K-solutions for Countersunk Holes using an *H-P* *F*inite Element Code and DoD HPC

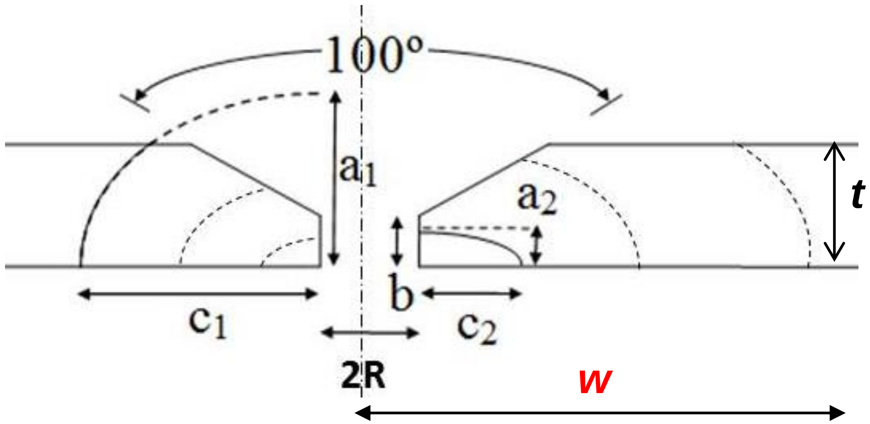
By

Börje Andersson[1], James Greer, Jr.[2]

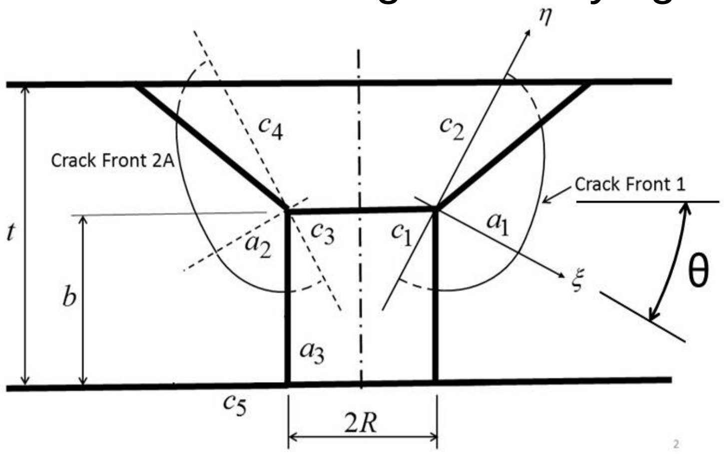
[1]***BARE, Sweden***
[2] ***U.S. Air Force Academy Centre for Aircraft Structural Life Extension, CAStLE, U.S.A.***

Contents

1. The present lecture: Derivation of ~800*M?* Accurate and Reliable *K*-functions for Cracks at a Countersunk Hole



- Cracks initiating at the faying surface



- Cracks initiating at the "knee"

c/a = 0.10,… 25 values…10.0

a/t = 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 0.975, 1.05, 1.15, 1.25, 1.50, 1.75, 2.0, 2.25, 2.5, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0,…, 800.0. 40 values

R/t = 0.10,… 26 values…10.0

b/t = 0.05, 0.25, 0.50, 0.75. 4 values

W/R = 2.4, 2.6, 2.8, 3.2, …100.0 20 values

c/a, a/t, R/t, b/t, W/R roughly as above, and

c_1/c_2=0.5, 0.75, 1.0, 1.333, 2.0

The Challenge:

