

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**

16-21 September 2018 | DoubleTree by Hilton, Cape Cod, Hyannis, MA, USA



AFRL Digital Twin Vision



CBM+SI & Airframe Digital Twin

Informational Briefing
Spring-Summer 2011

Dr. Eric J. Tuegel
Senior Aerospace Engineer
AFRL Air Vehicles Directorate
Eric.tuegel@wpafb.af.mil

Dr. Pamela A. Kobryn
Future Capability Lead for Sustainment
AFRL Air Vehicles Directorate
Pamela.kobryn@wpafb.af.mil



CBM+SI Far-term Vision

per AFRL CBM+SI Workshop - Feb 2009



“Digital Twin”: 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 **flied 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**.



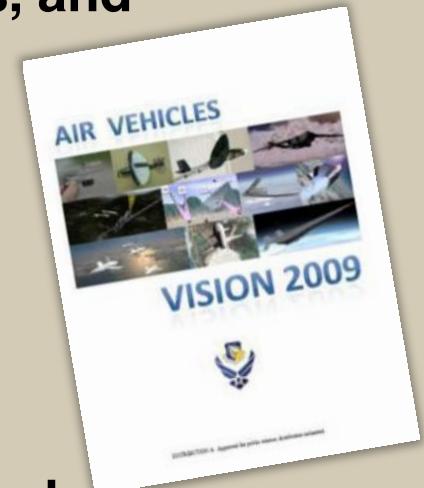


CBM+SI Mid-term Vision

per AFRL Air Vehicles Directorate's "Vision 2009"

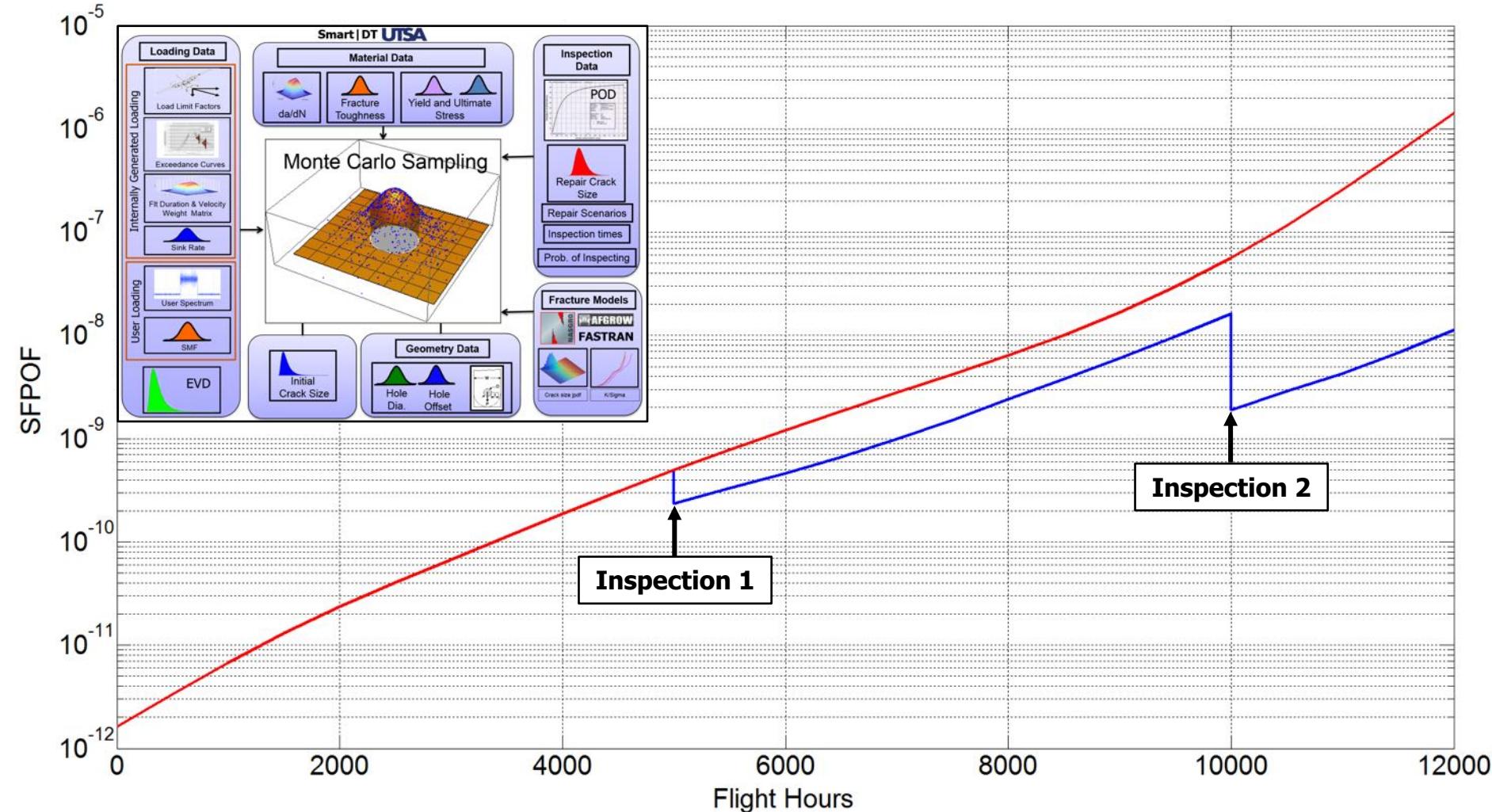


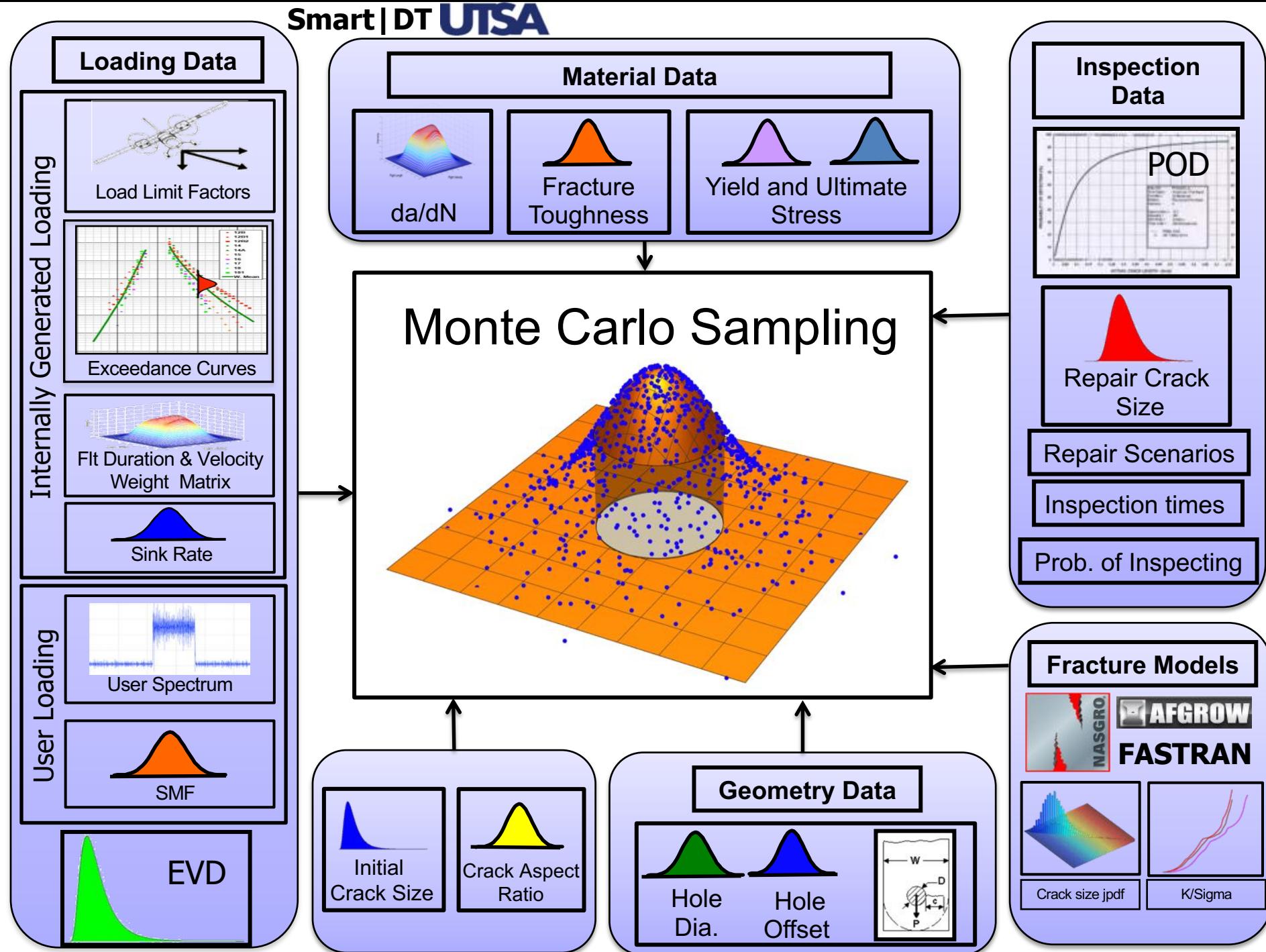
- In ten years, aircraft lifecycle management and maintenance practices will be completely transformed from an inefficient, inconsistent, disparate, and labor-intensive state to an **efficient, standardized, integrated, and semi-autonomous** state.
- The **structural capability** of individual airframes 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







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 lifting 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
- ✓ 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
- ✓ 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

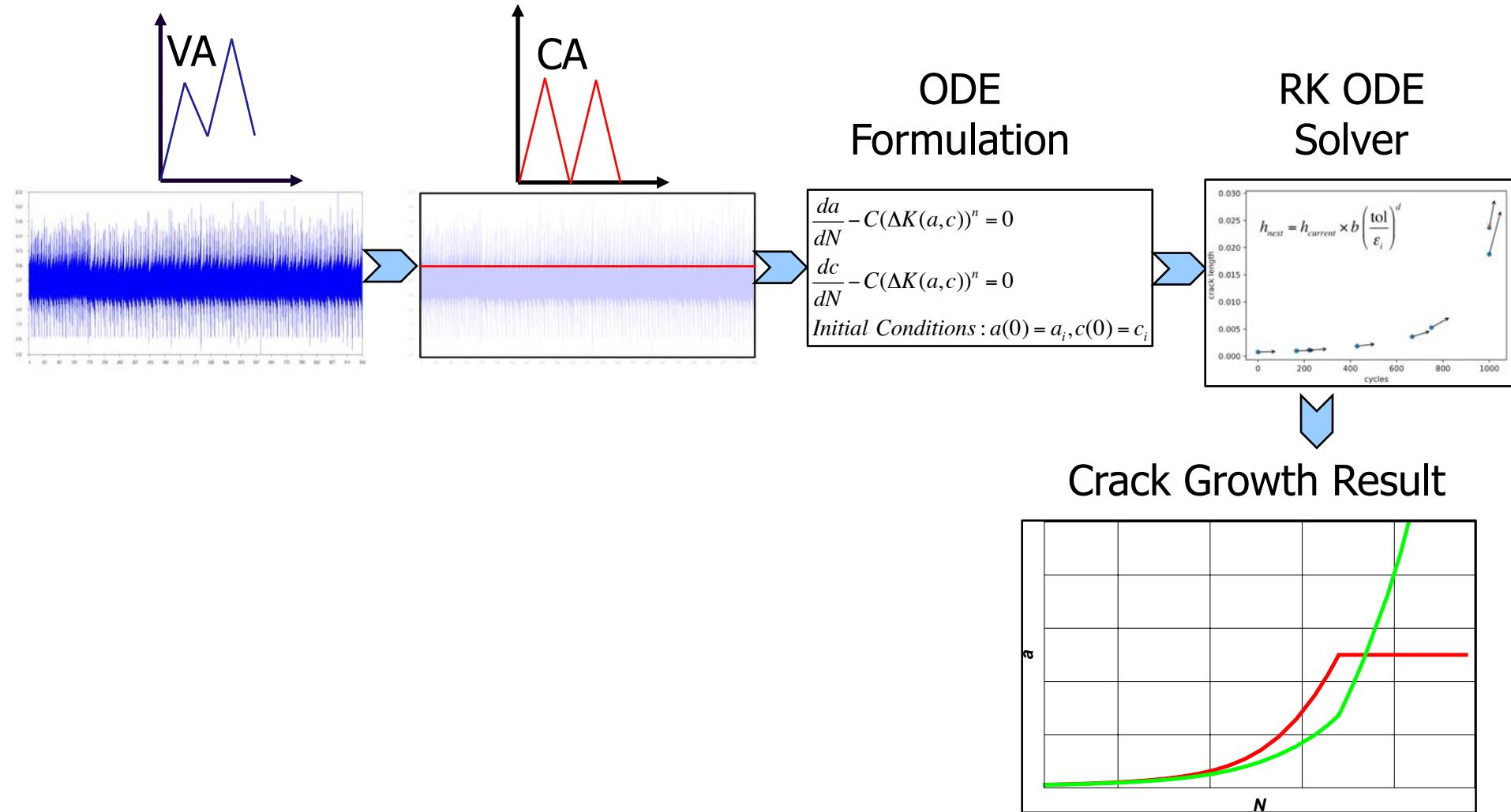


Ultrafast Approach

- 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

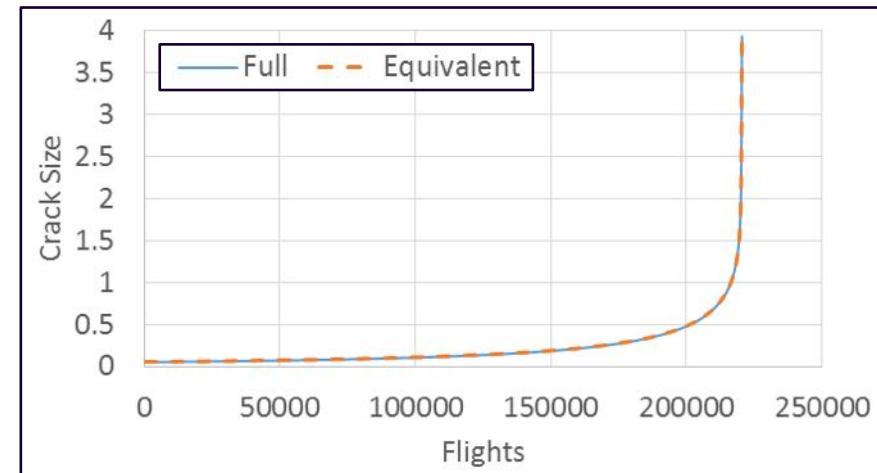
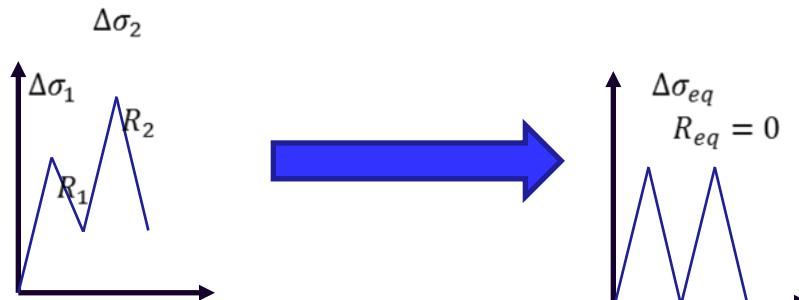


Internal CG Code



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.





