An ultrafast crack growth lifing model for efficient prognosis

Harry Millwater, Nathan Crosby
University of Texas at San Antonio

Juan D. Ocampo
St. Mary's University, San Antonio

INTERNATIONAL CONFERENCE ON FATIGUE DAMAGE OF STRUCTURAL MATERIALS XII
International Conference on Fatigue Damage of Structural Materials XII
16-21 September 2018 | DoubleTree by Hilton, Cape Cod, Hyannis, MA, USA

AFRL Digital Twin Vision

CBM+SI Far-term Vision

per AFRL CBM+SI Workshop - Feb 2009

<u>"Digital Twin":</u> Real-Time, High-Fidelity Operational Decisions for Individual Aircraft Enabled by Tail Number Health Awareness

- **When physical aircraft is delivered, a Digital Model of the aircraft – specific to that tail number, including deviations from the nominal design – will be delivered as well.**

- **The Digital Model will be flown virtually through the same flight profiles as recorded for the actual aircraft by its on-board SHM system.**
- **The modeling results will be compared to sensor readings recorded by the SHM system at critical locations to update / calibrate / validate the model.**
- **As unanticipated damage is found, it will be added to the Digital Model so that the model continually reflects the current state of the actual aircraft.**
- **Prognostics for the airframe will be developed by "flying" the Digital Model through possible future missions.**
- **The Digital Model will be used to determine when & where structural damage is likely to occur, and when to perform maintenance.**

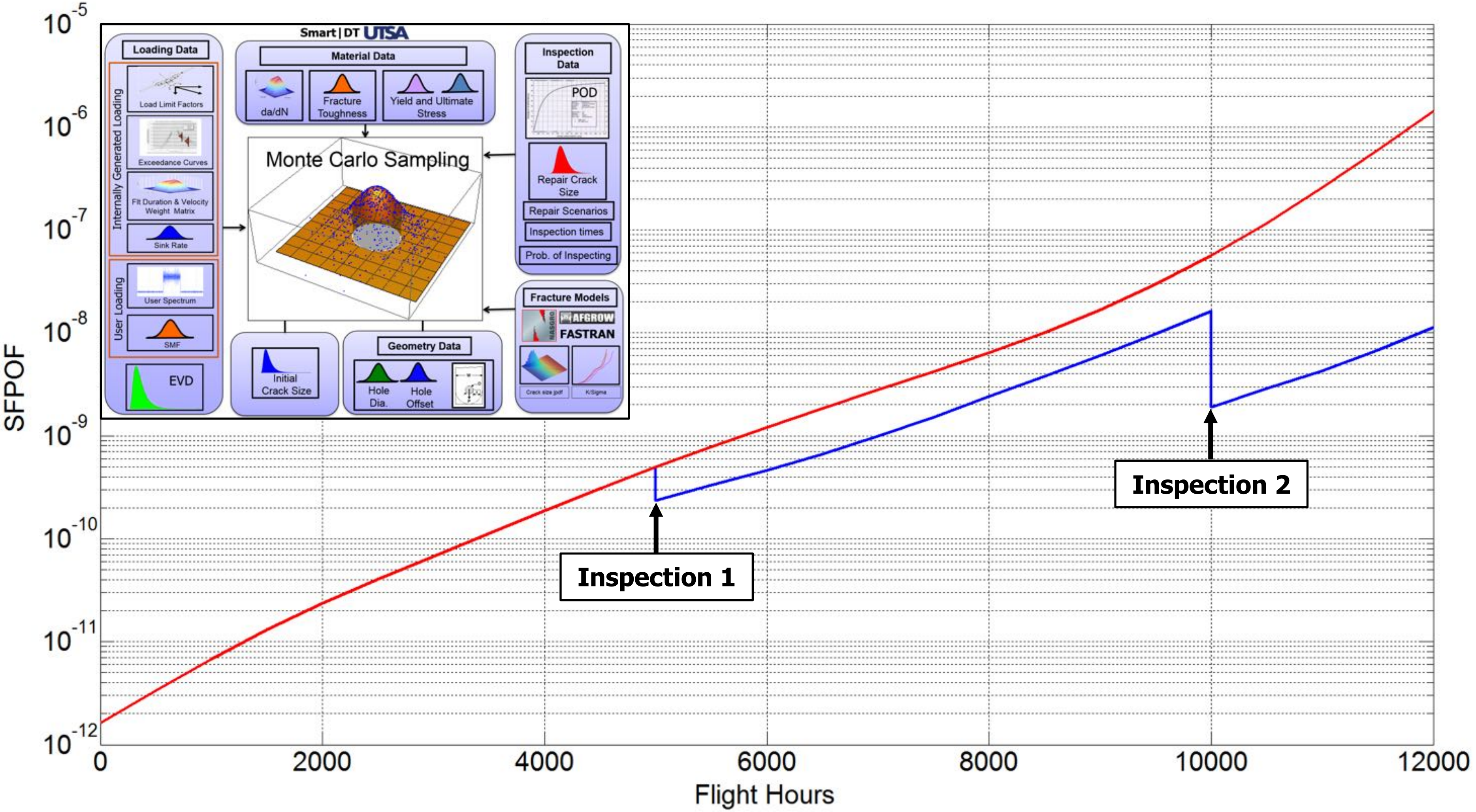
CBM+SI Mid-term Vision

per AFRL Air Vehicles Directorate's "Vision 2009"

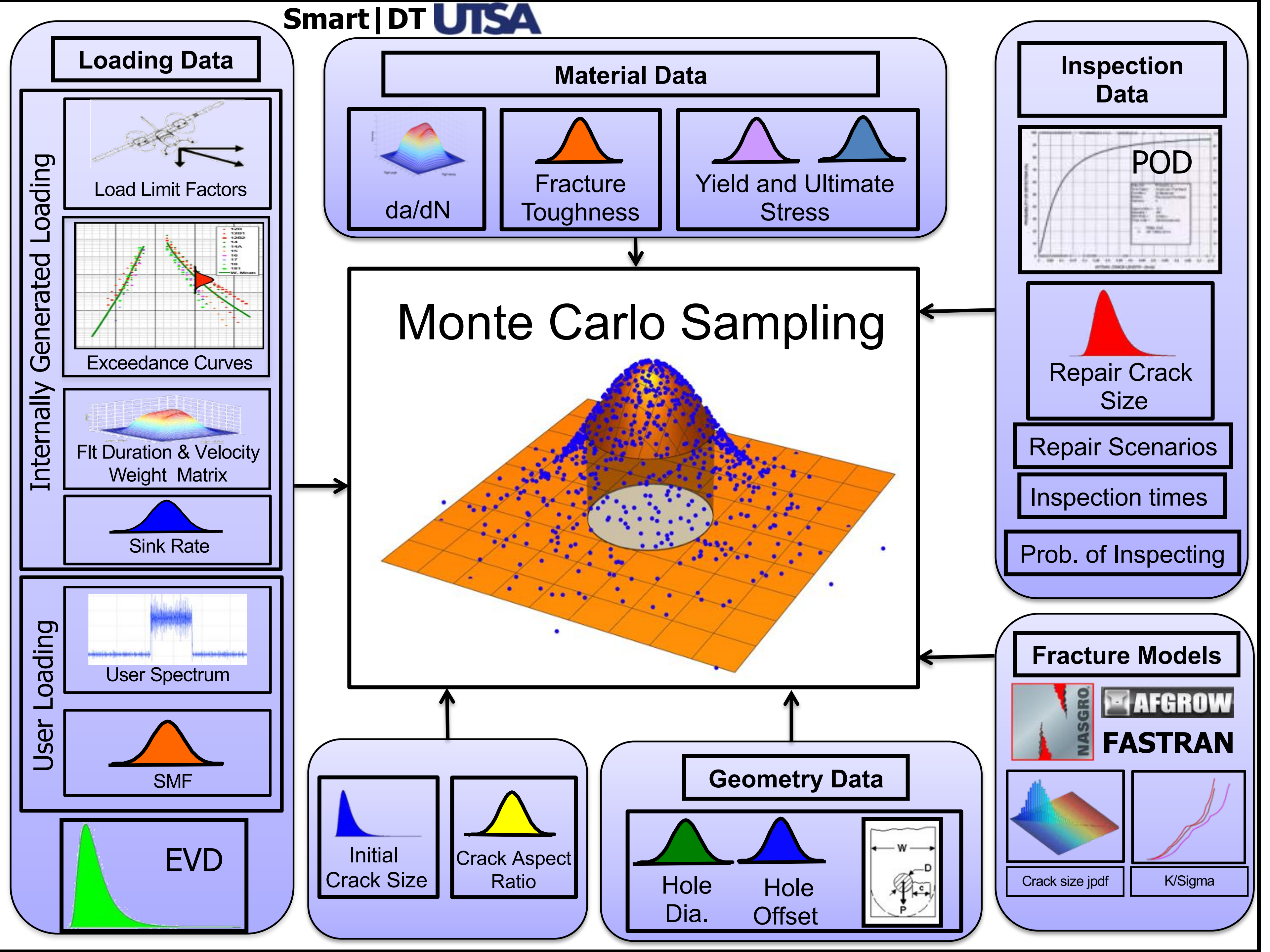
- In ten years, aircraft lifecycle management and maintenance practices will be completely transformed from an inefficient, inconsistent, disparate, and labor-intensive state to an efficient, standardized, integrated, and semi-autonomous state.
- The structural capability of individual airframes will be known and predictable based on the capability to characterize the current health status, predict the future health status, and plan usage and maintenance accordingly.

- Risk of structural failure will be quantified and safety will be enhanced.
- The inspection and repair burden will be diminished and cycle times for inspections, repairs, and modifications will be greatly reduced.
- The remaining useful life of airframes will be extended.
- As a result of all of these improvements, aircraft availability will be increased and O&S costs will be reduced.

Risk Assessment



Smart | DT UTSA



Smart | DT UTSA
Loading Data
Internally Generated Loading
Load Limit Factors
Exceedance Curves
Flt Duration & Velocity Weight Matrix
Sink Rate
User Loading
User Spectrum
SMF
EVD
Material Data
da/dN
Fracture Toughness
Yield and Ultimate Stress
Monte Carlo Sampling
Inspection Data
POD
Repair Crack Size
Repair Scenarios
Inspection times
Prob. of Inspecting
Fracture Models
NASGRO
AFGROW
FASTRAN
Crack size jpdf
K/Sigma
Initial Crack Size
Crack Aspect Ratio
Geometry Data
Hole Dia.
Hole Offset

Purpose

1. Probabilistic damage tolerance analysis requires very small probabilities, e.g., 1E-7, hence, a large number or samples are required.
2. Previous methods allow for a deterministic crack growth curve and do not consider randomness in crack growth rate properties and other random variables.
3. Surrogate models, e.g., Kriging, can be used to speed up the analysis but are still time consuming and the accuracy may not be sufficient.
4. SmartDT users may not have access to Afgrow and Nasgro.
5. As a result, an ultrafast internal crack growth lifing code and strategy was developed, implemented, and verified

Why our own Crack Growth Module

- Purpose: Smart|DT has a directly link with Nasgro and AFGROW. These codes are state-of-the-art and in use by the industry but are **too time consuming** to execute and require a license by the user.
  - Nasgro: File-based interface. Runs in parallel. Requires $4K license. Free for FAA users. Interface currently stuck at V7.1 (current version 8.11). *Expected new Nasgrow interface in 2018.*
  - Afgrow: COM interface: Windows only, no file transfers. Requires $1K License for all users. Does NOT run in parallel. Will stay up to date with Afgrow.
- Solution: Embed a crack growth lifing algorithm within Smart|DT for efficiency and convenience.
  - Should be MUCH faster, 1000X or more, than commercial codes.
  - Will not require any user fees or another software install.
  - Will run in parallel.
  - Requires significant coding and verification.
  - Will have only ESSENTIAL capabilities needed for PDTA.

Why our own Crack Growth Module

- Typical run times w Monte Carlo (1B samples):
- 1) Master Curve:
  - 1 CG (30 sec), 1B interpolations->3 hrs on 8 processors

(only 3 random variables)

(N random variables)

- 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

Current estimate

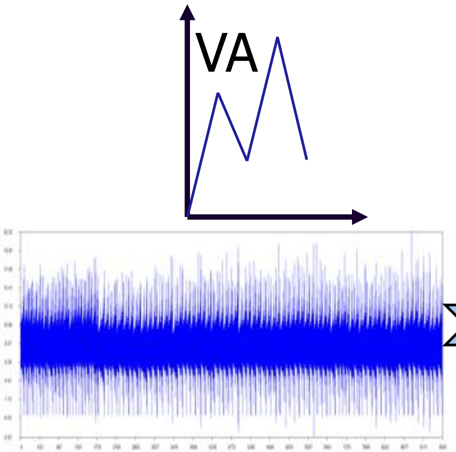
- 4) Numerical Integration
  - 100K CG -> 800 hrs on 1 processor
  - If internal CG code 1000x faster -> 0.8 hrs
- 5) Numerical Integration w Kriging
  - 400 ICG (2s), 100K interpolations-> 100s on 1 processor
- 6) Importance Sampling
  - Internal CG for optimization then 0K ICG -> 1 hr

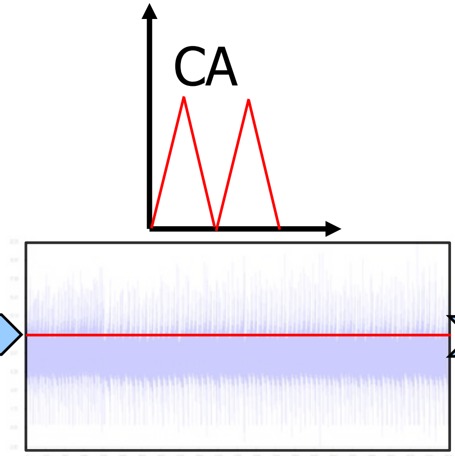
w/o inspection

Ultrafast Approach

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

Internal CG Code





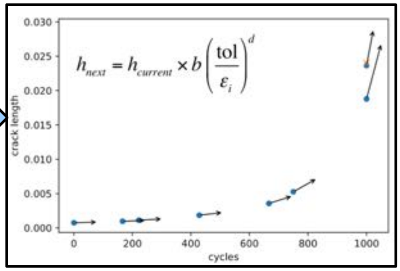
ODE Formulation

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

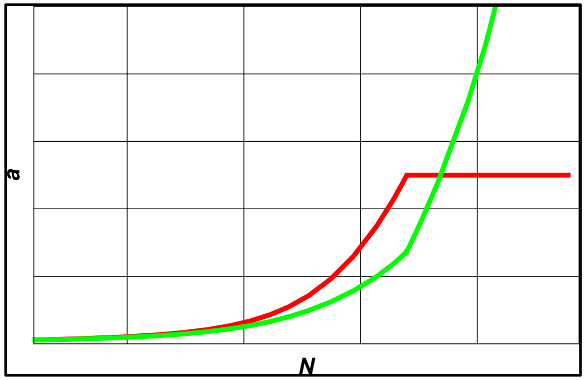
$$\frac{dc}{dN} - C(\Delta K(a,c))^n = 0$$

