An Ultrafast Crack Growth Lifing Model to Support Digital Twin, Virtual Testing, and Probabilistic Damage Tolerance Applications

Juan D. Ocampo
St. Mary's University

Harry Millwater, Nathan Crosby
University of Texas at San Antonio

Beth Gamble, Chris Hurst
Textron Aviation (Cessna)

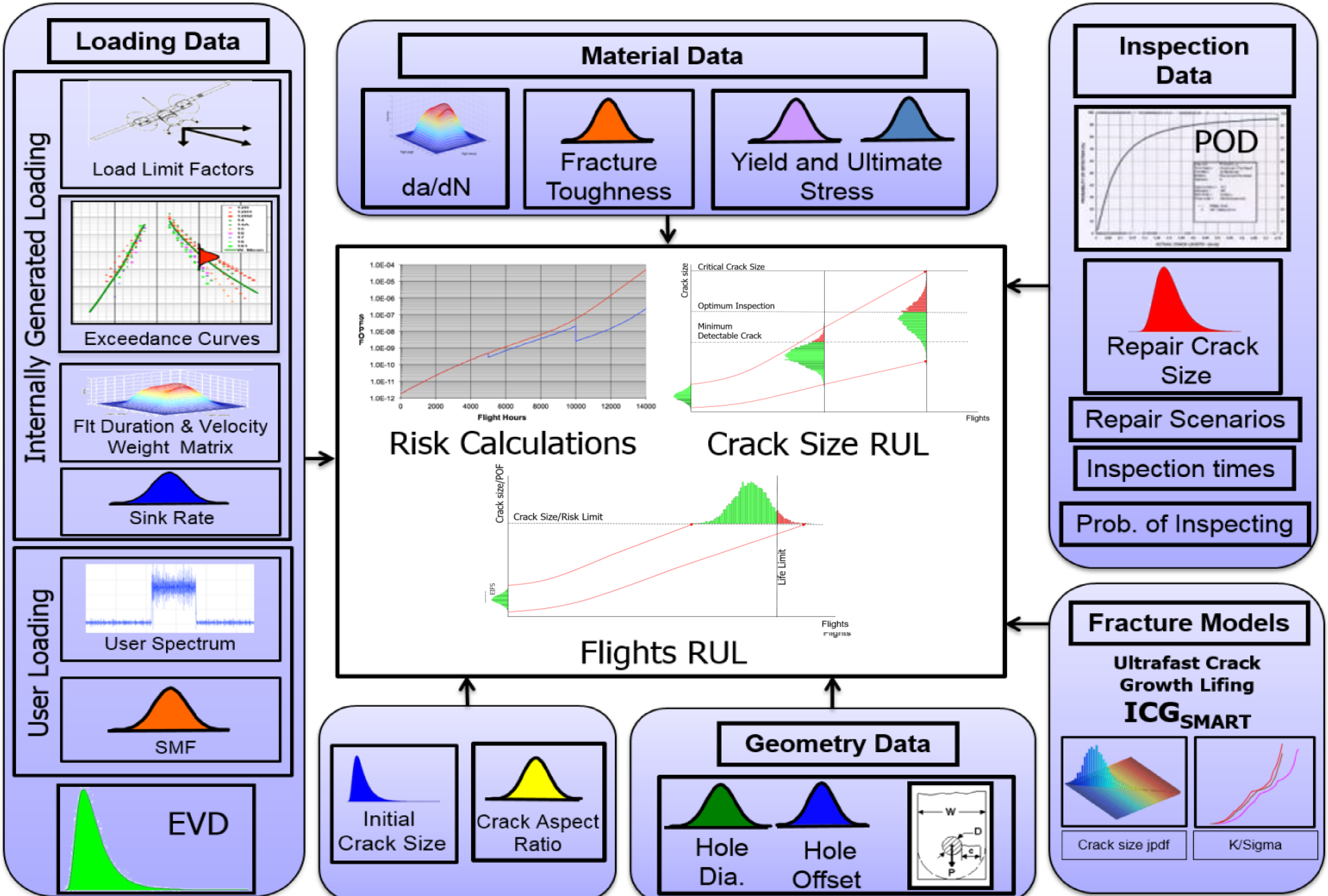
Marv Nuss
Nuss Sustainment Solutions

Michael Reyer, Sohrob Mottaghi
Federal Aviation Administration

30th ICAF Symposium – Kraków, Poland , 5–7 June 2019

Outline

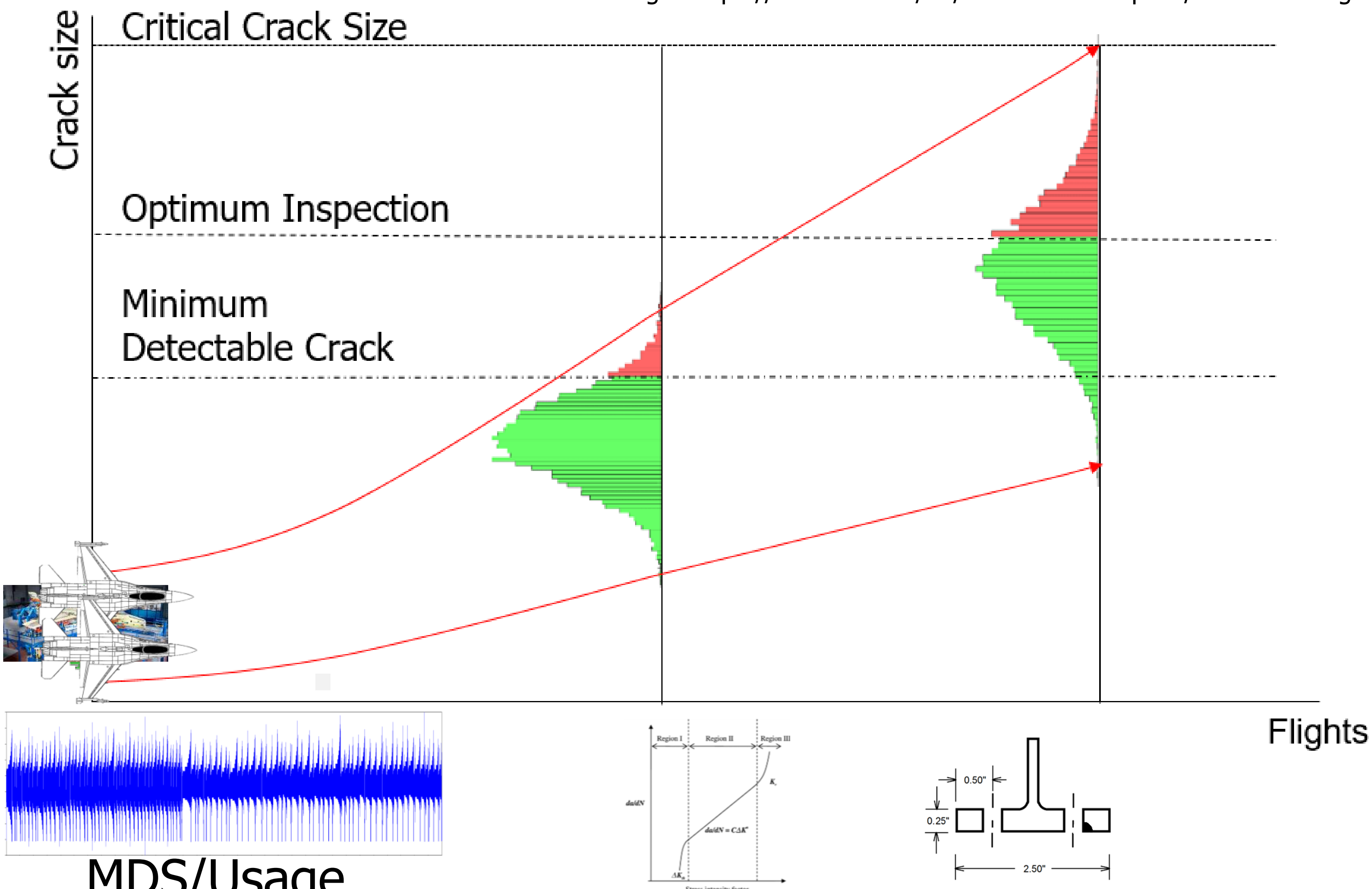
- SMART|DT Overview
- Motivation for an ultrafast crack growth
- Components of the ultrafast crack growth
  - Equivalent Spectrum
  - ODE solver
- Digital Twin, Virtual Testing, and Probabilistic Damage Tolerance Applications
- Conclusions
- Future Work



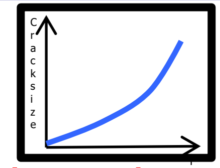
J. Ocampo et al. “Probabilistic Damage Tolerance for Aircraft Fleets Using the FAA-Sponsored SMART|DT” International Conference on Aeronautical Fatigue, Nagoya, Japan, June 2017.

Motivation

Image: https://mandanis.ch/en/bilder-und-beispiele/full-scale-fatigue-test/



Motivation



(only 3 random variables)

(N random variables)

- Typical run times w Monte Carlo (1B samples):
- 1) Master Curve:
  - 1 CG (30 sec), 1B interpolations->3 hrs on 8 processors
- 2) Kriging :
  - 400 CG (1/2 hr), 1B interpolations-> 20 hrs on 8 processors
- 3) Standard Monte Carlo, 1B samples
  - General CG: 30s/run on 8 processors = 43K days = 118 yrs!
  - If internal CG code 1000x faster -> 43 days
  - If internal CG code 10,000x faster -> 4.3 days
  - If internal CG code 100,000x faster -> 0.43 days = 10 hrs

Minimum improvement

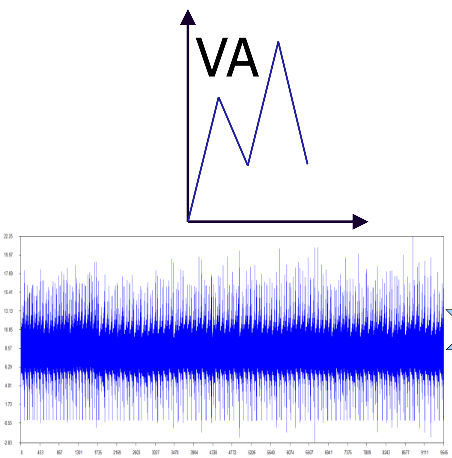
- 4) Numerical Integration
  - 100K CG -> 800 hrs on 1 processor
  - If internal CG code 1000x faster -> 0.8 hrs
- If internal CG code 1000x faster -> 0.8 hrs
- 5) Numerical Integration w Kriging
  - 400 ICG (2s), 100K interpolations-> 100s on 1 processor
- 6) Importance Sampling
  - Internal CG for optimization then 1K ICG -> 1 hr

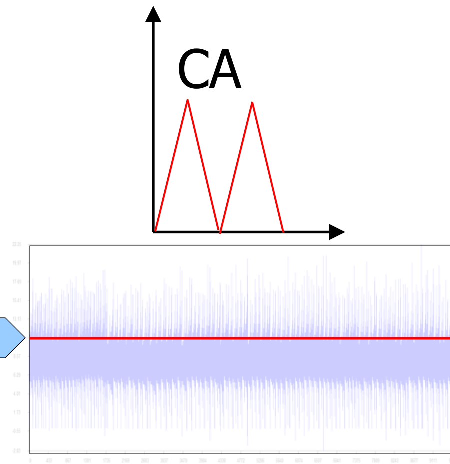
w/o inspection

Ultrafast Approach "Hypergrow"

1) Create an ***equivalent constant amplitude*** from an arbitrary spectrum
2) Use an internal ***adaptive time stepping*** Runge-Kutta algorithm to grow the crack (Cycles become the independent variable)
3) Collect the top 100 (or so) damaging realizations for further examination and potential reanalysis