Equivalent Stress Formulation

Variable Amplitude

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

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



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

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

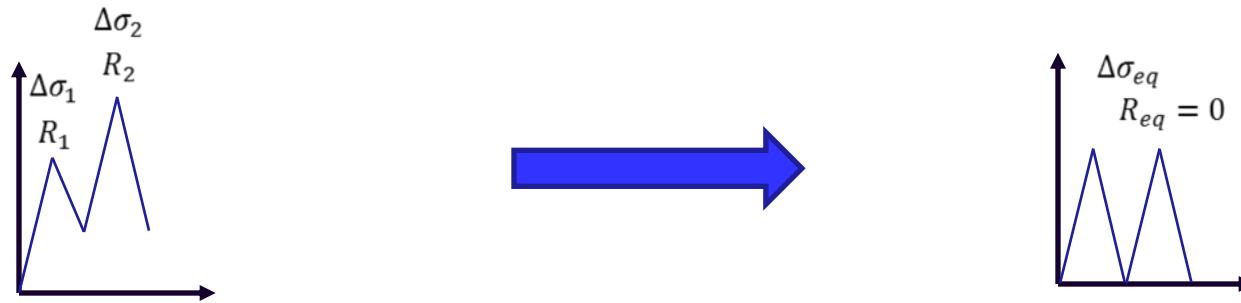
Constant Amplitude

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

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

- 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



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

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

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



Excel example

VA

CA

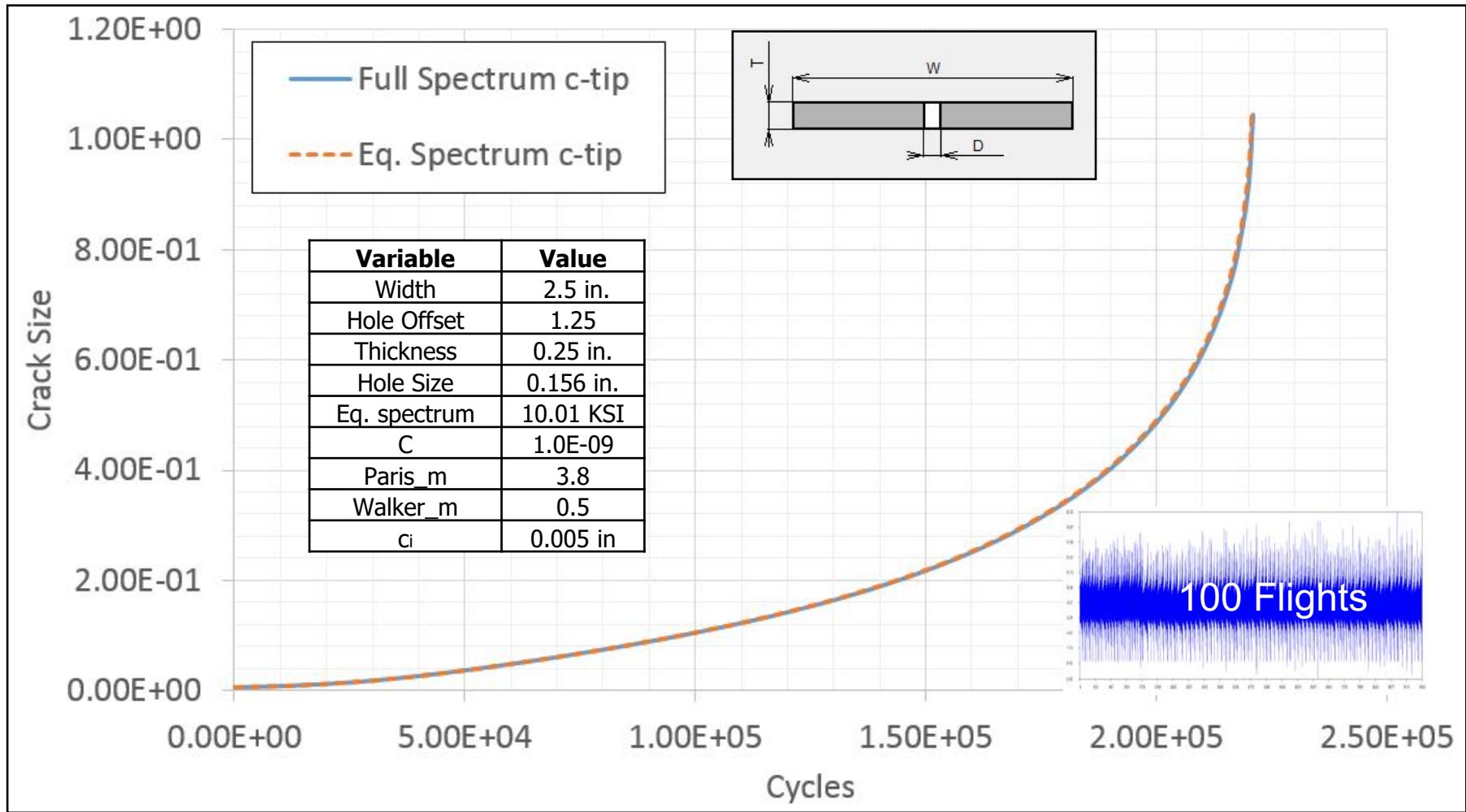
a0	0.05			VA Crack Growth				CA Crack Growth					
	ni	sigmaMin	sigmaMax	pi*delsigma^n	a	delta K	dadN	Delta a	a	delta K	dadN	Delta a	
1	1	40	55592.41		0.05	17.312	5.0781E-05	5.07811E-05		0.05	13.580	2.0187E-05	2.01872E-05
1	2	36	33005.77		0.050050781	15.100	3.0207E-05	3.02074E-05	0.050020187	13.583	2.0203E-05	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	
20			441996.8		0.050406667029		Sum	4.0666703E-04	0.05040670623		Sum	4.0670623E-04	
			30.594023726										



Eq. Stress Examples

Through Crack in a Hole

All solutions using Afgrow

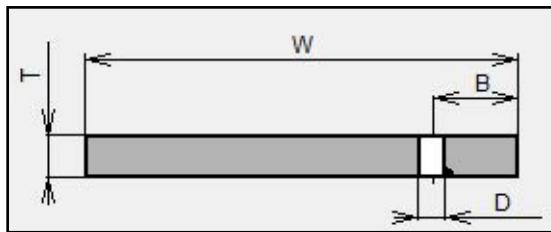




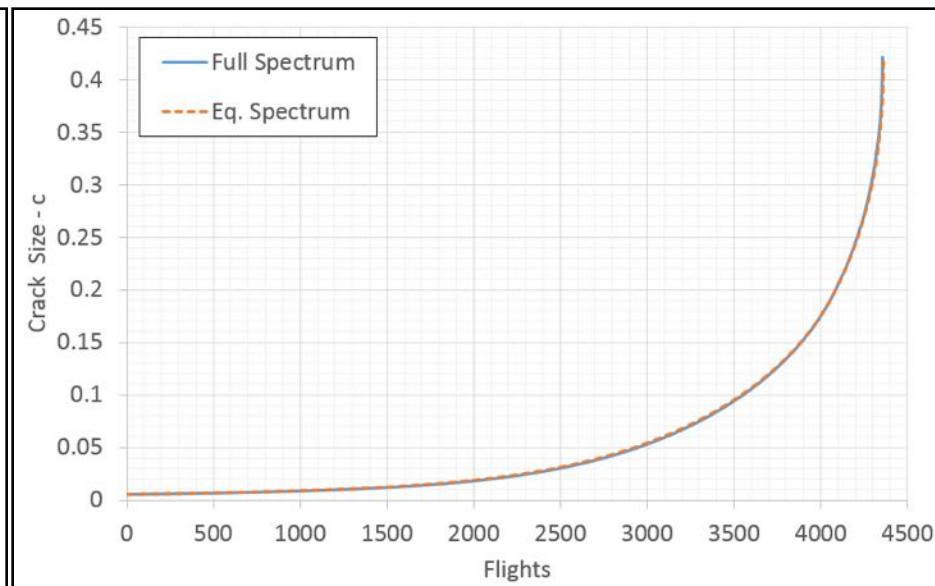
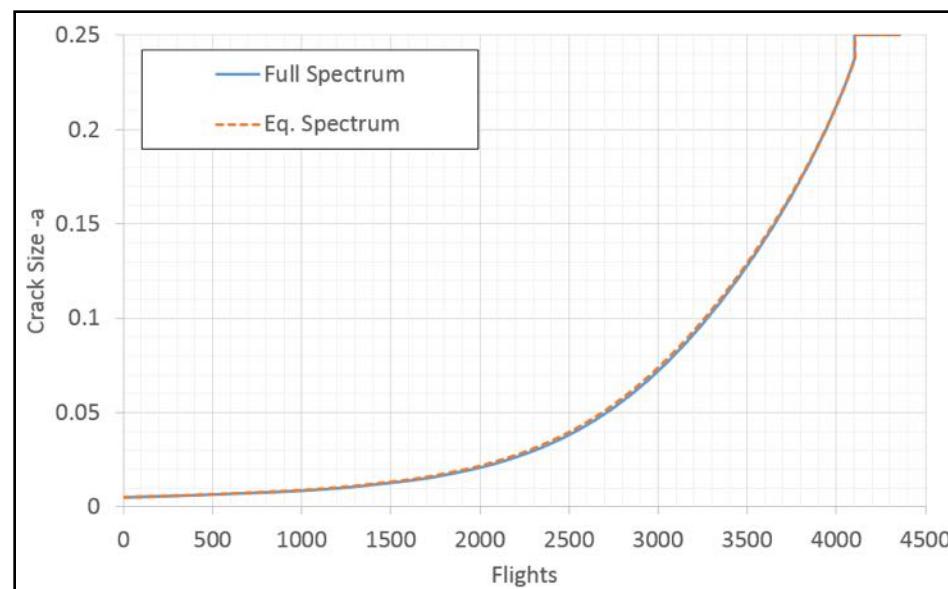
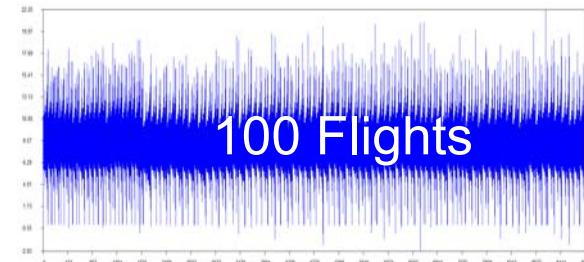
Eq. Stress Examples

Corner Crack in a Hole

All solutions using Afgrow

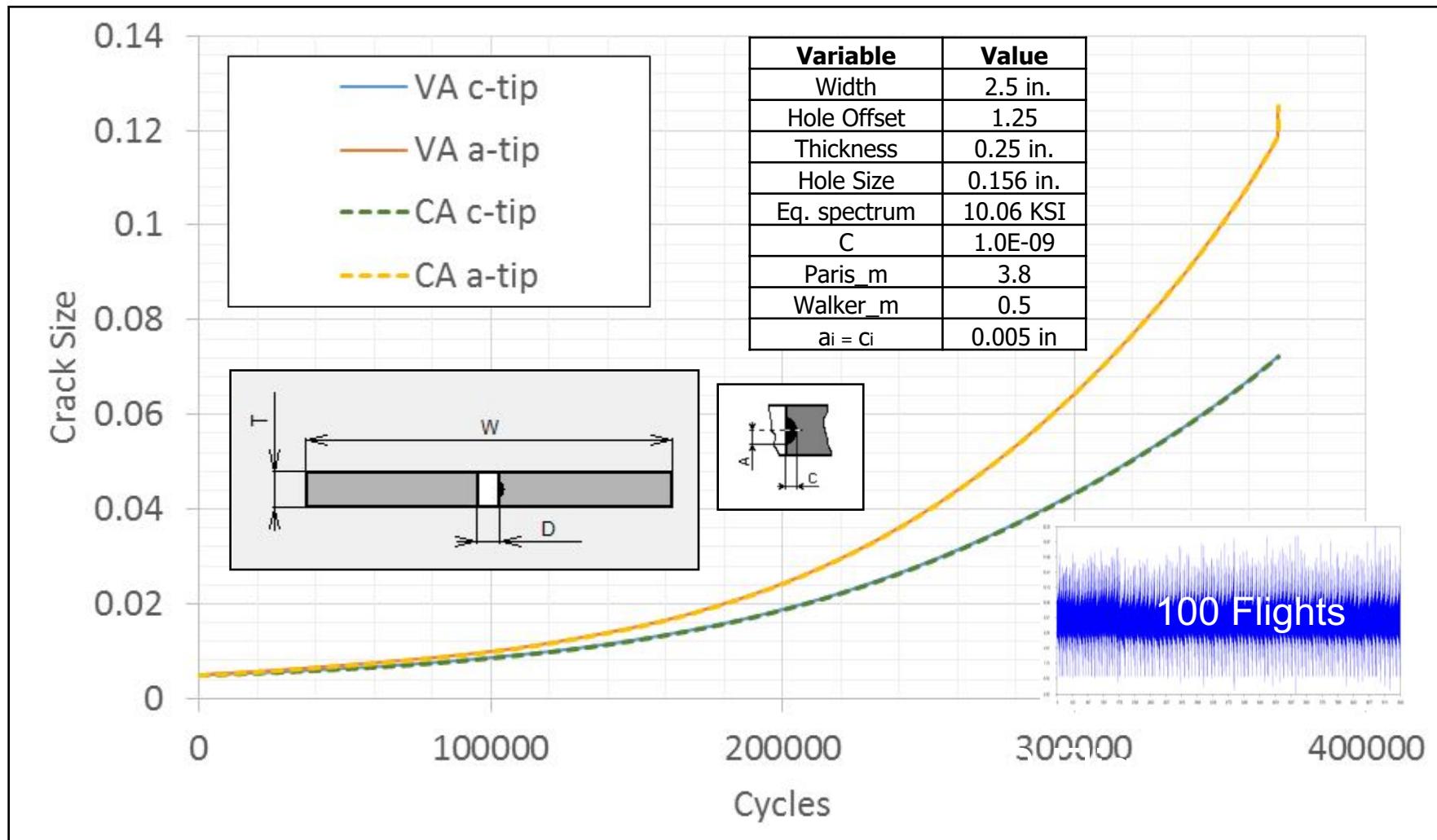


Variable	Value
Width	4 in.
Hole Offset	0.5
Thickness	0.25 in.
Hole Size	0.156 in.
Eq. spectrum	10.01 KSI
C	1.0E-09
Paris_m	3.8
Walker_m	0.5
$a_i = C_i$	0.005 in



