HILL ENGINEERING
Predict. Test. Perform.

BAMpF Workshop 2021

14 September 2021

Special Thanks!

- BAMpF Consortium Partners
  - ESRD
  - AFGROW
  - Boeing
  - Northrop Grumman
  - USAF

Agenda

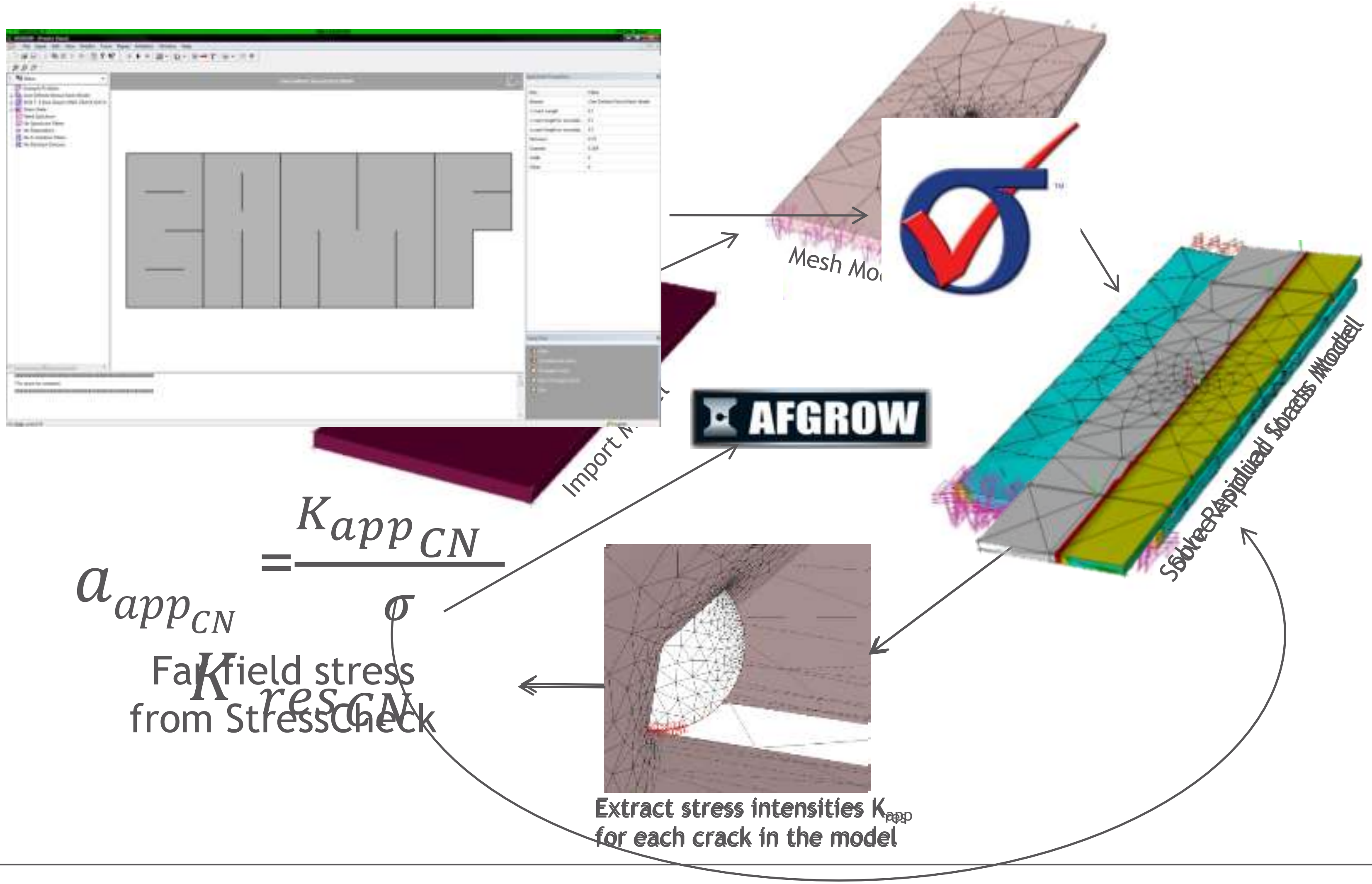
- ❑ **BAMpF Overview**
- ❑ **Major Events**
- ❑ **New Version: Updates and Features**
- ❑ **Previous Version: Bug Fixes**
- ❑ **BAMpF Utilization: From the User Community**
- ❑ **BAMpF Future**
  - ➢ Proposed Features and Upgrades

Overview: What is BAMpF?

BAMpF

Broad Application for Multi-point Fatigue

Overview: BAMpF Process Flow



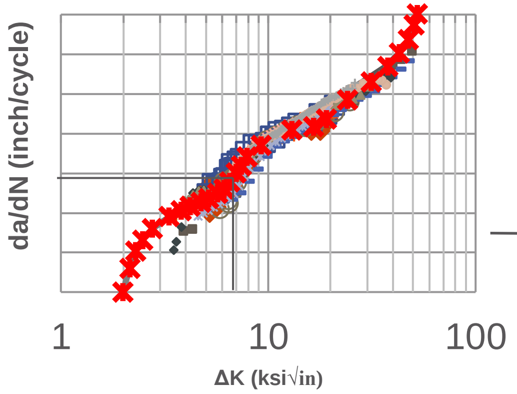
HILL ENGINEERING

Overview: BAMpF Process Flow

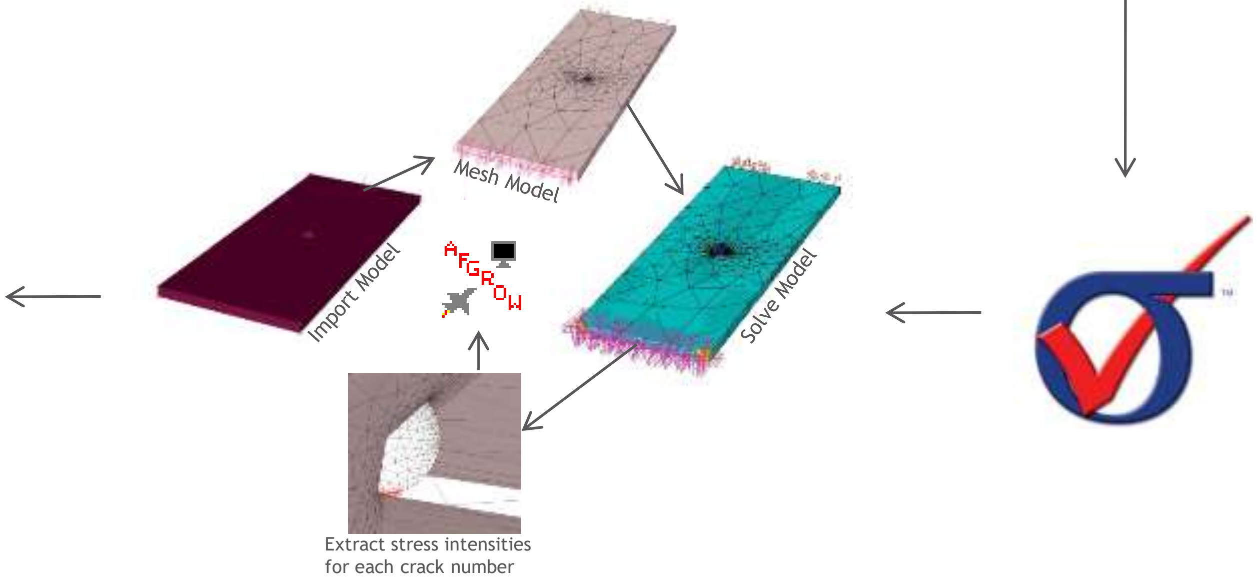
$$K_{min\,CN} = a_{app\,CN}\,\sigma_{\min} + K_{res\,CN}$$
$$K_{max\,CN} = a_{app\,CN}\,\sigma_{\max} + K_{res\,CN}$$

Where $\sigma_{min/max}$ is the AFGROW spectrum stress

$$R_{CN} = K_{min\,CN} / K_{max\,CN}$$



New Crack Lengths



Major Events

- AFGROW 5.3.5.24 (March 2020)
- StressCheck 11.1 (June 2021)
- BAMpF API beta 8.0 (June 2021)
- BAMpF 8.0 (July 2021)

Released 20 July 2021

BAMpF 8.0 FEATURES

Features

- ❑ Program Updates
- ❑ BAMpF API
- ❑ BAMpF K-array
- ❑ Surface Correction
- ❑ Retardation
- ❑ BAMpF Hidden Parameters

Updated Features

- **StressCheck 11.0 Compatibility**
  - Seamless integration if utilizing locate sets
  - Updating crack front meshing and extraction still needs to be added to BAMpF
- **AFGROW 5.3.4.23 Compatibility**
  - Updated to include retardation tracking for restarts and re-splining
  - Updated residual stress K’s to increase with crack length
  - Updated to track retardation parameters

BAMpF API

- In 2020 BAMpF went through an architectural change to allow for programmers to utilize an API to access functions utilized in BAMpF
- Programmers can now write their own code accessing specific BAMpF features
- Creation of API also cleaned up BAMpF code creating a faster and far more stable BAMpF experience
- The API was written for StressCheck 10.5 and newer
  - Previous versions of StressCheck will not be supported

BAMpF API Functions

- The BAMpF API Exposes 5 functions to the user
  - Initialize
    - Creates the output directories and starts the BAMpF User Form
  - configureBAMpFRun
    - Opens a StressCheck model and returns all of the BAMpF parameters set inside of StressCheck
  - setUpPrediction
    - Sets up any symmetry cracks the model has
  - runFirstBAMpFPrediction
    - Runs the initial StressCheck model with no crack growth information and returns the stress intensities used in the AFGROW prediction
  - runBAMpfPrediction
    - Takes the crack growth from AFGROW and uses it to grow the crack in the StressCheck model and run a BAMpF Prediction
- **Using these functions in conjunction with AFGROW in Excel a more tailored BAMpF application can be created**