"Hypergrow" CG Code





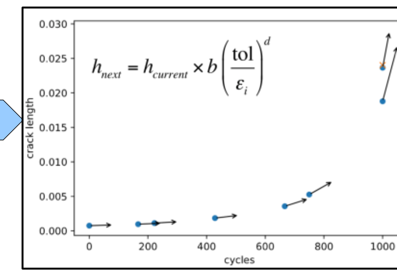
ODE Formulation

$$\frac{da}{dN} - C(\Delta K(a,c))^n = 0$$

$$\frac{dc}{dN} - C(\Delta K(a,c))^n = 0$$

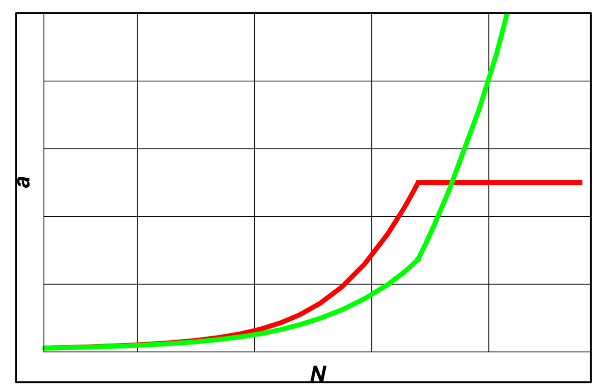
$$Initial\ Conditions: a(0) = a_i, c(0) = c_i$$

RK ODE Solver

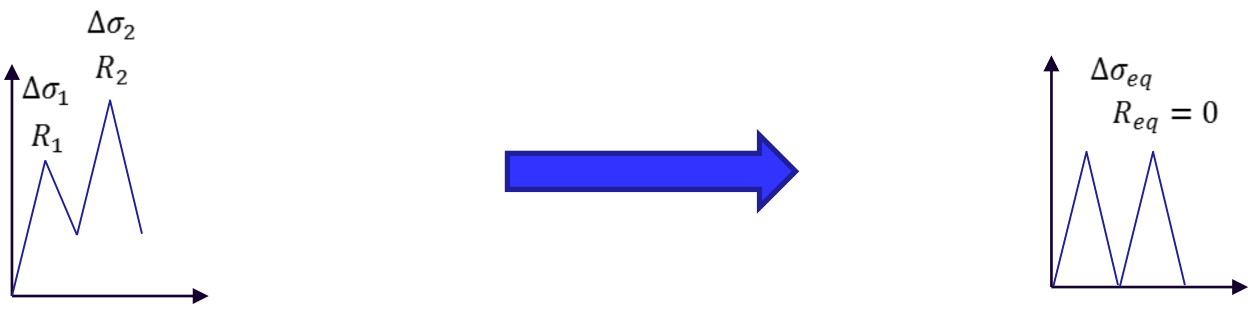


ICG Capabilities	
Method	4-5th order Runge-Kutta
Accuracy	Error controlled by user tolerance
Speed	~10,000/sec single proc.
Parallel	95% speedup on 8 proc.
K solutions	Newman-Raju, read beta tables

Crack Growth Result



Equivalent Stress



$$N_{total} = N_1 + N_2 = \int_{a_0}^{a_1} \frac{1}{f(\Delta\sigma_1, R_1, a)} + \int_{a_1}^{a_2} \frac{1}{f(\Delta\sigma_2, R_2, a)}$$

$$N_{total_{eq\_stress}} = \int_{a_0}^{a_2} \frac{1}{f(\Delta\sigma_{eq}, R_{eq}, a)}$$

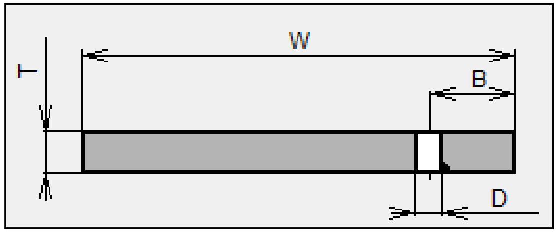
$$N_{total} = N_{total_{eq\_stress}}$$

$$\mathrm{Ds}_{eq} = \left[\sum_{i=1}^{K} \frac{n_i}{N_{Tot}} \left((1 - R_i)^{(m-1)n} \right) \mathrm{Ds}_i^{\,n} \right]^{1/n}$$

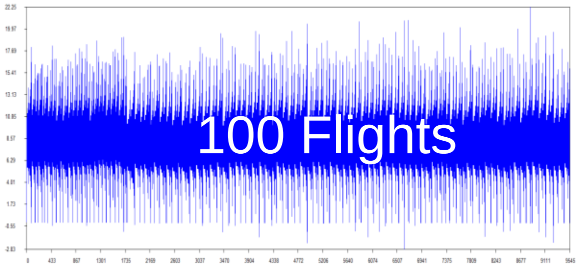
Liu. Y., Mahadevan, S. (2008), "Probabilistic fatigue life prediction using an equivalent initial flaw size distribution," International Journal of Fatigue, 31, 476–487.

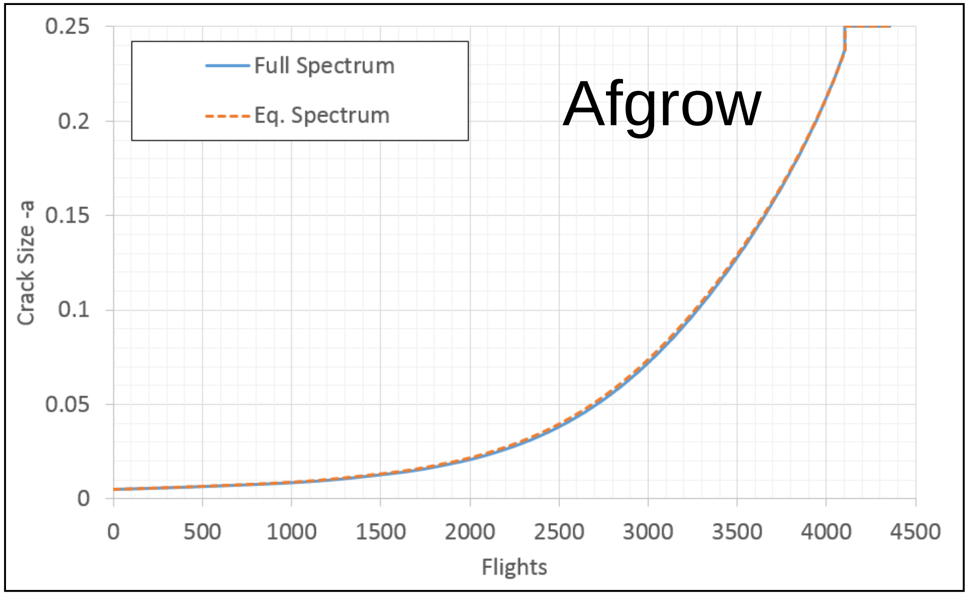
Eq. Stress Examples

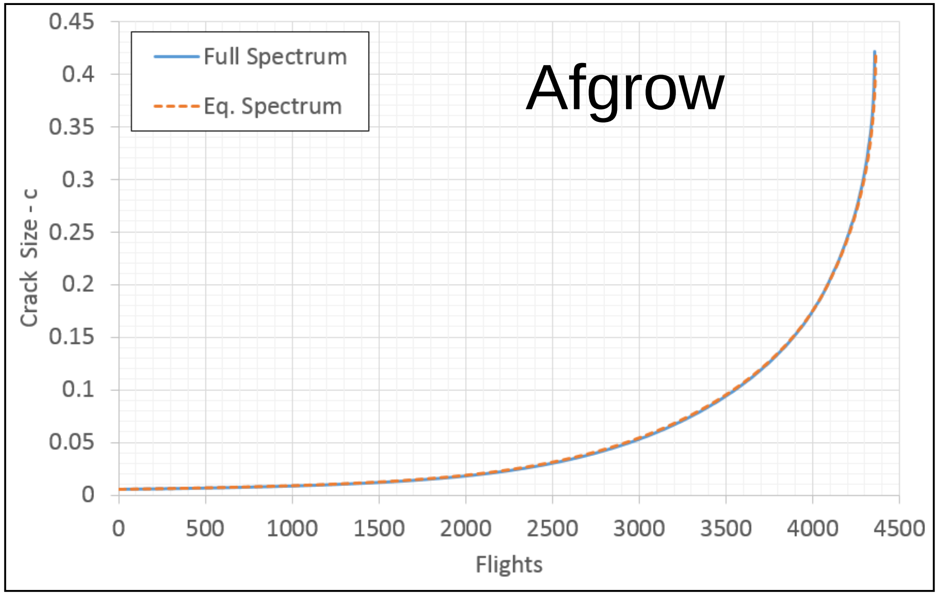
Corner Crack in a Hole



Variable	Value
Width	4 in.
Hole Offset	0.5
Thickness	0.25 in.
Hole Size	0.156 in.
Eq. spectrum	10.01 KSI
C	1.0E-09
Paris_m	3.8
Walker_m	0.5
$a_i = c_i$	0.005 in