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

RK ODE Solver

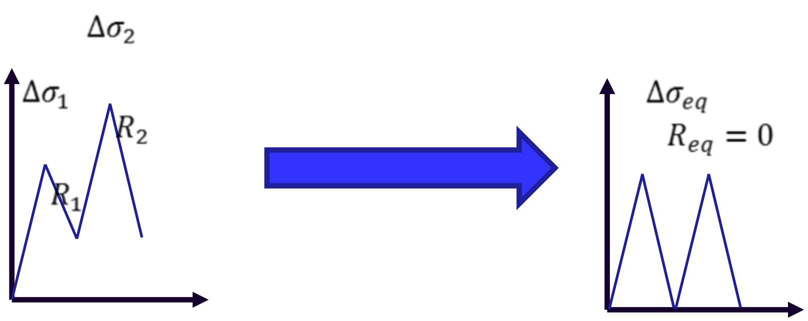


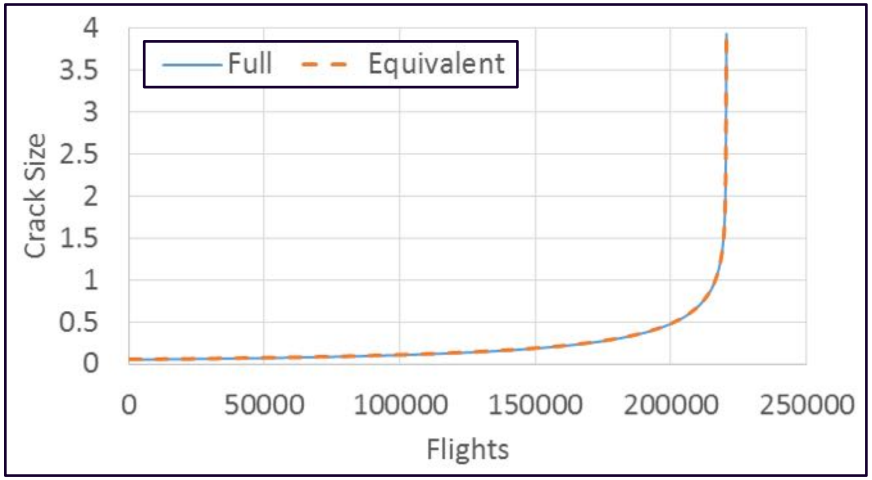
Crack Growth Result



Equivalent Stress

The key idea is to derive an equivalent stress transformation based on the statistical description of the random loading, such as the probabilistic distribution of applied stress range and stress ratio.





T. Swift, Repairs to Damage Tolerant Aircraft, Int. Symposium on Structural Integrity of Aging Airplanes, Atlanta, GA, March 1990
Y. Xiang and Y. Liu, Equivalent Stress Transformation for Efficient Probabilistic Fatigue-Crack Growth Analysis under Variable Amplitude Loadings, J. Aerospace Eng., 2016

Equivalent Stress Formulation

Variable Amplitude

$$\sum_{i=0}^{n-1} N_i = \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \frac{1}{f(a_i,\ \beta_i,\ \Delta\sigma_i,\ c_p,\ n_p,\ \dots)} da$$
$$= \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \frac{1}{c_p\ (\Delta\sigma_i\ \beta_i\ \sqrt{\pi\ a_i})^{n_p}} da$$
$$= \sum_{i=0}^{n-1} \frac{1}{c_p\ (\Delta\sigma_i\ \sqrt{\pi})^{n_p}} \int_{a_i}^{a_{i+1}} \frac{1}{(\beta_i\ \sqrt{a_i})^{n_p}} da$$

$$\sum_{i=0}^{n-1} N_i\ \Delta\sigma_i^{n_p} = \frac{1}{c_p\ (\sqrt{\pi})^{n_p}} \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \frac{1}{(\beta_i\ \sqrt{a_i})^{n_p}} da$$

Constant Amplitude

$$N_{total} = \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \frac{1}{f(a_i,\ \beta_i,\ \Delta\sigma_{eq},\ c_p,\ n_p,\ \dots)} da$$
$$= \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \frac{1}{c_p\ (\Delta\sigma_{eq}\ \beta_i\ \sqrt{\pi\ a_i})^{n_p}} da$$
$$= \sum_{i=0}^{n-1} \frac{1}{c_p\ (\Delta\sigma_{eq}\ \sqrt{\pi})^{n_p}} \int_{a_i}^{a_{i+1}} \frac{1}{(\beta_i\ \sqrt{a_i})^{n_p}} da$$

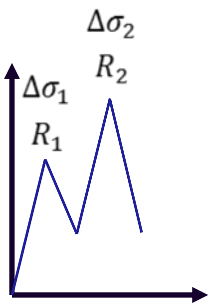
$$N_{total}\ \Delta\sigma_{eq}^{n_p} = \frac{1}{c_p\ (\sqrt{\pi})^{n_p}} \sum_{i=0}^{n-1} \int_{a_i}^{a_{i+1}} \frac{1}{(\beta_i\ \sqrt{a_i})^{n_p}} da$$

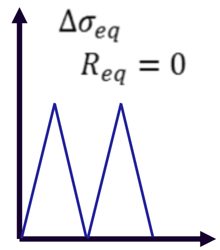
$$\sum_{i=0}^{n-1} N_i\ \Delta\sigma_i^{n_p} = N_{total}\ \Delta\sigma_{eq}^{n_p}$$

$$\Delta\sigma_{eq} = \left(\sum_{i=0}^{n-1} \frac{N_i}{N_{total}}\ \Delta\sigma_i^{n_p}\right)^{\frac{1}{n_p}}$$
$$= \left(\sum_{i=0}^{n-1} p_i(\Delta\sigma_i)\ \Delta\sigma_i^{n_p}\right)^{\frac{1}{n_p}}$$

- Above derivation using basic Paris Law with constant $R_i = \frac{s_{min}}{s_{max}}$ for clarity
- Extends to other crack growth laws – as long as there is no dependency between the stress spectrum and crack size

Equivalent Stress





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

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

$$\Delta\sigma_{eq} = \left[\sum_{i=1}^{K} \frac{n_i}{N_{Tot}} \left((1 - R_i)^{(m-1)n} \right) \Delta\sigma_i^n \right]^{1/n}$$

Equivalent Stress Transformation for Efficient Probabilistic Fatigue-Crack Growth Analysis under Variable Amplitude Loadings
Yibing Xiang and Yongming Liu

Excel example

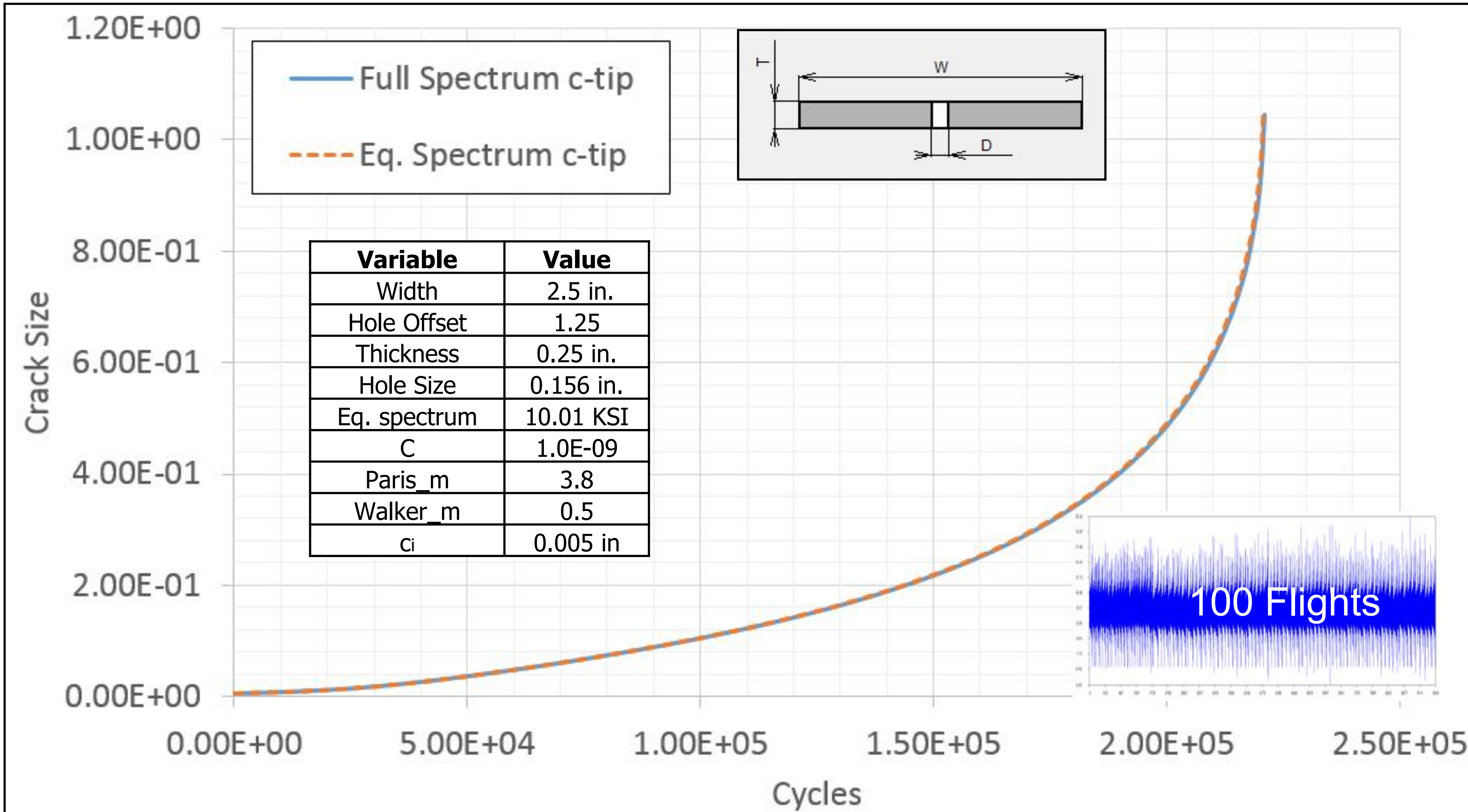
VA CA

a0	0.05				VA Crack Growth				CA Crack Growth		
ni	sigmaMin	sigmaMax	pi*delsigma^n	a	delta K	dadN	Delta a	a	delta K	dadN	Delta a
1	1	40	55592.41	0.05	17.312	5.0781E-05	5.07811E-05	0.05	13.580	2.0187E-05	2.01872E-05
1	2	36	33005.77	0.050050781	15.100	3.0207E-05	3.02074E-05	0.050020187	13.583	2.0203E-05	2.02027E-05
1	2	36	33005.77	0.050080988	15.105	3.0242E-05	3.02421E-05	0.05004039	13.586	2.0218E-05	2.02182E-05
1	2	36	33005.77	0.050111231	15.109	3.0277E-05	3.02768E-05	0.050060608	13.589	2.0234E-05	2.02337E-05
1	2	36	33005.77	0.050141507	15.114	3.0312E-05	3.03116E-05	0.050080842	13.591	2.0249E-05	2.02492E-05
1	2	36	33005.77	0.050171819	15.118	3.0346E-05	3.03464E-05	0.050101091	13.594	2.0265E-05	2.02648E-05
1	3	32	18033.60	0.050202165	12.899	1.6600E-05	1.65996E-05	0.050121356	13.597	2.0280E-05	2.02804E-05
1	3	32	18033.60	0.050218765	12.901	1.6610E-05	1.66101E-05	0.050141636	13.600	2.0296E-05	2.02960E-05
1	3	32	18033.60	0.050235375	12.903	1.6620E-05	1.66205E-05	0.050161932	13.602	2.0312E-05	2.03116E-05
1	3	32	18033.60	0.050251995	12.905	1.6631E-05	1.66309E-05	0.050182244	13.605	2.0327E-05	2.03272E-05
1	3	32	18033.60	0.050268626	12.907	1.6641E-05	1.66414E-05	0.050202571	13.608	2.0343E-05	2.03428E-05
1	3	32	18033.60	0.050285268	12.910	1.6652E-05	1.66519E-05	0.050222914	13.611	2.0359E-05	2.03585E-05
1	3	32	18033.60	0.05030192	12.912	1.6662E-05	1.66624E-05	0.050243272	13.613	2.0374E-05	2.03742E-05
1	3	32	18033.60	0.050318582	12.914	1.6673E-05	1.66728E-05	0.050263646	13.616	2.0390E-05	2.03899E-05
1	3	32	18033.60	0.050335255	12.916	1.6683E-05	1.66833E-05	0.050284036	13.619	2.0406E-05	2.04056E-05
1	3	32	18033.60	0.050351938	12.918	1.6694E-05	1.66938E-05	0.050304442	13.622	2.0421E-05	2.04213E-05
1	5	30	10259.87	0.050368632	11.138	9.5036E-06	9.50363E-06	0.050324863	13.625	2.0437E-05	2.04371E-05
1	5	30	10259.87	0.050378136	11.139	9.5070E-06	9.50704E-06	0.0503453	13.627	2.0453E-05	2.04529E-05
1	5	30	10259.87	0.050387643	11.140	9.5104E-06	9.51045E-06	0.050365753	13.630	2.0469E-05	2.04687E-05
1	5	30	10259.87	0.050397153	11.141	9.5139E-06	9.51386E-06	0.050386222	13.633	2.0484E-05	2.04845E-05
				0.050406667029				0.05040670623			
20			441996.8			Sum	4.0666703E-04			Sum	4.0670623E-04
			30.594023726								

Eq. Stress Examples

Through Crack in a Hole

All solutions using Afgrow

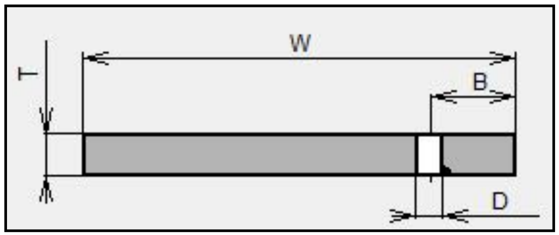


Variable	Value
Width	2.5 in.
Hole Offset	1.25
Thickness	0.25 in.
Hole Size	0.156 in.
Eq. spectrum	10.01 KSI
C	1.0E-09
Paris_m	3.8
Walker_m	0.5
c_i	0.005 in

