastic materials, no such set of equations exist.

ASTM E2899: Standard Test Method for Measurement of Initiation Toughness in Surface Cracks Under Tension and Bending



Test specimen and crack configurations



Four-point bend test configuration

The left half of this slide shows an example specimen for ASTM E2899, including a cross-section of its semi-elliptical surface crack.

The right half shows that specimen loaded into a four-point bending configuration.

- Starter crack machined into flat plate, fatigued to sharpen crack front
- CMOD monitored as tension or bending load increased monotonically
- Either specimen fails or start of stable crack tearing is detected
- Location where crack growth occurs is recorded
- Conditions classified as linear elastic, elastic-plastic, or fully-plastic
- If LEFM or EPFM, calculate constraint from tables
- If LEFM, calculate K from series of provided equations
- If EPFM, **use nonlinear FEA** to calculate J

In ASTM E2899, the first few steps to calculating stress conditions at the crack tip are purely related to experimental measurement.

For linear elastic conditions, a series of equations derived from experimental curve fits can be used to calculate K.

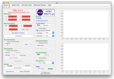
But for elastic-plastic conditions, no such series of equations exists to calculate J, and the standard requires finite element analysis to calculate it.

That last step is a major departure compared to other standards, and would normally require construction of models for every tested geometry and material.

ASTM E2899 is unusual in its analysis requirements

- requires results of elastic-plastic finite element analysis
- other standards require much simpler calculations or graphical constructions
- NASA TASC program satisfies requirements, but only for tension

Mechanics of bending is more complex than for tension (constraint, crack closure)



NASA TASC program

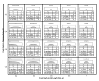
The vast majority of ASTM standards focus on experimental techniques, with some basic graphical constructions or simple equations to verify the test was done properly

But ASTM E2899 requires finite element analysis for elastic-plastic materials, and this can be a burden on those performing these kinds of tests.

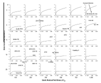
The NASA TASC program satisfies those analysis requirements of E2899, but only for tension, not for bending.

Bending analysis is more complex than tension analysis: cracks can close, and the state of stress around the crack front makes for variable constraint conditions.

TASC: interpolation code using database of 600 EPFM results for flat plates in tension



20 normalized geometries:
$0.2 \le \frac{a}{c} \le 1.0,\ 0.2 \le \frac{a}{t} \le 0.8$



30 normalized materials:
$100 \le \frac{E}{S_{ys}} \le 1000,\ 3 \le n \le 20$

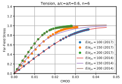
TASC is the tool for analysis of surface cracks.

TASC includes elastic-plastic results from 600 finite element analyses: all combinations of 20 normalized geometries and 30 normalized materials shown here.

It interpolates results for intermediate values, but only includes models for plates in tension.

No bending results are included, so work remained on satisfying the entire E2899 standard.

Verification of two TASC cases, applying procedure to a new material ($\frac{E}{\sigma_{ys}} = 150$)



Comparison of normalized FEA results, interpolated result, and TASC raw data

To get started, I got into the internals of the TASC database and extracted remote stress and CMOD results for two of the 600 models, plus the interpolated results for a third intermediate model.

The goal was to replicate the first two results with programmatically-created models, and to see how close the interpolated result was to a third purpose-built model, testing the limits of the interpolation method used in TASC.

On this figure, the lines represent TASC results for $E = 100$ and $E = 200$, with an interpolated result for $E = 150$.

The circles represent my model results at each modulus. You can see that the results are identical for $E = 100$ and $E = 200$, and that the interpolated TASC result is a few percent more compliant than the actual results at $E = 150$.

$t = 1 \quad 0.2 \le \frac{a}{c} \le 1.0 \quad 0.2 \le \frac{a}{t} \le 0.8$

$W = 5\max(c, t) \quad S_{\text{inner}} = W \quad S_{\text{outer}} = 2W \quad L = 1.1 S_{\text{outer}}$

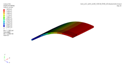


Plate model, $\frac{a}{c} = 0.2$, $\frac{a}{t} = 0.8$

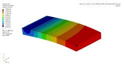


Plate model, $\frac{a}{c} = 1.0$, $\frac{a}{t} = 0.2$

The dimensions on each plate follow the same rules.