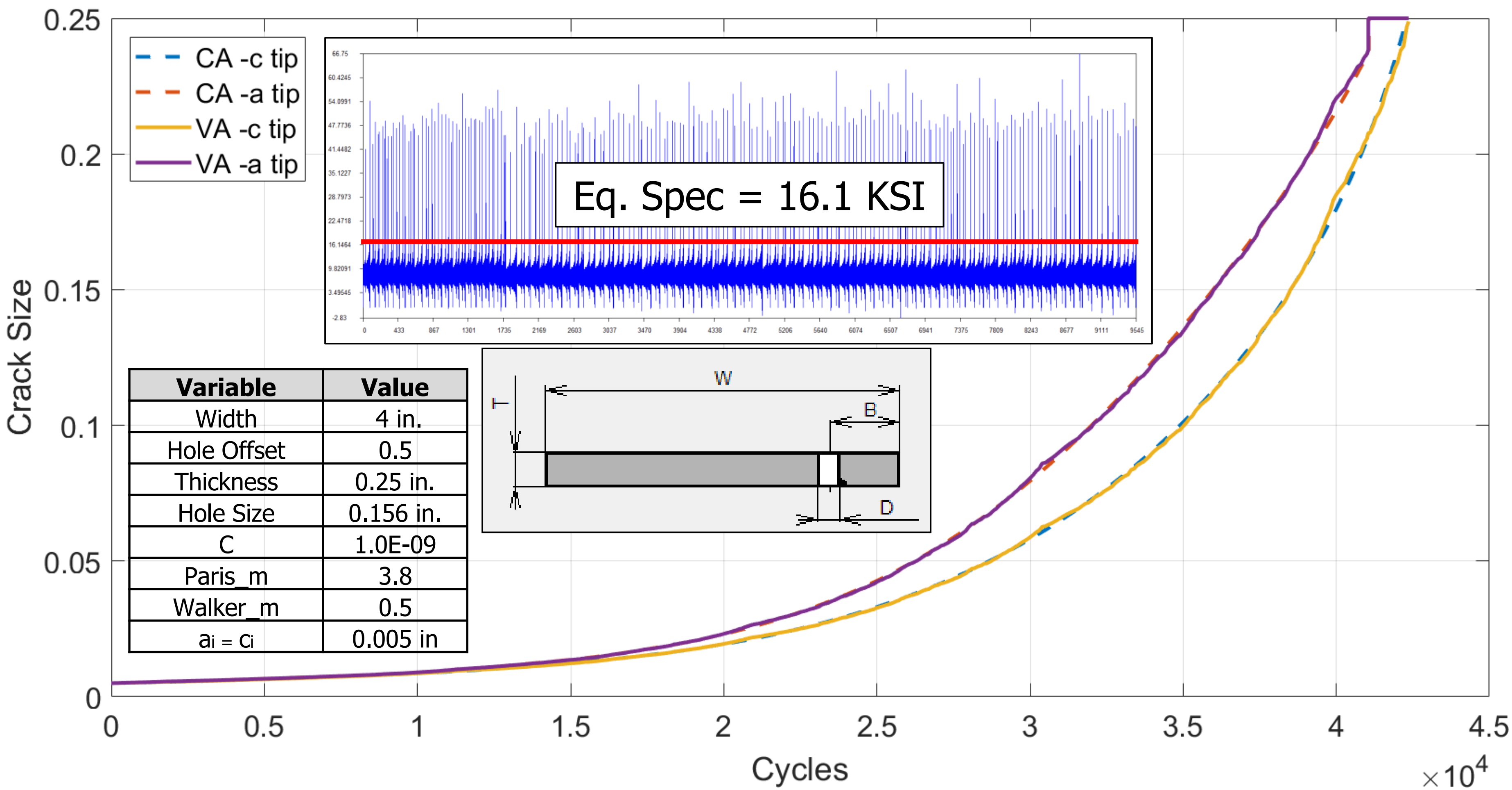






Eq. Stress Examples

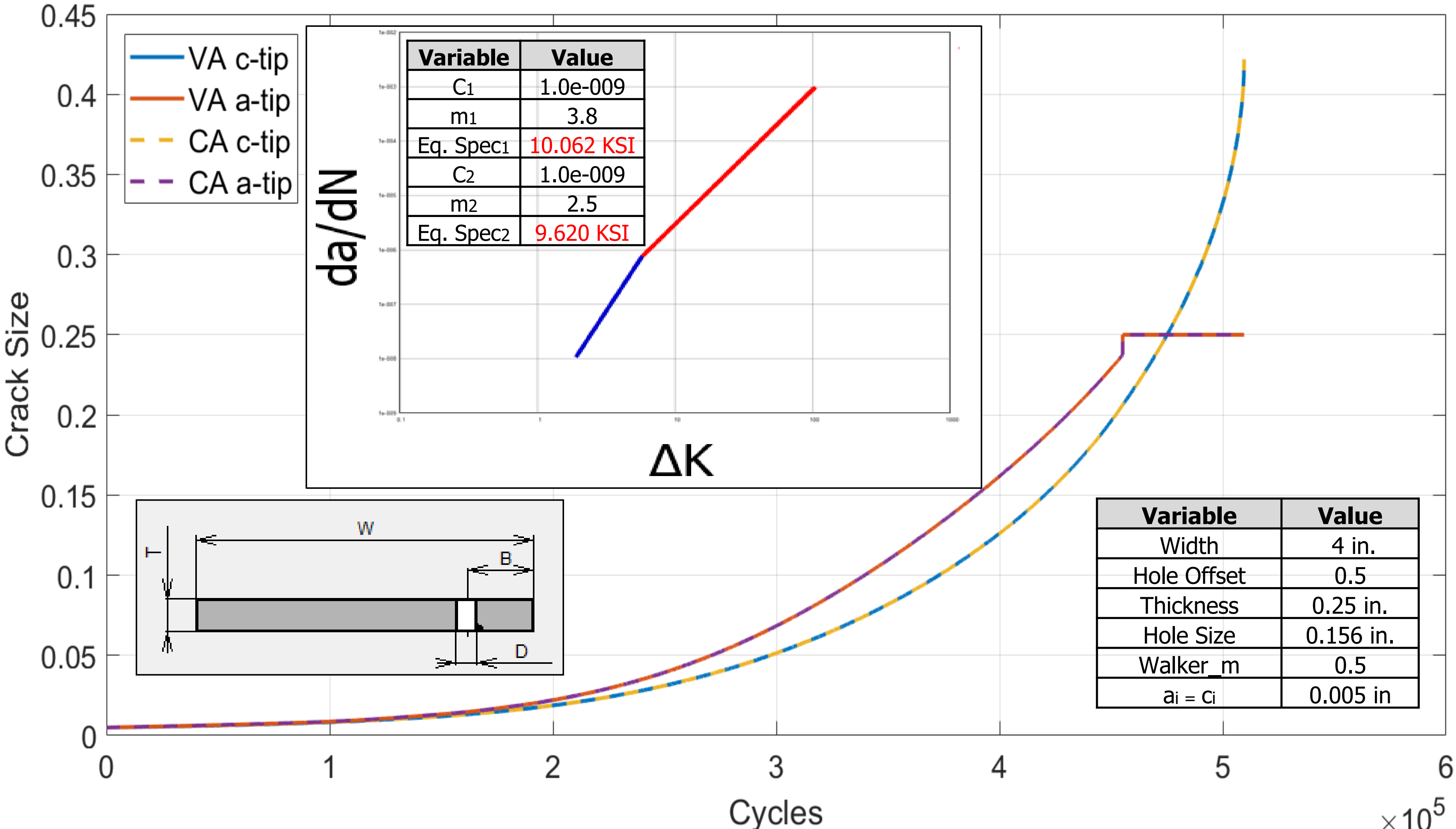
Over Load Example



Variable	Value
Width	4 in.
Hole Offset	0.5
Thickness	0.25 in.
Hole Size	0.156 in.
C	1.0E-09
Paris_m	3.8
Walker_m	0.5
$a_i = c_i$	0.005 in

Eq. Stress Examples

Bilinear Paris Example



Variable	Value
C_1	1.0e-009
m_1	3.8
Eq. $Spec_1$	10.062 KSI
C_2	1.0e-009
m_2	2.5
Eq. $Spec_2$	9.620 KSI

Variable	Value
Width	4 in.
Hole Offset	0.5
Thickness	0.25 in.
Hole Size	0.156 in.
Walker_m	0.5
$a_i = c_i$	0.005 in

Sigmoidal Crack Growth Law

- The equivalent stress is a function of the crack growth rate. Incorporate this relationship within the ODE solver.

$$\Delta\sigma_{eq}\big(n, a(N), c(N)\big)$$

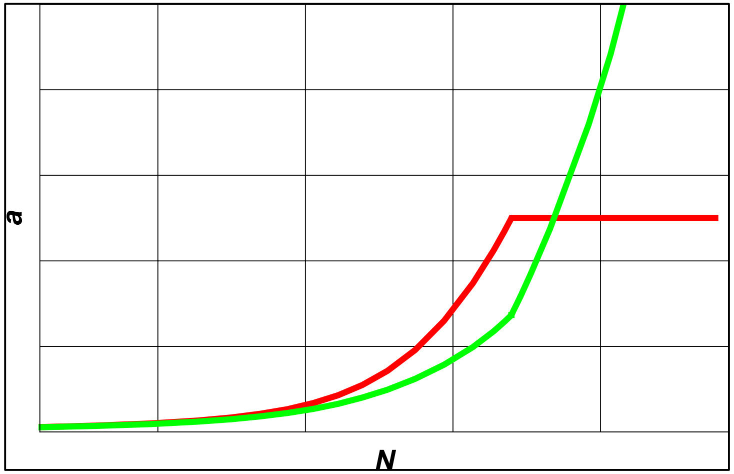
$$\frac{da}{dN} = C\left(\Delta K\left(\Delta\sigma_{eq}, a, c\right)\right)^{n} = 0$$

$$\frac{dc}{dN} = C\left(\Delta K\left(\Delta\sigma_{eq}, a, c\right)\right)^{n} = 0$$

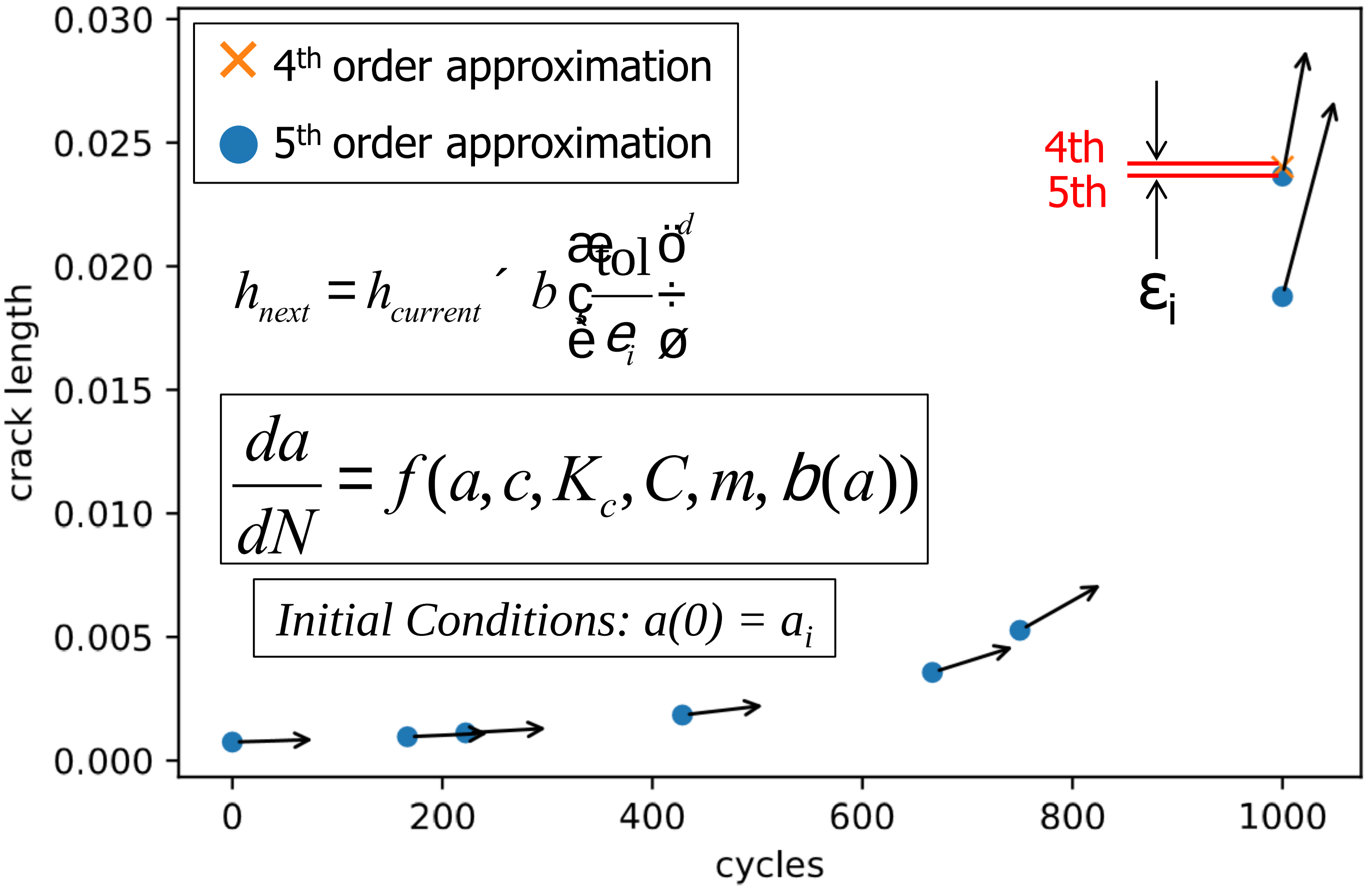
Initial conditions: $a(0) = a_i, \quad c(0) = c_i$

Fast ODE Solver

- Based on best practices from well known and available ODE solvers, e.g., Petsc, Sundials, RKSuite
- Paired Runge-Kutta implementations, 2(3), 4(5), 7(8), e.g., 4th and 5th order solutions computed simultaneously. Gives high quality error estimate.
- Automatically selects step size based on user input and error estimate. Produces large steps early in the life, smaller steps later.