API Example

Values obtained from AFGROW

FALSE			% Growth	2			182	Total Cycles	1391	
	AFGROW File	Crack Point #	CrackLength		K from StressCheck	Stress From StresCheck	Cycles	FinalC	DeltaP	
	C:\Users\JHodges\Desktop\NG BAMpF API Testing\Model3_2-1.dax	0	0.03773137	0.038485998	10444.22713	20000	182	0.0377	0.0000	Initialize BAMpF
	StressCheck File	1	0.038461254	0.039230479	10539.20474		182	0.0385	0.0000	
	C:\Users\JHodges\Desktop\NG BAMpF API Testing\Task3pt2.scp	2	0.038797908	0.039573866	10619.06891		182	0.0388	0.0000	
	Output Location	3	0.039235327	0.040020034	10650.75865		182	0.0392	0.0000	BAMpF Iteration
	C:\Users\JHodges\Desktop\NG BAMpF API Testing\	4	0.03980323	0.040599294	10679.37027		182	0.0398	0.0000	
		5	0.040502432	0.04131248	10702.9368		182	0.0405	0.0000	
		6	0.041508492	0.042338662	10732.94422		182	0.0415	0.0000	
		7	0.042470781	0.043320197	10772.61902		182	0.0424	0.0000	Run AFGROW
		8	0.043529932	[illegible]	[illegible]		182	0.0435	-0.0001	
		9	0.044670932	[illegible]	[illegible]		182	0.0446	-0.0001	
		10	0.045865597	0.046782909	10949.44627		182	0.0458	-0.0001	
		11	0.047367074	0.048314416	11053.75572		182	[illegible]	[illegible]	
		12	0.048649784	0.04962278	11160.90291		182	[illegible]	[illegible]	
		13	0.049951226	0.050950251	11294.8431		182	0.0499	-0.0001	
		14	0.051246369	0.052271296	11453.40537		182	0.0512	-0.0001	
		15	0.052535973	0.053586693	11625.49662		182	0.0524	-0.0001	
		16	0.05406543	0.055146738	11871.13883		182	0.0540	-0.0001	
		17	0.055281496	0.056387126	12076.7631		182	0.0552	-0.0001	
		18	0.056445321	0.057574228	12290.76994		182	0.0564	-0.0001	
		19	0.057487984	0.058637743	12482.96493		182	0.0574	-0.0001	
		20	0.058307951	0.05947411	12728.03546		182	0.0583	0.0000	
		21	0.058522942	0.059693401	12854.37836		182	0.0585	0.0000	

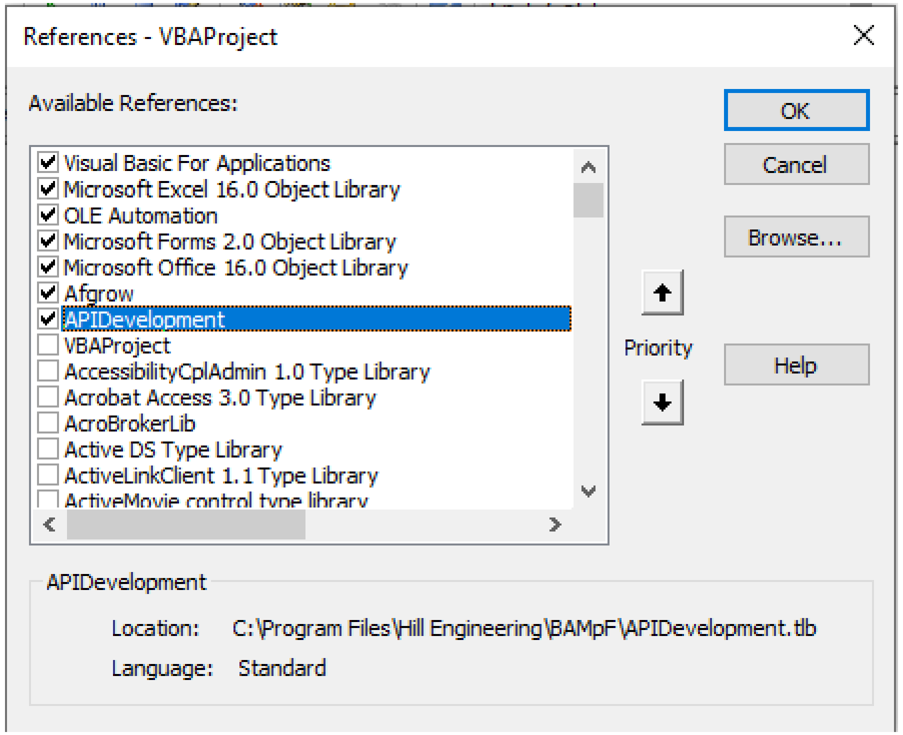
Growth Increment

User buttons

Input/output file locations

Values obtained from StressCheck

API Example




```
Dim mAfgrow As afgrow.Application
Dim i As Integer
Dim BAMpF As BAMpFAPI
Dim parameters As BAMpFStressCheckParameters
Dim result As IterationResult
Dim DeltaP(0, 21) As Double
Dim totalcycles As Long
Dim cycles As Double
Dim finalC As Double
Dim finalA As Double
Dim finalCt As Double
Dim finalKc As Double
Dim finalKa As Double
Dim finalKct As Double
Dim AFGROWresult As Boolean
Dim BTArray(1, 1) As Double
Dim middle As Variant
Dim SCModel As String
```

```
Public Sub Init_BAMpF_Click()

Application.DisplayAlerts = False
SCModel = Sheet1.Cells(5, 2)
Sheet1.Cells(1, 11) = 0
Sheet1.Cells(1, 1) = True

Set BAMpF = New BAMpFAPI

BAMpF.Initialize (Sheet1.Cells(7, 2))

parameters = BAMpF.configureBAMpFRun(SCModel, True)
BAMpF.setUpPrediction (parameters.crackSpecs)

End Sub
```

```
Public Sub BAMpF_Iter_Click()
Sheet1.Cells(1, 11) = Sheet1.Cells(1, 11) + Sheet1.Cells(1, 9)
Application.DisplayAlerts = False
If Sheet1.Cells(1, 1) = True Then
    result = BAMpF.runFirstBAMpFPrediction(1, 1, False)
    For i = 0 To parameters.crackSpecs.IPoints(0) - 1
        Sheet1.Cells(i + 3, 4) = Math.Sqr(result.crackParams.Px(0, i) ^ 2 + result.crackParams.Py(0, i) ^ 2)
        Sheet1.Cells(i + 3, 6) = result.kStress(0, i)
        Sheet1.Cells(i + 3, 3) = i
    Next i
    Sheet1.Cells(3, 7) = parameters.Stress
    Sheet1.Cells(1, 1) = False
Else

For i = 0 To parameters.crackSpecs.IPoints(0) - 1
DeltaP(0, i) = Sheet1.Cells(i + 3, 11)
Next i
result = BAMpF.runBAMpFPrediction(1, 1, False, Sheet1.Cells(1, 11), DeltaP())
    For i = 0 To parameters.crackSpecs.IPoints(0) - 1
        Sheet1.Cells(i + 3, 4) = Math.Sqr(result.crackParams.Px(0, i) ^ 2 + result.crackParams.Py(0, i) ^ 2)
        Sheet1.Cells(i + 3, 6) = result.kStress(0, i)
        Sheet1.Cells(i + 3, 3) = i
    Next i

End If

End Sub
```

API Example

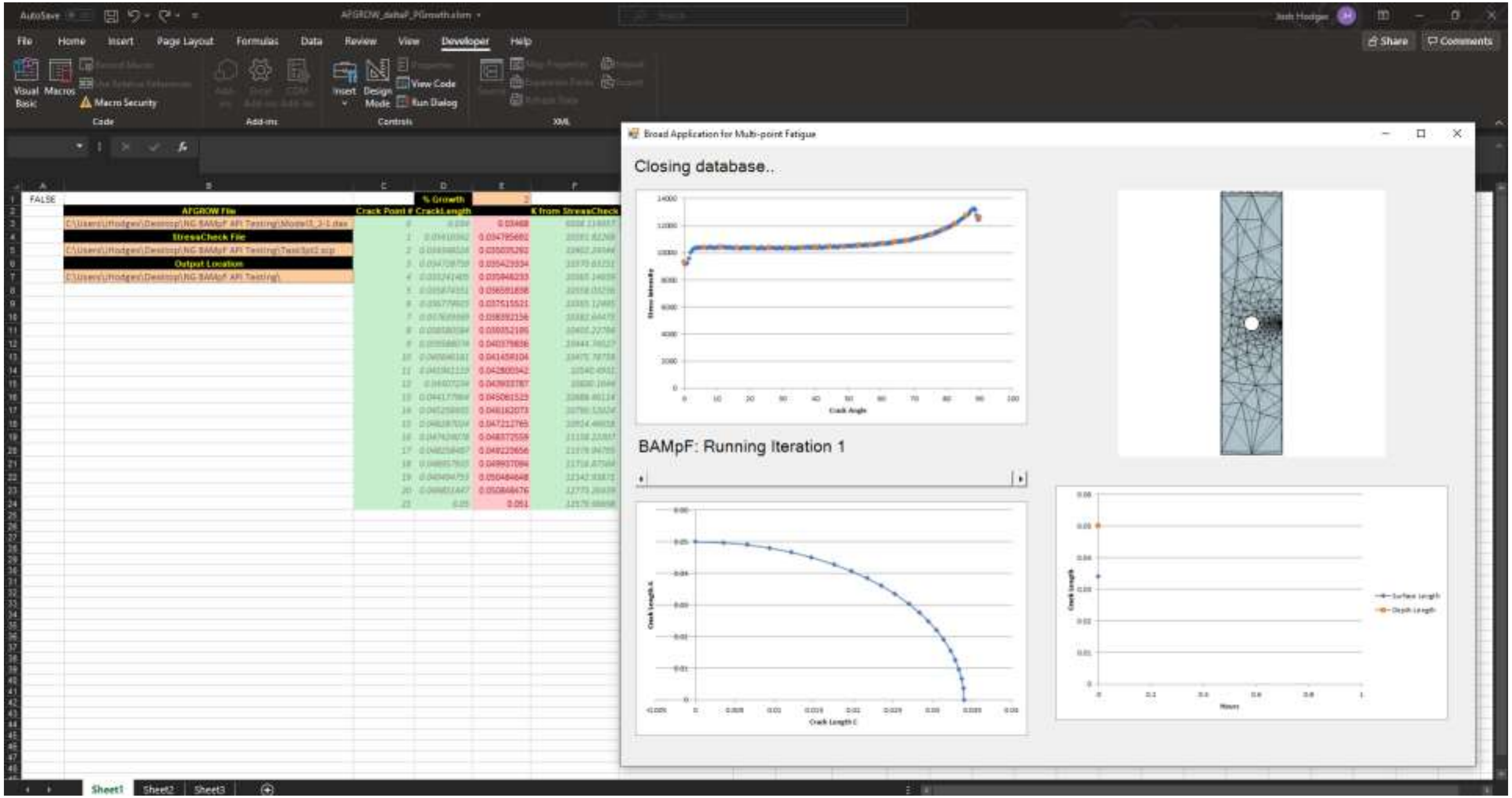
TRUE			% Growth	2				Total Cycles	0
	AFGROW File	Crack Point #	CrackLength		K from StressCheck	Stress From StresCheck	Cycles	FinalC	DeltaP
	C:\Users\JHodges\Desktop\NG BAMpF API Testing\Model3_2-1.dax			0					0.0000
	StressCheck File			0					0.0000
	C:\Users\JHodges\Desktop\NG BAMpF API Testing\Task3pt2.scp			0					0.0000
	Output Location			0					0.0000
	C:\Users\JHodges\Desktop\NG BAMpF API Testing\			0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000
				0					0.0000