Each model represents 1/4 of the full experimental plate, and has thickness of 1 inch.

The model width is 5 times larger than either the crack half width or the plate thickness, whichever is larger.

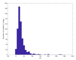
The plane where the inner rollers would appear will be a distance W from the center, and the line representing the outer rollers will be spaced at twice that distance.

The whole plate will extend 10% past the outer roller location.

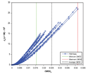
All these models will start from a single input file for the FEACrack program, and a Python script will run FEACrack with command-line parameters, adjusting dimensions and writing a new WARP3D input file for each of 20 geometries.

After that, the same Python script can adjust material properties in the 20 WARP3D input files and write a new input file for each of 30 materials.

allenwells2014 reported $M = \frac{r_\phi \sigma_{ys}}{J} < 25$ for tension



Histogram of M results from TASC tension model database

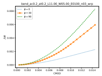


J-CMOD graph used for extrapolation

The TASC literature indicated models were loaded until J values were high enough to drive this dimensionless deformation formula below 25,

but looking at the full set of M values recorded in the TASC database shown in the left figure, it's clear many of the models didn't reach that deformation level.

The real criteria for loading is to load the model heavily enough to straighten the J-CMOD curves shown in the right figure so we can accurately extrapolate beyond the applied load.



$\frac{a}{c} = 0.2,\ \frac{a}{t} = 0.2,\ E = 100,\ n = 3$

Adjust boundary conditions until

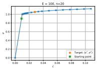
- slope of last 20% of J-CMOD curve is 20× larger than initial slope
- slope of last 20% of J-CMOD curve is $<$ 10% different than slope of previous 20%

at $\phi = 30\,^{\circ}$

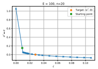
Since we have a full set of J-ϕ-CMOD data after running a model, my plan was to increase loads until the two criteria shown here were met at $\phi = 30\,^{\circ}$.

This ensures that the last 40% of the J-CMOD curve is consistently straight, and any major curvature is confined to the first 60% of the range.

Displacement control for tension models makes optimization easier



Example stress-strain curve using linear plus power law (LPPL) formulation



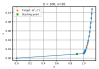
Transformed to find required strain level

Early versions of satisfying this load criteria for tension models used displacement boundary conditions, and some convenient optimizations could take place.

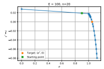
Because we're effectively adjusting strain to match a target stress, the left stress-strain curve can be transformed to the form on the right, and we can do a secant method to find the strain required to drive the transformed stress value to zero.

Just move on a tangent at wherever we are on the transformed curve, and we quickly end up at the desired strain value.

Load control for bending models makes optimization more difficult



Example stress-strain curve using LPPL formulation, transformed to stress-controlled



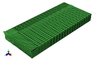
Transformed to find required stress level

Unfortunately, the bending models have to operate with traction boundary conditions, where we effectively adjust applied stress to match a target strain.

The secant method won't work, as our first step would run us way off the right side of the transformed graph.

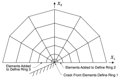
So we just have to take small steps and stop whenever we see the curve has straightened out enough.

Abaqus validation



Example Abaqus bending model from FEACrack

J convergence



Elements used in WARP3D J calculations

Since we're going to spend a lot of time finding J values for the models, it makes sense to do some checks to ensure the J values are accurate.

First, a subset of the models will be run in Abaqus with identical meshes.

Second, as J values can be calculated for any set of elements surrounding the crack front, those J values should converge to a final result as we add more elements and capture more of the strain energy and work done inside the plastic zone.

- Don't break anything already working for tension
- Make a `results_bending` database alongside the existing `results` database for tension
- Identify any equations only valid for tension models
- Replace with conditionals checking for model type, then use tension or bending equations as required
- Interpolation method should need no changes
- Validate a load-CMOD curve against existing bending experimental data

For modifications to the TASC program, the goal is to make the smallest set of changes required to support bending.