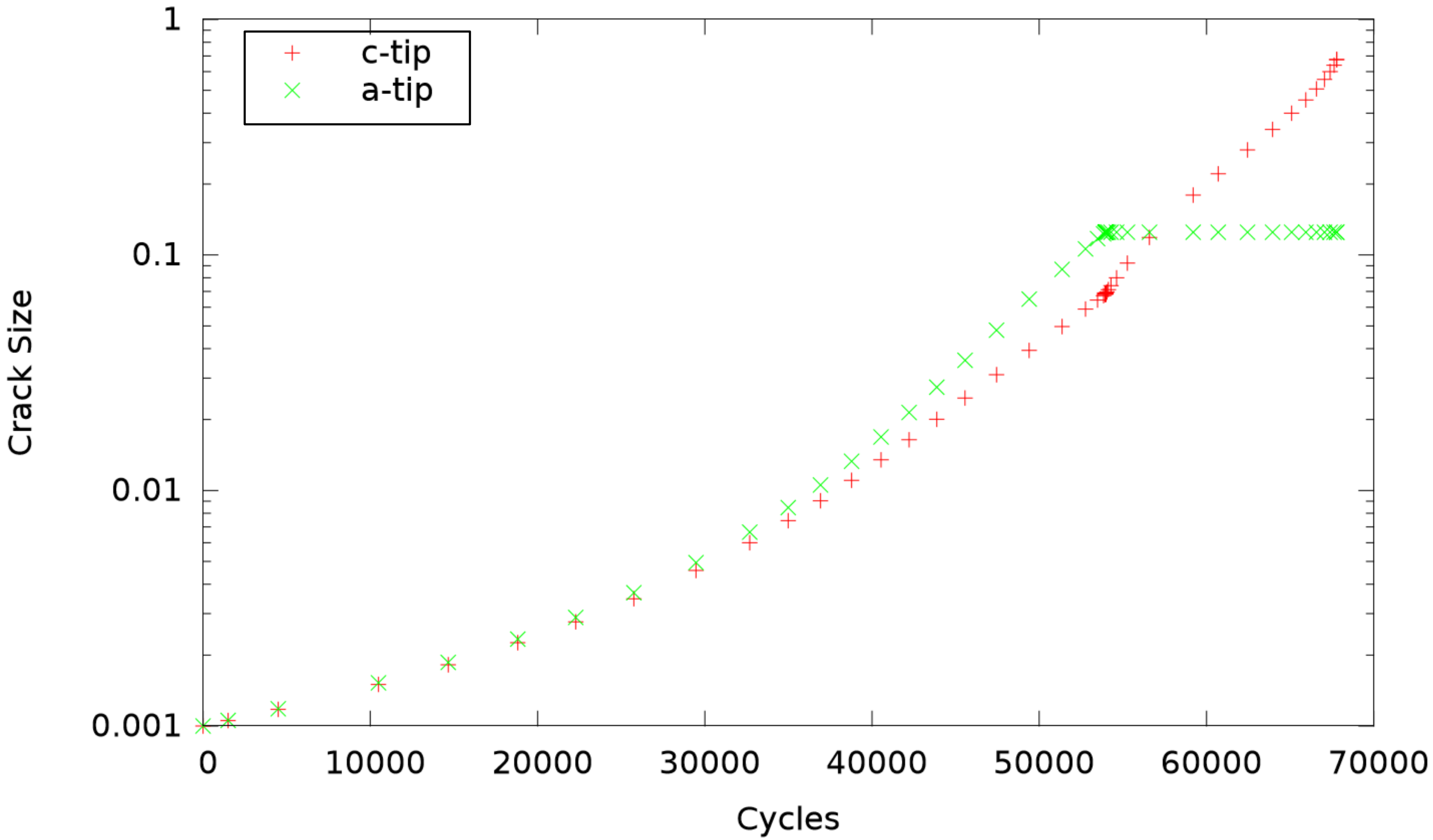
Adaptive Step Size Control



- ε_i is the absolute value of the difference between 5th and 4th order evaluations of the crack size
- Constants b and d determined empirically by the authors
- Step size is increased or decreased depending on the ratio of the user–defined tolerance to the error

Adaptive Step Size Control

Variable step sizes - corner crack integration

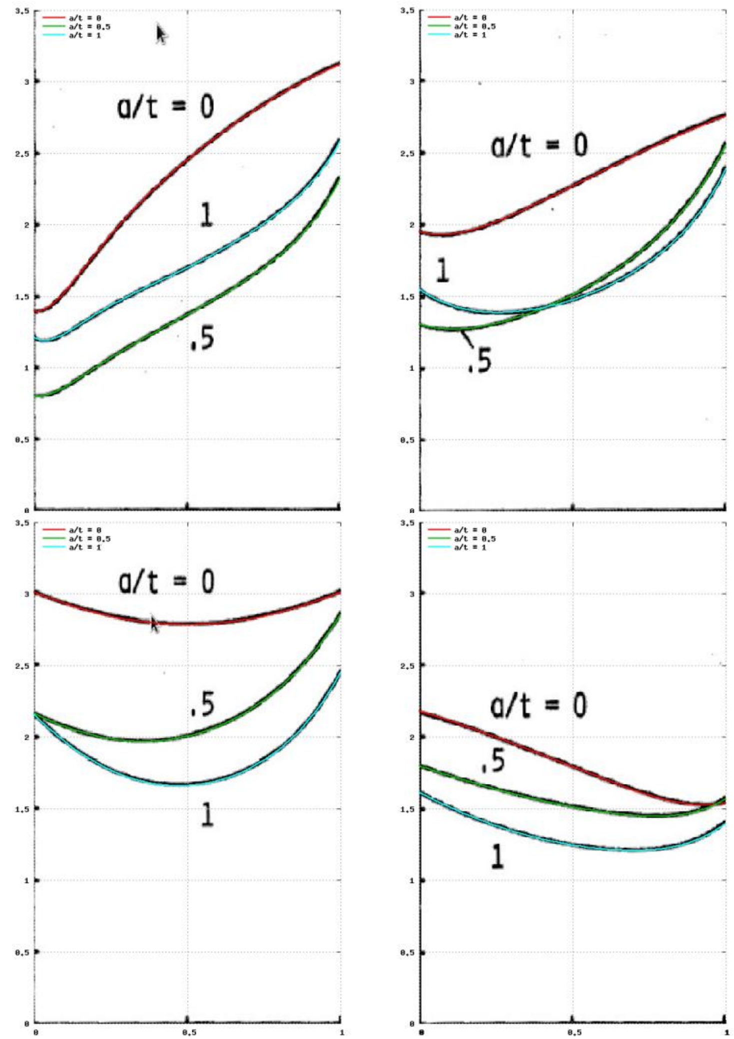


Internal K-Solutions

	Plate	Hole
Thru	✓	✓
Corner (Newman-Raju)	✓	✓
Surface (Newman-Raju)	✓	✓

- Tension Loading only, bending / pin loading not implemented yet
- Centered Hole only
- Weight functions not implemented

Newman-Raju



Beta Tables

! Thru crack betas

c_1	β_1
c_2	β_1
…	…
c_N	β_1

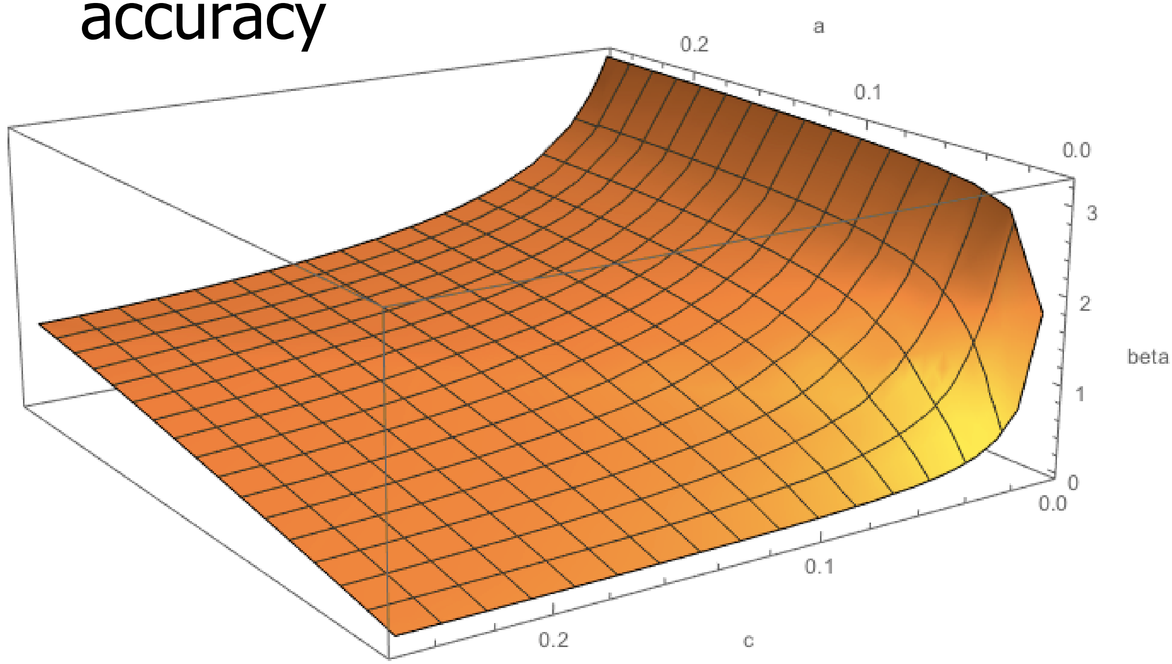
! C-tip direction

	a_1	a_2	…	a_N
c_1	β_{11}	β_{12}	…	β_{1N}
c_2	β_{21}	β_{22}	…	β_{2N}
…	…	…	…	…
c_N	β_{N1}	β_{N2}	…	β_{NN}

! A-tip direction

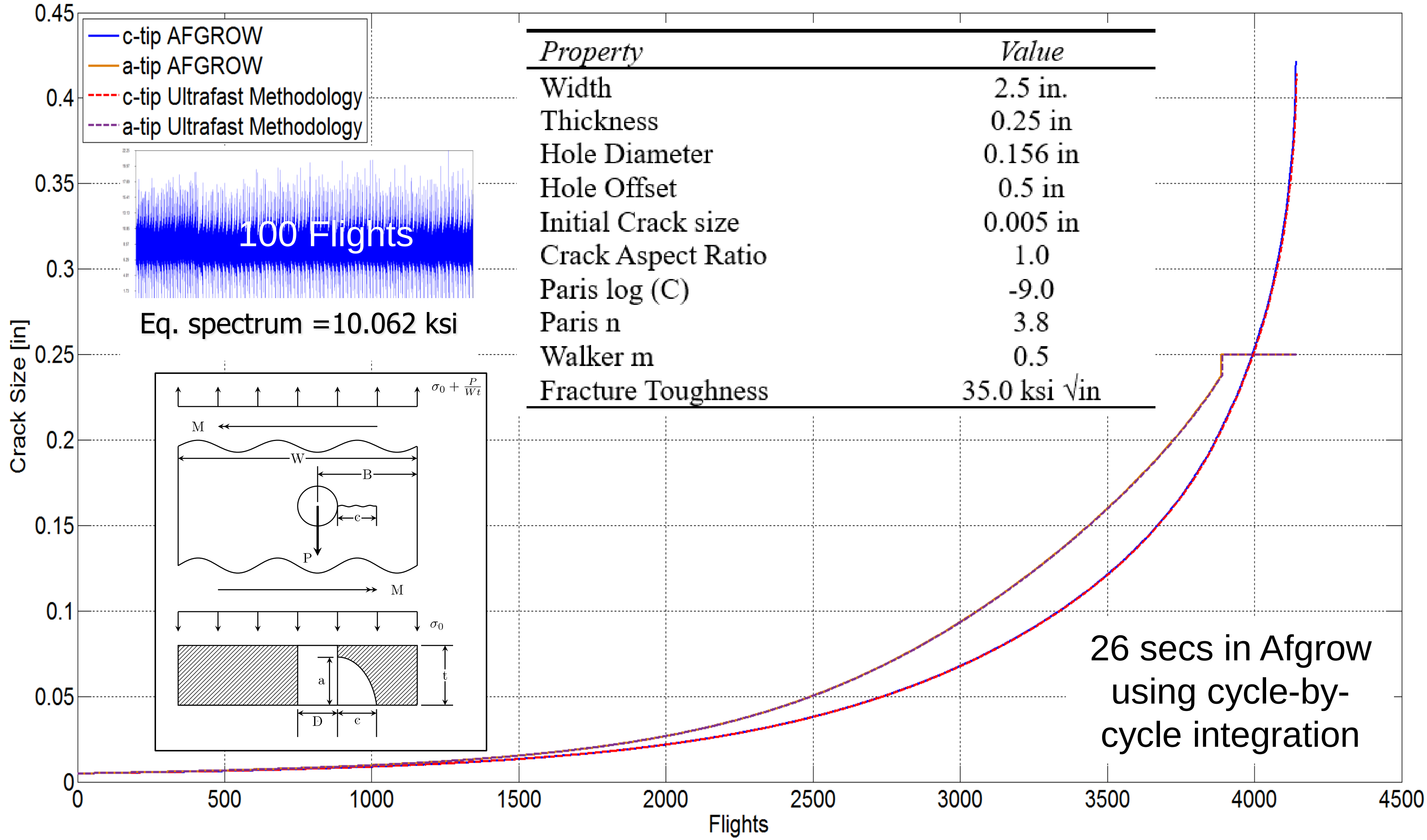
	a_1	a_2	…	a_N
c_1	β_{11}	β_{12}	…	β_{1N}
c_2	β_{21}	β_{22}	…	β_{2N}
…	…	…	…	…
c_N	β_{N1}	β_{N2}	…	β_{NN}