Eq. Stress Examples

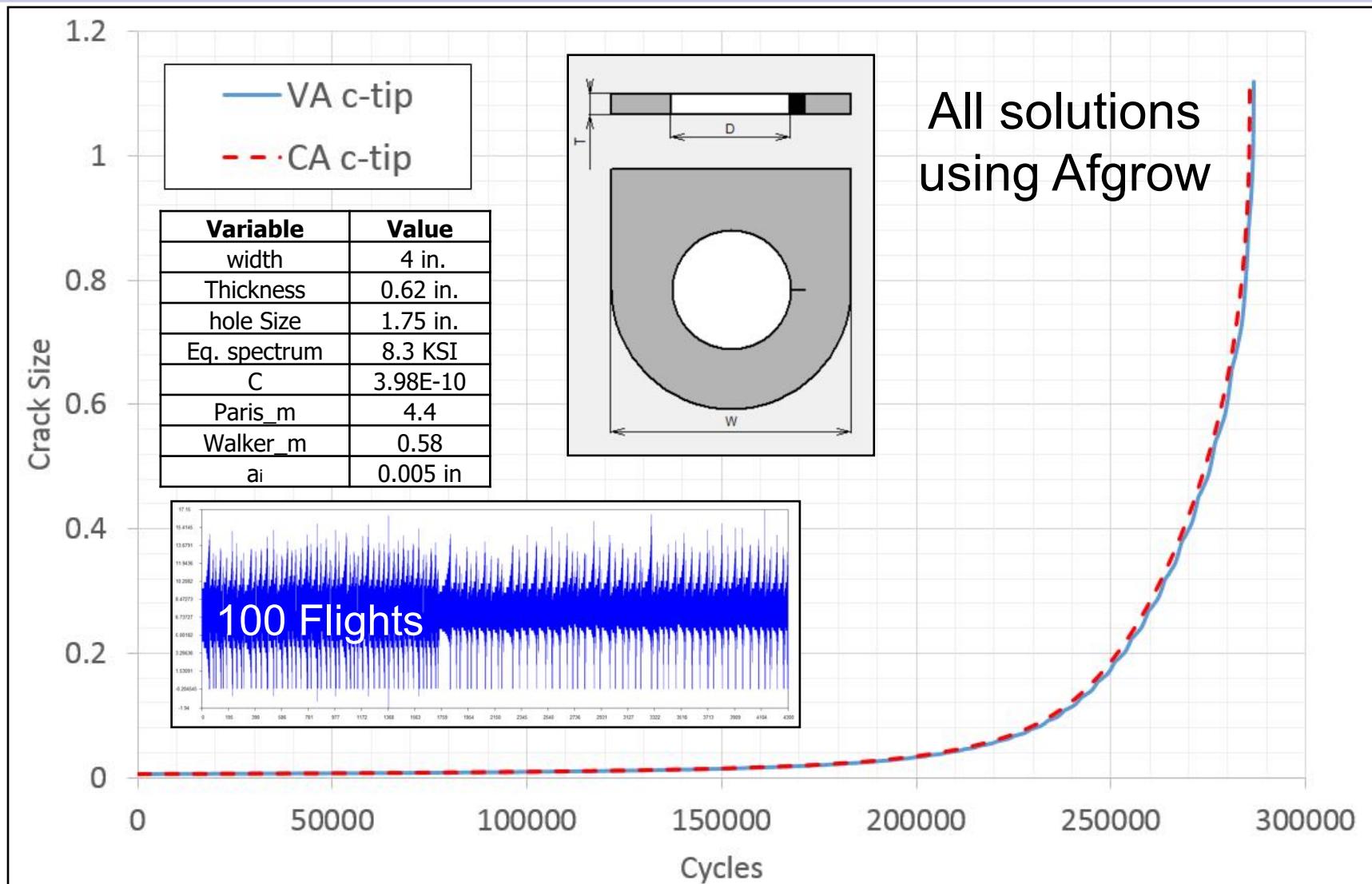
Surface Crack in a Hole



All solutions using Afgrow

Eq. Stress Examples

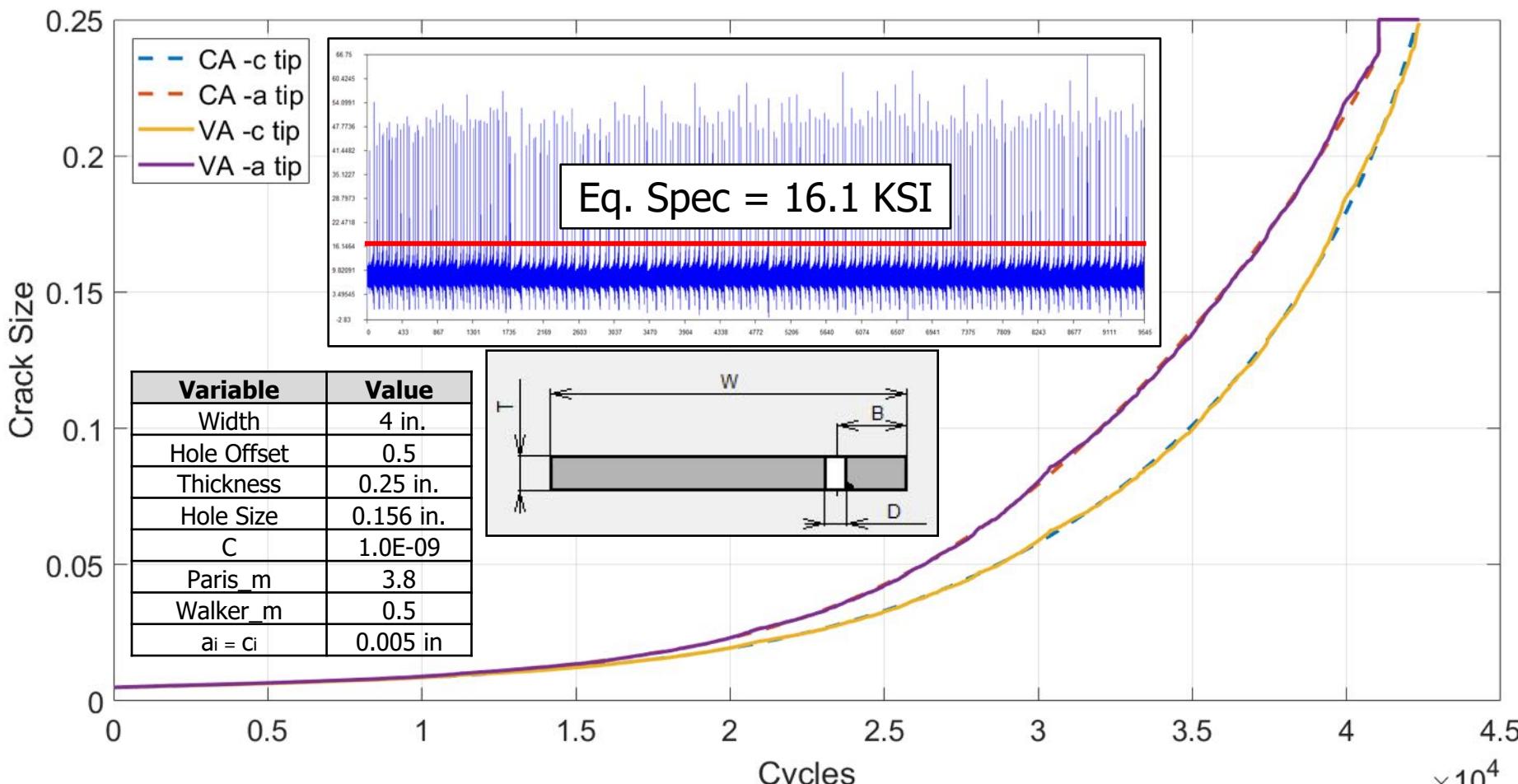
Through Crack in a Lug



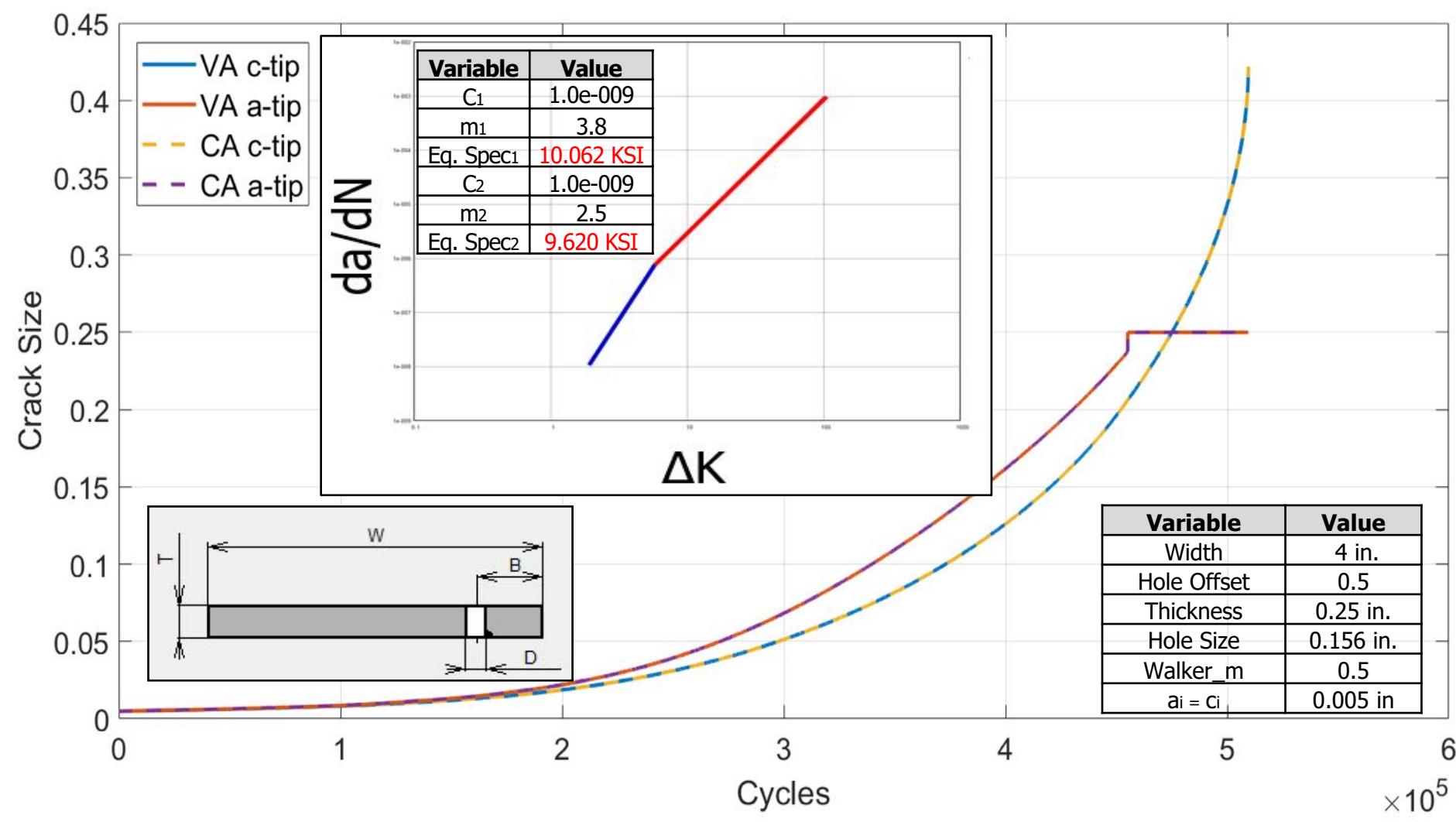
All solutions
using Afgrow

Over Load Example

All solutions using Afgrow



Bilinear Paris Example

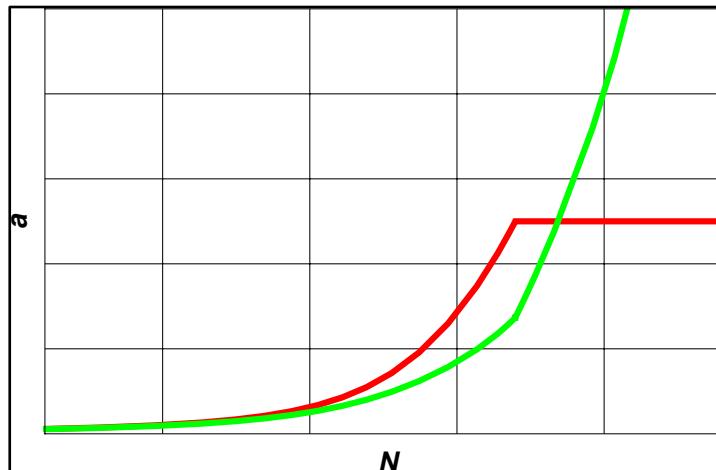


All solutions using Afgrow



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., 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.