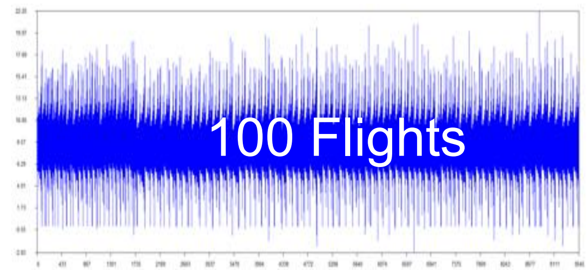
Eq. Stress Examples

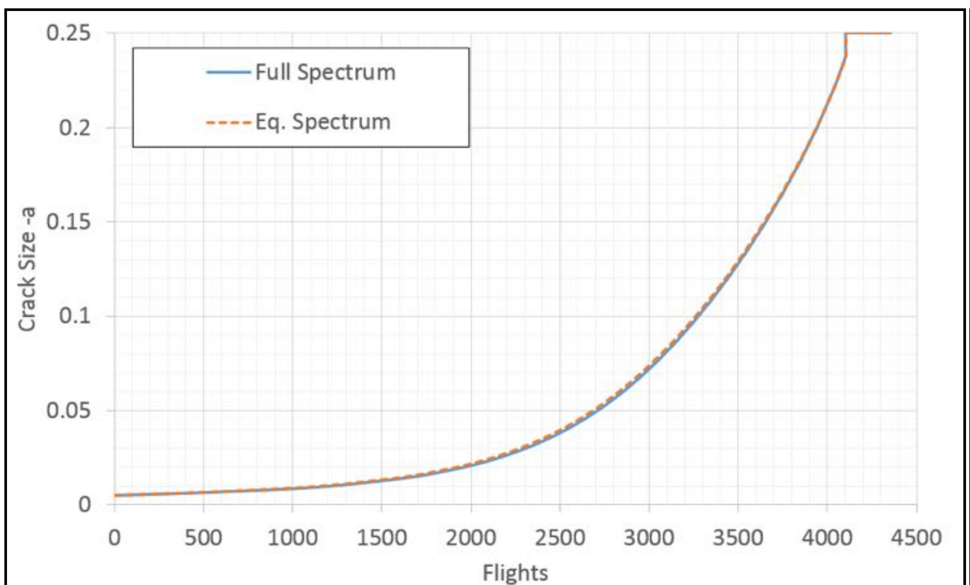
Corner Crack in a Hole

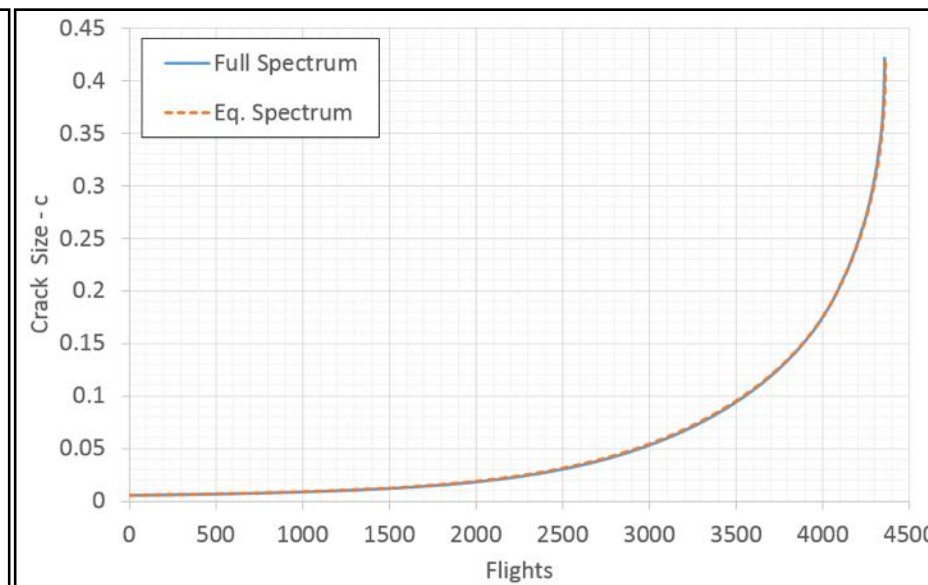
All solutions using Afgrow



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

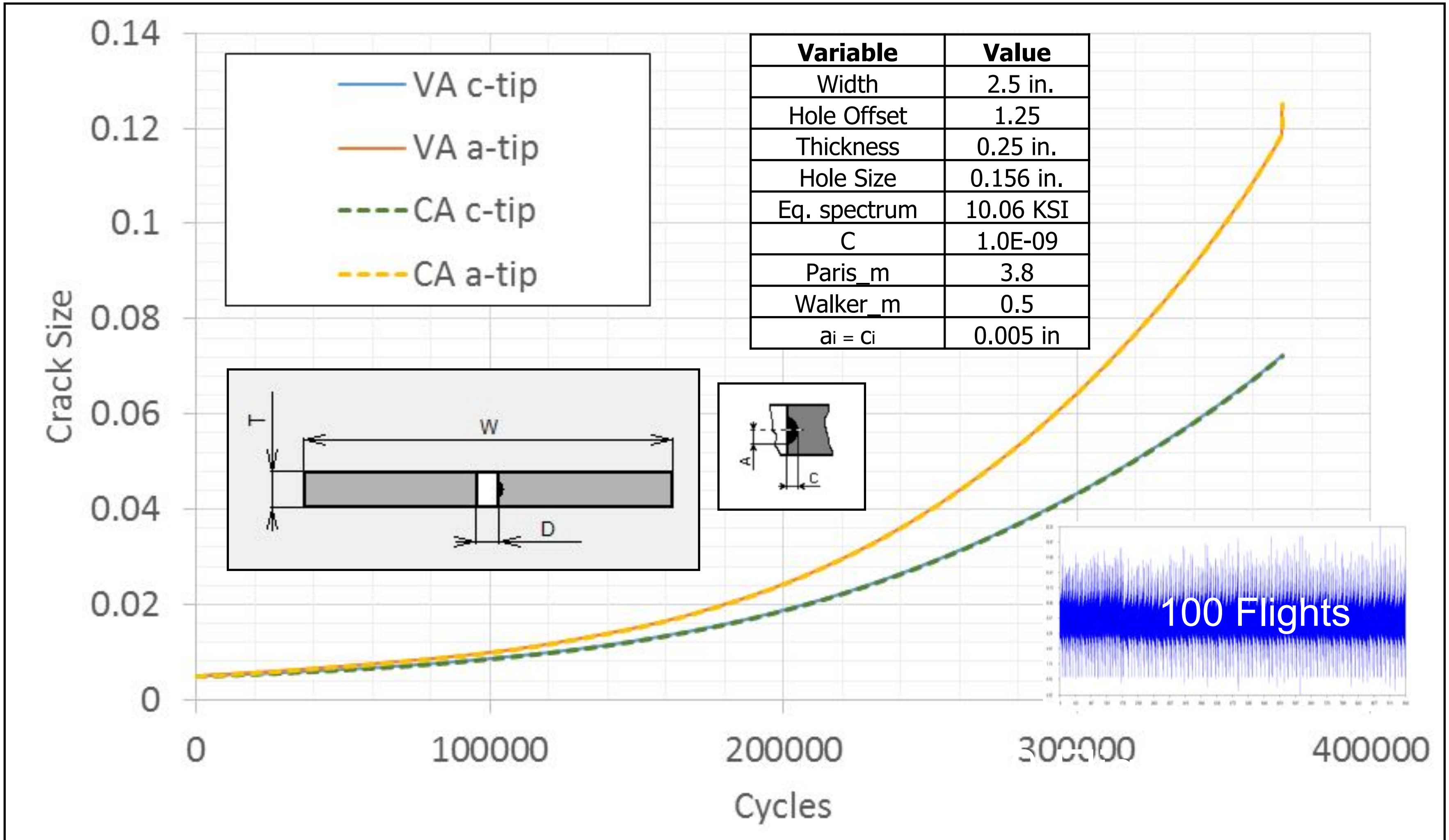






Eq. Stress Examples

Surface Crack in a Hole

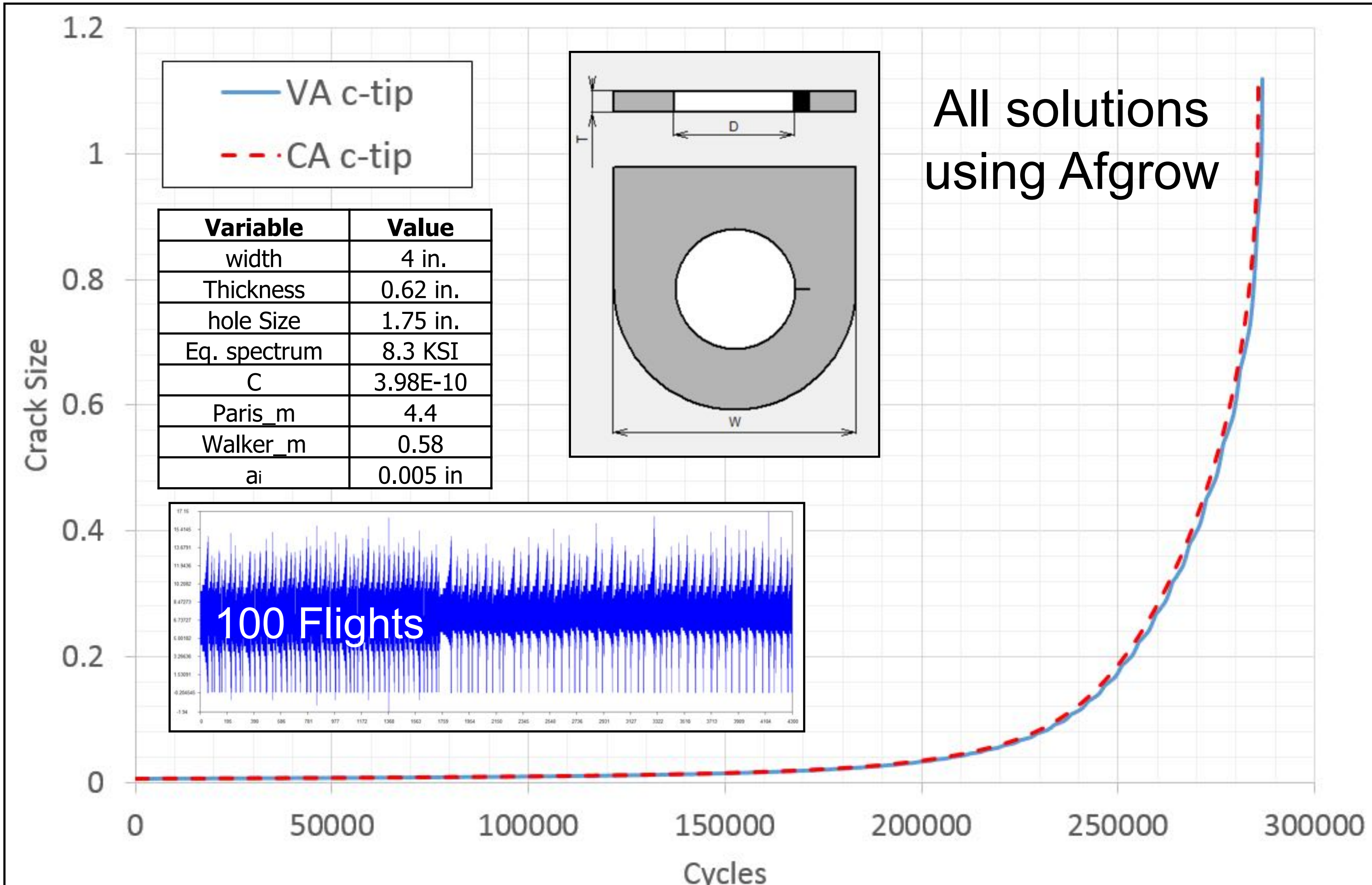


Variable	Value
Width	2.5 in.
Hole Offset	1.25
Thickness	0.25 in.
Hole Size	0.156 in.
Eq. spectrum	10.06 KSI
C	1.0E-09
Paris_m	3.8
Walker_m	0.5
$a_i = c_i$	0.005 in

All solutions using Afgrow

Eq. Stress Examples

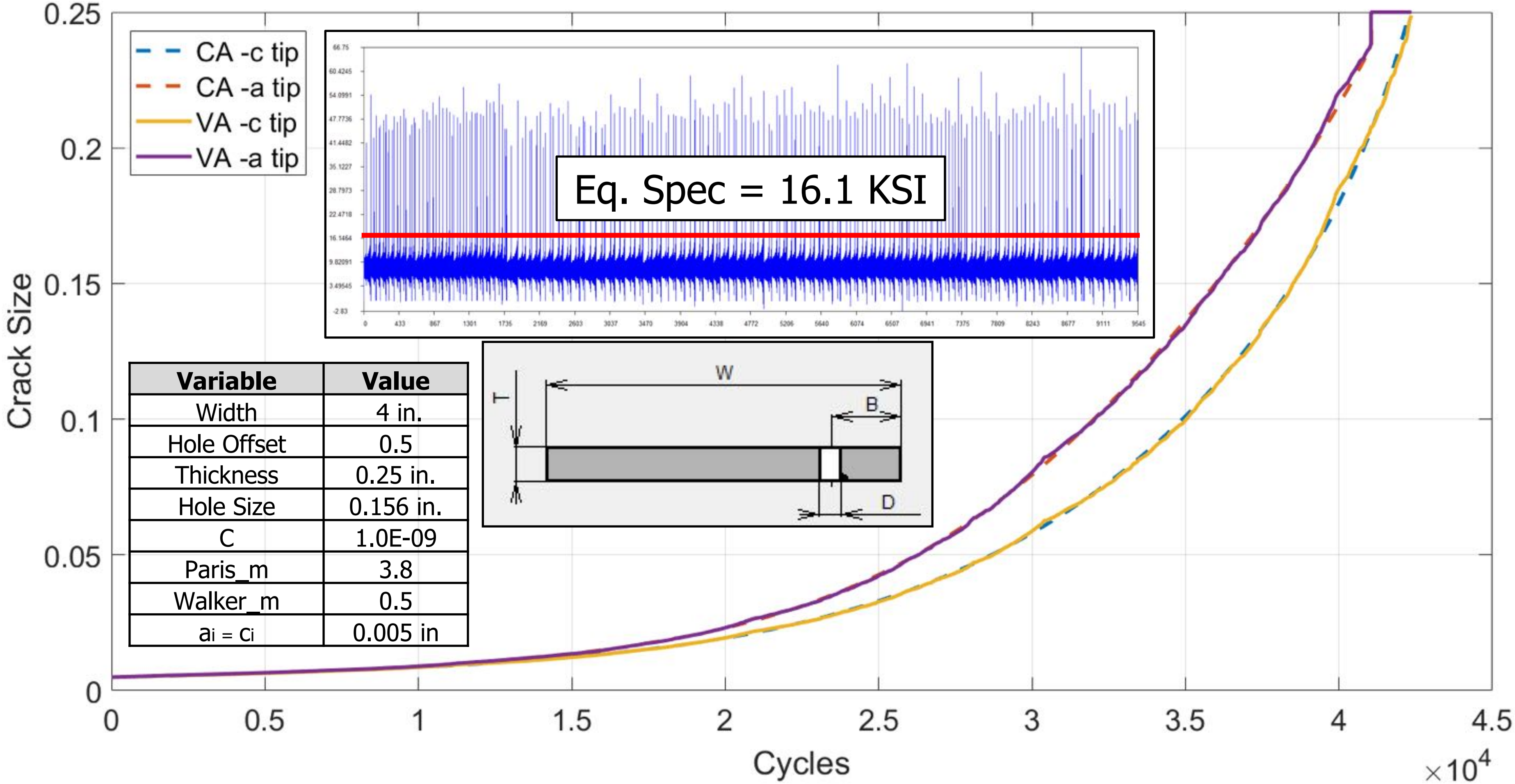
Through Crack in a Lug



Variable	Value
width	4 in.
Thickness	0.62 in.
hole Size	1.75 in.
Eq. spectrum	8.3 KSI
C	3.98E-10
Paris_m	4.4
Walker_m	0.58
a_i	0.005 in