- Use Afgrow/Nasgro/other to generate beta tables for any K solution. Hypergrow reads the table and interpolates to get betas.
- Can solve any crack model with high accuracy



Corner Crack at Hole

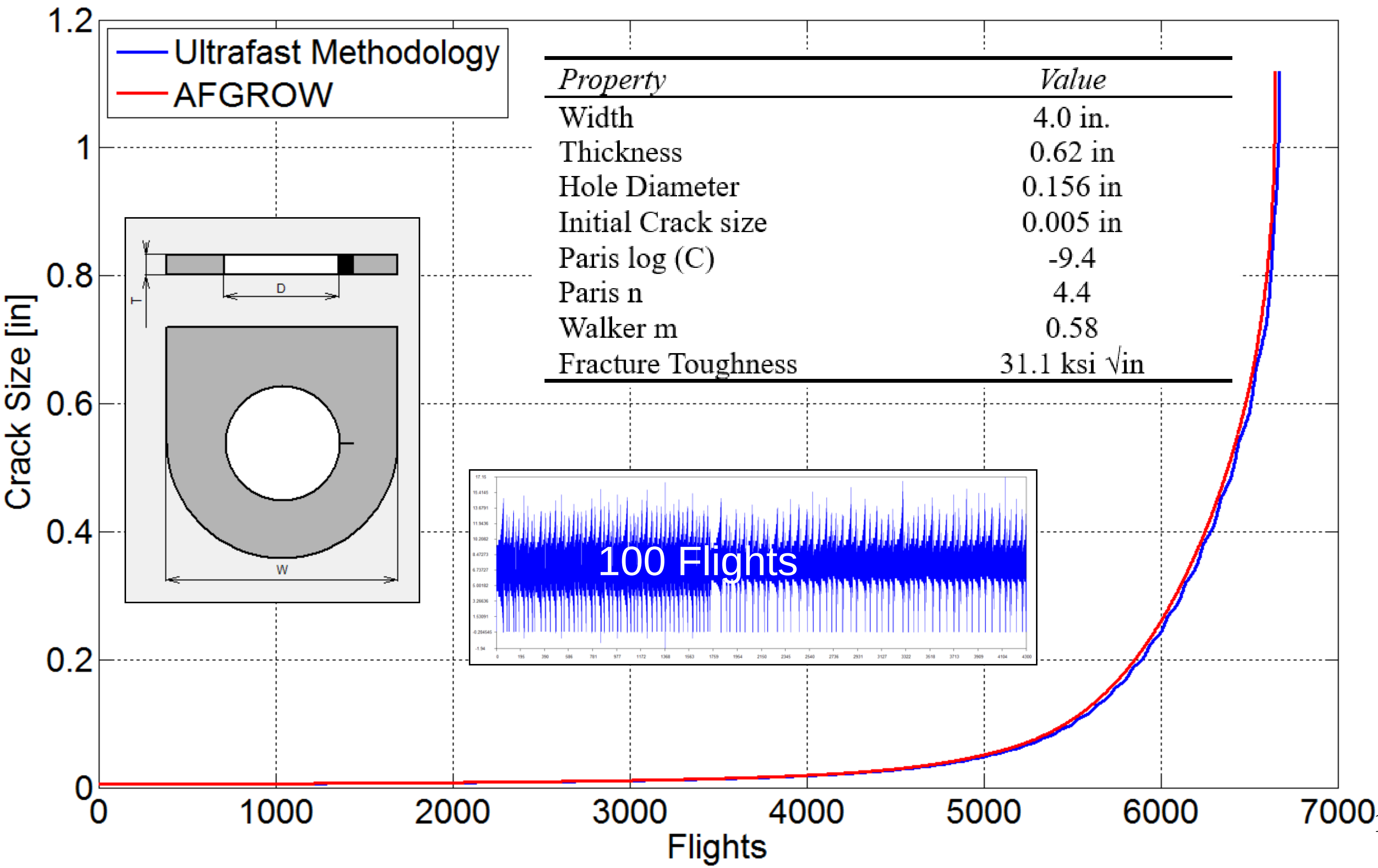
(Tension)



Property	*Value*
Width	2.5 in.
Thickness	0.25 in
Hole Diameter	0.156 in
Hole Offset	0.5 in
Initial Crack size	0.005 in
Crack Aspect Ratio	1.0
Paris log (C)	-9.0
Paris n	3.8
Walker m	0.5
Fracture Toughness	35.0 ksi √in

26 secs in Afgrow using cycle-by-cycle integration

Thru Crack at Lug
(Tension)



Property	*Value*
Width	4.0 in.
Thickness	0.62 in
Hole Diameter	0.156 in
Initial Crack size	0.005 in
Paris log (C)	-9.4
Paris n	4.4
Walker m	0.58
Fracture Toughness	31.1 ksi √in

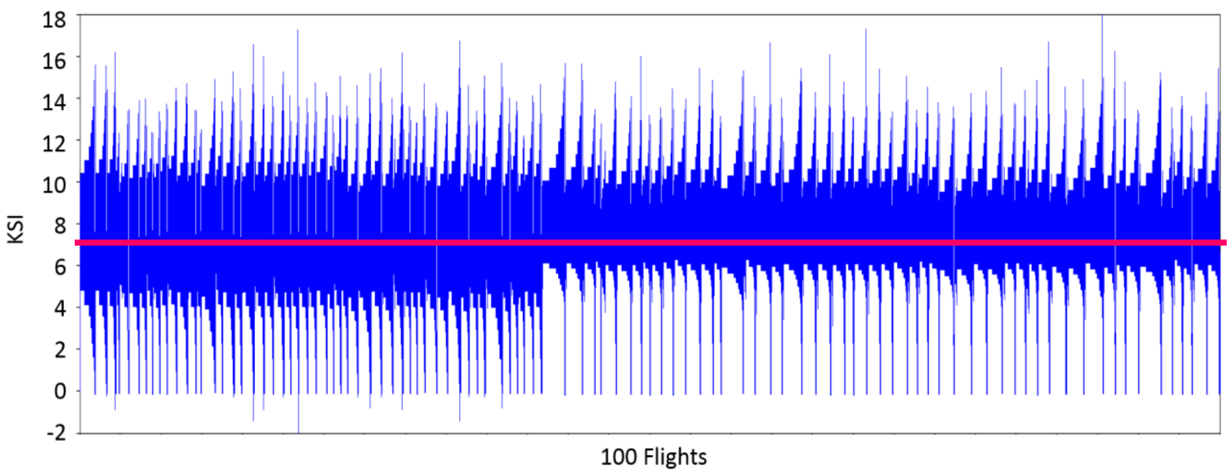
Digital Twin, Virtual Testing, and Probabilistic Damage Tolerance Applications

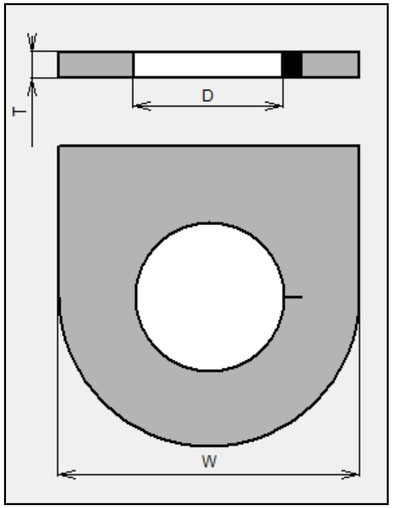
Example Problem (I)

Application to probabilistic damage tolerance and airframe digital twin

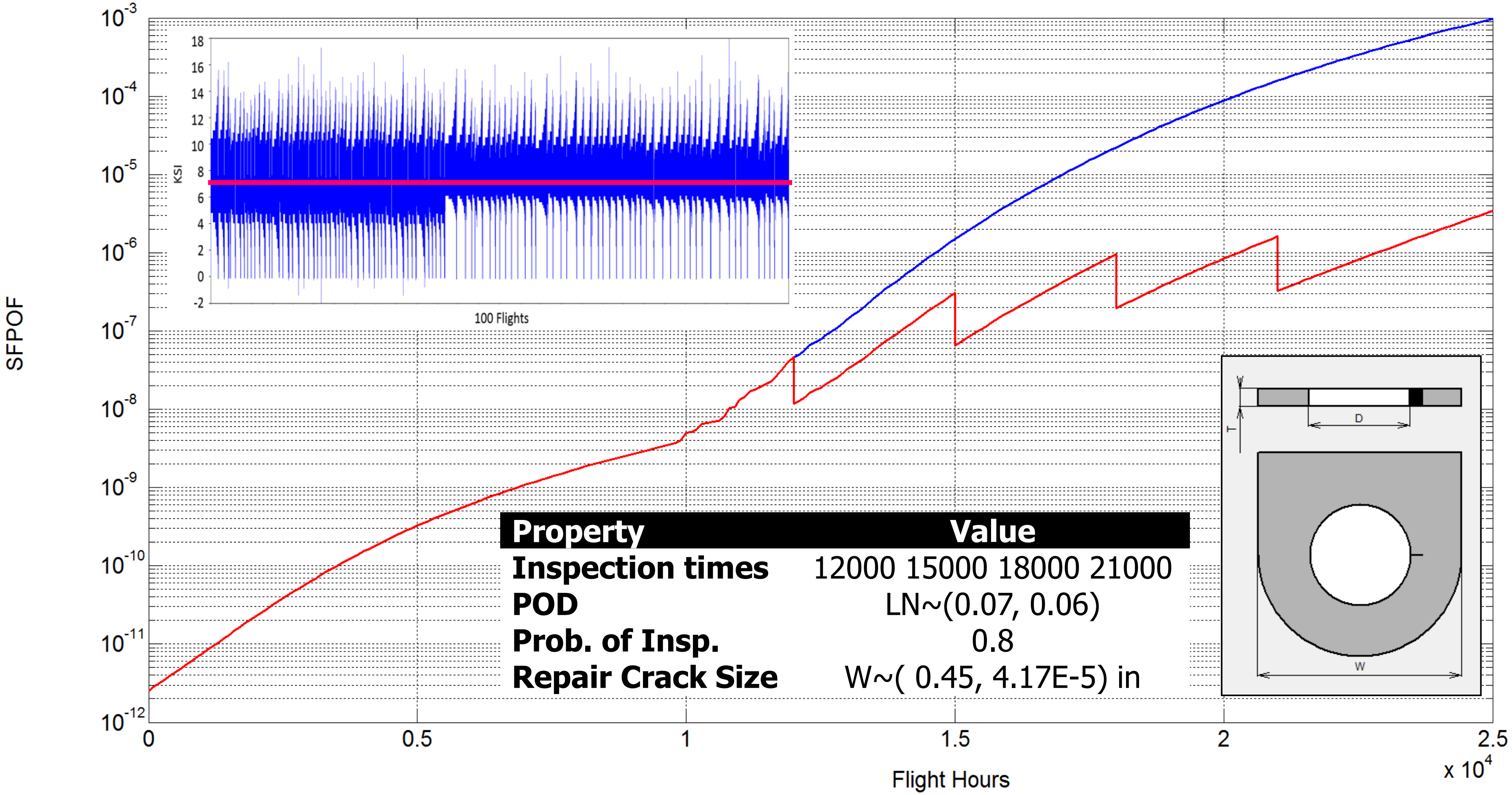
Property	Value
Width	5.0 in.
Thickness	0.2 in
Hole Diameter	0.156 in
Hole Offset	2.5 in
Initial Crack size	W~(0.45, 4.17E-5) in
Paris log (C)	N~(-9.0, 0.1)
Paris n	3.8
Walker m	1.0
Fracture Toughness	N~(35.0, 3.5) ksi √in
Loading	EVD~(13.37, 1.24, 0.09)