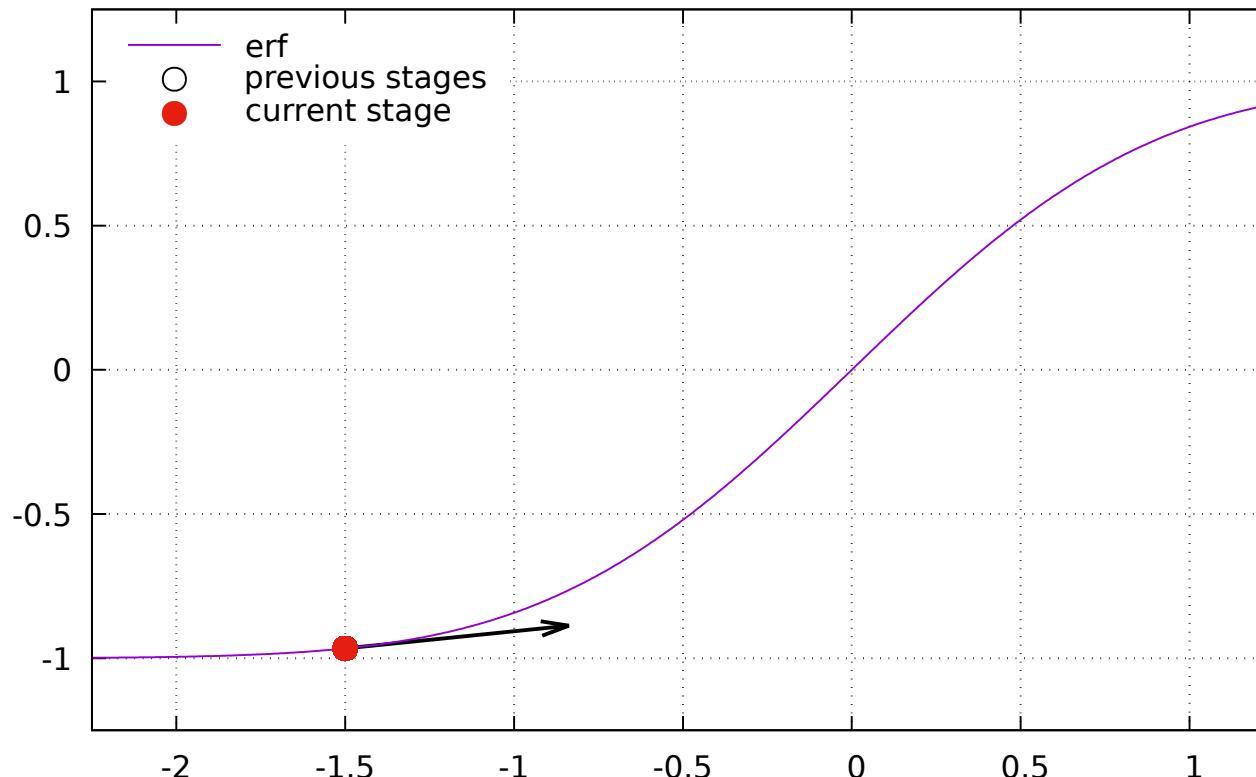


RKSUITE_90

- 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



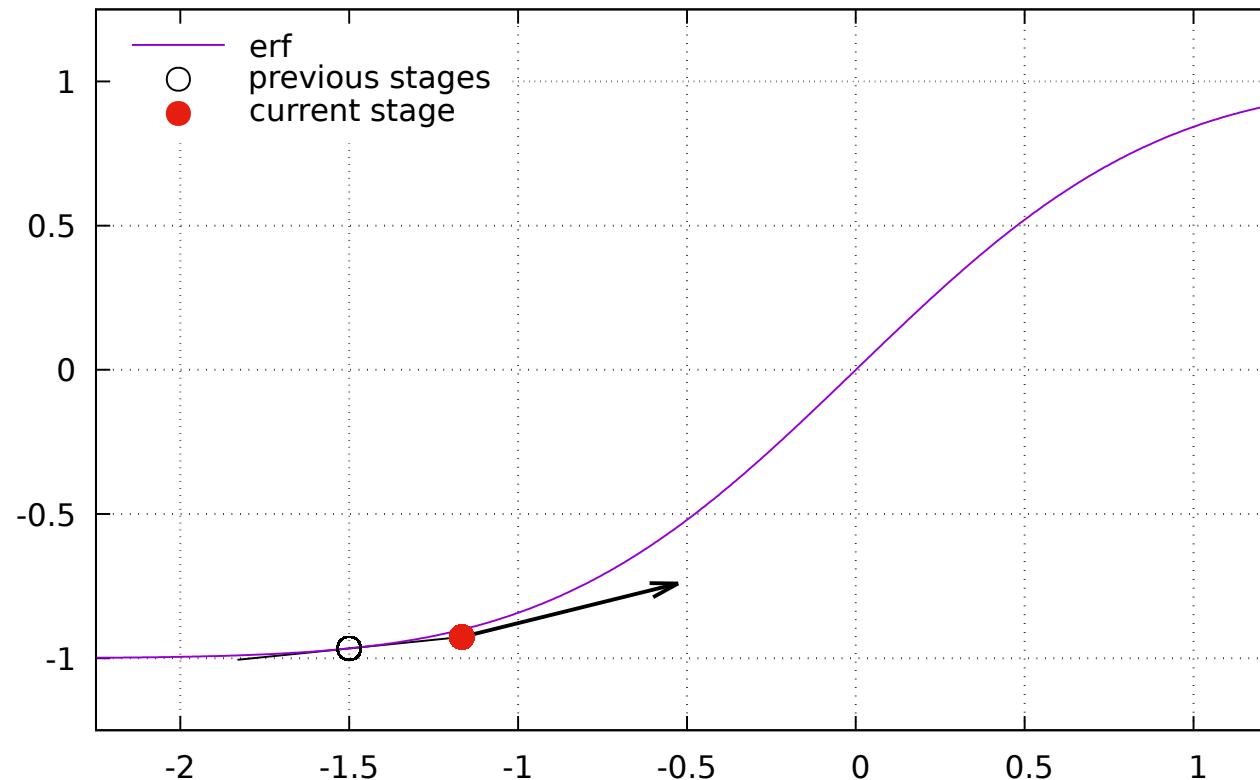
Example Bogacki-Shampine 4-5 pair



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



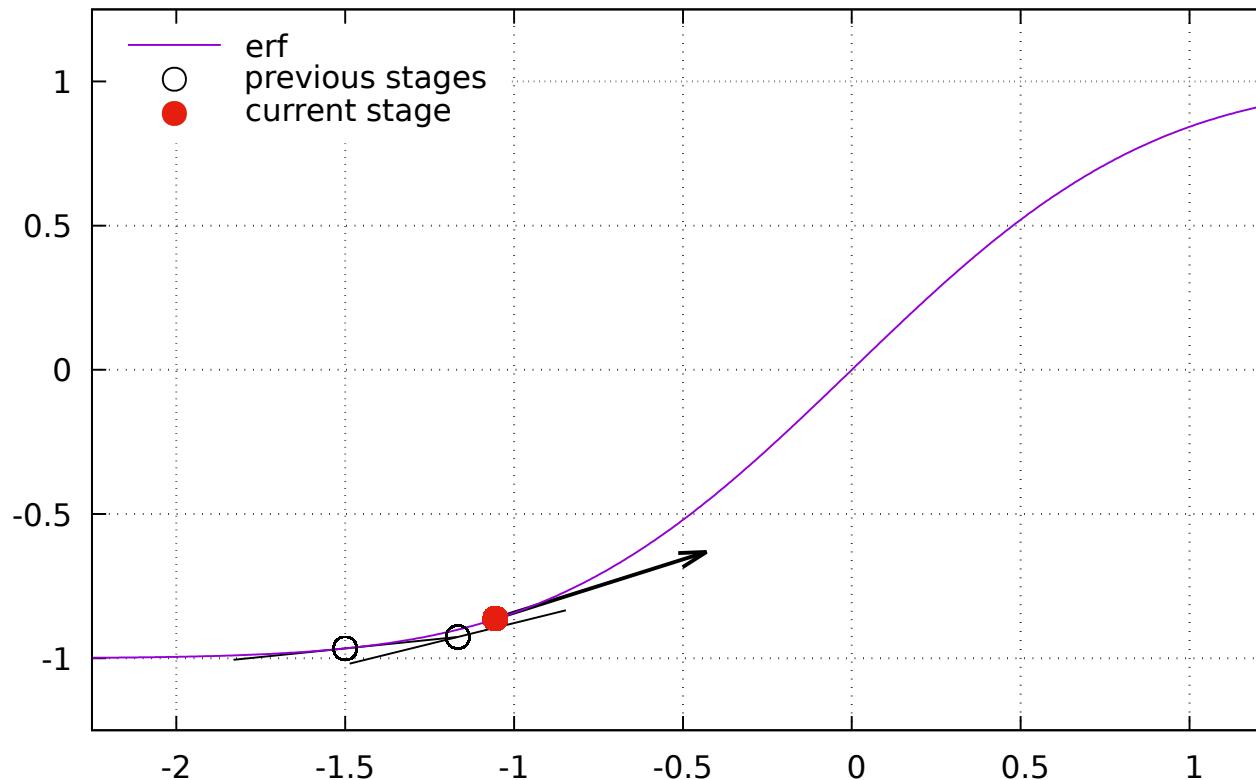
Example Bogacki-Shampine 4-5 pair



■ ... stage 2 ...



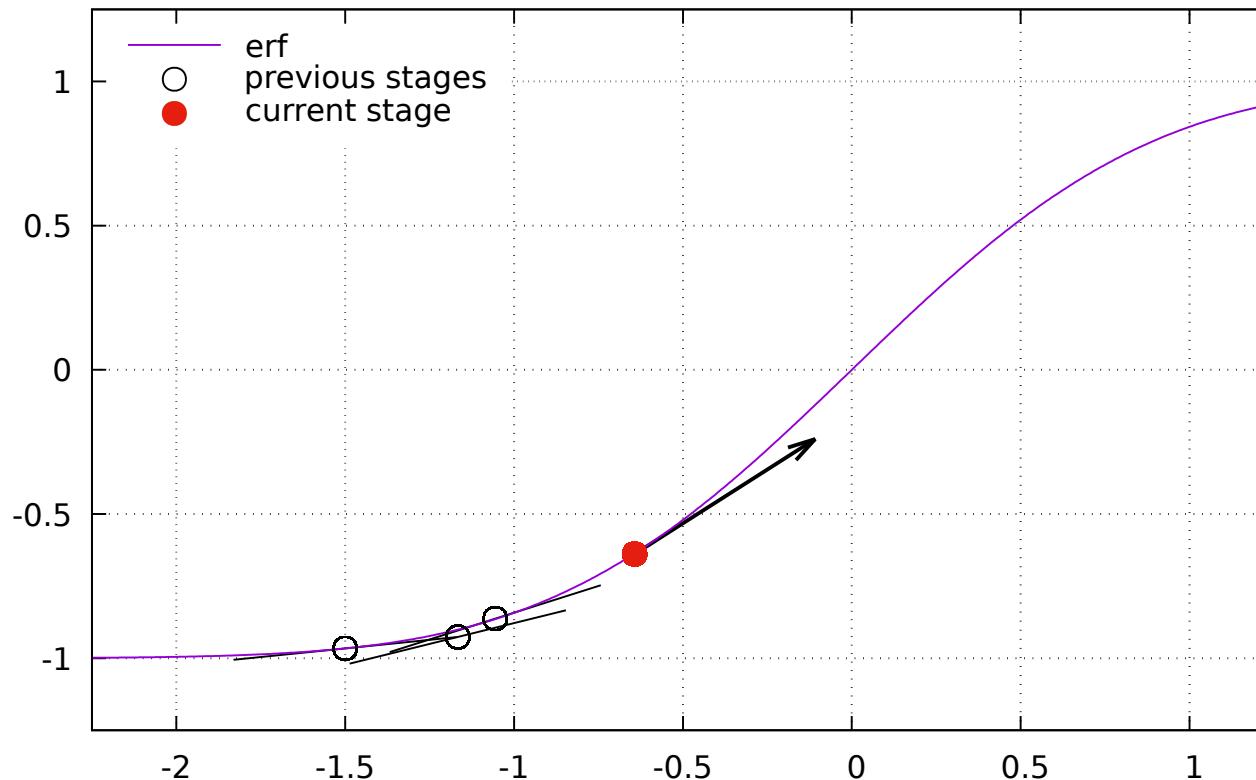
Example Bogacki-Shampine 4-5 pair



■ ... stage 3 ...