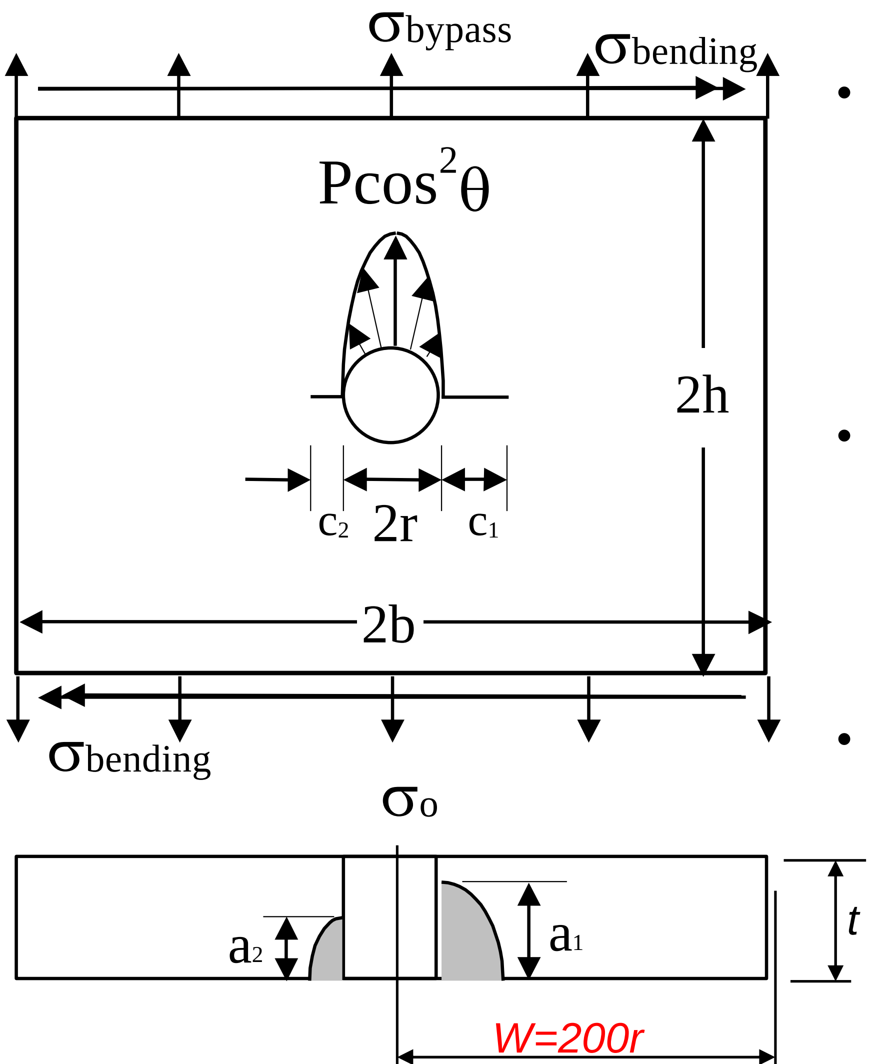
- Derive ~800*M*? *K*-functions reliably and with practically zero error (*i.e.* relative error in *K* <0.1%).

2.Tools used in present study:

- FE-software STRIPE (1984-) based on a *H-P* version on the finite element method (FEM)
- Advanced extraction procedures for calculation of stress intensity functions *K(θ)* along the entire crack front, i.e. including the crack front ends (vertex regions)
- Mathematical splitting method for fast and accurate analysis of domains with multiple (*i.e.* 2-20) cracks
- Mathematical splitting method for fast and accurate variable plate width (*W*) and height (*H*) analysis.
- DoD HPC-resources used to analyze 1/2*M* single crack problems.

3. Previous Work on Deriving large $K(\theta)$ Databases

4.9 *M K*-functions delivered to users via AFGROW (2004) and NASGRO (2012) (S. Fawaz and B.Andersson)