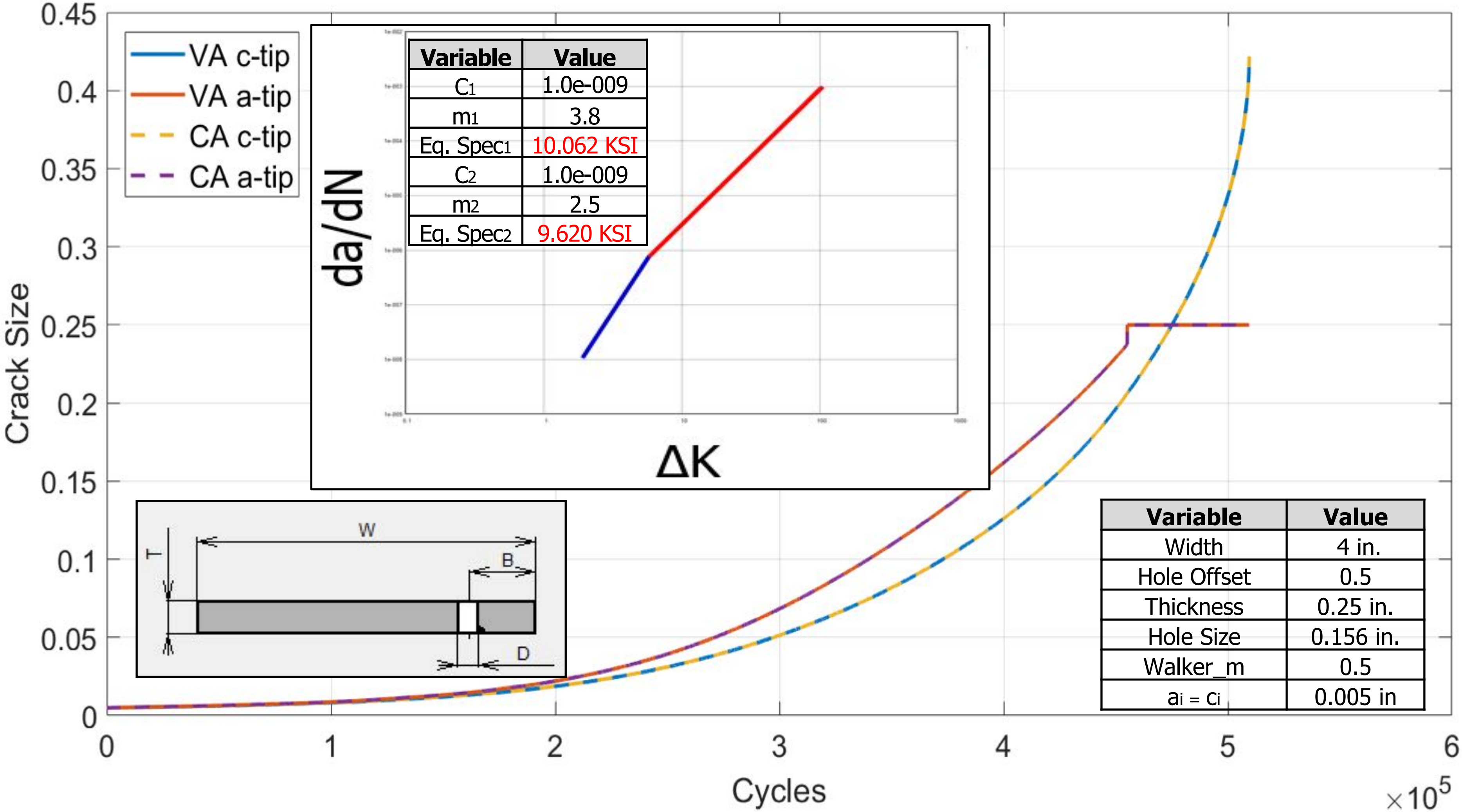
Over Load Example

All solutions using Afgrow



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

Bilinear Paris Example



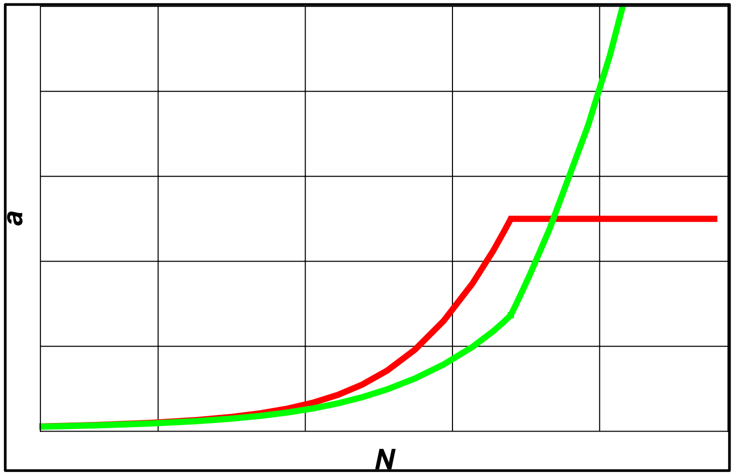
Variable	Value
C_1	1.0e-009
m_1	3.8
Eq. $Spec_1$	10.062 KSI
C_2	1.0e-009
m_2	2.5
Eq. $Spec_2$	9.620 KSI

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

All solutions using Afgrow

Fast ODE Solver

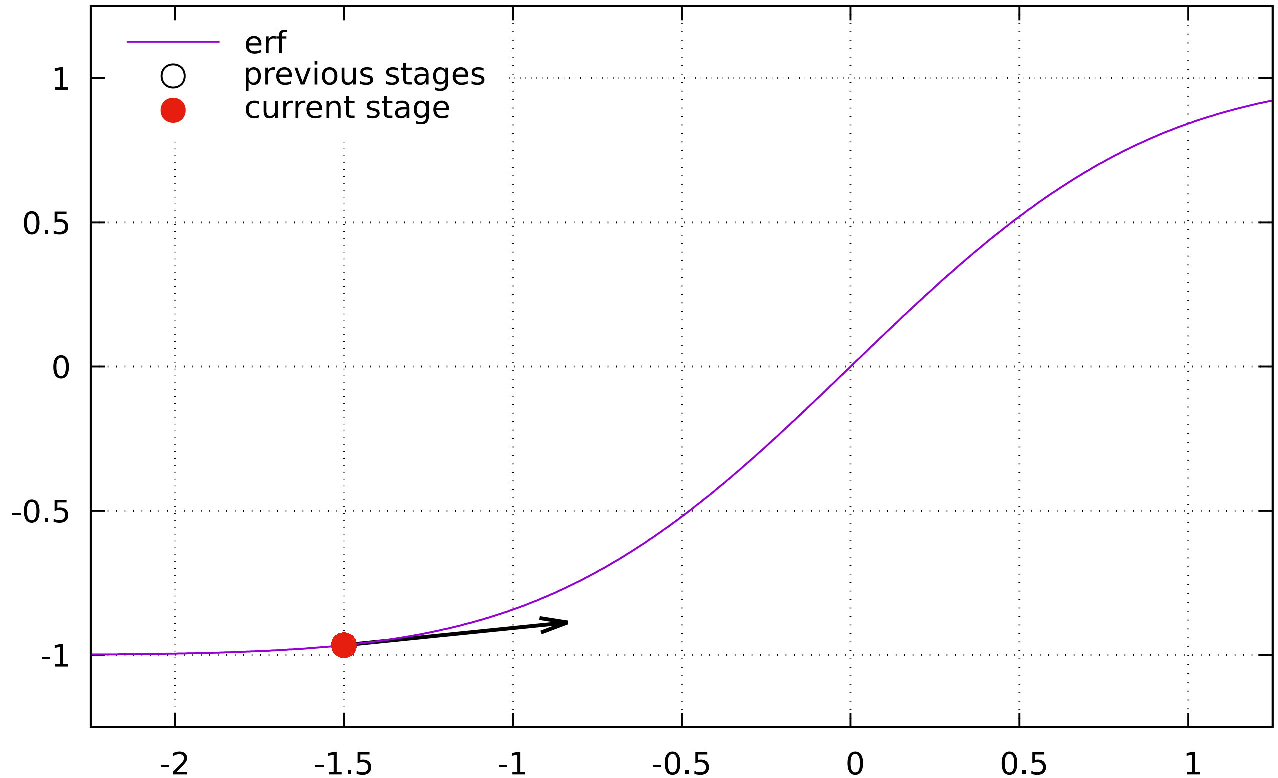
- Based on best practices from well known and available ODE solvers, e.g., Petsc, Sundials, RKSuite
- Paired Runge-Kutta implementations, 2(3), 4(5), 7(8), e.g., 4^{th} and 5^{th} order solutions computed simultaneously. Gives high quality error estimate.
- Automatically selects step size based on user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



RKSUITE_90

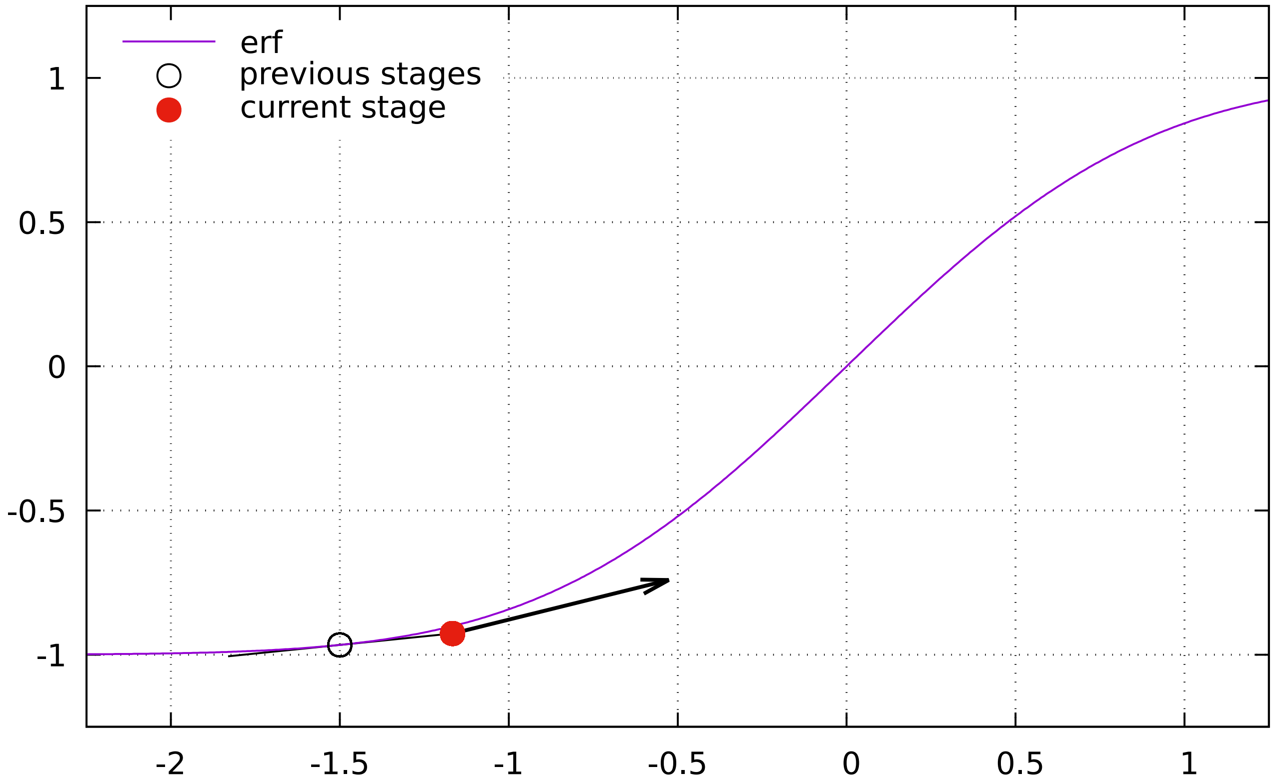
- Developed by Richard Brankin and Ian Gladwell
- Incorporated into NAG ODE library
- 3 paired Runge-Kutta implementations (2-3, 4-5, 7-8)
- Range and step integration methods
- Automatically selects initial step size
- Initially used module variables for integral parameters
  - For Parallelization in SMART|DT, the code was modified to use Fortran 2008 extensible types to pass integral parameters