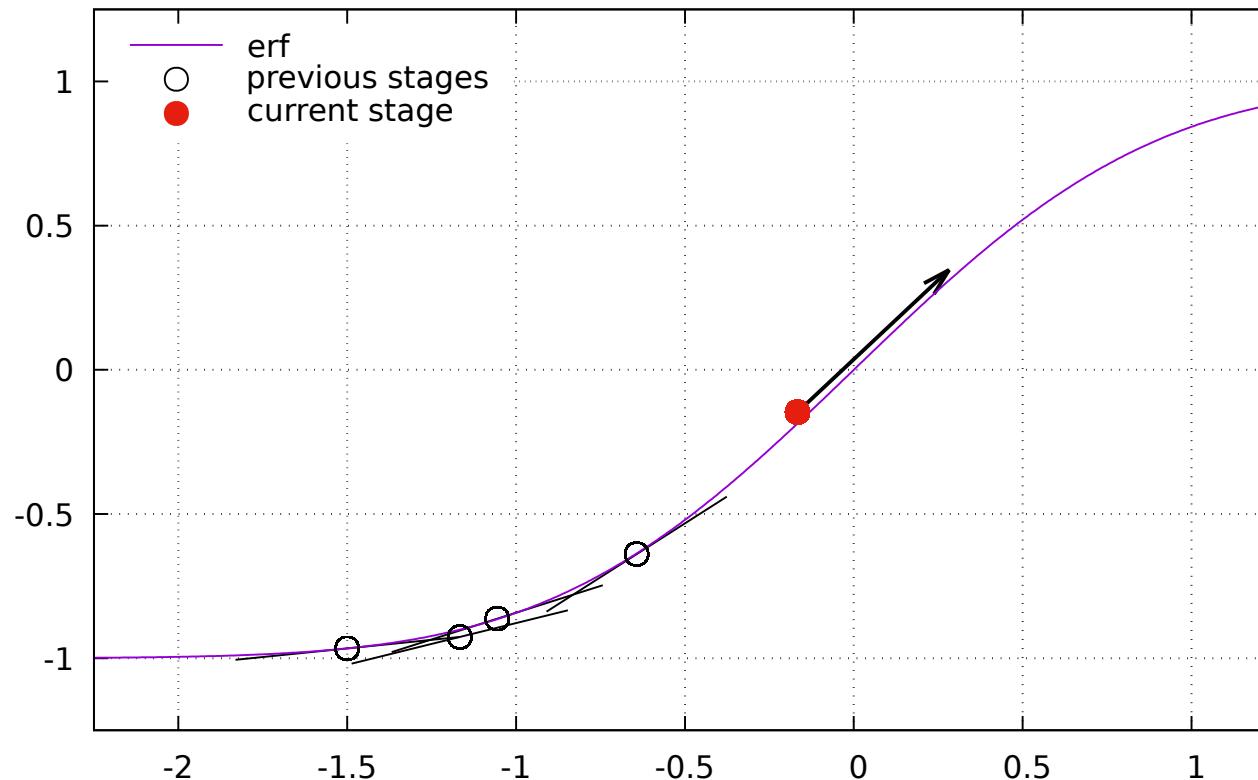
Example Bogacki-Shampine 4-5 pair



■ ... stage 4 ...



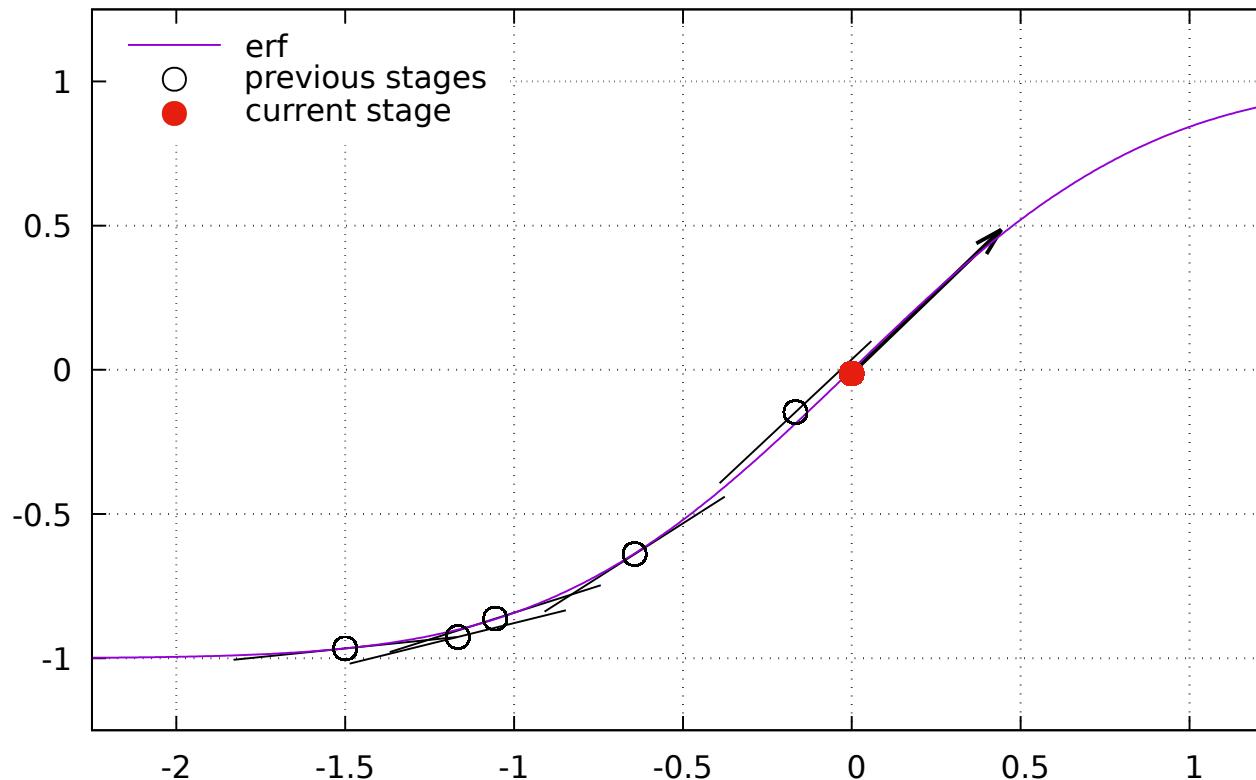
Example Bogacki-Shampine 4-5 pair



■ ... stage 5 ...



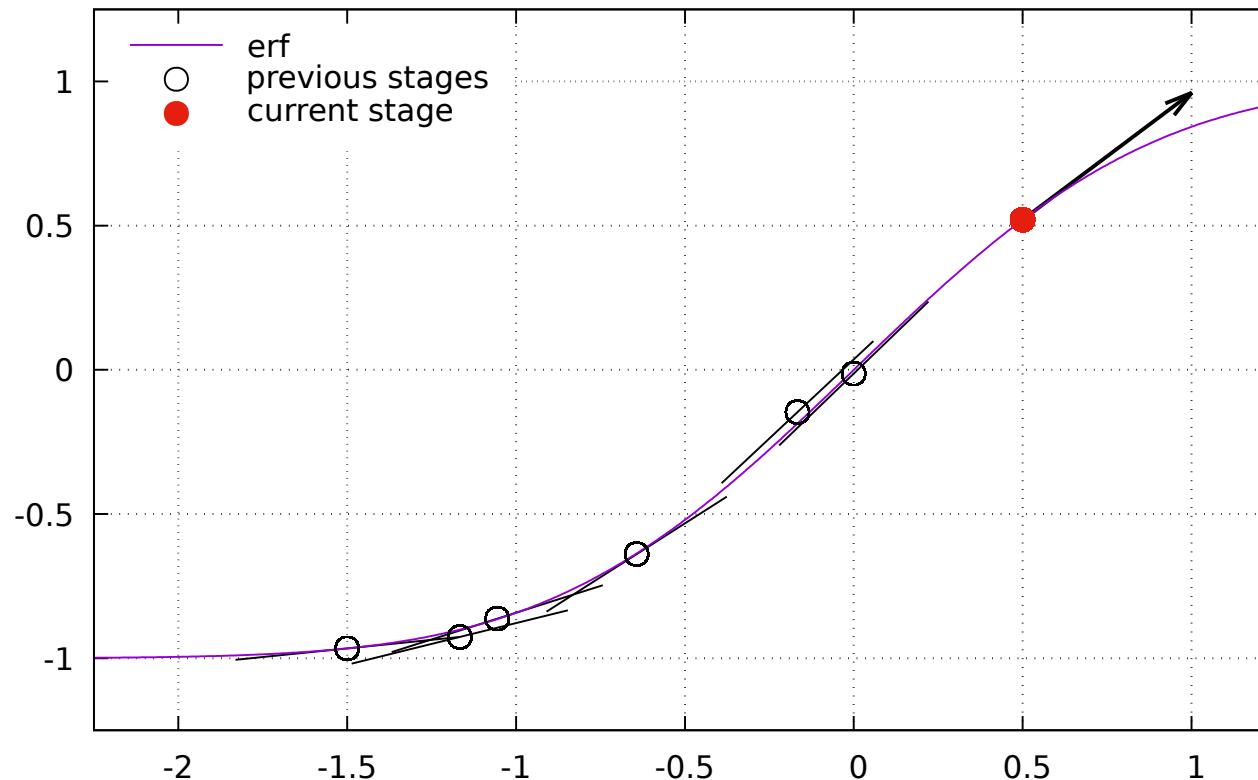
Example Bogacki-Shampine 4-5 pair



■ ... stage 6 ...



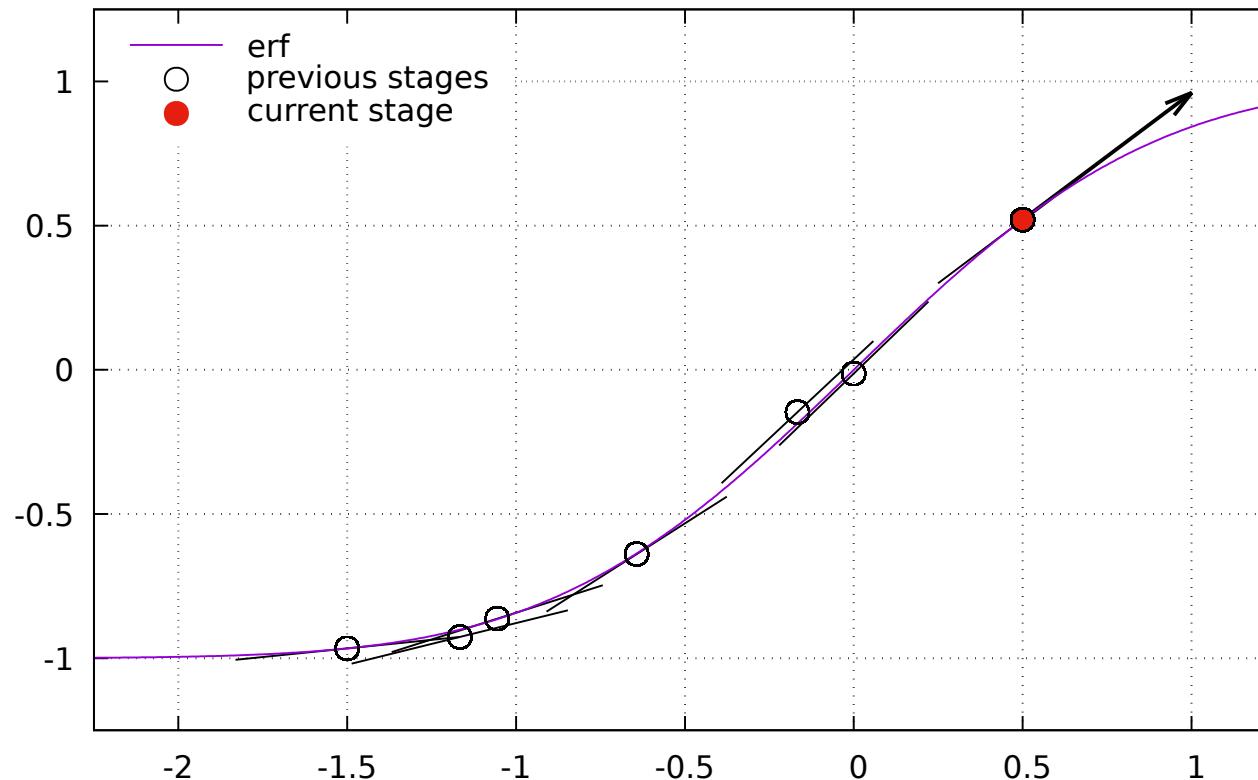
Example Bogacki-Shampine 4-5 pair



■ ... stage 7 ...