Initialize BAMpF

BAMpF Iteration

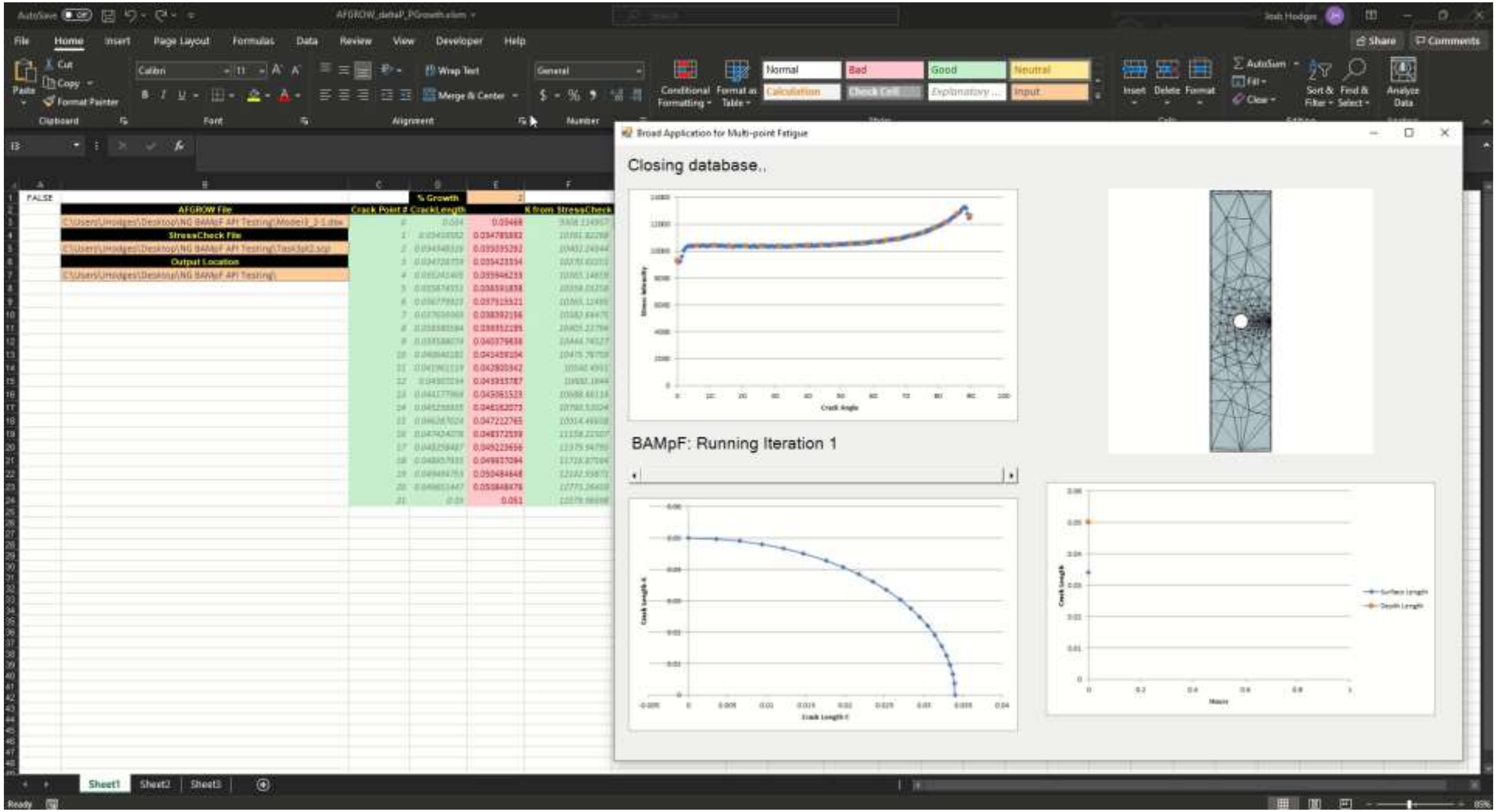
Run AFGROW

API Example

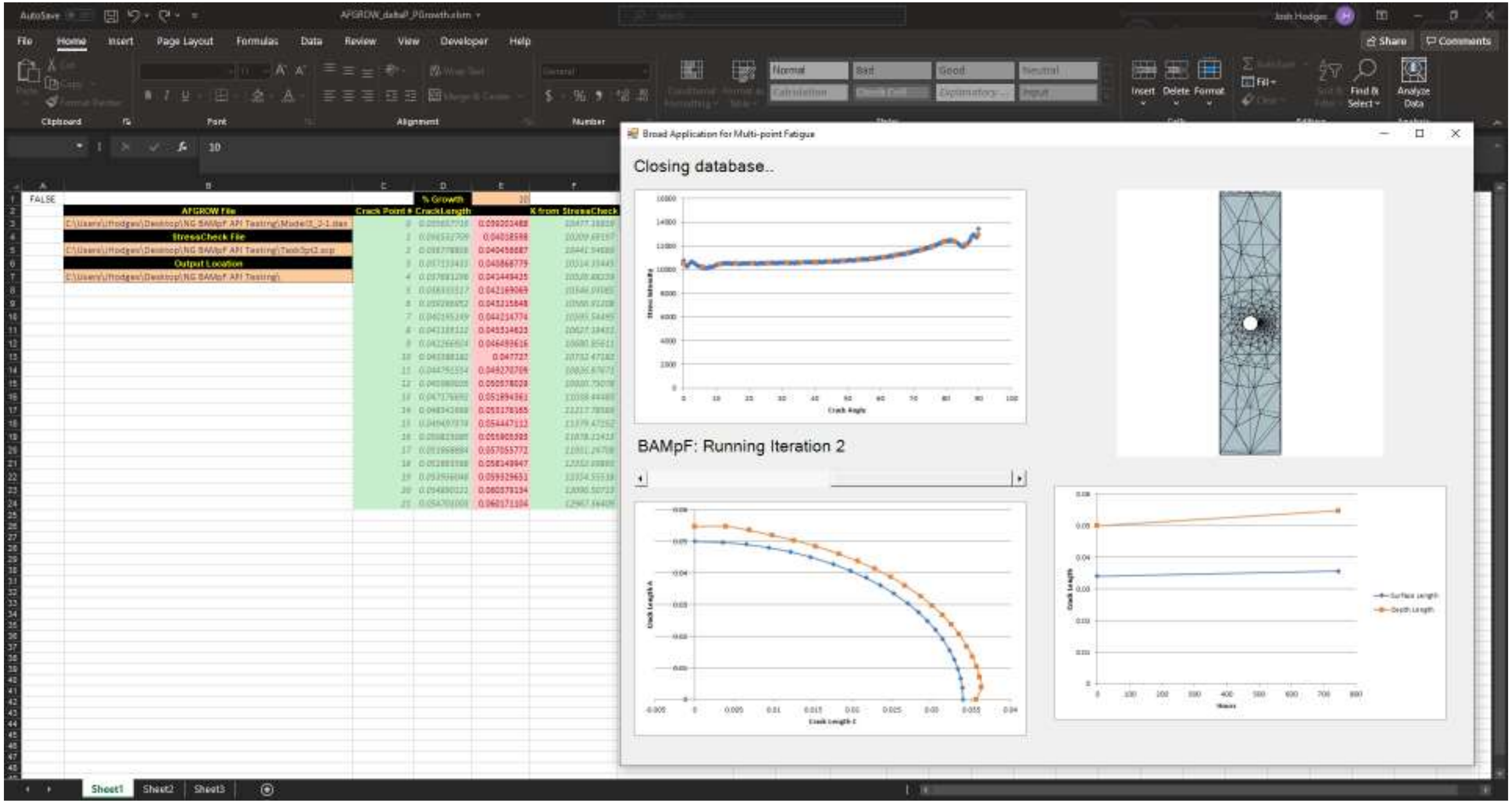


Broad Application for Multi-point Fatigue
Closing database..
BAMpF: Running Iteration 1