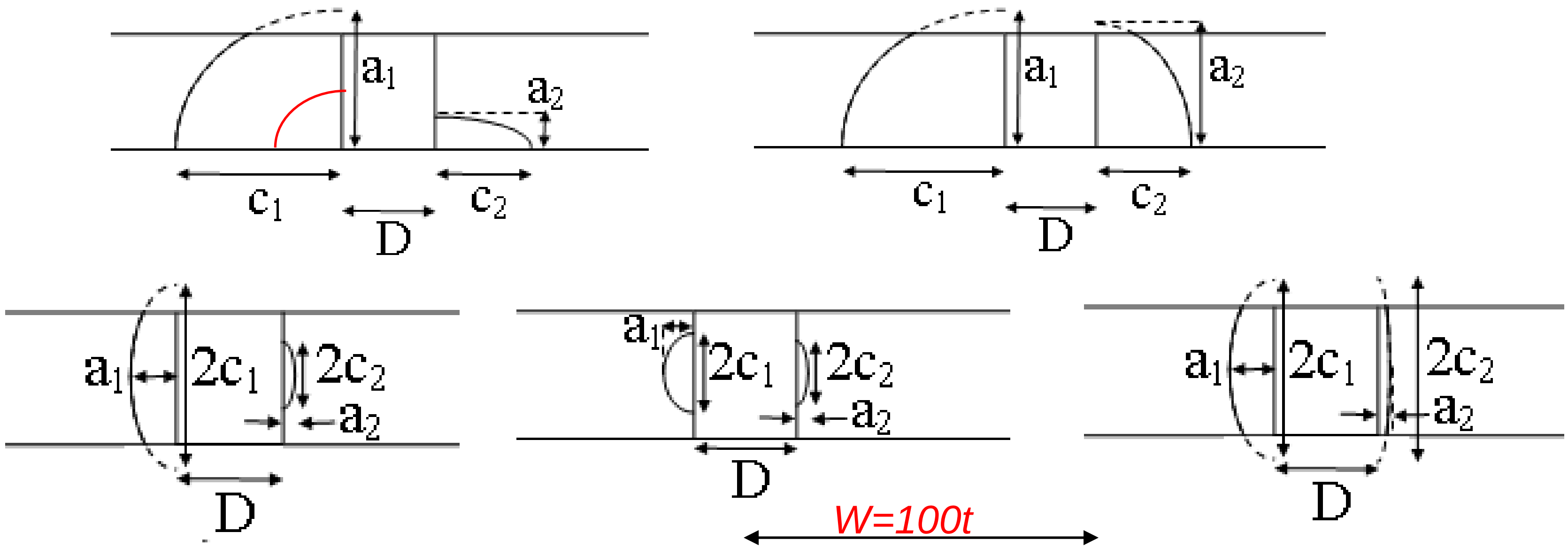


- Geometry
  - Centrally Located Straight Shank Hole
  - $0.1 \le r/t \le 10.0$ — 26 values
    - 0.1, 0.111, 0.125, 0.1428, 0.1667, 0.2, 0.25, 0.333, 0.5, 0.667, 0.75, 0.8, 1.0, 1.25, 1.333, 1.5, 1.667 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0
  - Finite Width/Height Plate
    - r/h = 0.0025
    - r/b = 0.0025
- Crack Shapes — 25 values
  - $0.1 \le a/c \le 10.0$
    - 0.1, 0.111, 0.125, 0.1428, 0.1667, 0.2, 0.25, 0.333, 0.5, 0.667, 0.75, 0.8, 1.0, 1.25, 1.333, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0
  - $0.1 \le a/t \le 0.99$ — 10 values
    - 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99
- Load Conditions — 3 cases
  - Tension
  - Bending*
  - Pin Loading (Bearing)

$$2x3x26x(25x10)^2 = 9.8M$$

Continuation of 2003 years work:

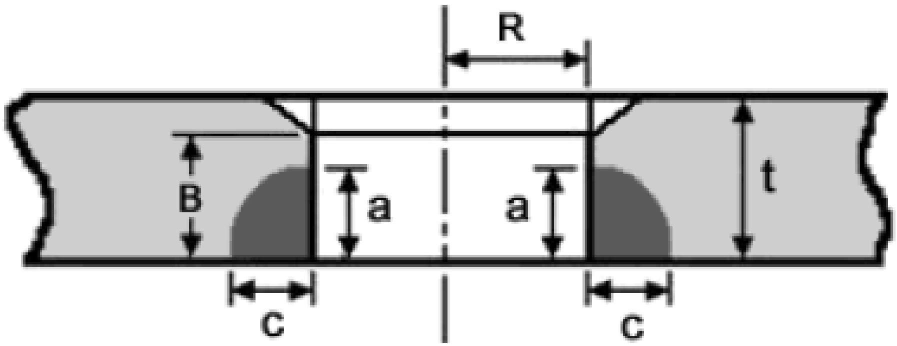
- Derivation of a total of 88 *M K*-functions (2014-2017)



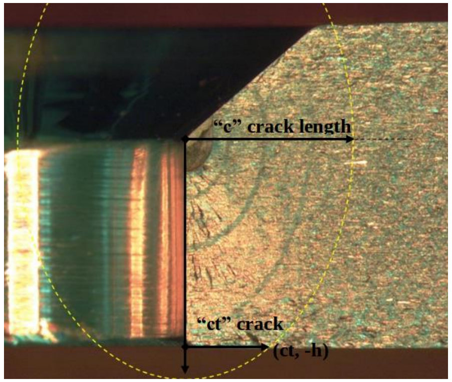
- The *K*-solutions of 2014-2017 verify the 2004 years solutions (and correct errors in bending solutions for *R/t=7,8,9*).

Previous *K*-work on countersunk hole geometries:

- *3k* crack scenario delivered to *AFGROW by Dr Reinier de Rijck*, Thesis NLR, Amsterdam 2014 (work contains a large experimental program)



- *3k* crack scenario analyzed by J Cronenberger, Thesis, San Antonio, TX 2011 (the thesis contains a validation program)



(Present work several 100 *M K*-solutions)

4. H-P Version of the Finite Element Method and the STRIPE Software

STRIPE FE-software based on a *hp*-version of FEM

- Software developed at The Aeronautical Research Institute of Sweden FFA (now FOI) 1984- for static, dynamic, linear, nonlinear (material, geometry, contact) analysis. Objective: Large 3D structures.
- Used since 1986- for advanced analysis of Swedish Fighter Aircraft
- Based on a *hp*-version of FEM (error control, exponential convergence)
- STRIPE use several novel computational schemes, fracture mechanics,…
- STRIPE is a hybrid code that uses MPI+OpenMP,STRIPE is written in FORTRAN and C, 300k statements
- STRIPE solves efficiently GDOF-problems on massively parallel computers.
- STRIPE is a module in the multi-scale analysis system BABEL

STRIPE's HPC-History:

CRAY-1A,CRAY1-S,CRAY-YMP,SGI,COMPAQ,IBM, SUN, **SGI/O3K/ERDC, IBM-kraken-babbage-davinci -…-Armstrong…**

1984 1986 1988 1990 2000 **2002 2004 2006 2008 2014**

1p 1p 1p 8p 64p-2000p **512p 1600p, …………....5000p 24000p**

US DoD HPC-systems used in the present work: Gordon, Conrad, Onyx and Excalibur at ERDC, ARFL, NAVO