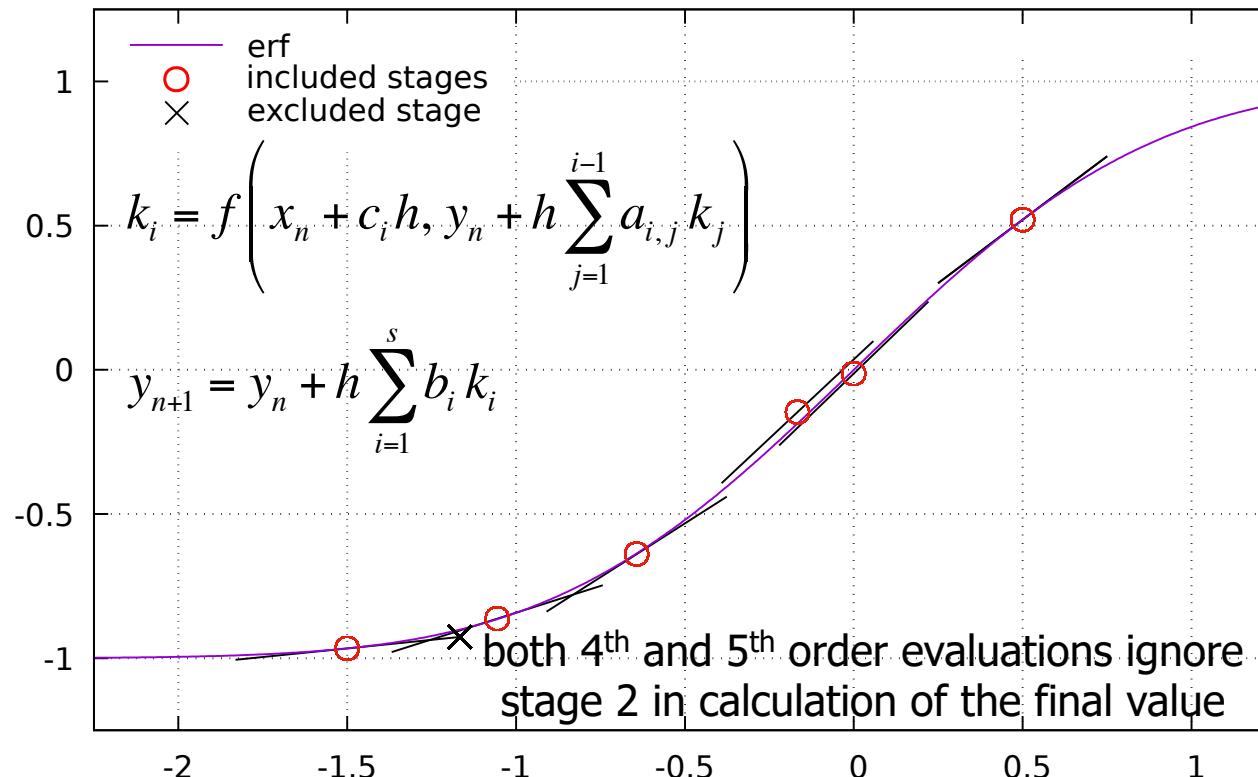


Example Bogacki-Shampine 4-5 pair



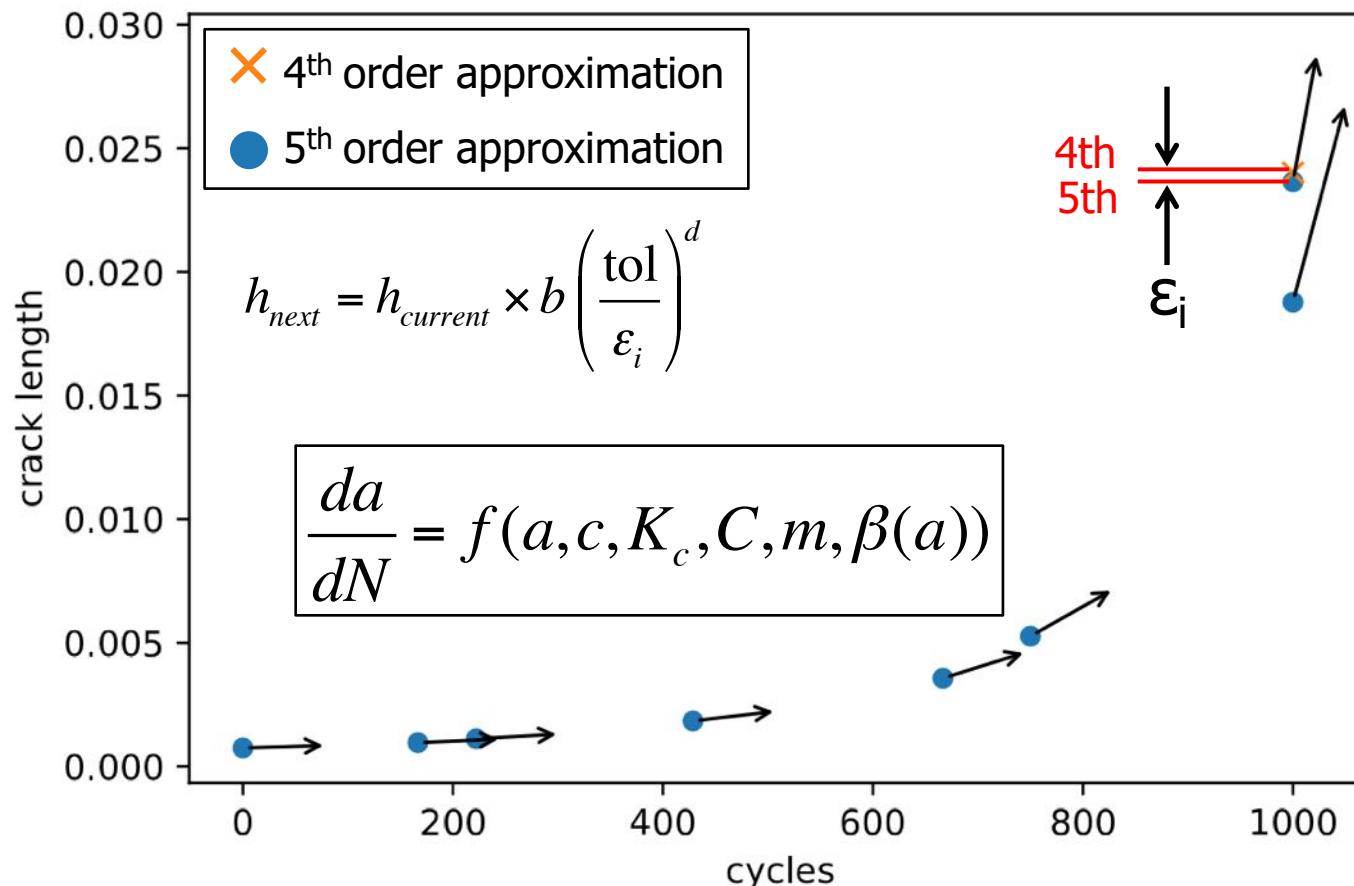
■ ... stage 8 ...

Example Bogacki-Shampine 4-5 pair



- 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}

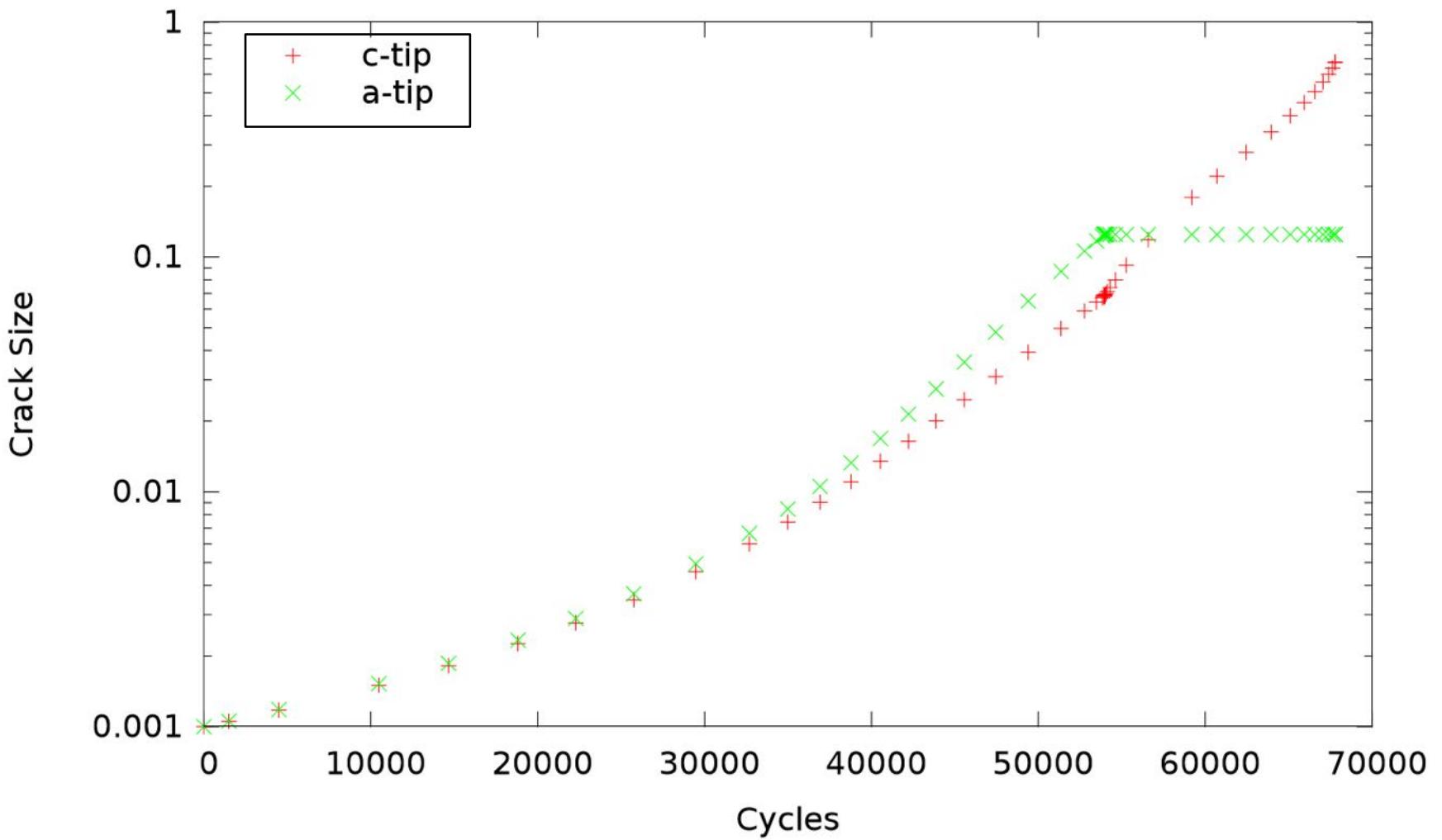
Adaptive Step Size Control



- ◻ 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

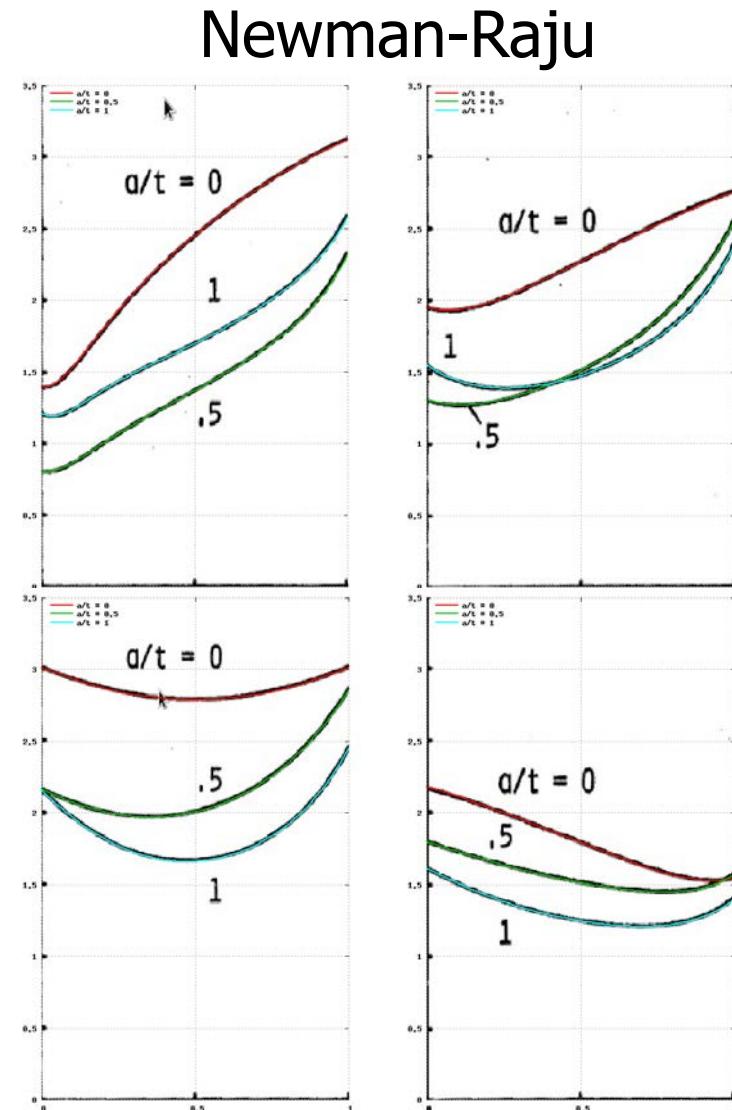
Variable step sizes - corner crack integration