API Example

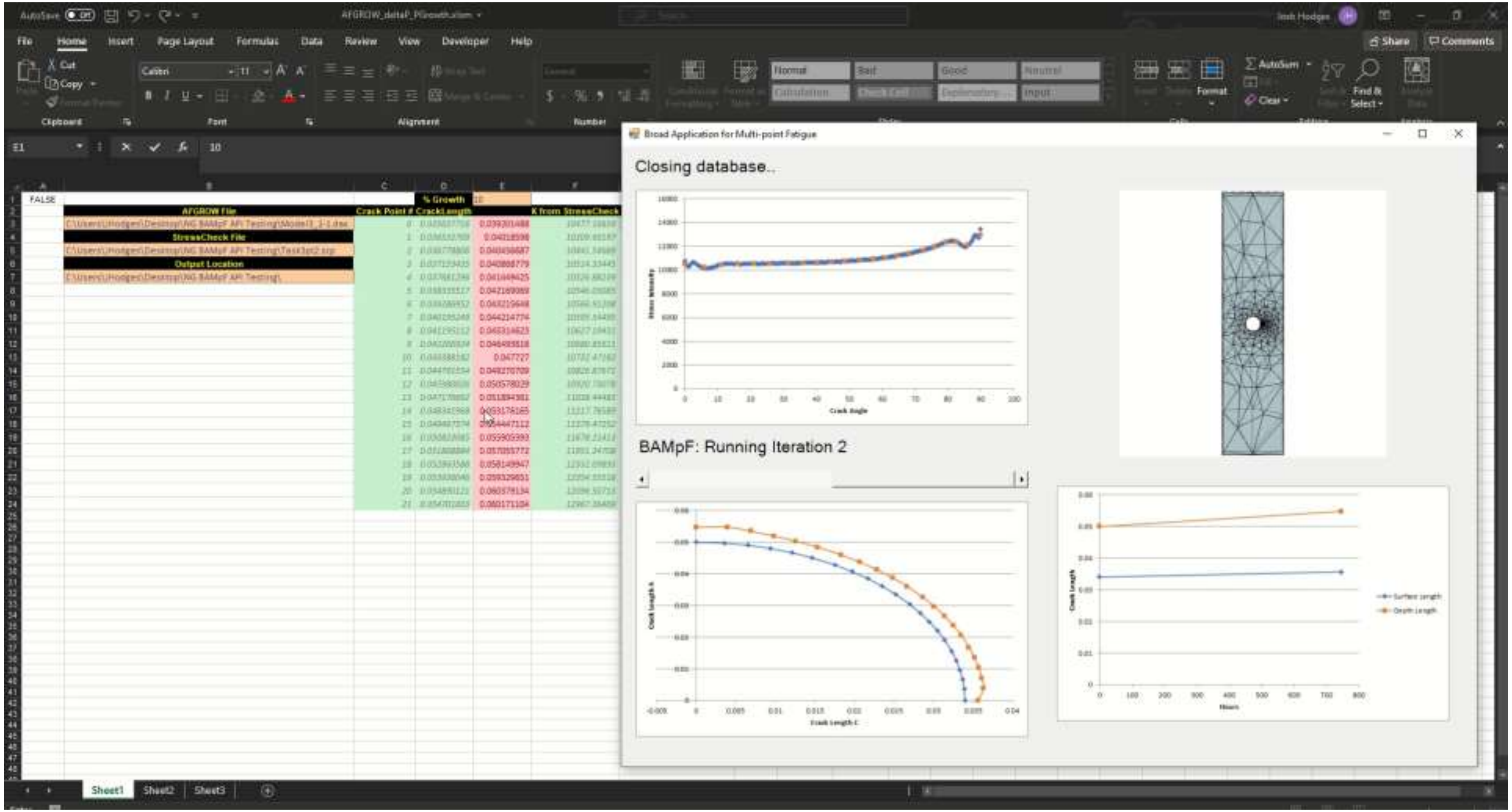

Broad Application for Multi-point Fatigue
Closing database..
BAMpF: Running Iteration 1
AFGROW File
StressCheck File
Output Location
% Growth
Crack Point #
CrackLength
K from StressCheck
Sheet1
Sheet2
Sheet3

API Example

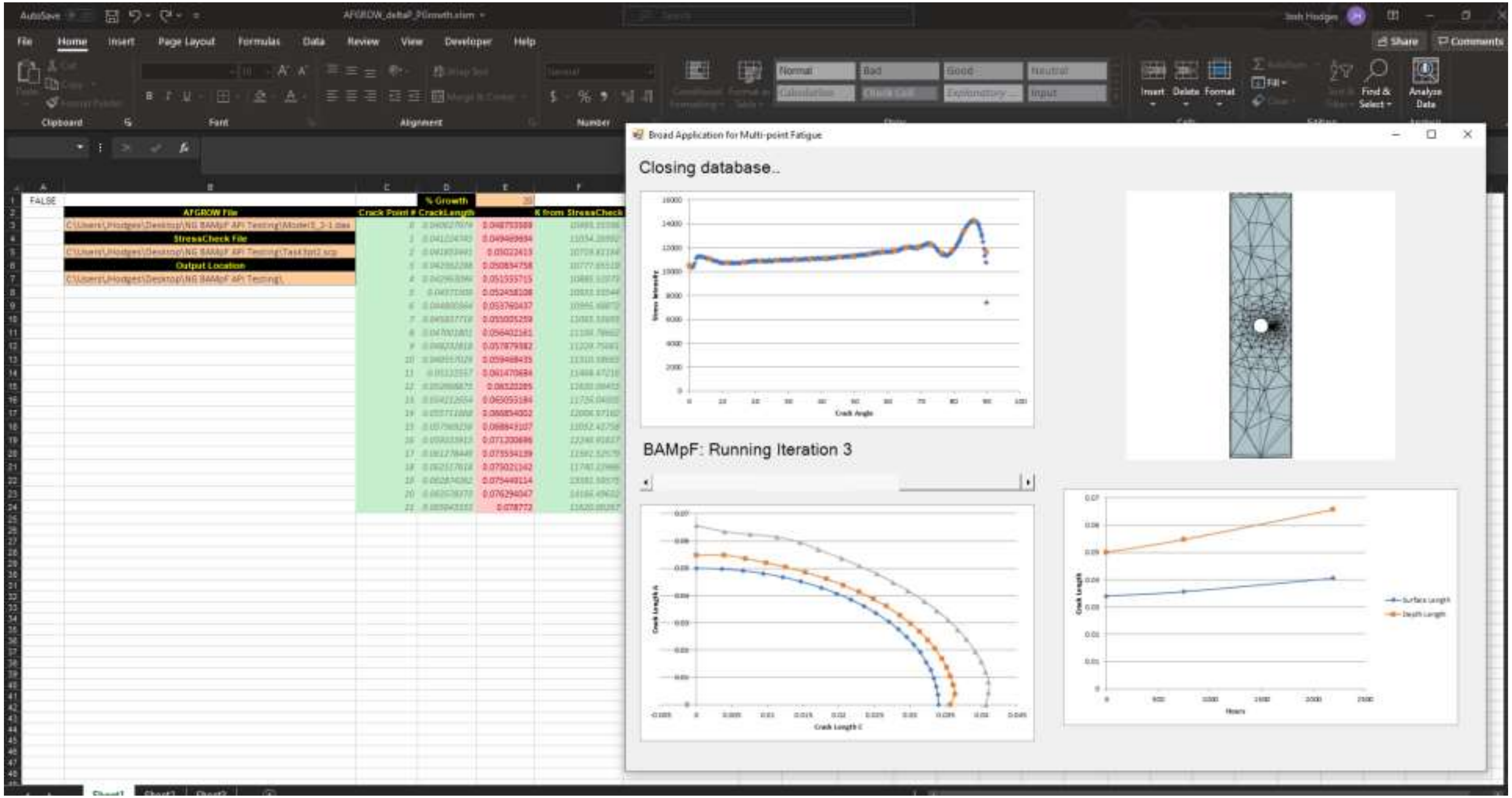


Broad Application for Multi-point Fatigue
Closing database..
BAMpF: Running Iteration 2

API Example


Broad Application for Multi-point Fatigue
Closing database..
BAMpF: Running Iteration 2

API Example

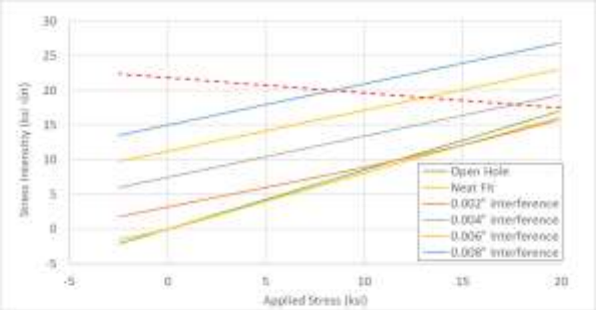


Broad Application for Multi-point Fatigue
Closing database..
BAMpF: Running Iteration 3

BAMpF K_{total}-array

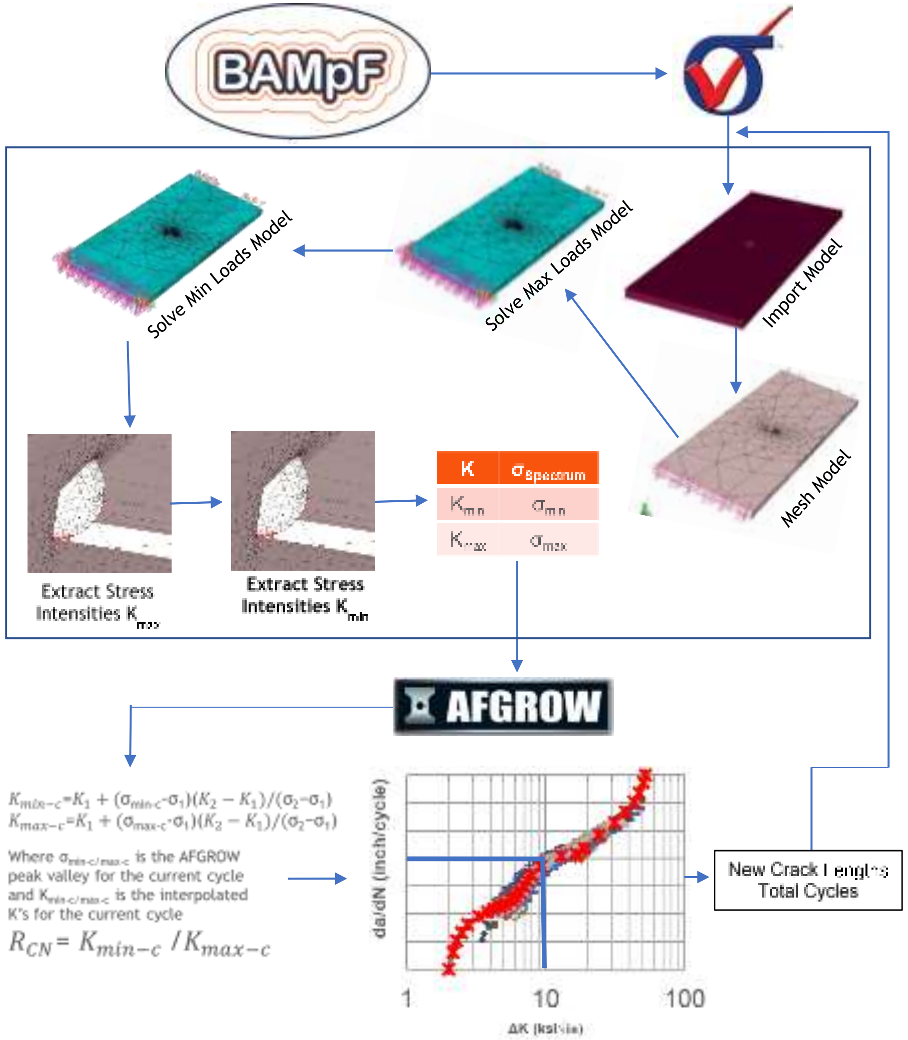
- **AFGROW traditionally uses a K/σ for plug-in**
  - Assumes K is linear with σ
  - Assumes K=0 at no applied loads
- **Stress intensities at interference fit fasters and neat fist fasteners do not satisfy these assumptions**
- **Capability has been added to AFGROW plug-in and BAMpF to account for these situations**
  - Rather than passing K/σ for each point an array is passed to AFGROW which can account for non-linear stress intensities over the entire spectrum
  - AFGROW will interpolate to get K_{max} and K_{min} for growth calculations