Example: Two large-scale STRIPE applications

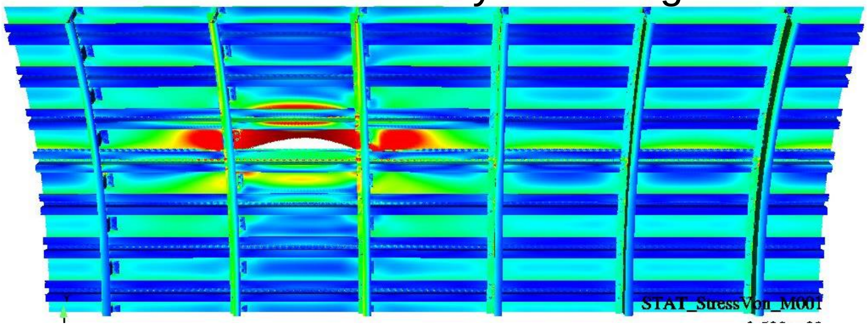
- Residual strength analysis of 1-bay crack in civil aircraft
- Solution of GDOF-size structural problems, C130, A380,

STRIPE Residual Strength Analysis

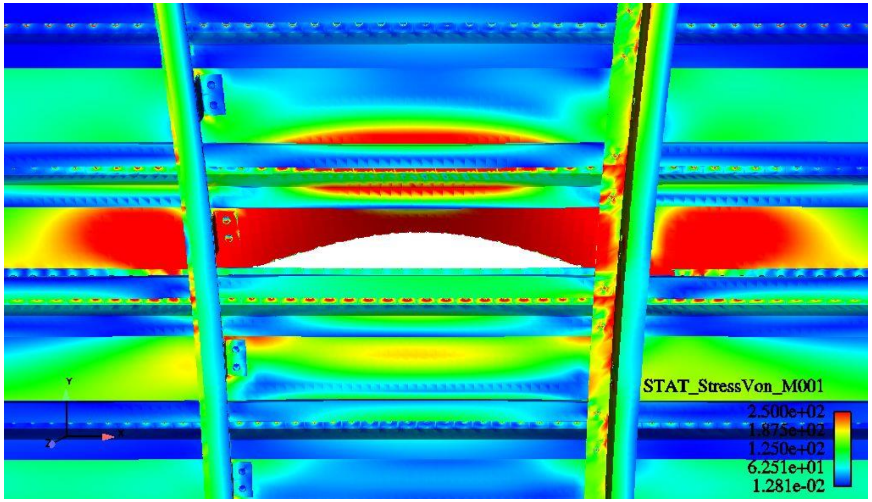
US DoD HPC Challenge Project C2G 2005-2008

Fawaz, Andersson, UGC 2008

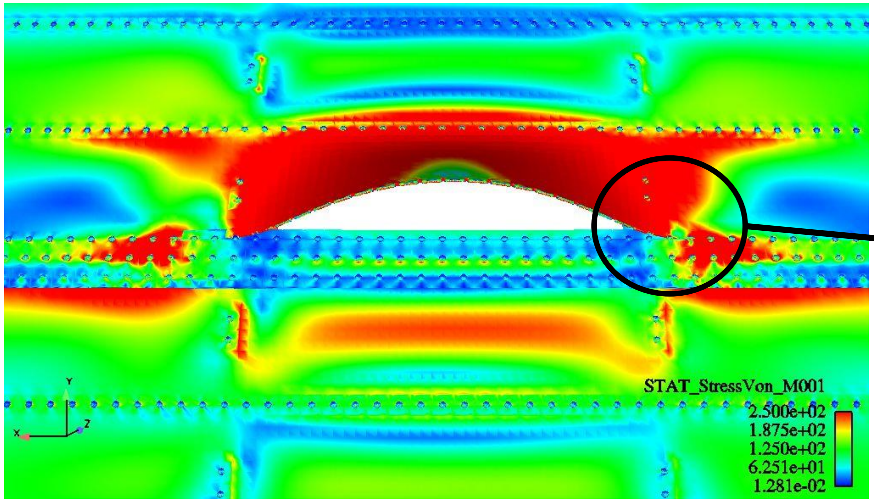
Model for Nonlinear analysis having 225 MDOfs



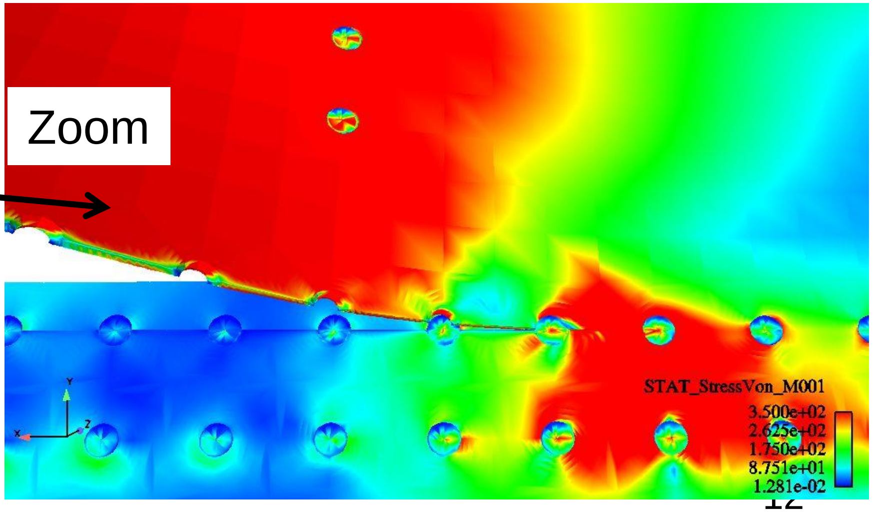
1-bay crack, all countersunk fasteners are modelled as 3D objects