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





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

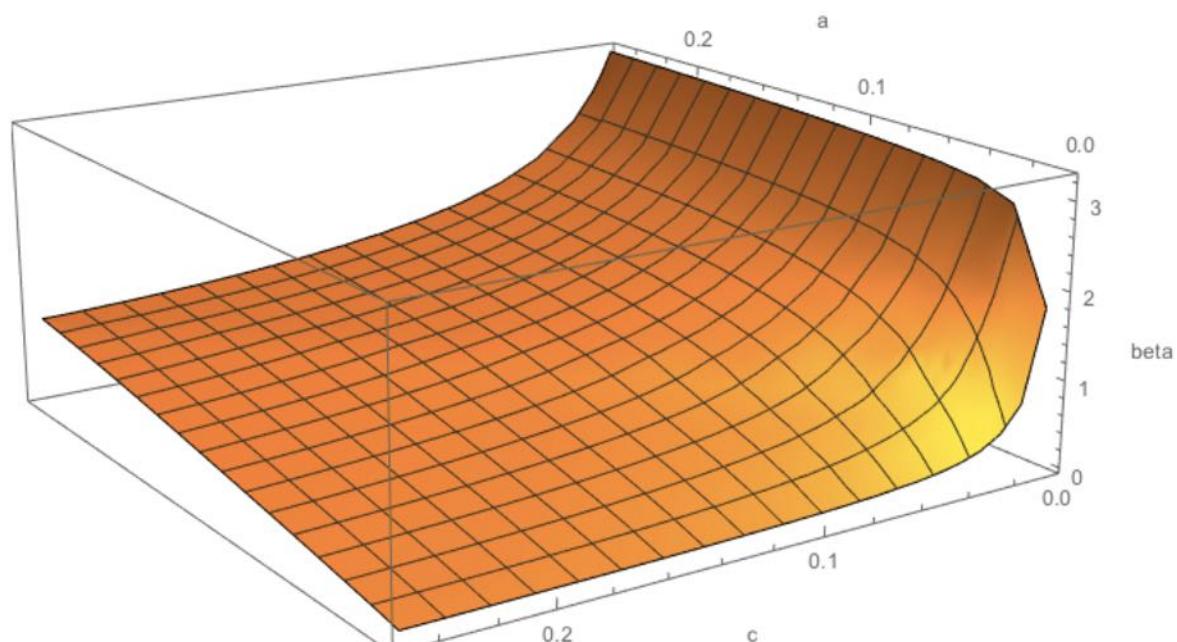
c1 β_{11} β_{12} ... β_{1N}

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

beta c		Crack a Direction																																
Model	1030	0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04	0.045	0.05	0.055	0.06	0.065	0.07	0.075	0.08	0.085	0.09	0.095	0.1	0.105	0.11	0.115	0.12	0.125	0.13	0.135	0.14	0.145	0.15			
Crack dir c	0.005	1.80545	2.41897	2.67039	2.79924	2.87637	2.9278	2.96482	2.99034	3.07549	3.03396	3.04957	3.06304	3.07487	3.08541	3.09491	3.10366	3.1115	3.11885	3.12568	3.13208	3.1381	3.14378	3.14916	3.15427	3.15915	3.16382	3.16828	3.17257	3.1767	3.18068			
	0.01	0.98293	1.66217	2.01601	2.2304	2.36918	2.46454	2.53336	2.59852	2.6254	2.65762	2.68403	2.70613	2.72489	2.74109	2.75256	2.76778	2.77895	2.78901	2.80647	2.81442	2.82192	2.82775	2.83387	2.8396	2.84498	2.85005	2.85484	2.85938	2.8637				
	0.05	6.98987	1.1061	15.45451	1.78375	19524	2.0747	2.16656	2.2376	2.293	2.33787	2.37467	2.40535	2.4329	2.45351	2.47275	2.49858	2.50455	2.51767	2.52952	2.54022	2.54993	2.55879	2.5744	2.58133	2.59877	2.59377	2.59538	2.60464	2.60959				
	0.02	0.53631	0.85948	1.13232	1.4508	16.2579	17.5959	18645	19.4735	2.01407	2.06861	2.1383	2.1518	2.18045	2.2172	2.2357	2.25665	2.2751	2.2948	2.31924	2.3311	2.34187	2.3517	2.3607	2.36898	2.37662	2.3837	2.39029	2.39643	2.40217				
	0.025	0.43077	0.70264	0.91706	1.12843	13.752	15.033	16.887	17.076	17.813	18.4245	18.8378	19.3735	19.7466	2.00687	2.03492	2.0595	2.0812	2.0948	2.11769	2.13315	2.1471	2.15975	2.1727	2.1818	2.19147	2.20037	2.2085	2.21621	2.2233	2.2299			
	0.03	0.38933	0.52327	0.77617	0.94245	1.11493	1.30975	14.992	15.1058	15.8733	16.22517	17.0702	17.5555	17.9625	18.3202	18.6336	18.9088	19.1546	19.3277	19.5681	19.7438	19.902	20.0467	2.0178	2.0298	2.04082	2.05036	2.06899	2.07704	2.08454				
	0.035	0.30628	0.51012	0.67335	0.81046	0.9504	1.0974	1.0756	1.2565	1.3475	1.42505	1.49181	1.54949	1.59598	16.4333	16.9713	17.1561	17.4965	17.7242	17.9763	18.197	18.7374	18.5494	18.8095	18.9894	19.1213	19.3225	19.3301	19.4247	19.517	19.6008			
	0.04	0.26748	0.44764	0.59402	0.7194	0.83517	0.9541	10.7534	12.169	12.8988	13.5881	14.4451	14.6605	15.1148	15.517	15.8744	16.1934	16.4793	16.7363	16.9683	17.1785	17.3696	17.5438	17.7045	17.8491	18.0249	18.1929	18.3223	18.4249	18.5214				
	0.045	0.23769	0.39671	0.53076	0.64504	0.748	0.8476	0.9462	0.94782	1.0752	12.3293	12.9971	13.0503	13.3918	14.3854	14.7557	15.0882	15.3078	15.6587	15.9043	16.2177	16.3316	16.518	16.6991	16.8465	16.9897	17.126	17.2505	17.3661	17.4743				
	0.05	0.21422	0.35646	0.47916	0.58643	0.67862	0.76688	0.85411	0.94397	0.93895	1.04074	1.0884	1.1984	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632	
	0.055	0.19534	0.32353	0.43637	0.53442	0.62155	0.70208	0.7796	0.8576	0.93821	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632	
	0.06	0.17985	0.28191	0.4004	0.492	0.57356	0.6482	0.7824	0.7885	0.8594	0.93207	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632
	0.065	0.16851	0.27322	0.36881	0.45686	0.53248	0.61651	0.68449	0.7322	0.7954	0.85977	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632
	0.07	0.15600	0.2537	0.34353	0.42427	0.49889	0.56307	0.58479	0.7424	0.80309	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632	
	0.075	0.14677	0.23634	0.32075	0.39686	0.46575	0.52873	0.58733	0.643	0.67901	0.75041	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632
	0.08	0.13871	0.22442	0.30096	0.37275	0.43825	0.49893	0.5542	0.6069	0.6572	0.7075	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632
	0.085	0.13675	0.20975	0.28336	0.35141	0.41381	0.47812	0.5246	0.57494	0.62316	0.67007	1.02275	1.0848	1.1729	12.9508	12.9757	13.3953	13.7735	14.1613	14.425	14.7663	15.1988	15.4988	15.7495	16.0456	16.3243	16.5872	16.9564	16.1111	16.2543	16.3874	16.5114	16.6271	16.7632
	0.09	0.12548	0.19869	0.26768	0.3324	0.39194	0.44694	0.49815	0.53649	0.59233	0.63699	0.68066	0.72394	0.7672	0.81074	0.85474	0.89933	0.94455	0.98903	1.01739	1.0405	1.0672	1.08997	1.1144	1.13172	1.15089	1.16904	1.18622	1.20252	1.218	1.2327			
	0.095	0.11977	0.18872	0.25407	0.31537	0.37227	0.42505	0.47423	0.52067	0.56484	0.60743	0.64937	0.68979	0.73059	0.7713	0.81225	0.85365	0.89527	0.93743	0.98	0.10514	1.0219	1.05336	1.07333	1.0936	1.11292	1.1319	1.14854	1.16503	1.1807	1.19566			
	0.1	0.11505	0.1792	0.24171	0.30005	0.3545	0.40521	0.45625	0.49375	0.53939	0.58079	0.62054	0.65953	0.69011	0.73652	0.77498	0.81036	0.85494	0.90213	0.95271	0.97973	0.99428	1.01367	1.0303	1.04622	1.0622	1.07977	1.0962	1.10367	1.1203	1.14615			
	0.105	0.11063	0.17203	0.2306	0.2862	0.33836	0.38716	0.4239	0.47609	0.5178	0.5566	0.59481	0.63271	0.6689	0.7055	0.7413	0.7834	0.8184	0.85161	0.8884	0.92554	0.96847	0.99597	1.0137	1.03391	1.0543	1.0719	1.08905	1.11045	1.12931				
	0.11	0.10662	0.16492	0.22055	0.27362	0.32366	0.37065	0.41856	0.46561	0.50349	0.54349	0.57017	0.60733	0.64266	0.67757	0.71227	0.74688	0.7814	0.8167	0.85094	0.88759	0.92067	0.95553	0.97632	0.99636	1.01543	1.03423	1.05462	1.07426	1.09162	1.10951			
	0.115	0.10284	0.15849	0.21943	0.26216	0.3021	0.35551	0.39284	0.4377	0.47723	0.5148	0.54984	0.58643	0.61567	0.65223	0.68458	0.71855	0.75154	0.78452	0.81751	0.85051	0.88759	0.92067	0.95553	0.97632	0.99636	1.01543	1.03423	1.05462	1.07426	1.09162	1.10951		
	0.12	0.09864	0.15264	0.20213	0.25321	0.29797	0.34158	0.3823	0.4293	0.46567	0.49545	0.53008	0.56375	0.59668	0.62907	0.66108	0.69324	0.72446	0.75759	0.79747	0.8193	0.85033	0.88616	0.91287	0.94388	0.96232	0.98048	0.99791	1.0164	1.0307	1.04614			
	0.125	0.09859	0.14731	0.19533	0.24206	0.2865	0.32872	0.36877	0.40684	0.4422	0.47908	0.5176	0.54446	0.5764	0.60777	0.63871	0.66939	0.70098	0.73029	0.76129	0.7922	0.82047	0.85044	0.88242	0.91287	0.94352	0.97061	0.99068	1.00671	1.02234				
	0.13	0.09377	0.14141	0.19214	0.24204	0.2865	0.32302	0.36381	0.40163	0.4374	0.4793	0.51694	0.54744	0.5763	0.60809	0.63864	0.66956	0.70098	0.73029	0.76129	0.7922	0.82047	0.85044	0.88242	0.91287	0.94352	0.97061	0.99068	1.00671	1.02234				
	0.135	0.09048	0.13730	0.18813	0.23059	0.27337	0.31412	0.35466	0.39456	0.43075	0.46704	0.5057	0.53504	0.56578	0.6022	0.6323	0.6623	0.6924	0.72279	0.75285	0.78339	0.81371	0.84371	0.87379	0.90379	0.93371	0.96379	0.99379	1.02027	1.04905				
	0.14	0.08874	0.13756	0.18797	0.23071	0.27337	0.3142	0.35466	0.39456	0.43075	0.46704	0.5057	0.53504	0.56578	0.6022	0.6323	0.6623	0.6924	0.72279	0.75285	0.78339	0.81371	0.84371	0.87379	0.90379	0.93371	0.96379	0.99379	1.02027	1.04905				
	0.145	0.08649	0.13291	0.17073	0.21045	0.25491	0.29859	0.33541	0.37082	0.40512	0.4374	0.4793	0.51694	0.54744	0.5763	0.60809	0.63864	0.66956	0.70098	0.73029	0.76129	0.7922	0.82047	0.85044	0.88242	0.91287	0.94352	0.97061	0.99068	1.00671	1.02234			
	0.15	0.08373	0.1892	0.17165	0.22037	0.26756	0.30957	0.34584	0.38455	0.41505	0.4425	0.4793	0.51694	0.54744	0.5763	0.60809	0.63864	0.66956	0.70098	0.73029	0.761													



Example Problems

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

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

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

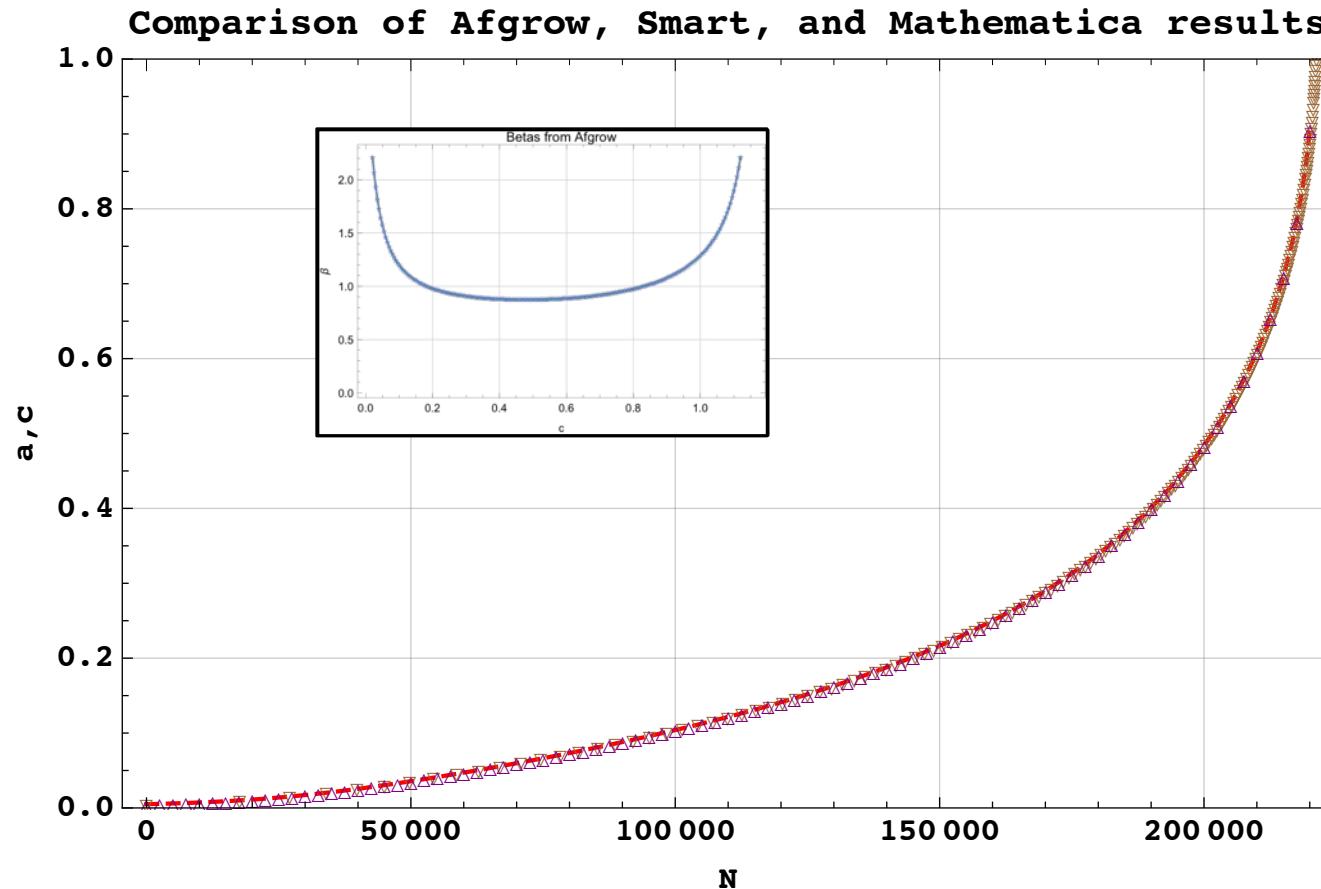
Initial Conditions : $a(0) = a_i, c(0) = c_i$



Through Crack at Hole

(Tension)