Applied Load (ksi)	K (ksi-√in)
-2.48	8.48
0	9.6
19.8	19.3



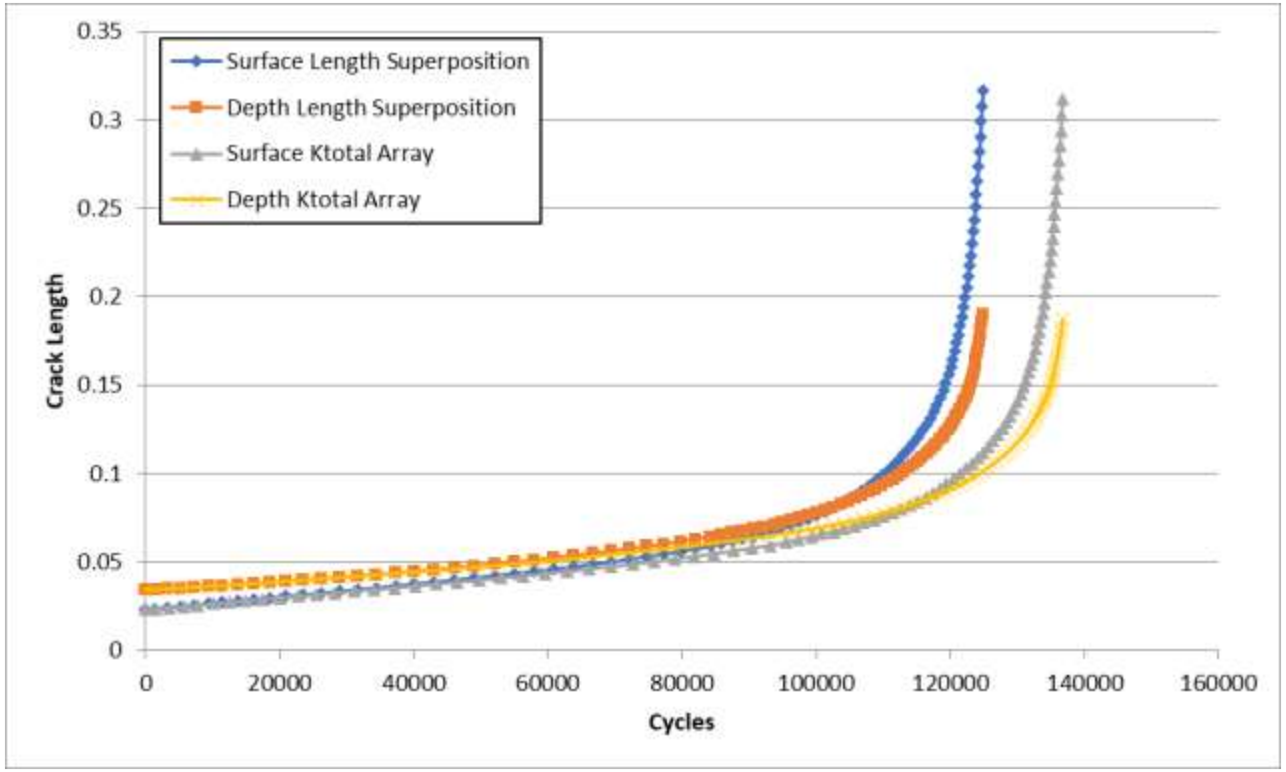
BAMpF K_{total}-array flow chart

- Models are complex and can include contact
- ERSI Round Robin #2 models solved were:
  - 88 growth steps
  - 2 solutions per step
  - 20 contact iterations per solution
  - ~3500 solutions total
  - 60 hours of run time
- For this case, only 2 stress levels are needed since the spectrum is constant amplitude with R=0.1
- Other scenarios could require more than 2 stress levels



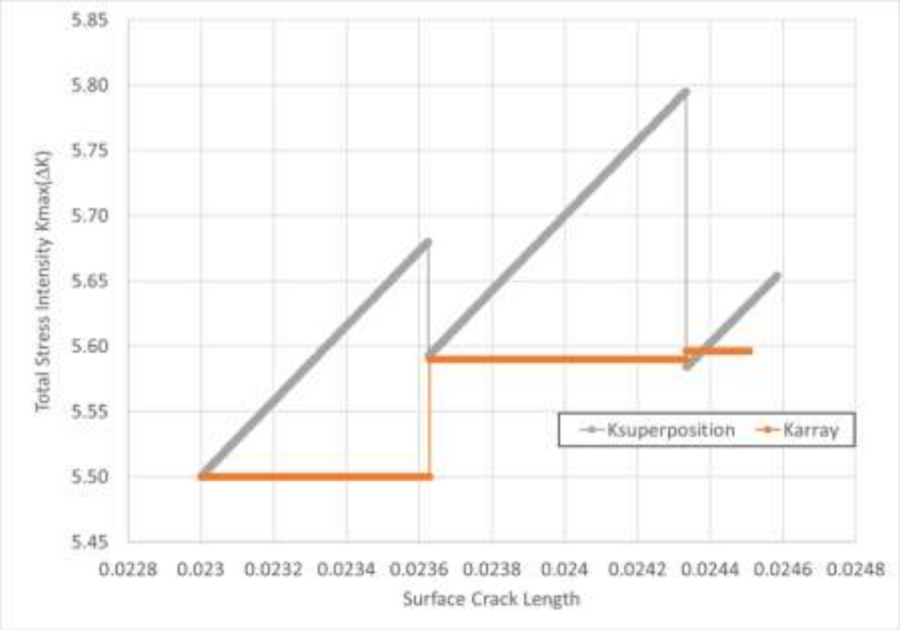
AFGROW Residual Stress Calculations

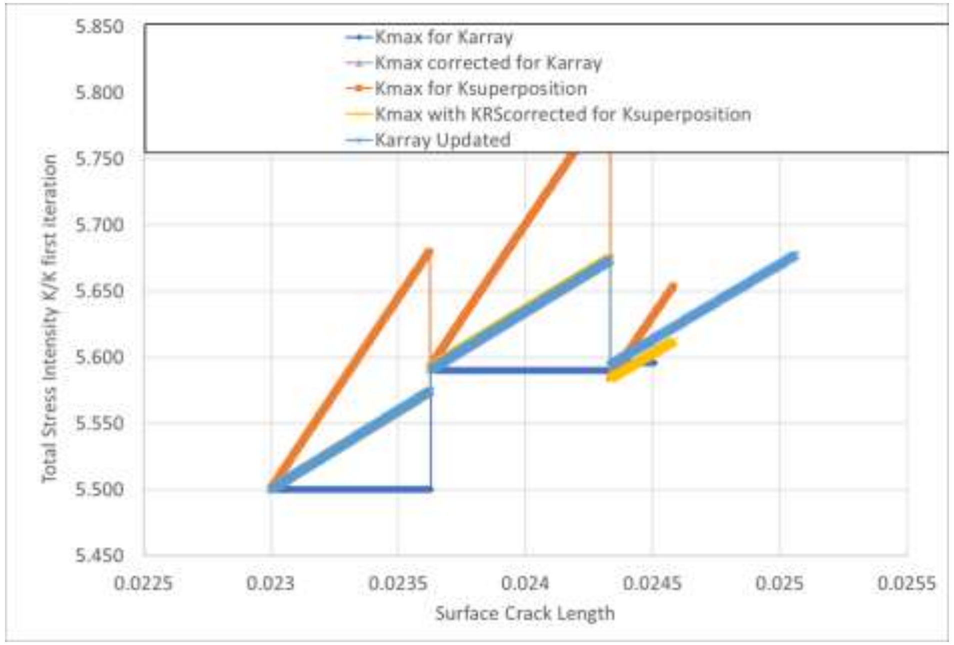
- **Testing the K_{total}-Array method with ERSI Round Robin #1, Condition 2**
  - If method is implemented correctly, it should produce identical results to current superposition method used in AFGROW/BAMpF
  - Unfortunately, there were differences