Example Bogacki-Shampine 4-5 pair



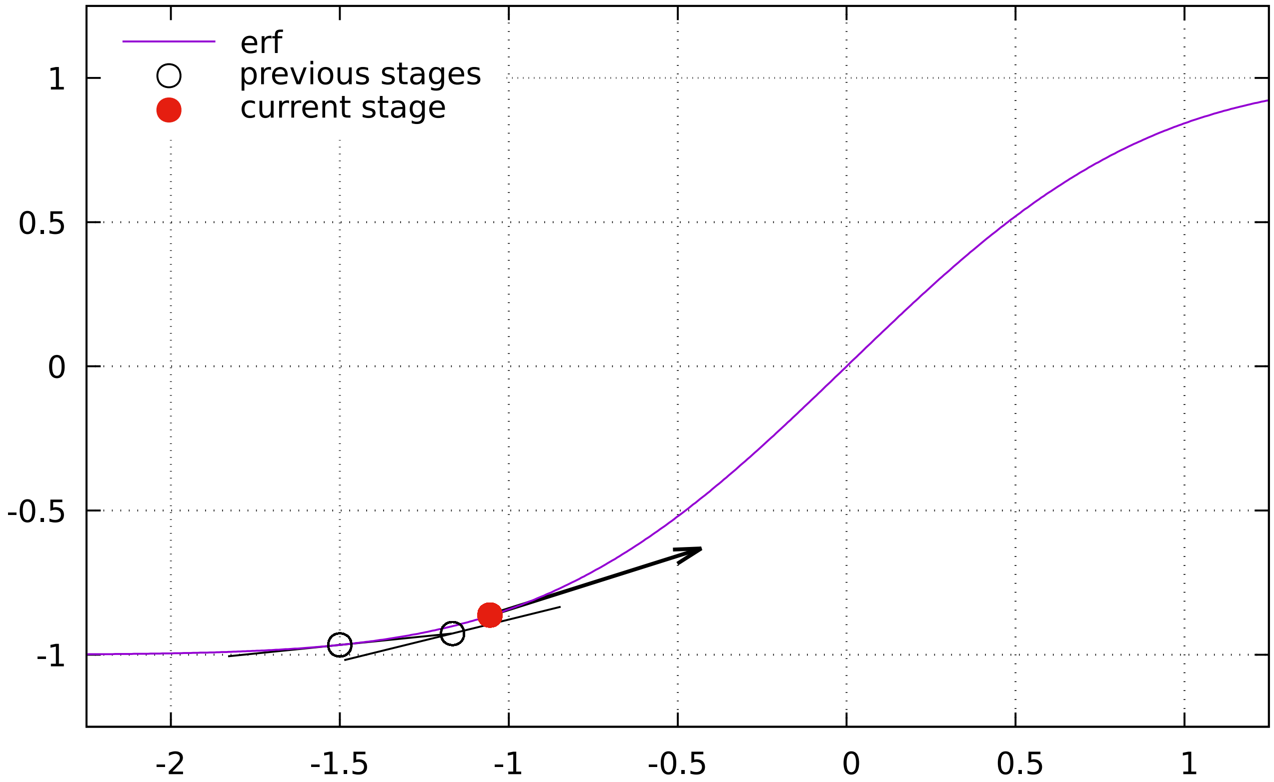
- Quick example using erf, starting at x=-1.5, stepping to x=0.5
- This RK pair uses 8 stages per step
- … stage 1 …