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



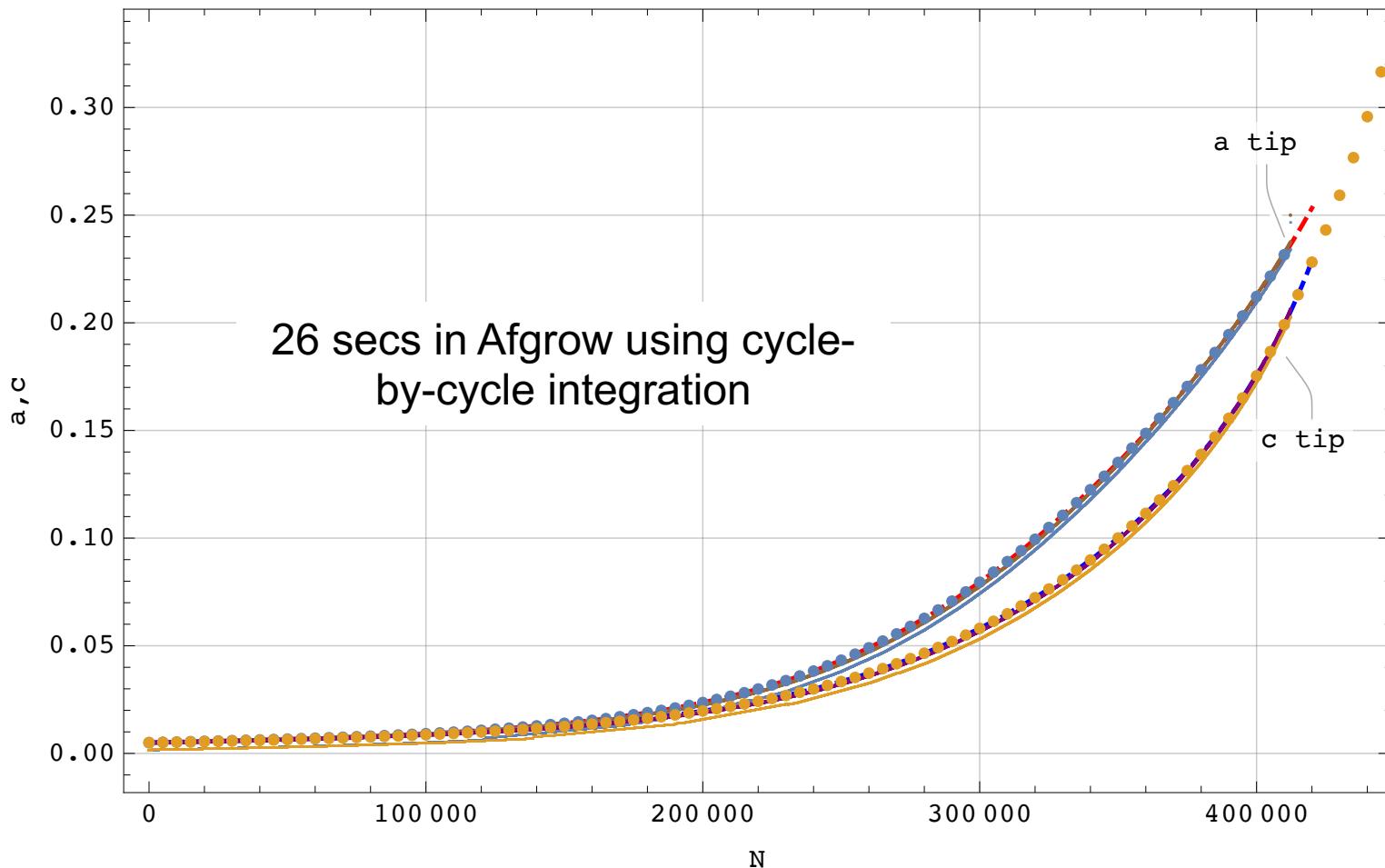


Corner Crack at Hole

(Tension)

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

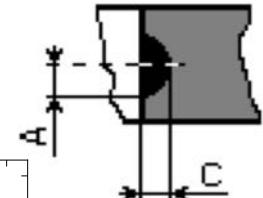
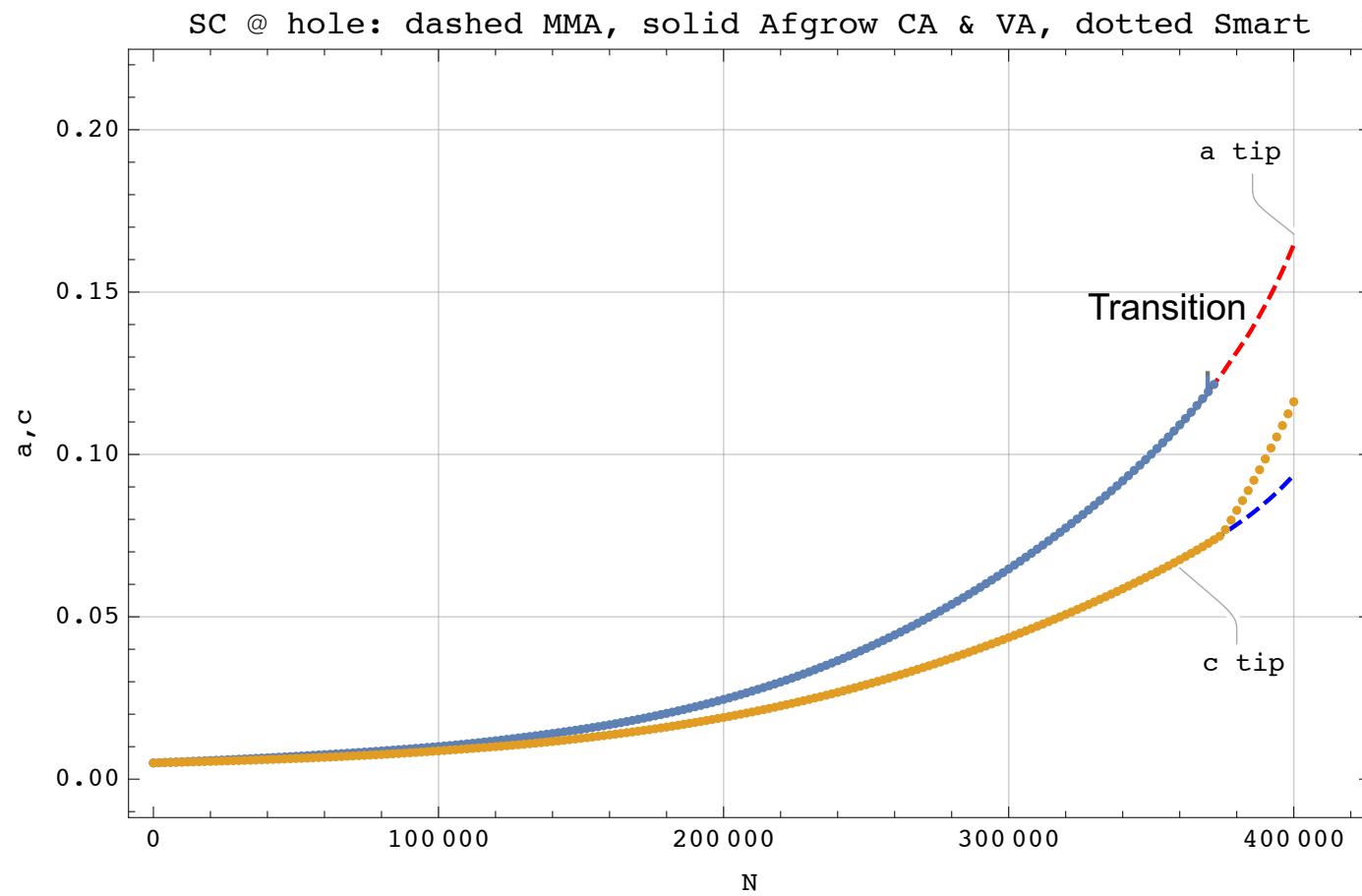
CC @ hole: dashed MMA, solid Afgrow CA & VA, dotted Smart





Surface Crack at Hole (Tension)

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

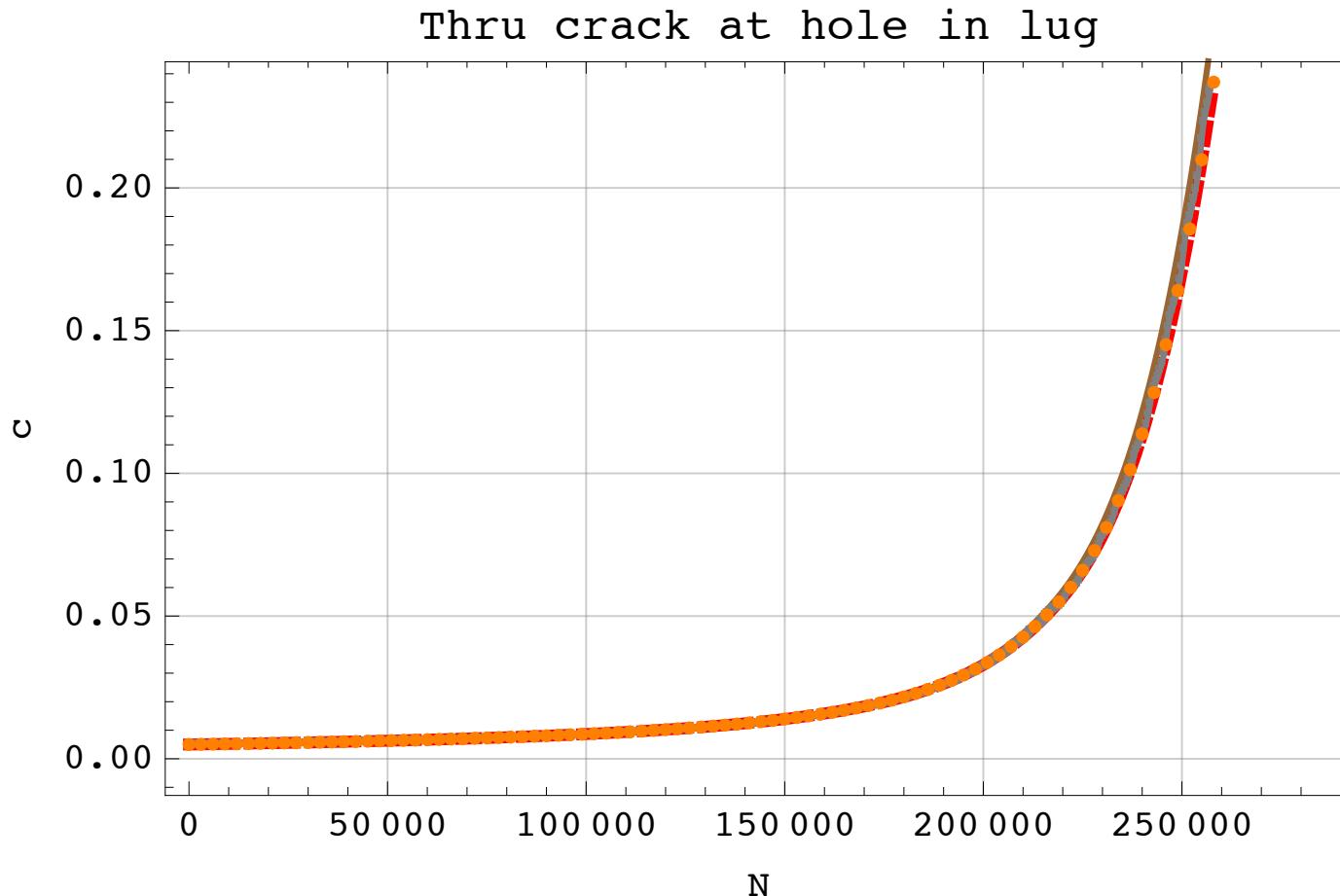




Thru Crack at Lug

(Tension)

- $C_{paris} = 10^{-9}$, $n_{paris} = 3.8$, $\Delta\sigma = 8.3$ ksi



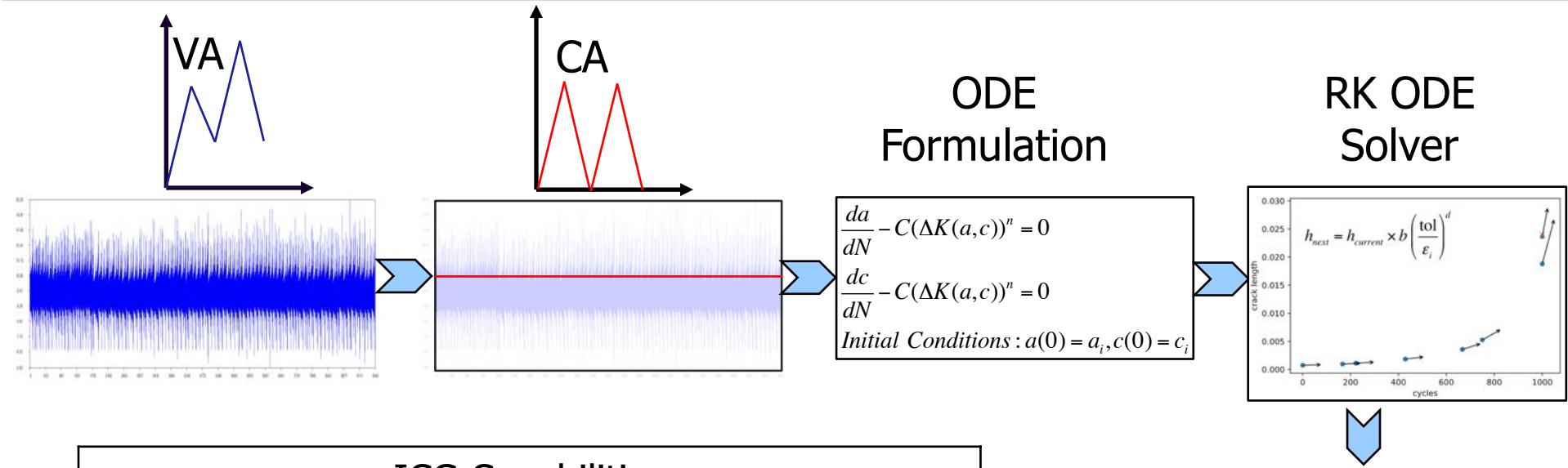


Crack Growth Capabilities

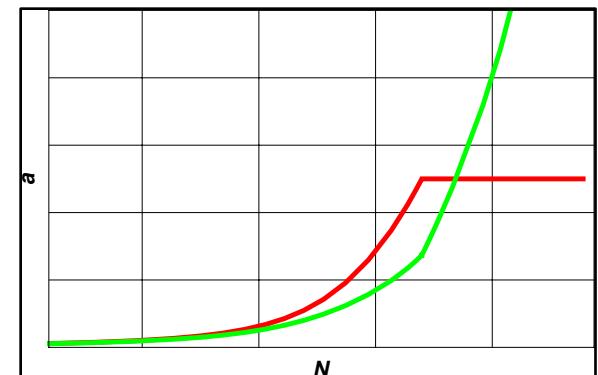
	Afgrow	Nasgro	ICG
Create avsn	Y	Y	coming
MCS	Y	Y	Y
NI	Y	Y	coming
Kriging	Y	Y	coming
RUL	Y	Y	coming
K solutions	Comprehensive	Comprehensive	Newman-Raju Read Beta tables (tension only)
Weight functions	Comprehensive	Comprehensive	N
Net section yield	Y	Y	coming
Retardation	Y	Y	N
Adaptive error control	%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



Internal CG Code



ICG Capabilities	
Method	4-5 th 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





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