Property	Value
Inspection times	12000 15000 18000 21000
POD	LN~(0.07, 0.06)
Prob. of Insp.	0.8
Repair Crack Size	W~(0.45, 4.17E-5) in





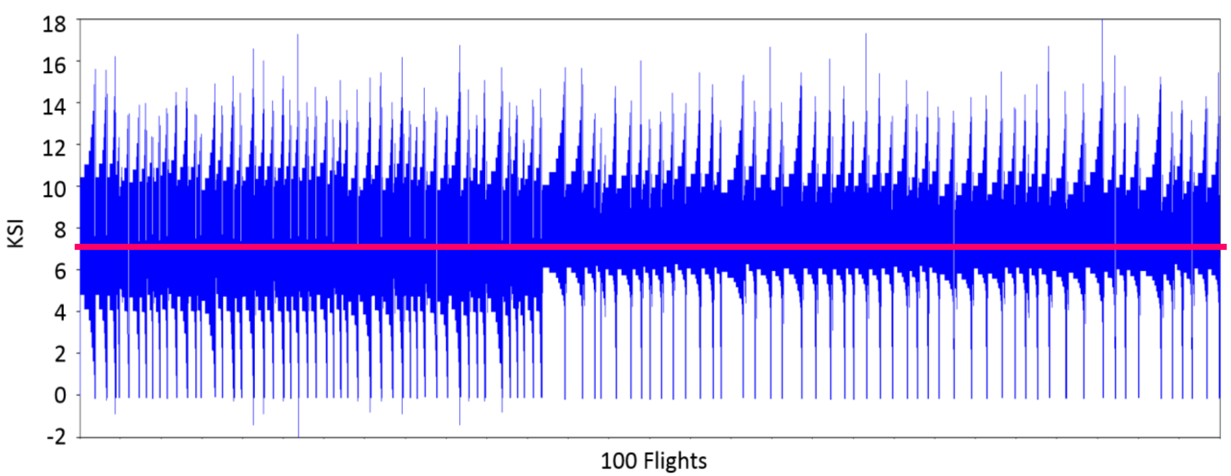
Example Problem (I)

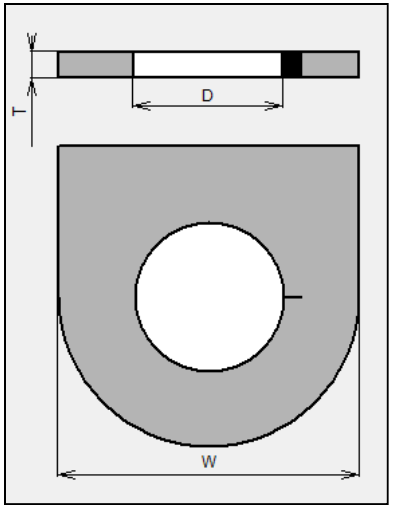


Property	Value
Inspection times	12000 15000 18000 21000
POD	LN~(0.07, 0.06)
Prob. of Insp.	0.8
Repair Crack Size	W~(0.45, 4.17E-5) in

Example Problem (II)

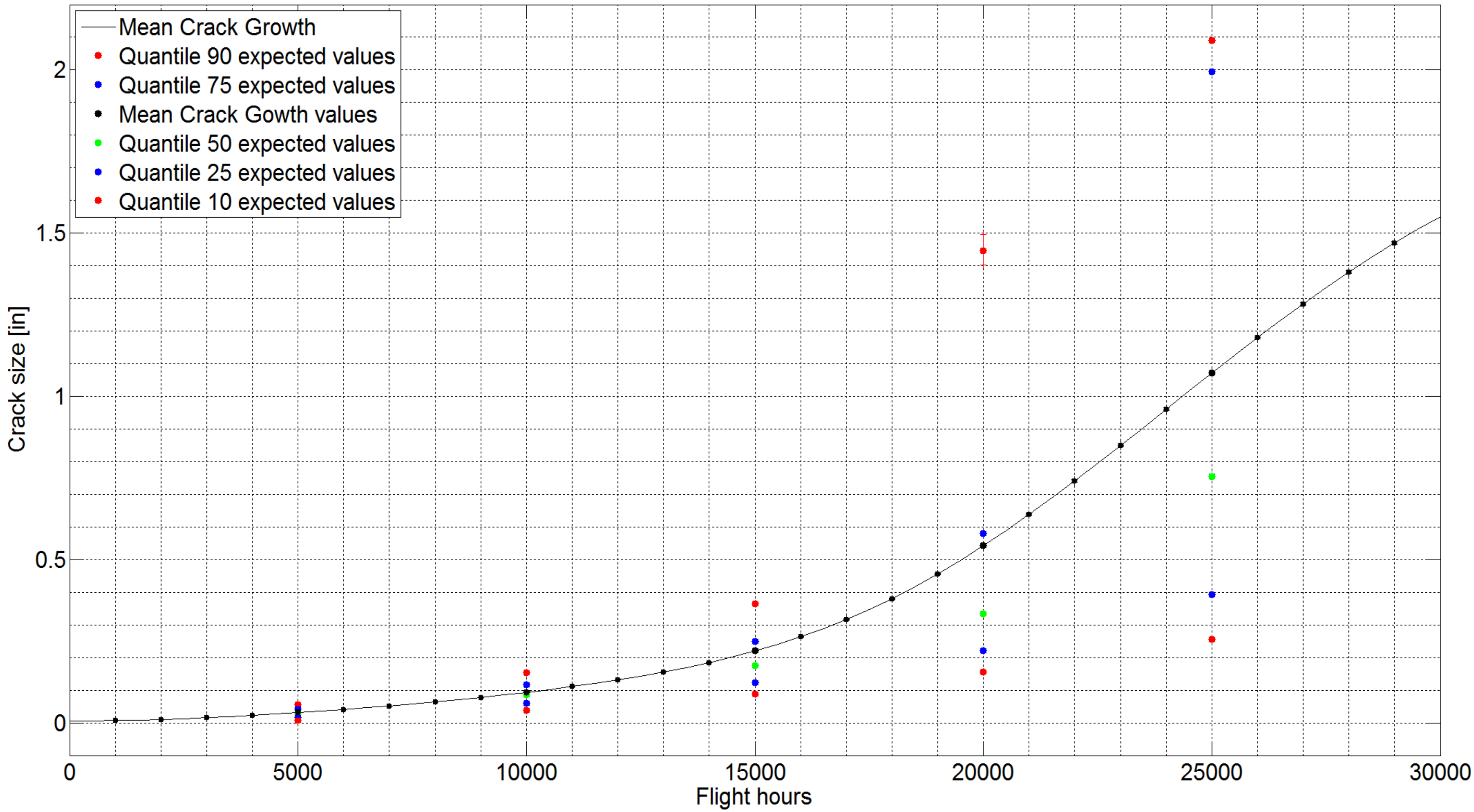
Property	Value
Width	5.0 in.
Thickness	0.2 in
Hole Diameter	0.156 in
Hole Offset	2.5 in
Initial Crack size	LN~(0.005, 0.003) in
Paris log (C)	N~(-9.0, 0.1)
Paris n	3.8
Walker m	1.0
Fracture Toughness	N~(35.0, 3.5) ksi √in
Loading	EVD~(13.37, 1.24, 0.09)





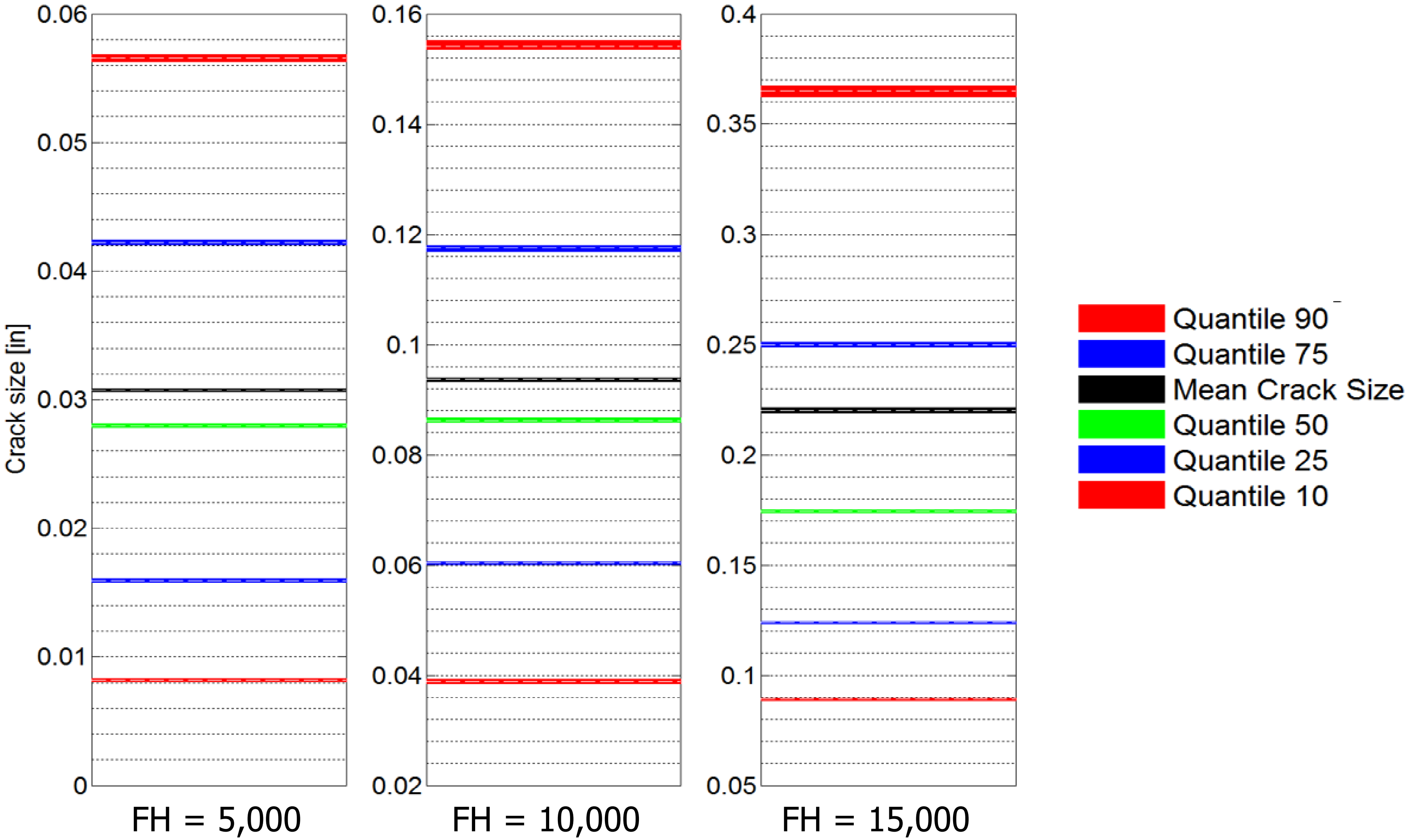
Example Problem (II)