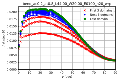
We can't break tension results, and most of the code assumes a certain arrangement of data in the database.

So we'll make a results_bending database alongside the existing results database.

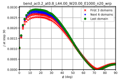
Any equations that are only applicable for tension will be replaced with a check for which type of analysis we're running, and then picking the right equation

There shouldn't need to be any changes to the interpolation parts of TASC

and ideally, we'll be able to reproduce an experimental load-CMOD curve from a four-point bend experiment with good accuracy.



Convergence of J across 10 domains

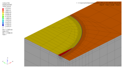


Anomalous J convergence graph

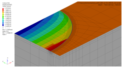
During the J convergence study, some of the models had local minima in the J values at weird locations, like the one on the right.

They did converge well as we used more elements around the crack front, but converged to unexpected values sometimes.

Why is J higher at $\phi = 90$ in some cases? What happens to plate deflection?



First load step



Last load step

Looking at how the plate deformed,

where orange is basically zero deflection,

red is positive deflection, and

yellow and below are negative deflections,

it's clear that regions deep in the crack are pushing out past the symmetry plane, which is physically impossible.

This never happened on tension models, as the plate cross section was always being stretched. But in bending, the underside of the plate is effectively being compressed, and it's pushing some of the cracked material along with it.

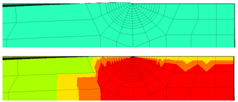
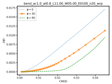


Plate deflection before and after addition of elastic boundary

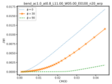
So an elastic boundary was added to the symmetry plane.

A rigid boundary plane would be more common, but WARP3D doesn't support those.

The effect of the boundary on displacement can be seen here, where now all the plate material stays below the symmetry plane.

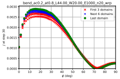


Before addition of elastic boundary

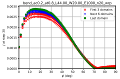


After addition of elastic boundary

In terms of J values, the plane has no effect toward the free surface (the solid and dotted lines), but greatly reduces the J values seen deep in the crack (the dashed line)

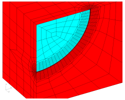


Before addition of elastic boundary

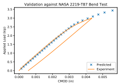


After addition of elastic boundary

Looking at the effect on J around the crack front, it's clear this removed the local minimum for J that was the original symptom of the problem.



Crack front mesh of purpose-built experimental validation model

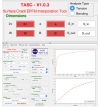


Comparison of predicted load and CMOD between purpose-built model and experiment

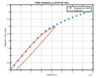
Next up is making sure that FEACrack and WARP3D can reproduce a load-displacement curve from a four-point bend experiment.

A plate model with the right material properties and a crack of the correct depth and aspect ratio was created, and the blue x marks show its load-CMOD curve.

The solid line represents a NASA four-point bend experiment, and matches the prediction within a few percent at most.



Modified TASC Program



Comparison of TASC and experiment

Finally, in terms of TASC work, the very minimal user interface changes can be seen in the top left,

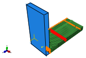
where a radio button switching between tension and bending analysis has been added,

as well as two text entries for roller spans.

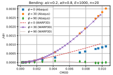
After entering in the correct crack geometry, plate dimensions, and material properties, the bottom left figure shows the predicted load-CMOD curve.

The same curve is shown on the right and compared to the original experimental data.

Again, the prediction is within percentage points all along the experimental curve.



Sample Abaqus validation model



Sample J comparison between Abaqus and WARP3D

For validating J values from WARP3D against Abaqus, you can see a representative Abaqus model on the left figure.

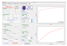
A rigid plane has been added similar to the elastic boundary added in WARP3D.

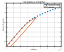
Looking at some of the J values from Abaqus shown with solid markers, versus the WARP3D results shown with lines,

the J values are coincident except at the free surface where $\phi = 0$, and even those are pretty close.