Example Bogacki-Shampine 4-5 pair



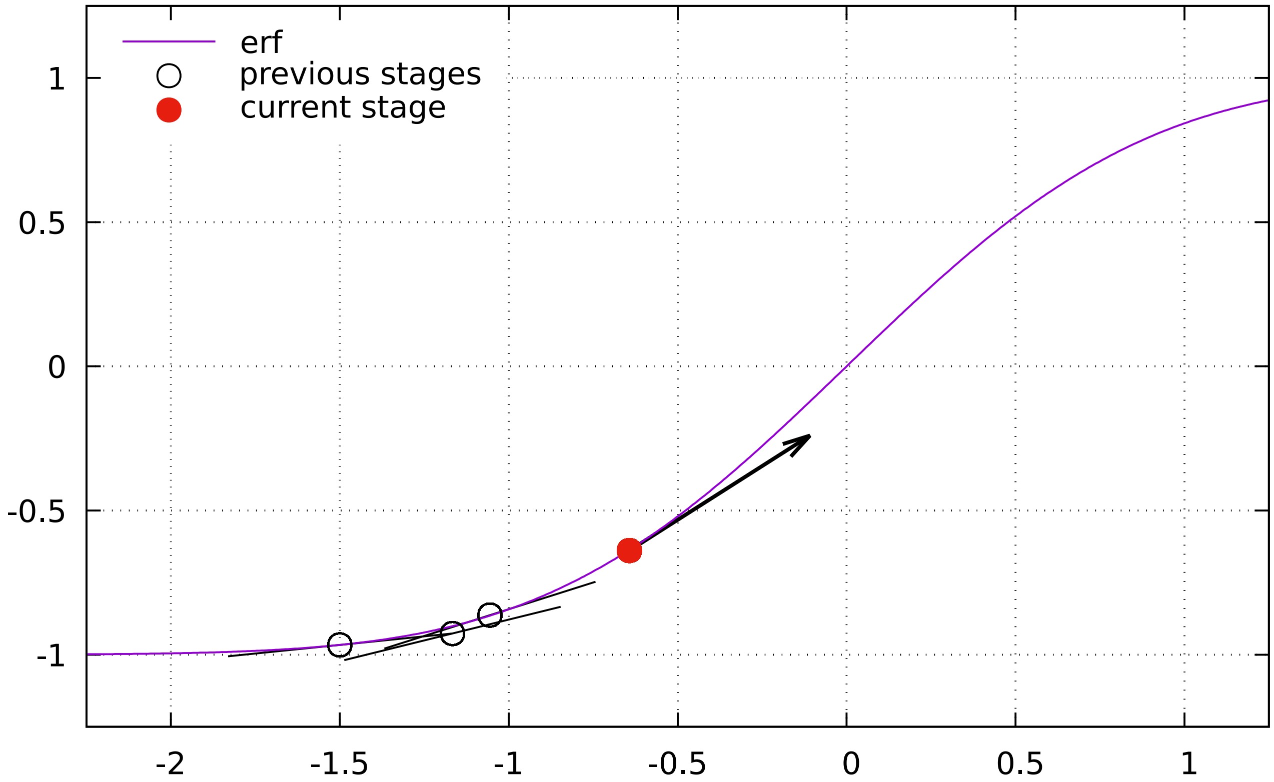
- … stage 2 …

Example Bogacki-Shampine 4-5 pair



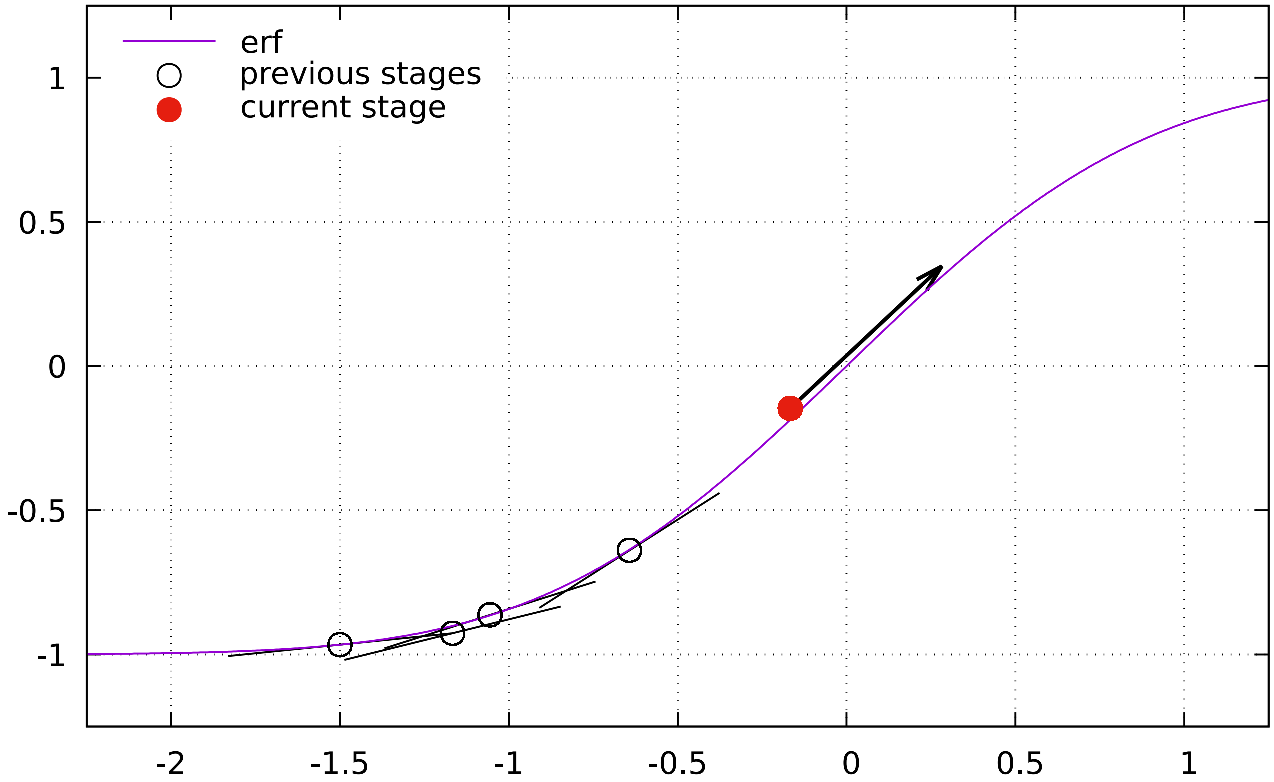
- … stage 3 …

Example Bogacki-Shampine 4-5 pair



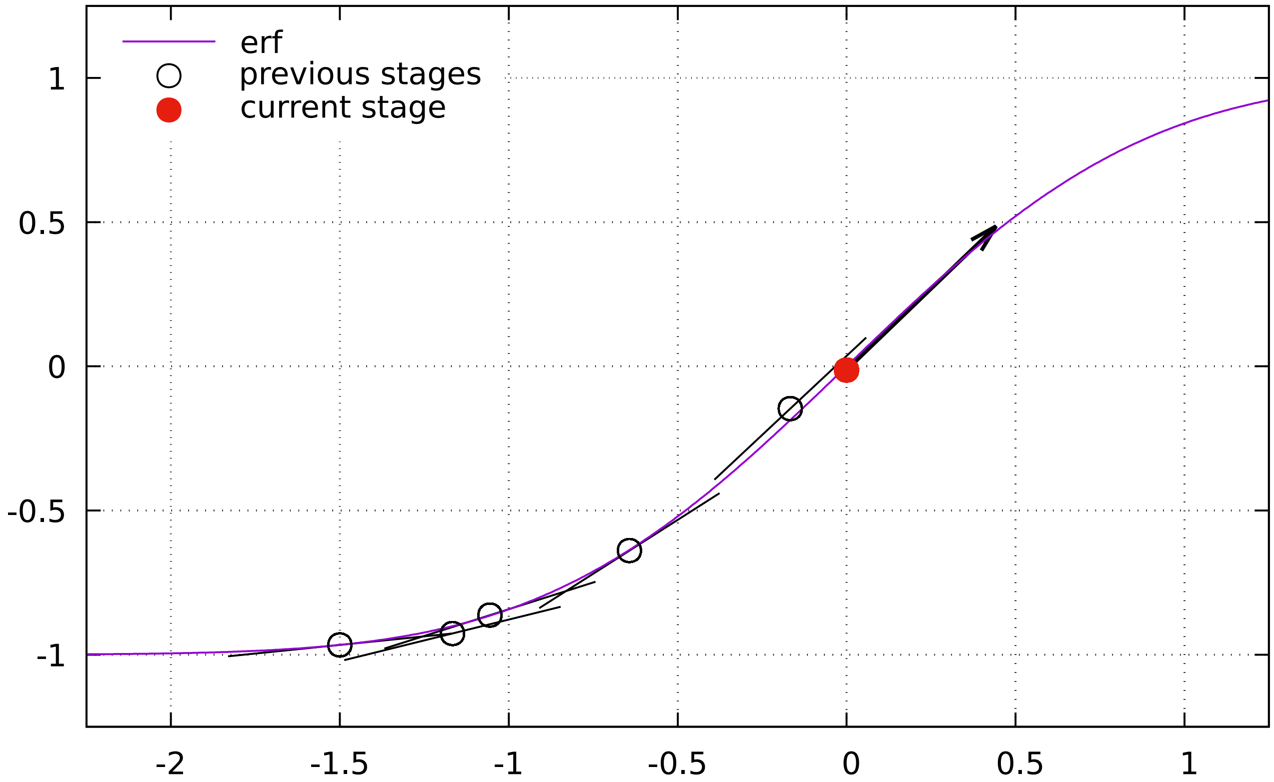
- … stage 4 …

Example Bogacki-Shampine 4-5 pair



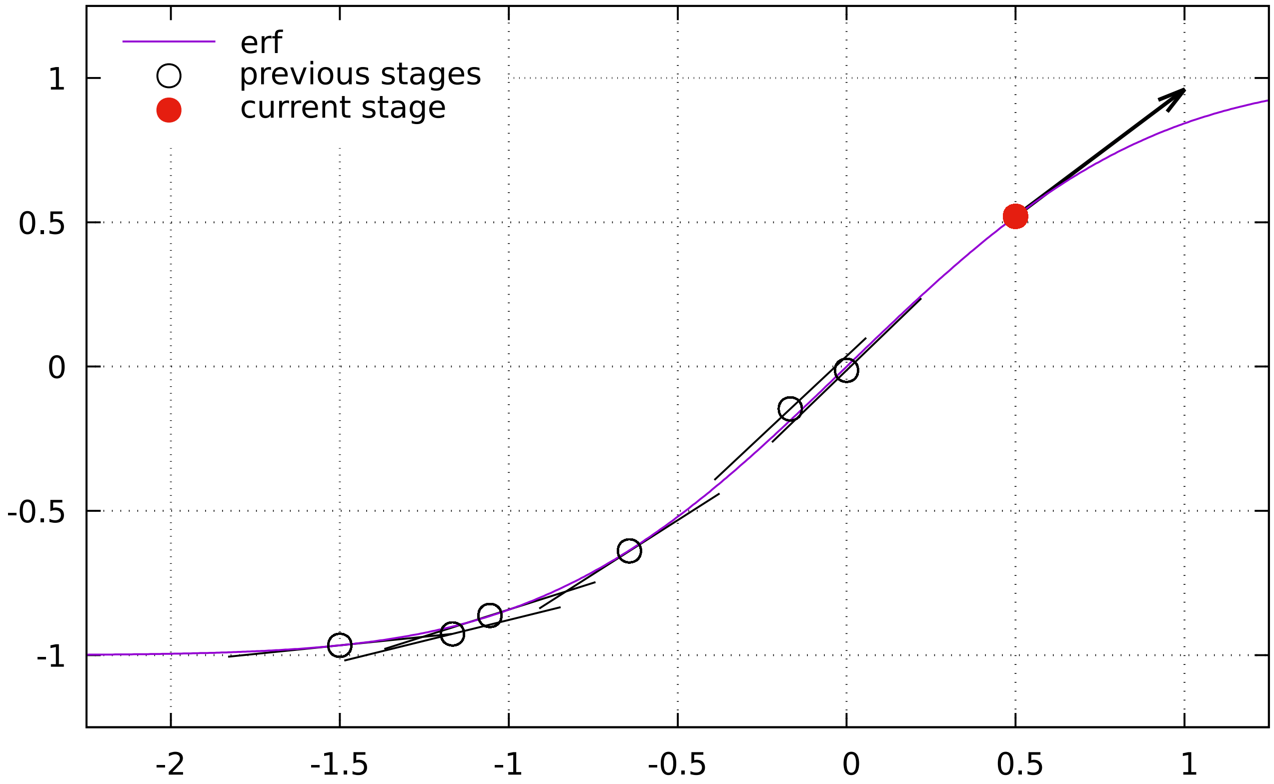
- … stage 5 …

Example Bogacki-Shampine 4-5 pair



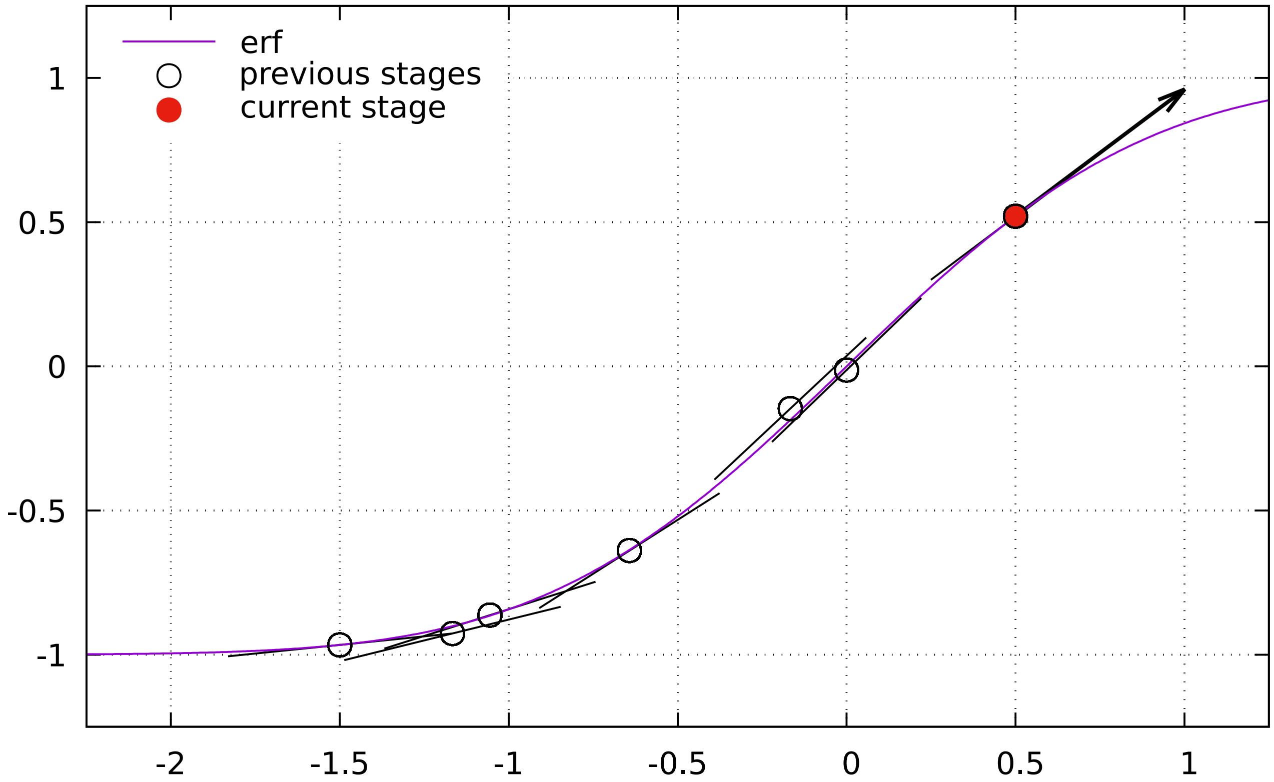
- … stage 6 …

Example Bogacki-Shampine 4-5 pair



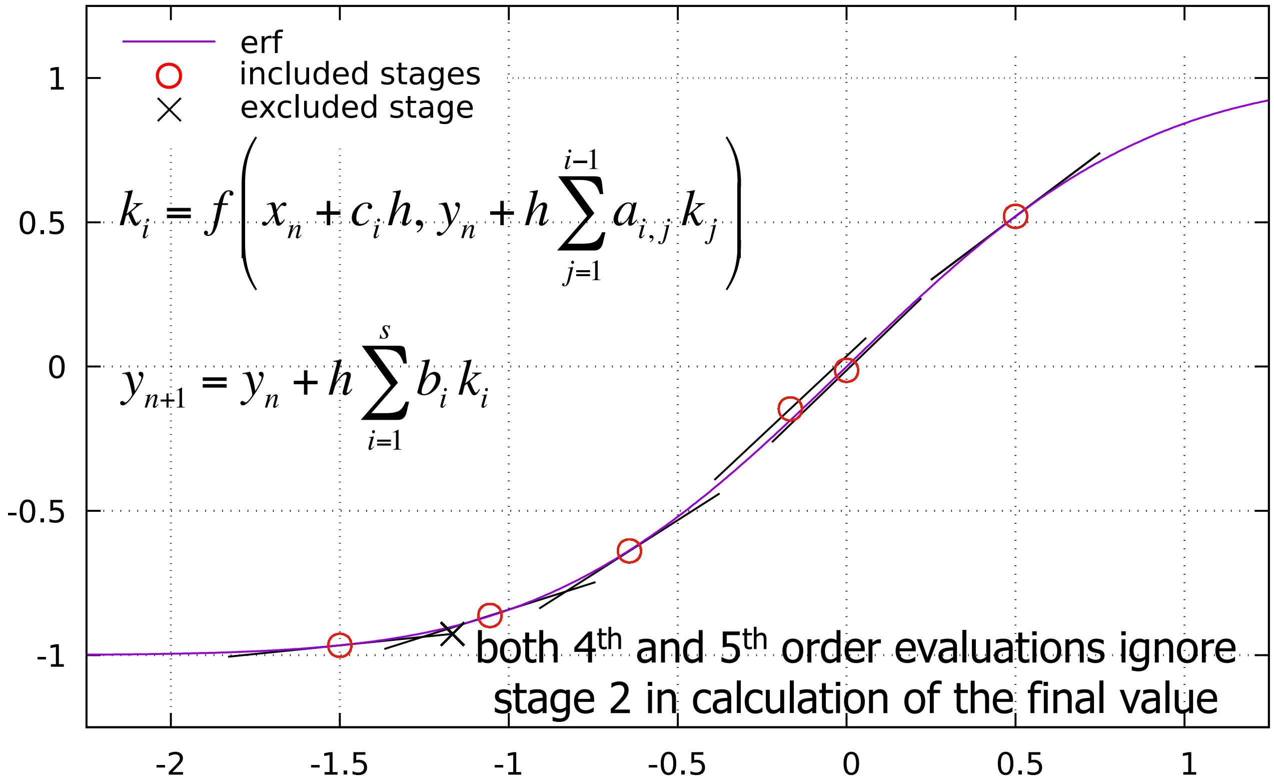
- … stage 7 …

Example Bogacki-Shampine 4-5 pair



- … stage 8 …

Example Bogacki-Shampine 4-5 pair

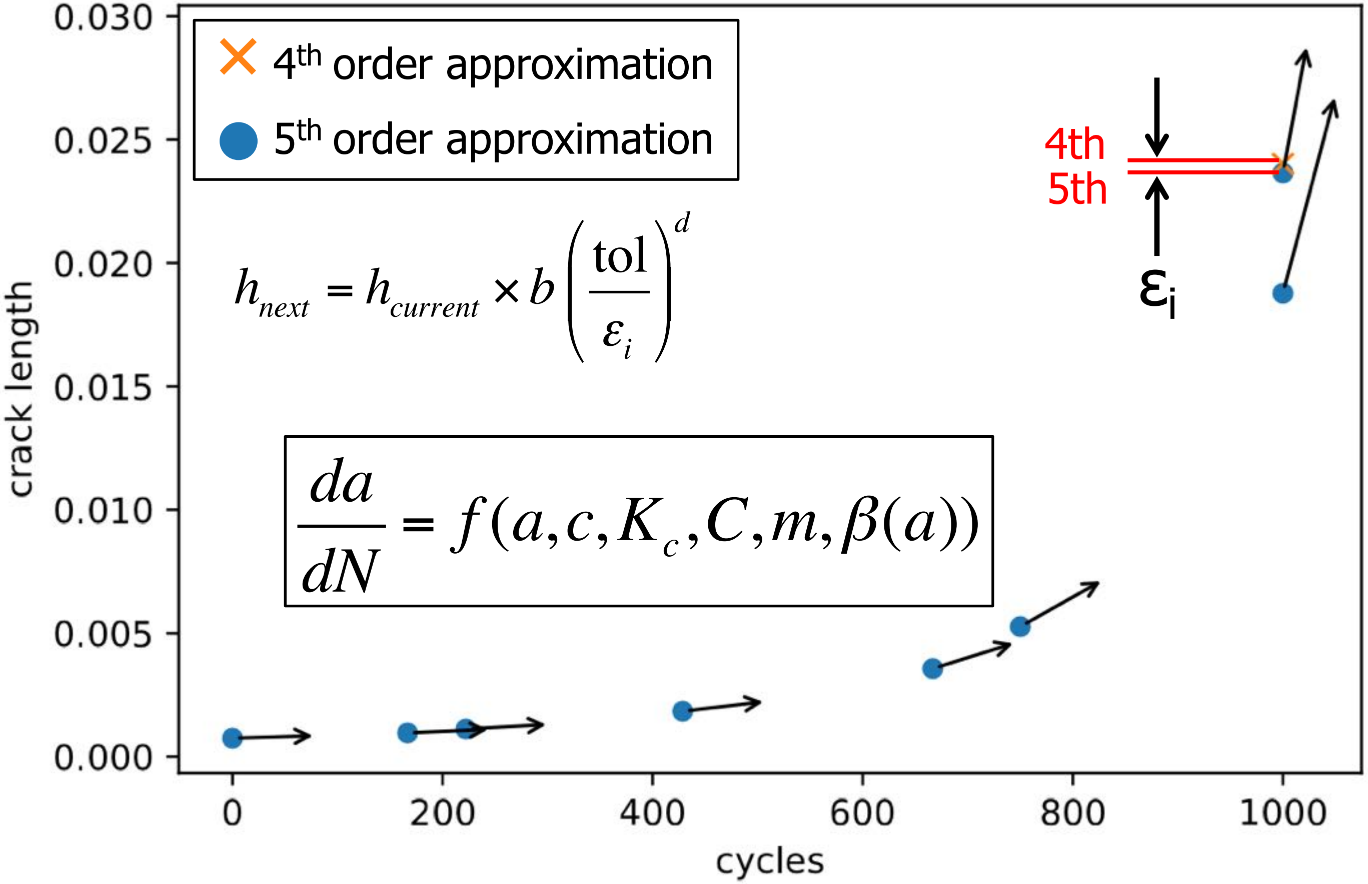


$$k_i = f\left(x_n + c_i h, y_n + h\sum_{j=1}^{i-1} a_{i,j} k_j\right)$$

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i$$

- For Runge-Kutta formulas, the order of the method is determined by constraints satisfied by the coefficients
- Different linear combinations of the same stages can produce both 4th and 5th order estimates of y_{n+1}

Adaptive Step Size Control



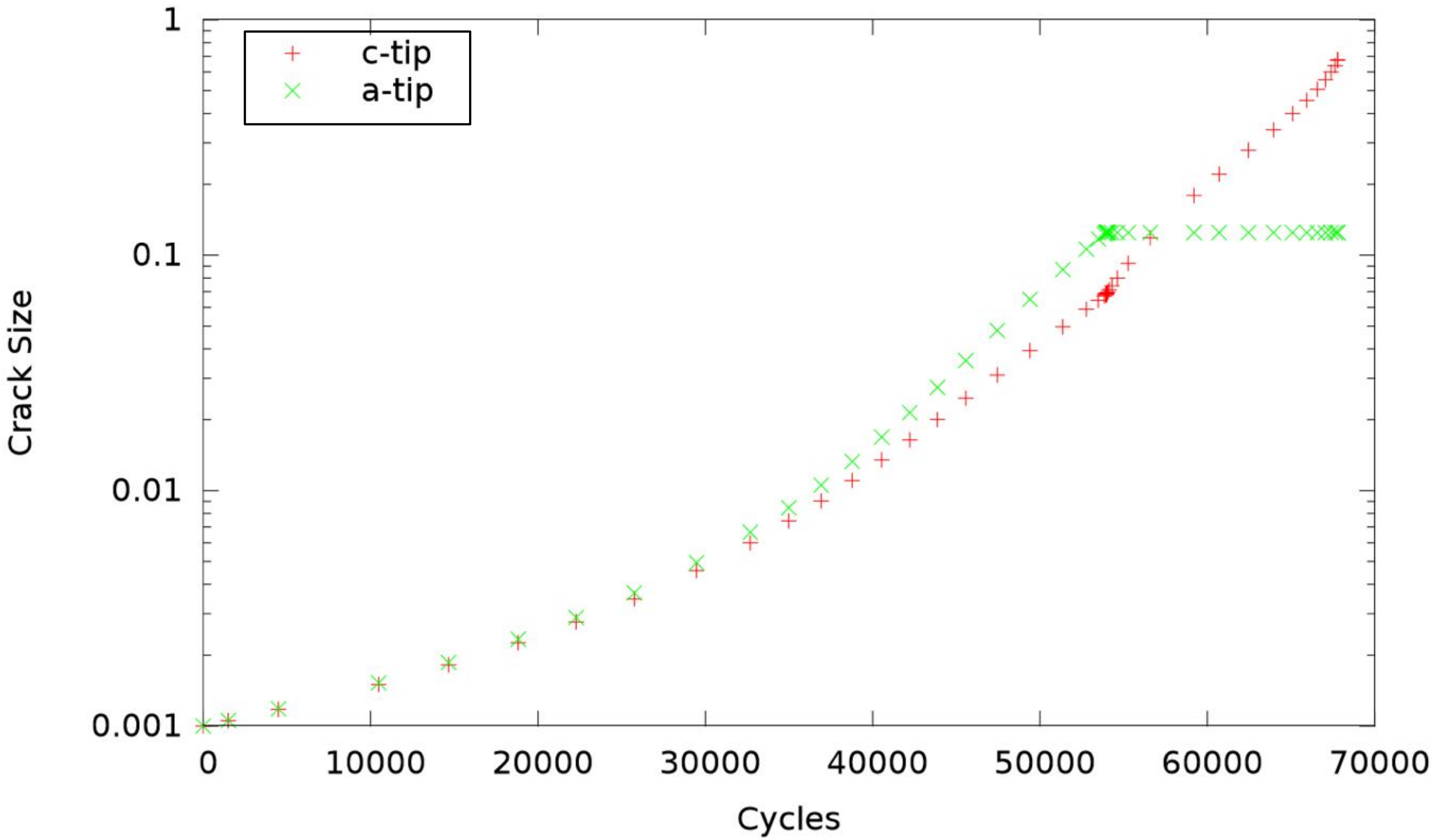
$$h_{next} = h_{current} \times b \left(\frac{\text{tol}}{\varepsilon_i} \right)^d$$