Seen from inside



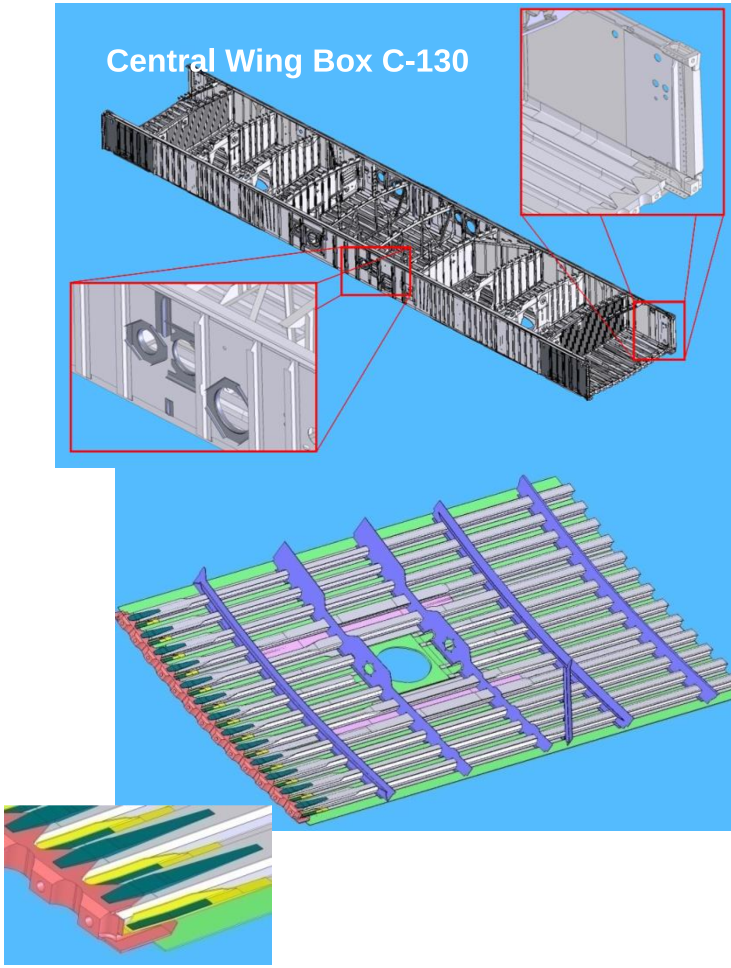
Seen from outside

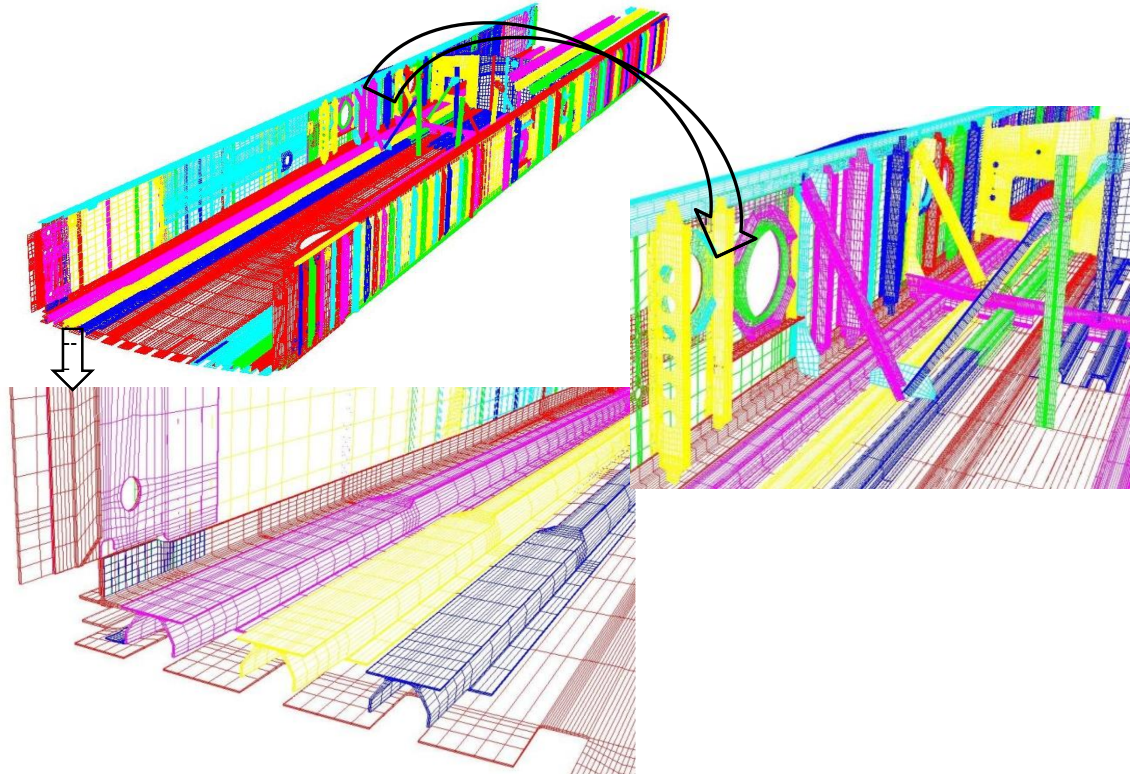


STRIPE Large-Scale Analysis, GDOF Challenge

US DoD HPC Challenge Project C2G 2005-2008

Fawaz, Andersson, UGC 2008



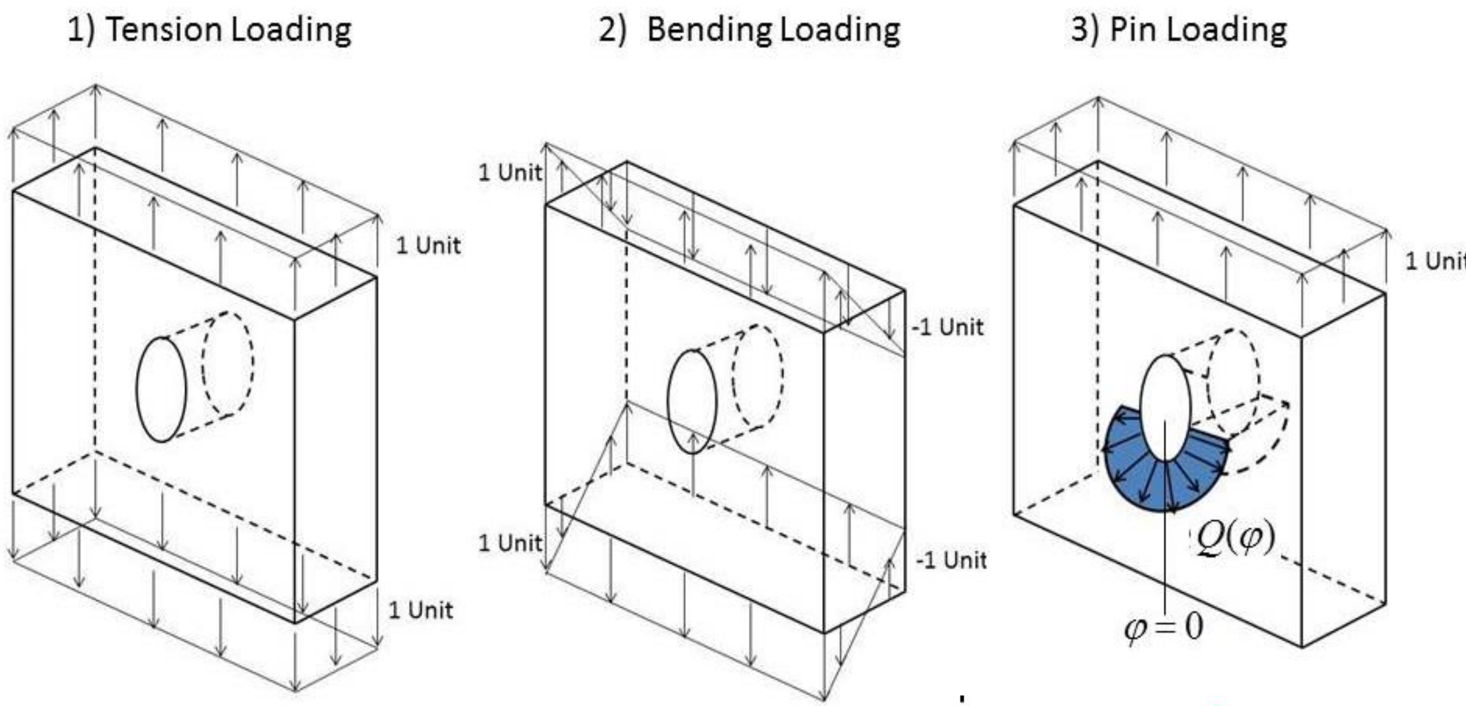


- Aircraft Structural Model
    - > 1000 components
    - > 2000 interfaces
    - > 25 million finite elements
    - > 5 GDoF

- 3D Analysis of an 20 m long AIRBUS A380 ***composite*** fuselage section with fasteners
- Statistical fatigue analysis of aircraft frames
- …