Crack Growth Quantiles



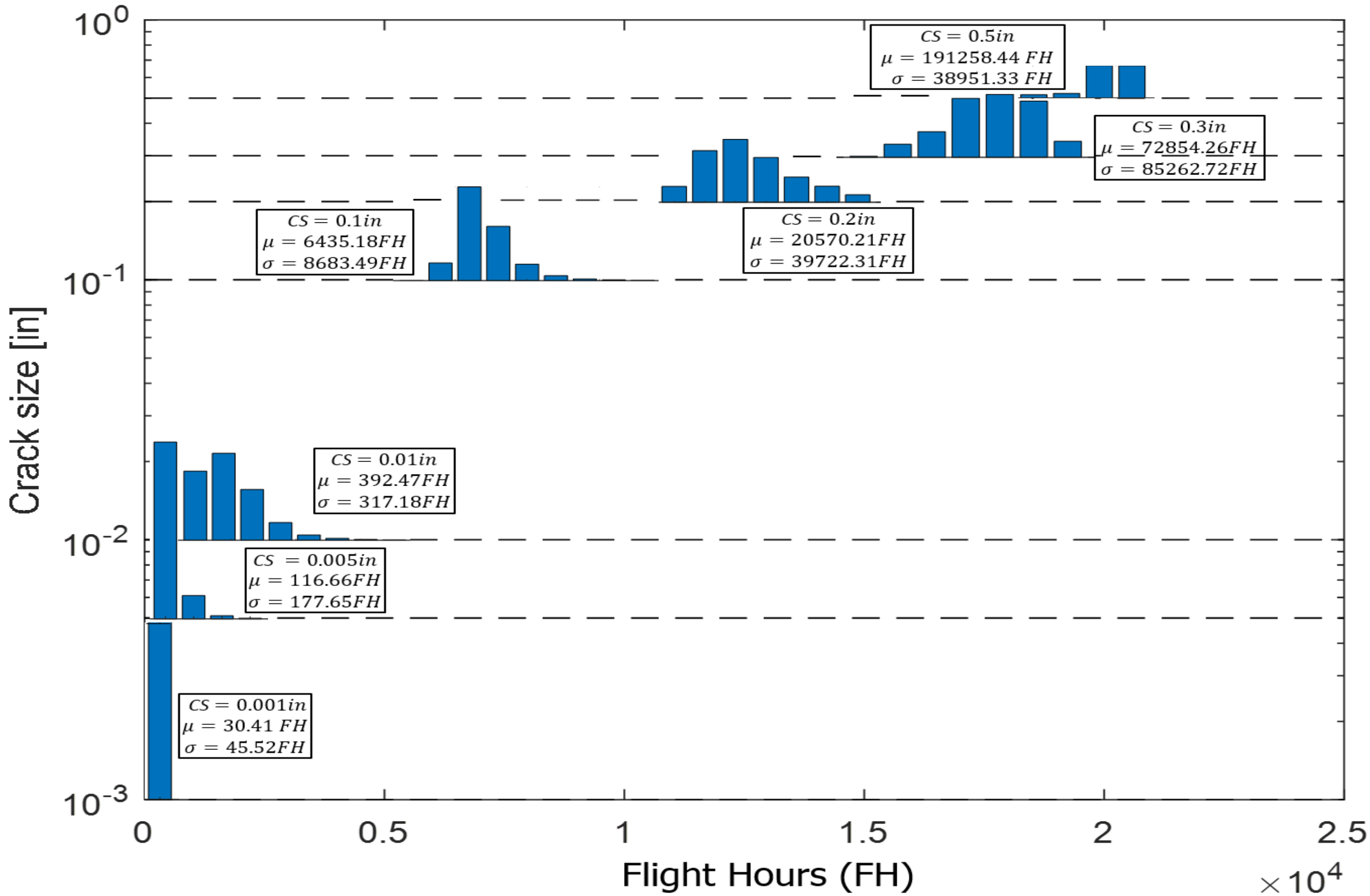
Example Problem (II)

Crack Growth Quantiles Confidence Bounds

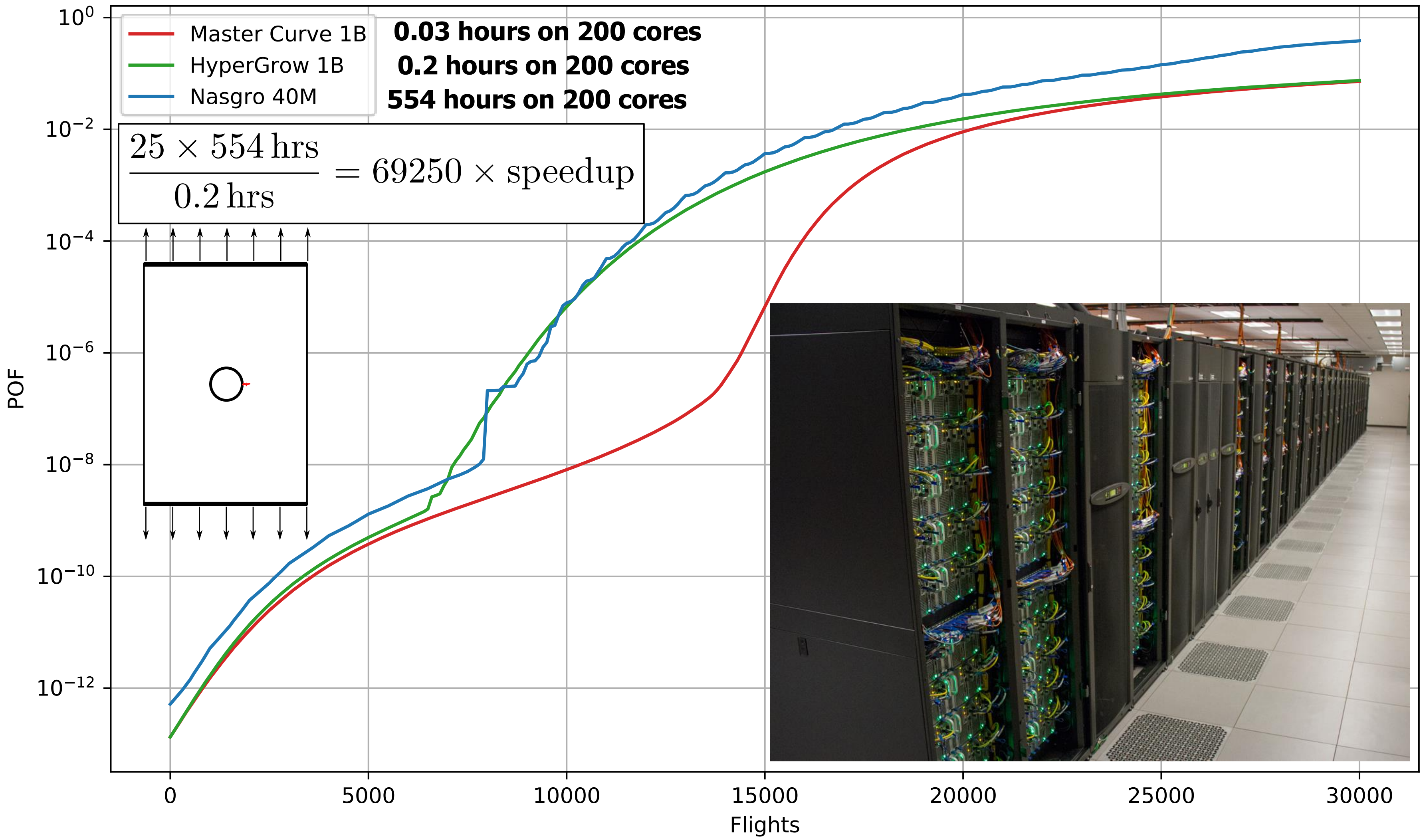


Example Problem (II)

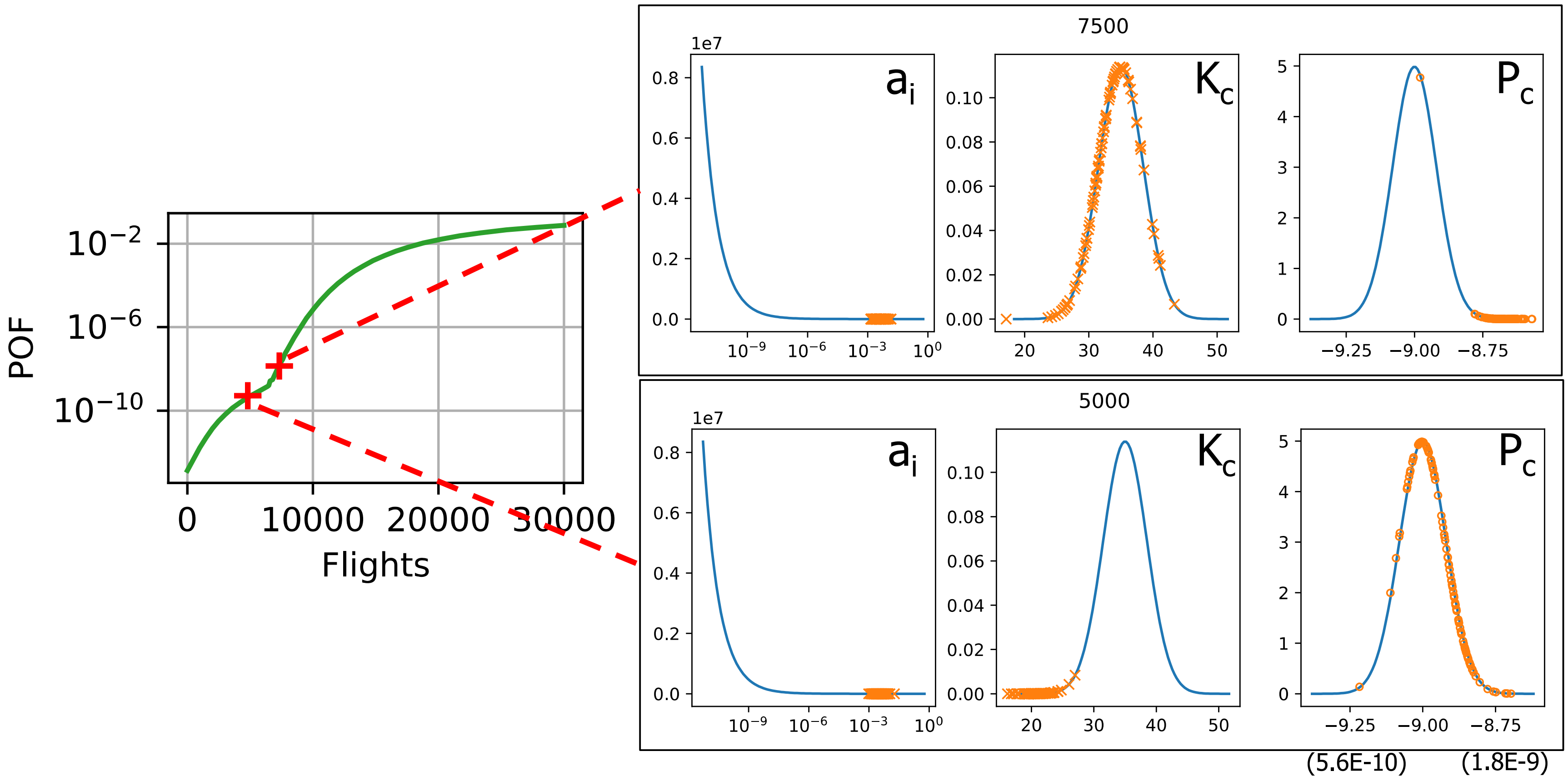
Remaining Useful Life



Uninspected POF Output



Most Influential Realizations



- Visualizes which realizations contribute most to POF

Ultrafast Approach Conclusions

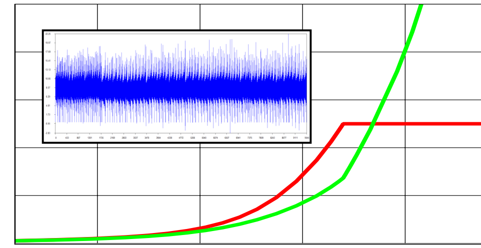
1) Equivalent constant amplitude is accurate at predicting variable amplitude crack growth – *for all problems to date.*
2) Adaptive RK algorithm to grow the crack is very effective (~10,000 evaluations/sec/proc)
    - Capability to read beta tables provides an attractive method to incorporate a variety of crack models.
3) The top 100 (or so) damaging realizations can be further examined for potential reanalysis