$$\frac{da}{dN} = f(a, c, K_c, C, m, \beta(a))$$

- 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

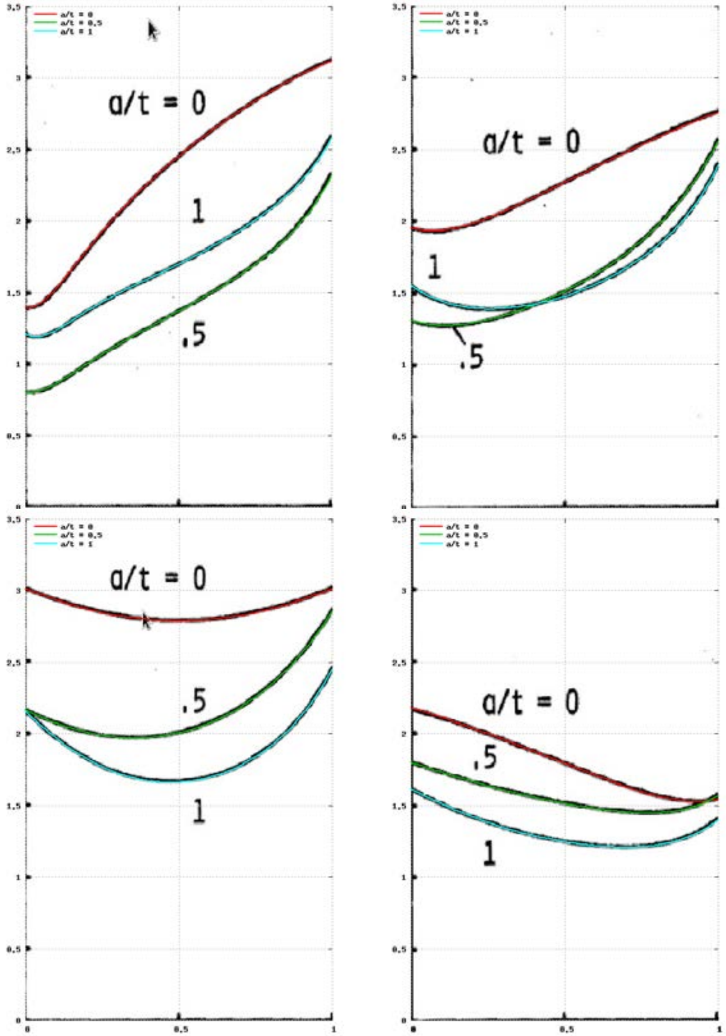


Internal K-Solutions

	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

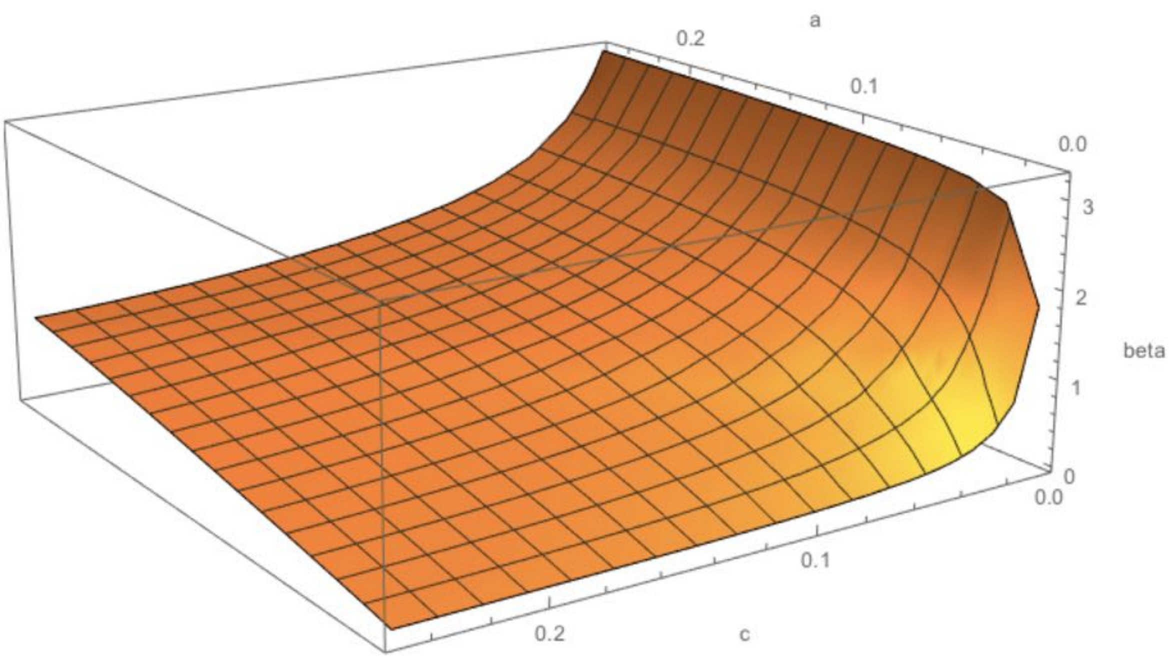
Newman-Raju



Beta Tables

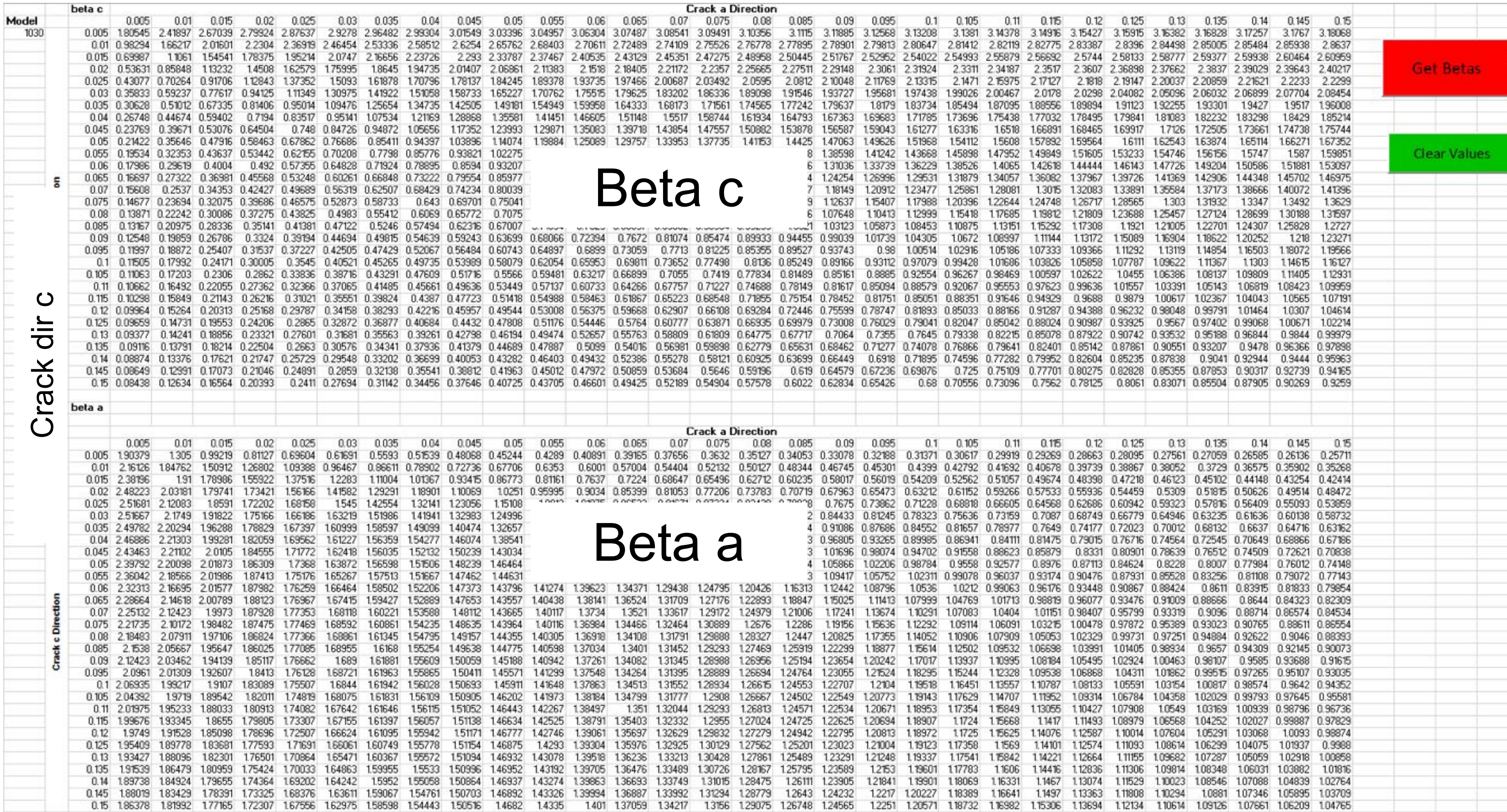
! Thru crack betas				
c1	β_1			
c2	β_1			
…	…			
cN	β_1			
! C-tip direction				
	a1	a2	…	aN
c1	β_{11}	β_{12}	…	β_{1N}
c2	β_{21}	β_{22}	…	β_{2N}
…	…	…	…	…
cN	β_{N1}	β_{N2}	…	β_{NN}
! A-tip direction				
	a1	a2	…	aN
c1	β_{11}	β_{12}	…	β_{1N}
c2	β_{21}	β_{22}	…	β_{2N}
…	…	…	…	…
cN	β_{N1}	β_{N2}	…	β_{NN}

- Can use AFGROW / NASGRO to generate beta tables for any solution
- Allows ICG to solve any crack models with high accuracy



Beta Table

Crack dir a



Example Problems

- Solve the first order ODE (thru crack) or the coupled system of ODEs (corner, surface).
- N – independent variable
- a,c – dependent variables.

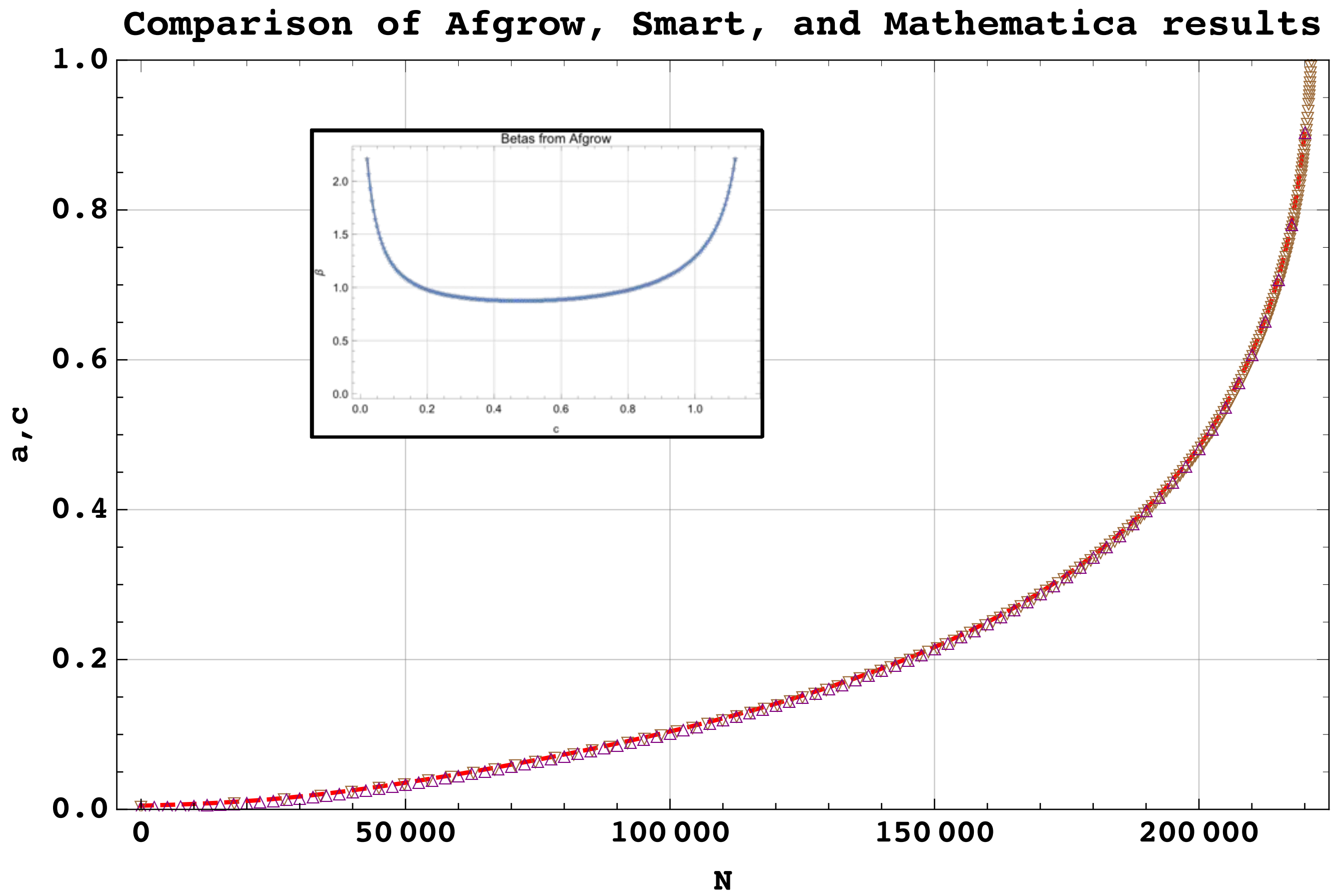
$$\frac{da}{dN} - C(\Delta K(a,c))^m = 0$$