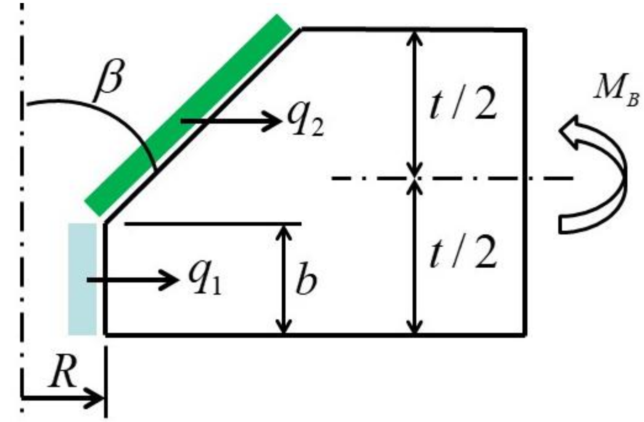
5. Loads

Three Load Cases, straight shank hole



$$Q(\varphi) = \begin{cases} q \cdot \cos^2(\varphi) \quad \text{for} \quad -\pi/2 \le \varphi \le \pi/2 \\ 0 \;\; \text{for} \;\; |\varphi| > \pi/2 \end{cases}$$

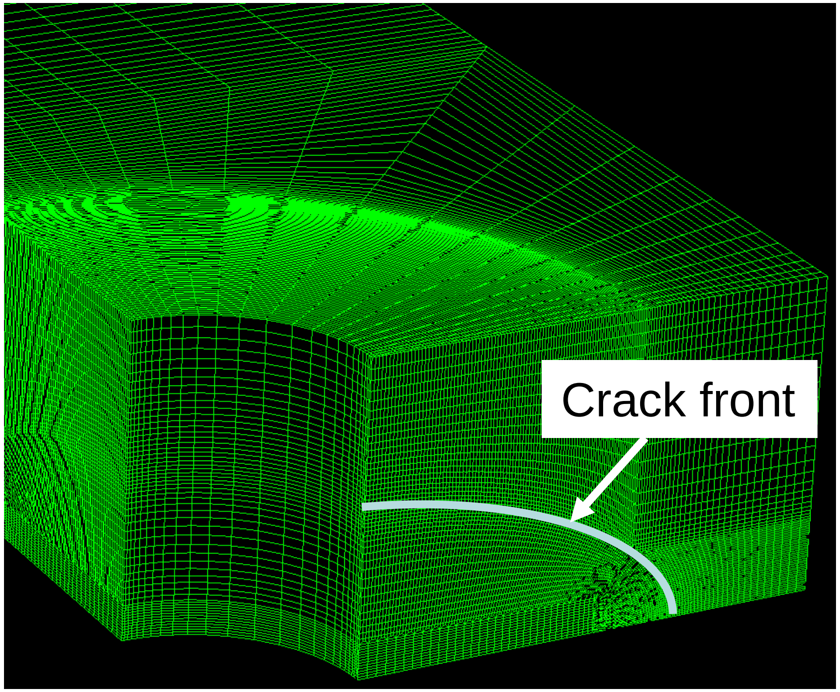
$$2 \cdot W \cdot t \cdot 1 = q \cdot t \cdot \int_{-\pi/2}^{\pi/2} R \cdot \cos^2(\varphi) \cdot \cos(\varphi) \cdot d\varphi$$