AFGROW Residual Stress Calculations

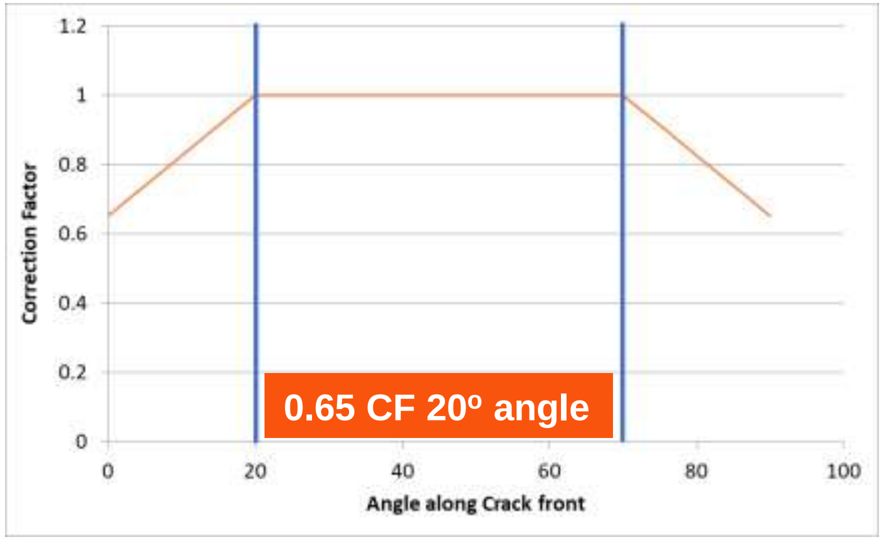
- **Differences in lives related to different K calculation methods**
  - For $K_{superposition}$ AFGROW utilized a constant beta between recalculations
    - Will increase $K_{applied}$ until a new beta is calculated at a rate of $\frac{\sqrt{a_j}}{\sqrt{a_1}} = CF$
    - Only $K_{applied}$ inreases, K_{rs} remain constant
  - For K_{total}-array AFGROW utilized a constant K between recalculations
  - AFGROW release fixed both the constant K_{rs} for $K_{superposition}$ and the K_{total} for the K array method by $\frac{\sqrt{a_j}}{\sqrt{a_1}} = CF$

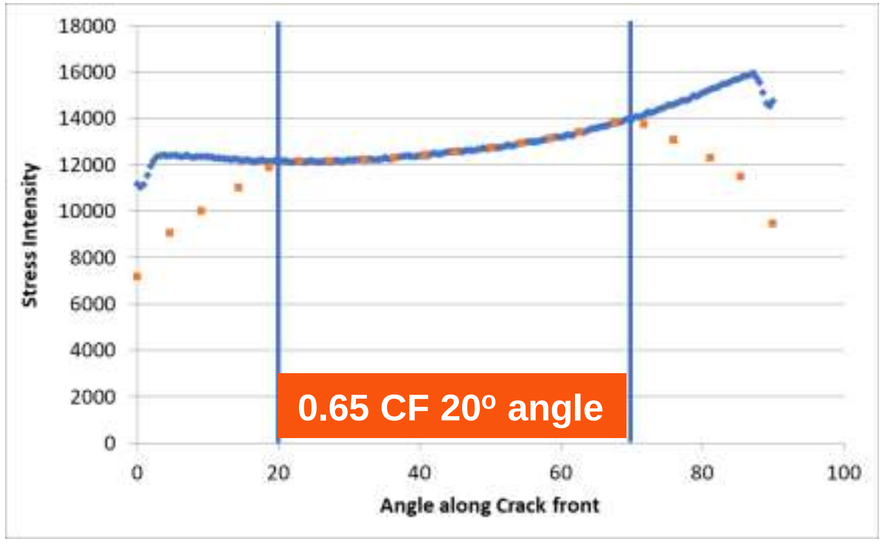




BAMpF Surface Corrections

- Utilizing marker band and fatigue test data determine what is necessary to match multi-point crack growth analysis to test data
- A 20º interpolation angle with a correction factor that appears to be a function of plane strain and plane stress fracture toughness
- Corrections are made to the extracted K's from StressCheck





BAMpF Surface Corrections

- ❑ Corrections have been updated for through thickness and unique crack shapes
- ❑ Understanding of what these corrections physically represent are on-going

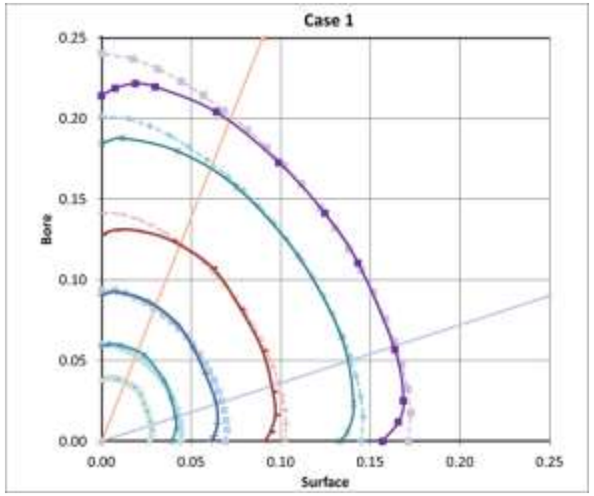
StressCheck Model Information

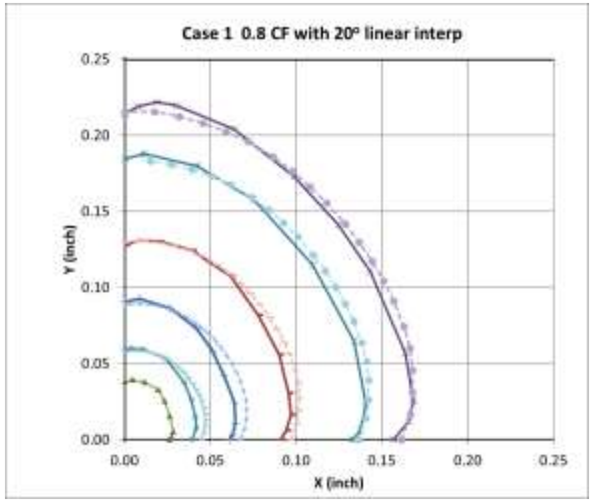
Model Info | Parameters | Rules

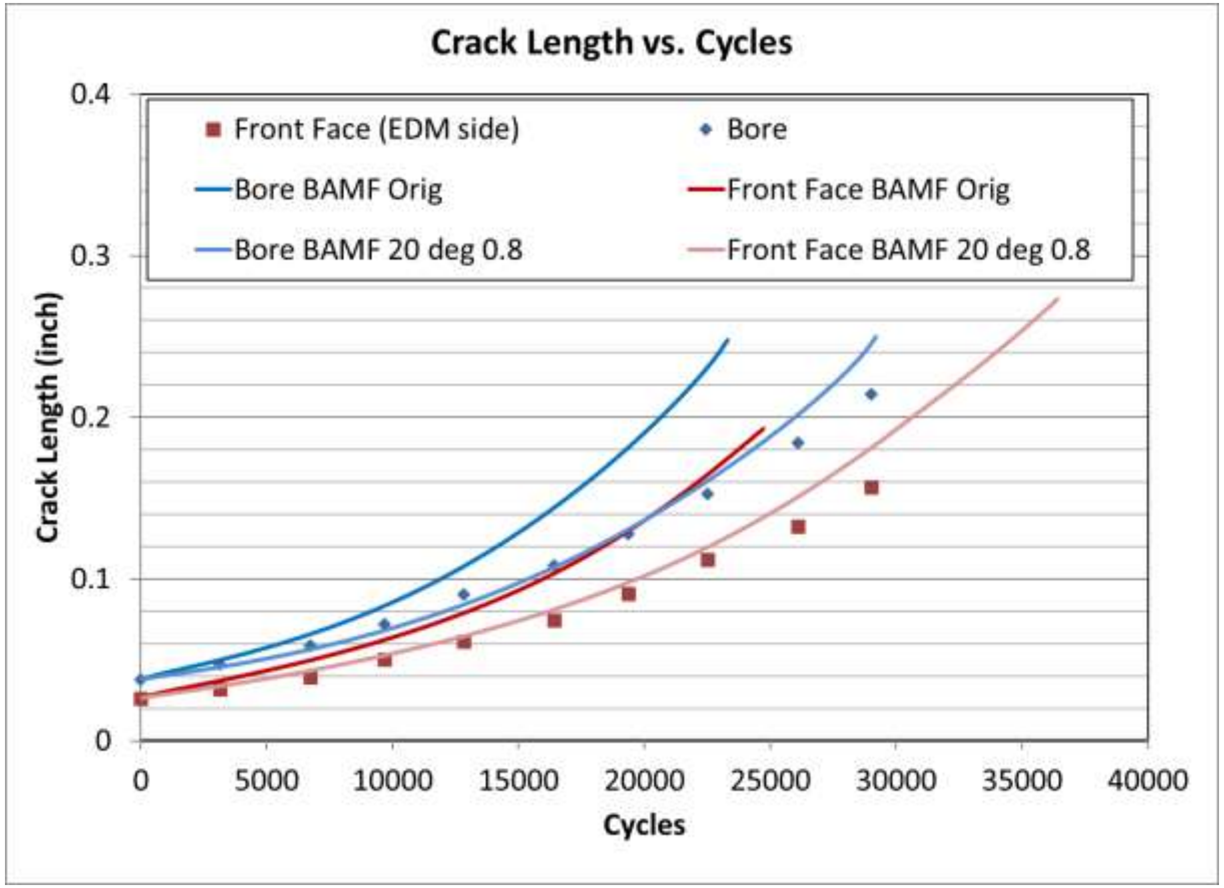
Name	Description	Expression	Value	Limit	Class	Sort
BAMF_CSP_Angle1	MaxAngle		2.0000e+01		General	
BAMF_CSP_Factor1	SurfaceCorrectionFactor		6.5000e-01		General	

```
Public Function SPCFEquation(ByVal PointAngle As Double, ByVal MaxAngle As Double, SurfaceCorrectionFactor As Double) As Double

    Return ((1 - SurfaceCorrectionFactor) / MaxAngle) * PointAngle + SurfaceCorrectionFactor

End Function
```