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

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

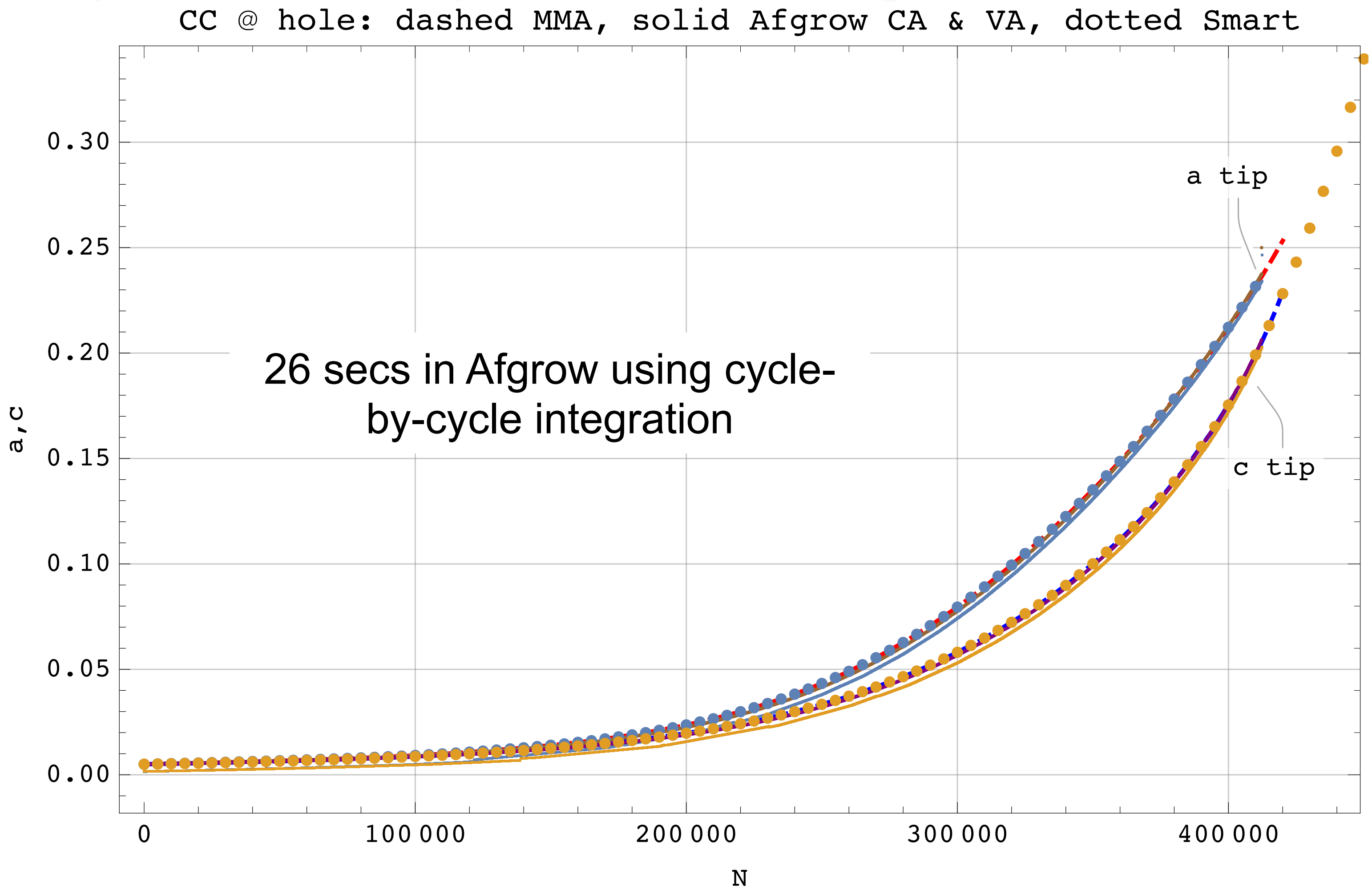
Through Crack at Hole
(Tension)

- Cparis = 10^{-9}, nparis = 3.8, delsigma =10.062 ksi



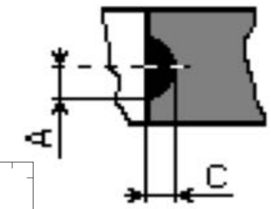
Corner Crack at Hole
(Tension)

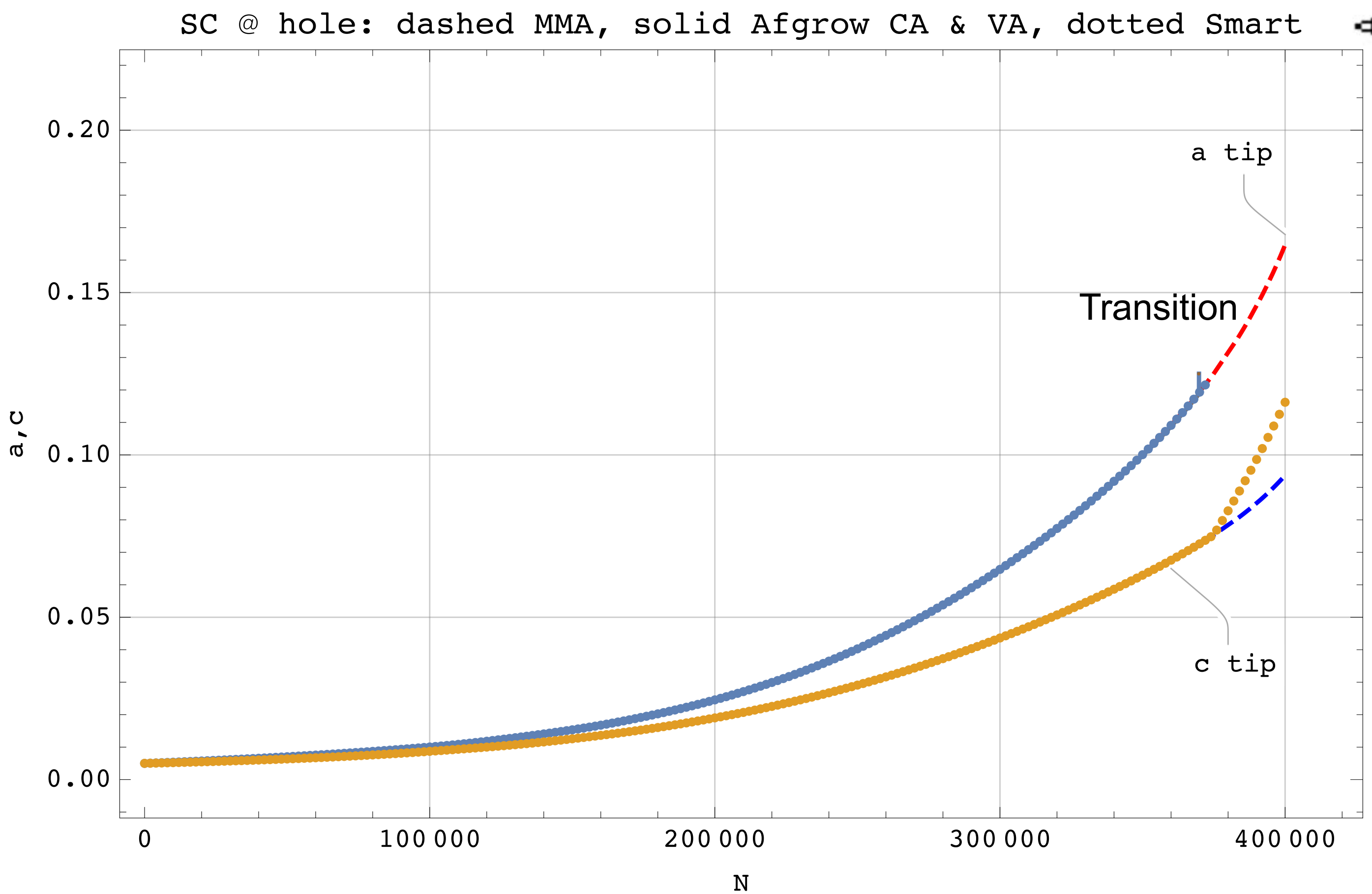
- Cparis = 10^{-9}, nparis = 3.8, delsigma =10.062 ksi



Surface Crack at Hole
(Tension)

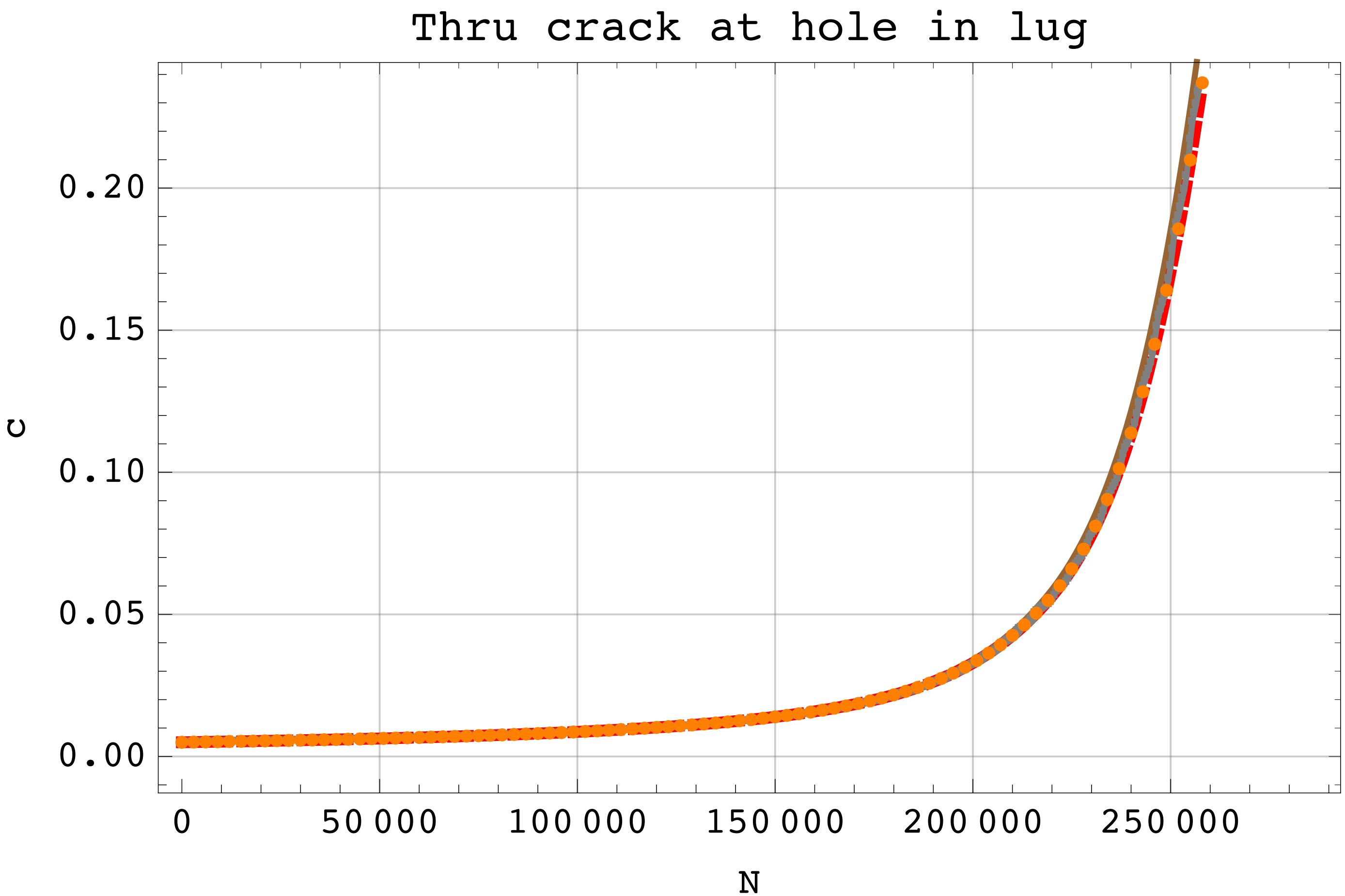
- Cparis = 10^{-9}, nparis = 3.8, delsigma =10.062 ksi





Thru Crack at Lug
(Tension)

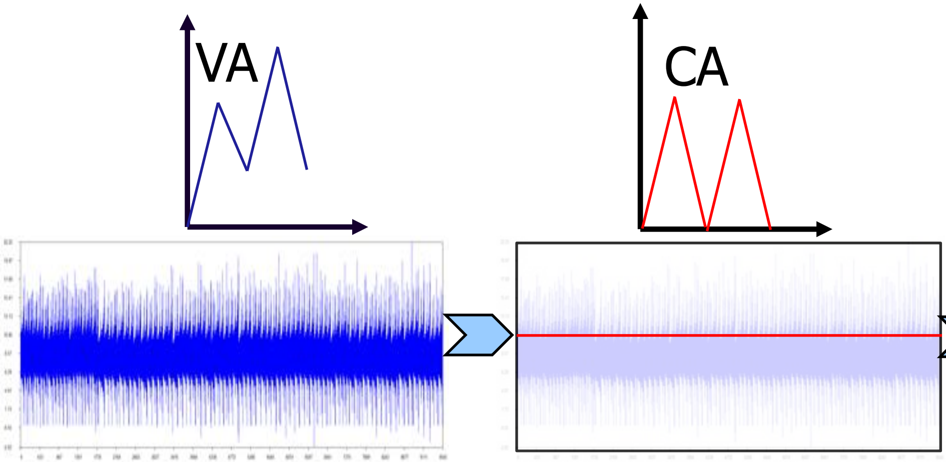
- Cparis = 10^{-9}, nparis = 3.8, delsigma =8.3 ksi



Crack Growth Capabilities

	Afgrow	Nasgro	ICG
Create avsn	Y	Y	coming
MCS	Y	Y	Y
NI	Y	Y	coming
Kriging	Y	Y	coming
RUL	Y	Y	coming
K solutions	Comprehensive	Comprehensive	Newman-Raju Read Beta tables (tension only)
Weight functions	Comprehensive	Comprehensive	N
Net section yield	Y	Y	coming
Retardation	Y	Y	N
Adaptive error control	% Δa	% Δa	RK4(5)
Parallel capable	N	Y	Y (multi-threaded)

Internal CG Code



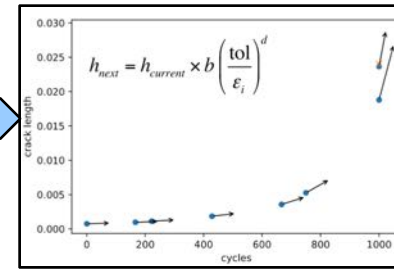
ODE Formulation

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

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

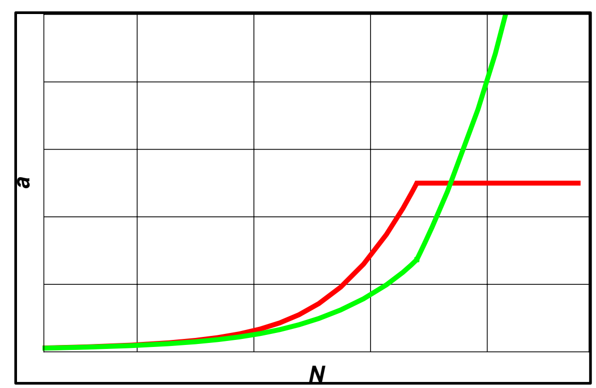
$$Initial\ Conditions: a(0) = a_i, c(0) = c_i$$

RK ODE Solver



ICG Capabilities	
Method	4-5th order Runge-Kutta
Accuracy	Error controlled by user tolerance
Speed	~20000/sec single proc.
Parallel	95% speedup on 8 proc.
K solutions	Newman-Raju, read beta tables

Crack Growth Result



Conclusions

- Equivalent constant is extremely accurate at predicting variable amplitude crack growth – for all problems to date
- Adaptive RK algorithm to grow the crack is very effective (~20000 evaluations/sec)
  - Capability to read beta tables provides an attractive method to incorporate a variety of crack models.
- The top 100 (or so) damaging realizations can be further examined for potential reanalysis