Database of 600 elastic-plastic finite element results for surface cracks in bending

- More challenging than tension models
- Subset of results verified and validated against Abaqus and experimental data
- Modified TASC program
- Possible to satisfy requirements of ASTM E2899 for tension **and** bending without constructing purpose-built EPFM models
- Greatly reduces analytical burden on anyone doing ASTM E2899 tests





We now have a database of 600 elastic-plastic FEA results for surface cracks in bending.

These models had some extra challenges compared to tension models, and those were overcome.

A subset of the results have been verified and validated against Abaqus models and against experimental data

The TASC program has been modified to support bending, at least as far as predicting load-CMOD curves.

All of this means it's now possible to satisfy all the requirements of ASTM E2899 in both tension and bending, without building custom models for each test,

and this greatly reduces the analytical burden on anyone performing tests under ASTM E2899.

1. TASC and E2899
   - Finish integrating bend data into TASC, beyond load-CMOD validation
   - Additional values for material and/or crack geometry
   - Investigate other interpolation methods (piecewise cubic?)
   - Replace traction boundary conditions with rigid rollers and contact modeling
2. **More experimental data needed for these.**

For future work related to this research,

there's still some work to be done in TASC modifications and testing, as I didn't modify anything beyond what was necessary to validate a load-CMOD curve

some additional data points for crack geometry and material models may be worthwhile,

as well as seeing if an alternative interpolation method will provide better agreement with experimental results.

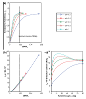
As WARP3D supports contact modeling, it may be worth replacing the traction conditions used in bending models with contact modeling of rigid rollers to more closely match experimental conditions

Thanks to:

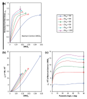
- Tennessee Tech University's:
  - Center for Manufacturing Research (Computer Aided Engineering)
  - Information Technology Services (High Performance Computing)
- Quest Integrity's FEACrack
- UIUC's WARP3D
- Dassault's Abaqus
- Anaconda's Python distribution

Facilities from both Tennessee Tech's Center for Manufacturing Research and Information Technology Services were instrumental at different phases of this research.

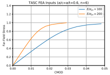
The research would not have been possible without Quest Integrity's FEACrack, UIUC's WARP3D, Dassault's Abaqus, and the Anaconda Python distribution.



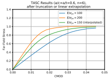
Gap in results for widest aspect ratios



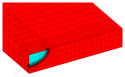
Gap in results for lowest E values



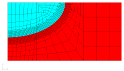
Raw FEA results used in TASC



Interpolated FEA results displayed by TASC



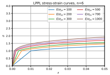
Isometric view of overall mesh



Detailed view of crack front

$$\frac{\epsilon}{\epsilon_{\mathsf{ys}}} = \begin{cases} \dfrac{\sigma}{\sigma_{\mathsf{ys}}}, & \epsilon \leq \epsilon_{\mathsf{ys}} \\ \left(\dfrac{\sigma}{\sigma_{\mathsf{ys}}}\right)^n, & \epsilon > \epsilon_{\mathsf{ys}} \end{cases}$$

where $\epsilon_{\mathsf{ys}} = \frac{\sigma_{\mathsf{ys}}}{E}$.



Set of LPPL stress-strain curves

$$M = \frac{r_\phi \sigma_{\mathrm{ys}}}{J}$$

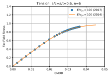
Applied displacement values for verification models

$\frac{E}{\sigma_{\mathrm{ys}}}$	Displacement	ϕ	M using $r_{\phi a}$	M using $r_{\phi b}$
100	0.1028	30°	15.9833	36.4241
		90°	22.6234	15.0822
200	0.0550	30°	24.7288	56.3542
		90°	34.9604	23.3069

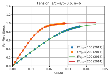
- 30 load steps
- `warp3d < file.inp > file.out`
- 21.6 minutes to solve on laptop, 2.2 minutes on HPC node

Python program

- run `packet_reader` to export displacements, forces
- extract node 1 z displacement, double to get CMOD
- identify nodes on $z = 0$ from input file
- extract z reactions from all identified nodes, sum to reaction force
- divide reaction force by plate cross section area to get stress



Verification of stress and CMOD relationship for first model



Verification of stress and CMOD relationship for second model