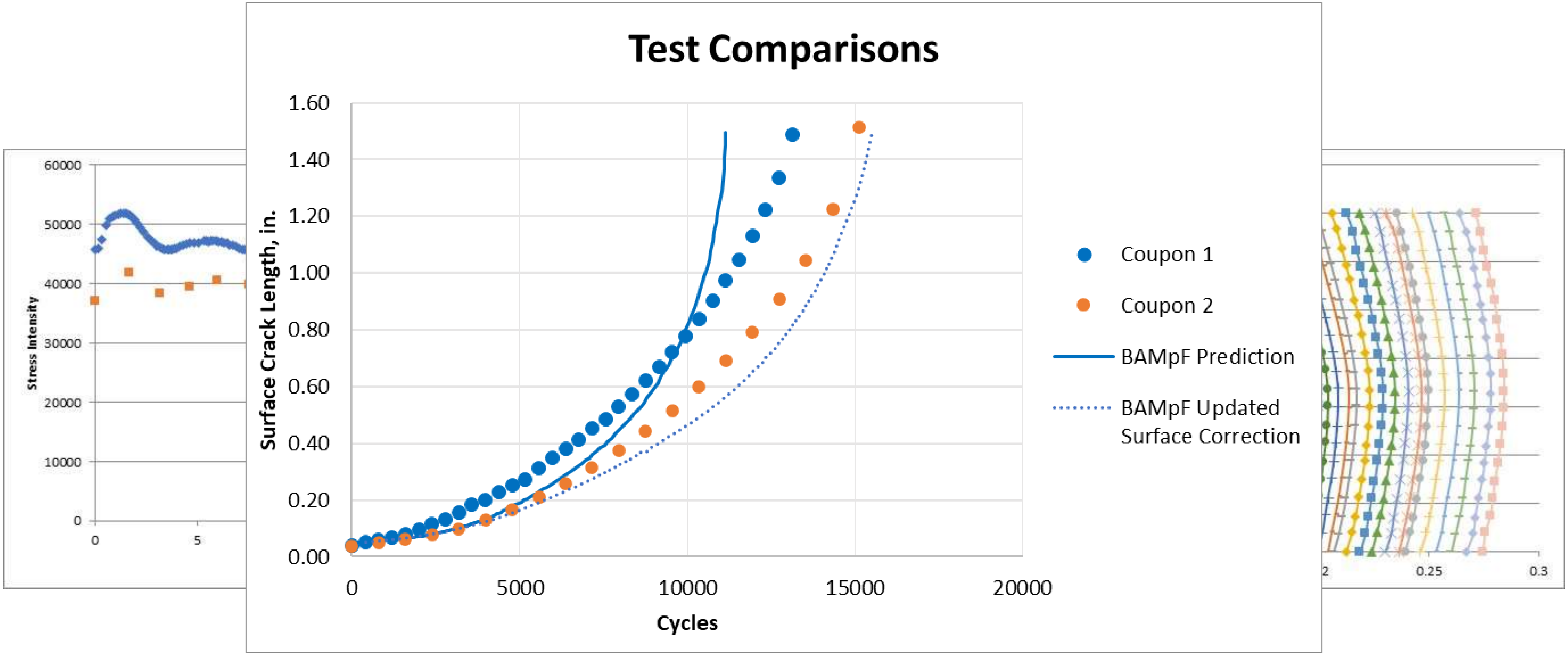
BAMpF Surface Corrections







BAMpF Surface Corrections



Retardation and BAMpF

- ❑ **When utilizing BAMpF with retardation if points are not evenly spaced issues can occur with crack growth**
- ❑ **Every iteration points are respaced and updated in AFGROW**
  - ➢ This causes issues with retardation as the retardation state is not updated
  - ➢ If the new crack length becomes less than the previous crack length
    - ❑ Occurs when transition through geometry or on the 1st iteration if points are not spaced evenly
    - ❑ It will lead to effective stress ratios that are >1 causing no growth

Retardation and BAMpF: What fixes were implemented?

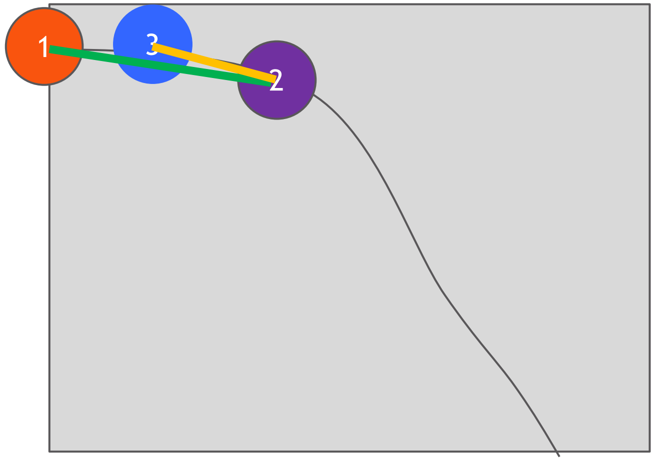
- **AFGROW added capability to get and set the retardation parameters**
  - ret_eff_length_old: previously saved effective length (length + yield)
  - ret_length_old: previously saved length
  - ret_length_cf_old: previously saved crack length + 0.25 of the yield zone
  - ret_length_ol: crack length at the last overload
- **BAMpF added capability to:**
  - Get the parameters
  - Interpolate between the parameters
  - Set the parameters

```
If Count > 0 Then
    ret_eff_length_old(k, i) = ISPredictManage.GetCrackDirectionParameter("Crack " & k, "P" & i, "ret_eff_length_old")
    ret_length_old(k, i) = ISPredictManage.GetCrackDirectionParameter("Crack " & k, "P" & i, "ret_length_old")
    ret_length_cf_old(k, i) = ISPredictManage.GetCrackDirectionParameter("Crack " & k, "P" & i, "ret_length_cf_old")
    ret_length_ol(k, i) = ISPredictManage.GetCrackDirectionParameter("Crack " & k, "P" & i, "ret_length_ol")
Else

If Count > 1 Then
    ISPredictManage.SetCrackDirectionParameter("Crack " & k, "P" & i, "ret_eff_length_old", ret_eff_length_old(k, i))
    ISPredictManage.SetCrackDirectionParameter("Crack " & k, "P" & i, "ret_length_old", ret_length_old(k, i))
    ISPredictManage.SetCrackDirectionParameter("Crack " & k, "P" & i, "ret_length_cf_old", ret_length_cf_old(k, i))
    ISPredictManage.SetCrackDirectionParameter("Crack " & k, "P" & i, "ret_length_ol", ret_length_ol(k, i))
End If
```

Retardation and BAMpF: What fixes were implemented?

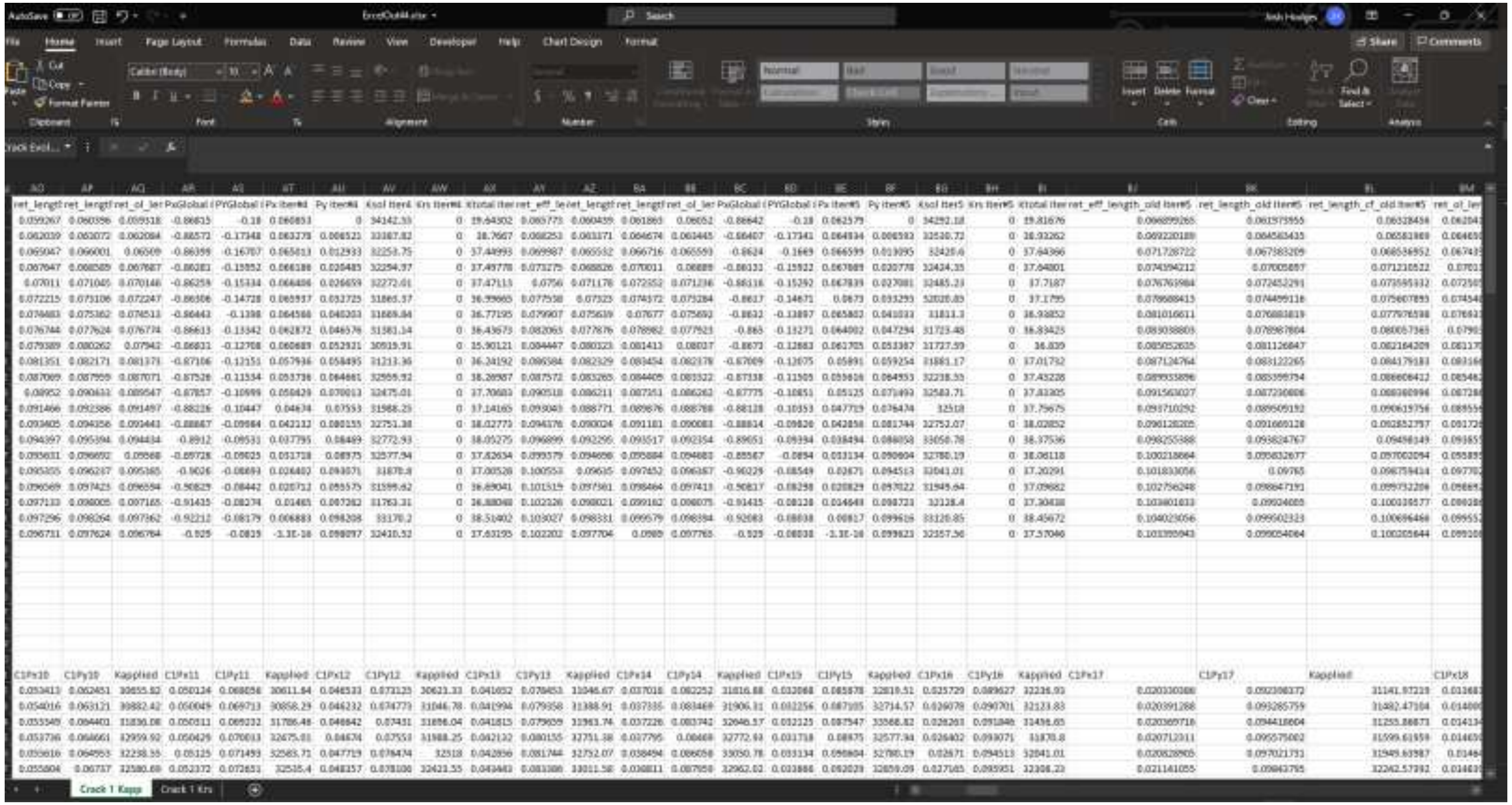
- ❑ **Interpolation required on the crack front (x,y,retardation parameter)**
- ❑ **Find the distance between points and interpolate on those**
  - ➢ Assumes curvature on crack front is small between points
  - ➢ Assumes retardation parameters are linear between points

$$R_3 = R_1 + \sqrt{(Px_3 - Px_1)^2 + (Py_3 - Py_1)^2}\,\frac{R_2 - R_1}{\sqrt{(Px_2 - Px_1)^2 + (Py_2 - Py_1)^2}}$$



Retardation and BAMpF:BAMpF Output

- Excel output to track retardation values
- Allows for future improvements to properly restart BAMpF runs that utilize retardation



BAMpF Hidden Feature Parameters-Crack Features

- BAMpF_ElementCF#: Changes the default number of elements along the crack face. Default is 1.25x # of points
- BAMpF_Ktol#:Defines the number of points to use in the K tolerance smoothing algorithm. Default is 4.
- BAMpF_Layers#:Allows the user to change the number of layers the boundary layer refinment uses. Default is 2
- BAMpF_Kconverge: Allows BAMpF to solve 1 solution above and 1 below the set p-level. The value the
- BAMpF_SymCrack#: Allows the user to define that a crack surface is a symmetry face
- BAMpF_MDCrackAngle#: Input value that allows for multi-directional crack growth.
- BAMpF_Refine:Provides the user capability to add to utilize guidelines from StressCheck for a more refined boundary layer mesh. This increases the elements of a model around 35%.