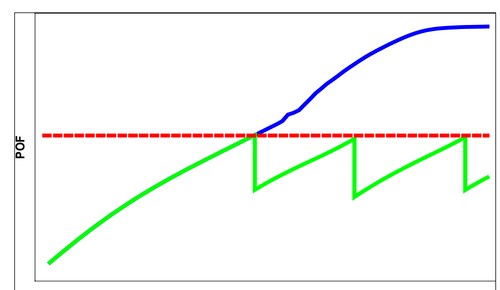
Future Work

- Verify using more geometries and a larger variety of spectra. **Open to suggestions.**
- Compute beta tables on-the-fly with Afgrow & Nasgro.
- Build library of highly-used beta tables to include with the software.
- Expand the equivalent stress method to work with varying crack growth laws, e.g., bilinear Paris, Nasgro equation, and tabular da/dN input.
- Include retardation

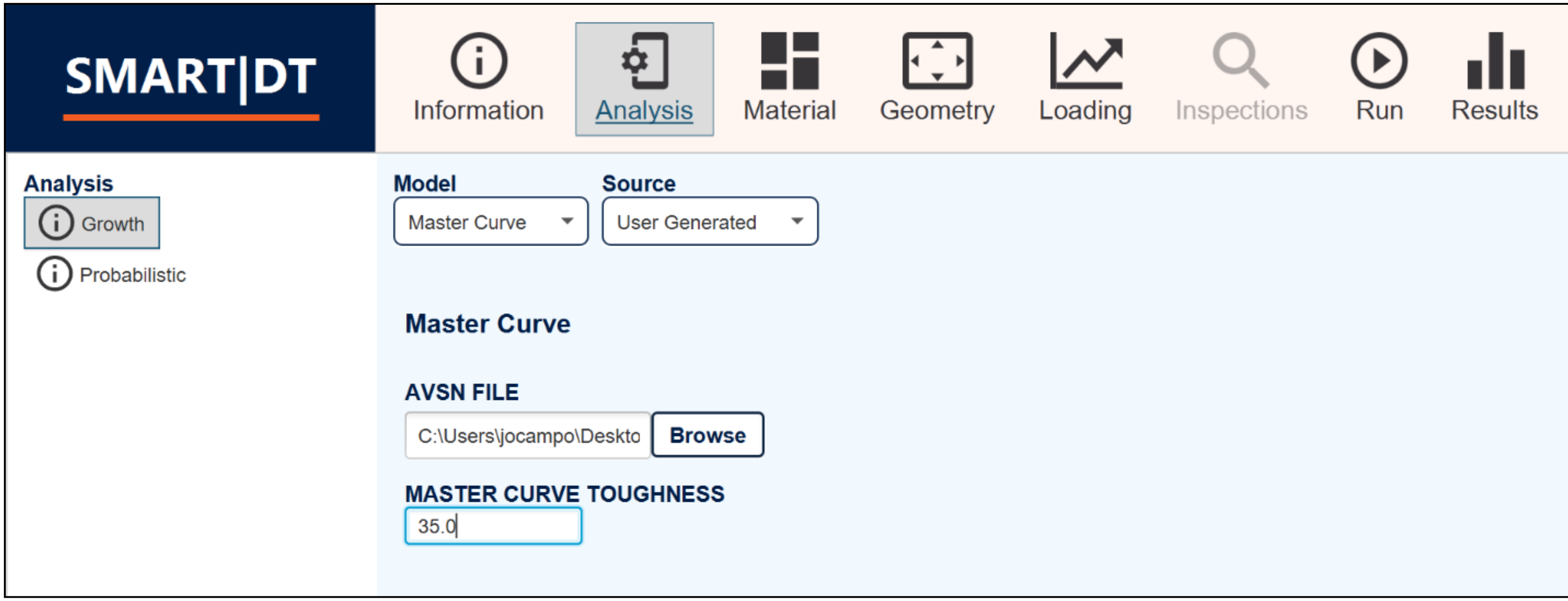
SMART|DT Current Development Activities

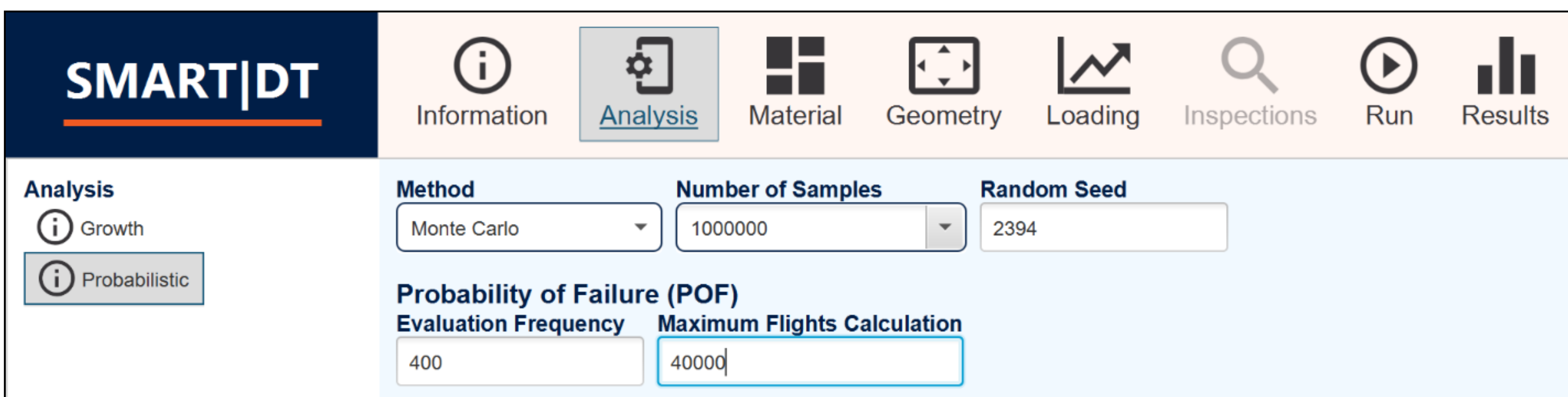
- Ultrafast crack growth code
- Probabilistic data base
  - (EIFS, POD, Kc, da/DN, etc.)
- MPI version for clusters
- New Java-based GUI
- Risk based inspections
- Importance Sampling
- Fleet management





NEW GUI

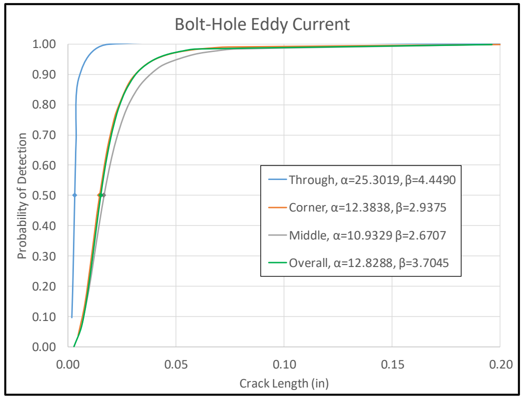

SMART|DT
Information
Analysis
Material
Geometry
Loading
Inspections
Run
Results
Analysis
Growth
Probabilistic
Model
Master Curve
Source
User Generated
Master Curve
AVSN FILE
C:\Users\jocampo\Deskto
Browse
MASTER CURVE TOUGHNESS
35.0


SMART|DT
Information
Analysis
Material
Geometry
Loading
Inspections
Run
Results
Analysis
Growth
Probabilistic
Method
Monte Carlo
Number of Samples
1000000
Random Seed
2394
Probability of Failure (POF)
Evaluation Frequency
400
Maximum Flights Calculation
40000

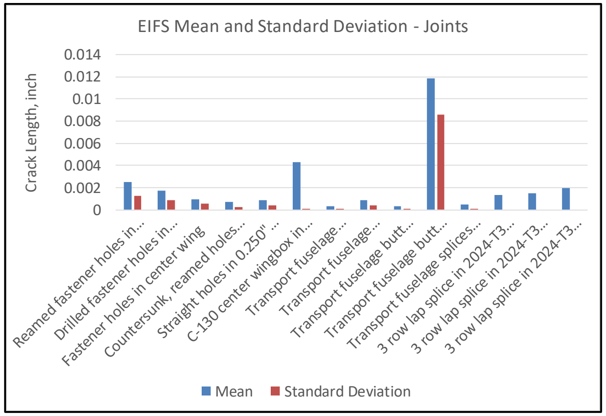
Probabilistic Database

- **Flaw size**: Size (EIFS); Crack aspect ratio (a/c)
- **Geometry**: Fastener hole diameter; Edge distance
- **Material Properties:** da/dN; Fracture toughness; Yield/ultimate strength
- **Inspections:** Probability of detection

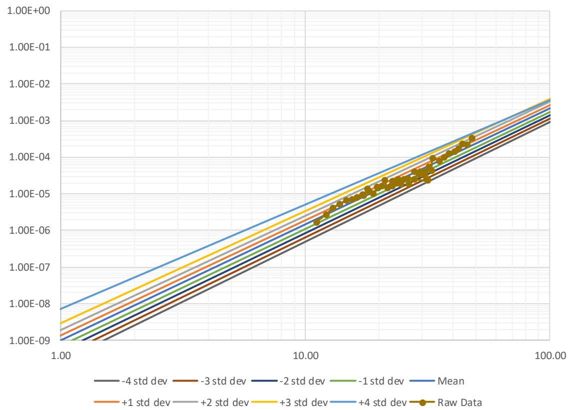
Probability of Detection



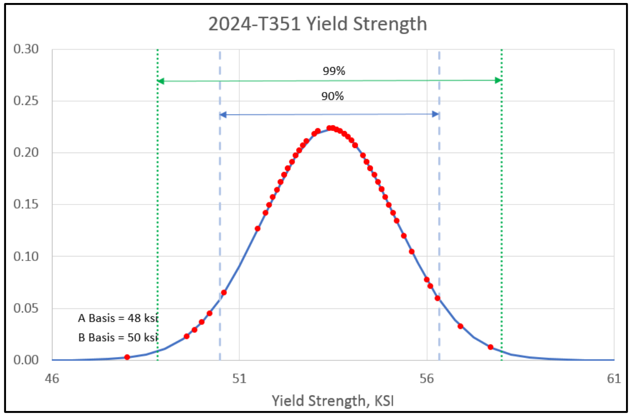
Flaw Size



da/dN



Strength



Goal: Community effort to expand and make the database available to the public

Acknowledgements

- Probabilistic Fatigue Management Program for General Aviation, Federal Aviation Administration, Grant 12-G-012