BAMpF Hidden Feature Parameters-New Features

- BAMpF_CSP_Angle#: Allows the user to correct the surface points. Currently the correction will go to the input angle linear interpolated from the surface by the input correction factor.
- BAMpF_CSP_Factor#: Allows the user to correct the surface points. Currently the correction will be maximum correction at the surface linear interpolated to the input angle.
- BAMpF_KtotalArray: Triggers BAMpF to solve multiple solutions for the Ktotal array approach, value is set to the number solutions to be solved.
- BAMpF_Contact#: Allows the user to ignore extraction points on the surface when utilizing contact in models. Contact adjacent to a crack has shown to unnaturally alter the first couple of points in an extraction. The value is set to the number of points to ignore.

BAMpF Hidden Feature Parameters-Debugging

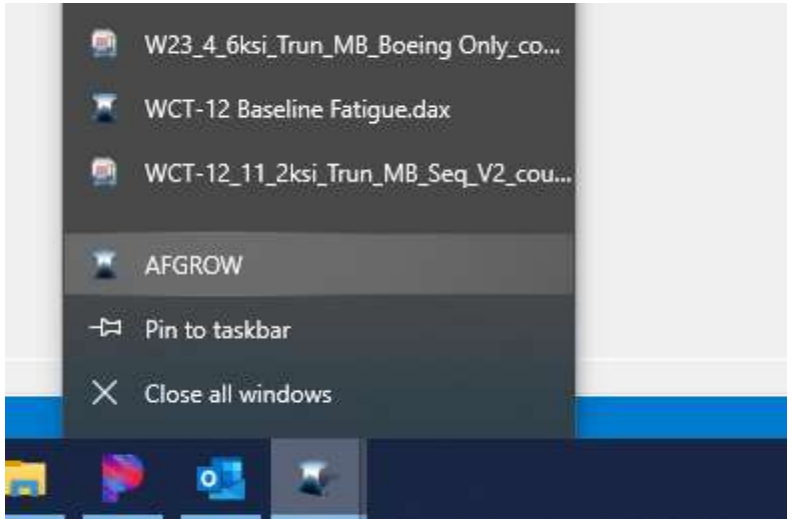
- ~~BAMpF_Debug: Triggers BAMpF to enter Debug mode. This provides data that can be given to BAMpF technical support to aide~~
- BAMpF_Locate: Allows the user to use selection sets if desired (currently not recommended)
- BAMpF_SaveSCP: Saves a solved .scp file for each iteration. Good for models that take a long time to run and reviewing individual models is desired
- BAMpF_Respline: Triggers BAMpF to not enter the re-spline routine, this aides in debugging issues to determine where the model is failing
- BAMpF_Visible: Allows the user to visualize StressCheck while BAMpF is operating

Released 20 February 2020

BAMpF 7.0 REPORTED BUGS AND ISSUES

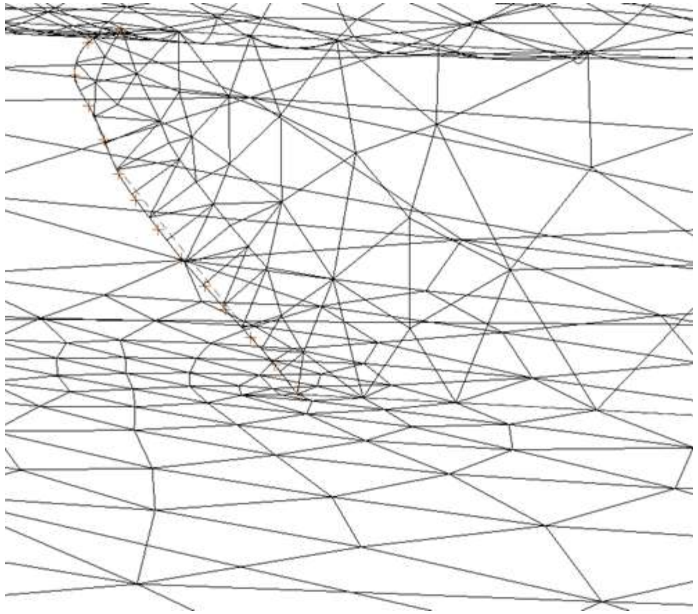
Bug: Launching AFGROW while BAMpF is Running

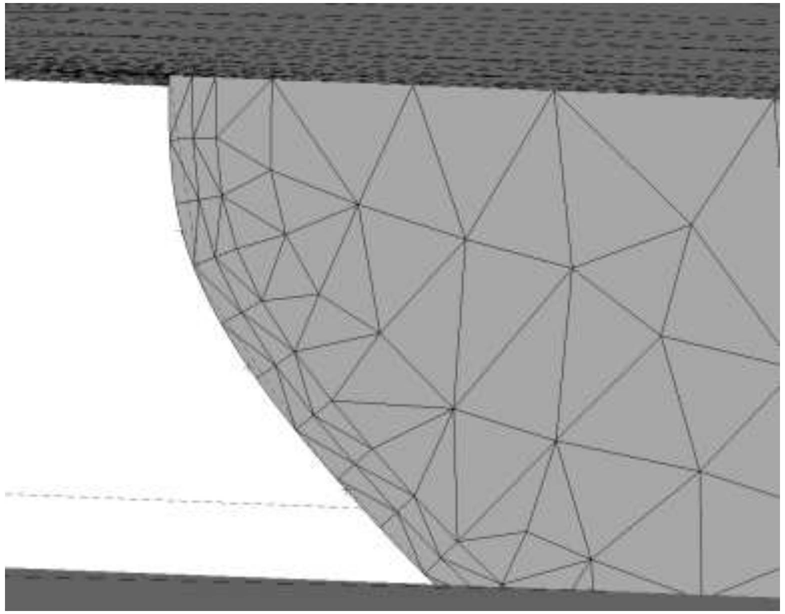
- **Bug: If AFGROW is launched through double clicking an analysis file BAMpF will freeze on the next iteration**
  - Extremely frustrating when running an analysis in the background
  - Work around, launch AFGROW through the task bar



Bug: Boundary Layer Refinement

- **Bug: Boundary layer refinement for cracks is not stable and not consistent**
  - Updates to meshsim's boundary layer parameters caused BAMpF's default meshing to be poor for through cracks causing refinements became large and unusable
  - Additionally, default meshes showed larger errors in handbook solutions (Stanfield AFGROW 2019)
    - Stanfield and StressCheck proposed utilizing a refine mesh based on updates to StressCheck Master Guide



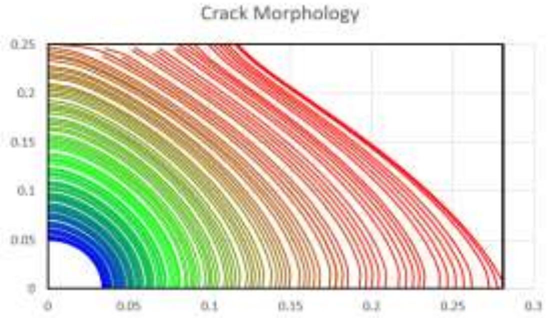


Fixed: Boundary Layer Refinement

- **Fixed: Added refinement capability:**
  - When transition to through thickness occurs thickness, rather then crack length, is utilized for To and T-total calculations. This produces smaller boundary layers which are better at picking up small changes in crack shape
  - Refinement capability was added by using BAMpF_Refine parameter
    - Default Mesh parameters:
      - Ratio=0.8/(number of points)
      - Layers=2
      - To=0.15^2*Surface Length
      - T-Total=2* To
    - Refine Mesh parameters:
      - Ratio=0.4/(number of points)
      - Layers=3
      - To=0.15^3*Surface Length
      - T-Total=6* To

Bug: API BAMpF Bugs

- **Bug: The BAMpF_Debug parameter does not work**
  - Updates to BAMpF have made this an obsolete feature
  - Plans to utilize an error log to help the user in debugging their BAMpF runs
- **Bug: BAMpF_Kconvergence parameter triggers a convergence check every iteration**
  - The parameter is intended to trigger a convergence check every # of iterations based on the parameter
  - The convergence check when triggered would be performed every iteration
  - The flag for convergence checking was not getting reset after the first instance forcing a convergence check every iteration
  - Bug has been patched
- **Bug: BAMpF Graphs were not working as intended after transition to through thickness**
  - Plotting features caused errors in finding the boundaries
  - Error was found in writing arrays to excel
  - Bug has been patched



Release 2022

BAMpF 8.1 UPGRADES

Proposed Feature List from Previous Workshop

Feature	Description
Multi-Channel Loading	Solve for multiple loading scenarios
~~Integration with other FE tools~~	~~Ansys, NX (Funding Sources Phase 2 landing gear, A-10)~~
K extrapolation	Extrapolate K to radius of 0 using two different K extractions
Multi-Crack Auto-Coalescence	(MCAC) When crack fronts intersect-automatically coalesce cracks
~~Filled Hole Analysis capability~~	~~Kmax and Kmin may not be linear when contact (neat/interference fit)~~
3D crack growth	**Crack turning, out of plane crack growth**
~~K colored plots~~	~~Implement K colored plots into AFGROW outputs~~
Restart Analysis	**Utilizing Extraction folder restart at last analysis point**
Pause Analysis for UI	**Allows user to pause an analysis to update the start.scw file**
Custom Output files	**Use a customizable .ini file to allow the user to tailor output files**
Crack front splitting	When the crack front splits into multiple cracks
Error Log file	**Log file tracks when exceptions have been reached in the code**
Updates to API	**Continue to Update API to create 1 version of BAMpF Code**