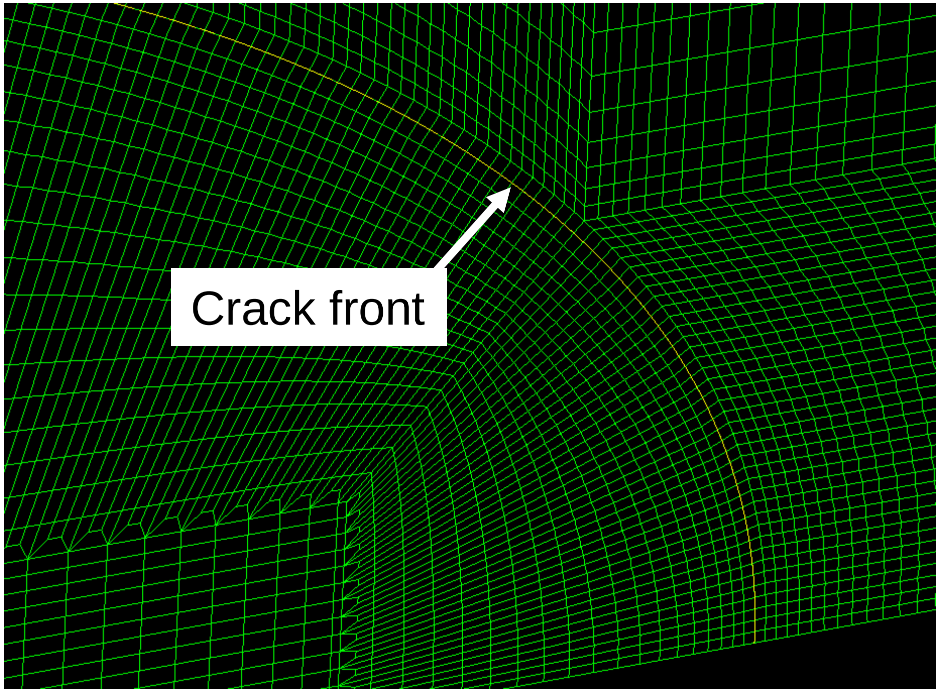


6. Derivation of stress intensity functions $K(\theta)$ with small *guaranteed* error is difficult!

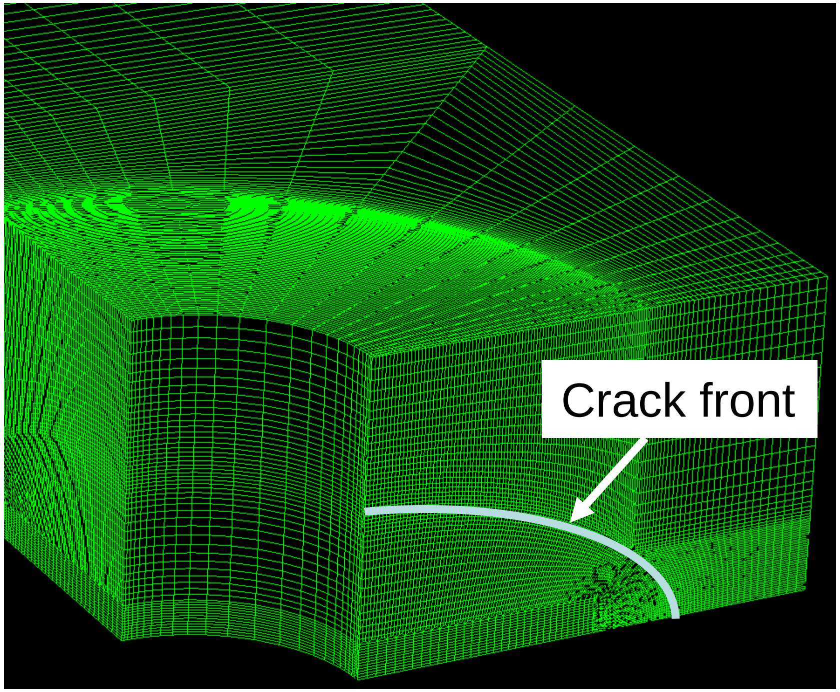
At the AFGROW Conference 2013, Hammond et al used extremely detailed meshes and DoD HPC hardware (with simple elements (p=1)) to derive K-solutions.

Question: Are calculated K-solutions accurate?

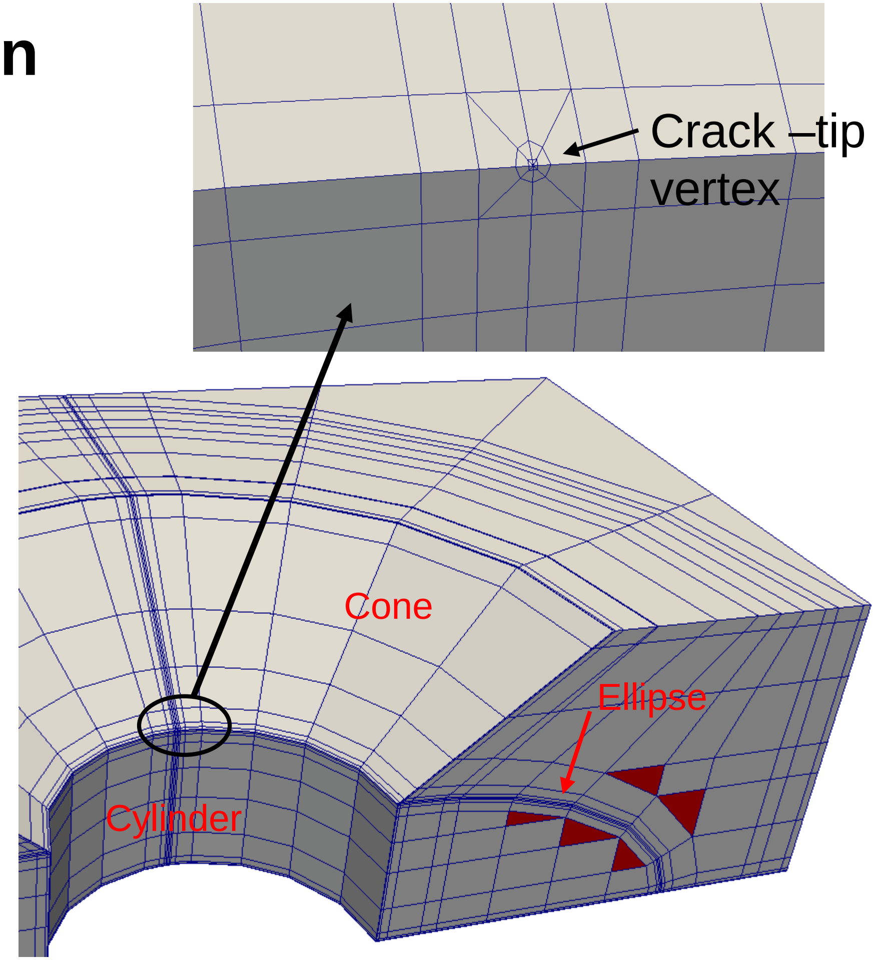




Meshes for *h-* and *hp*-version of FEM.



Mesh for *h*-version of FEM

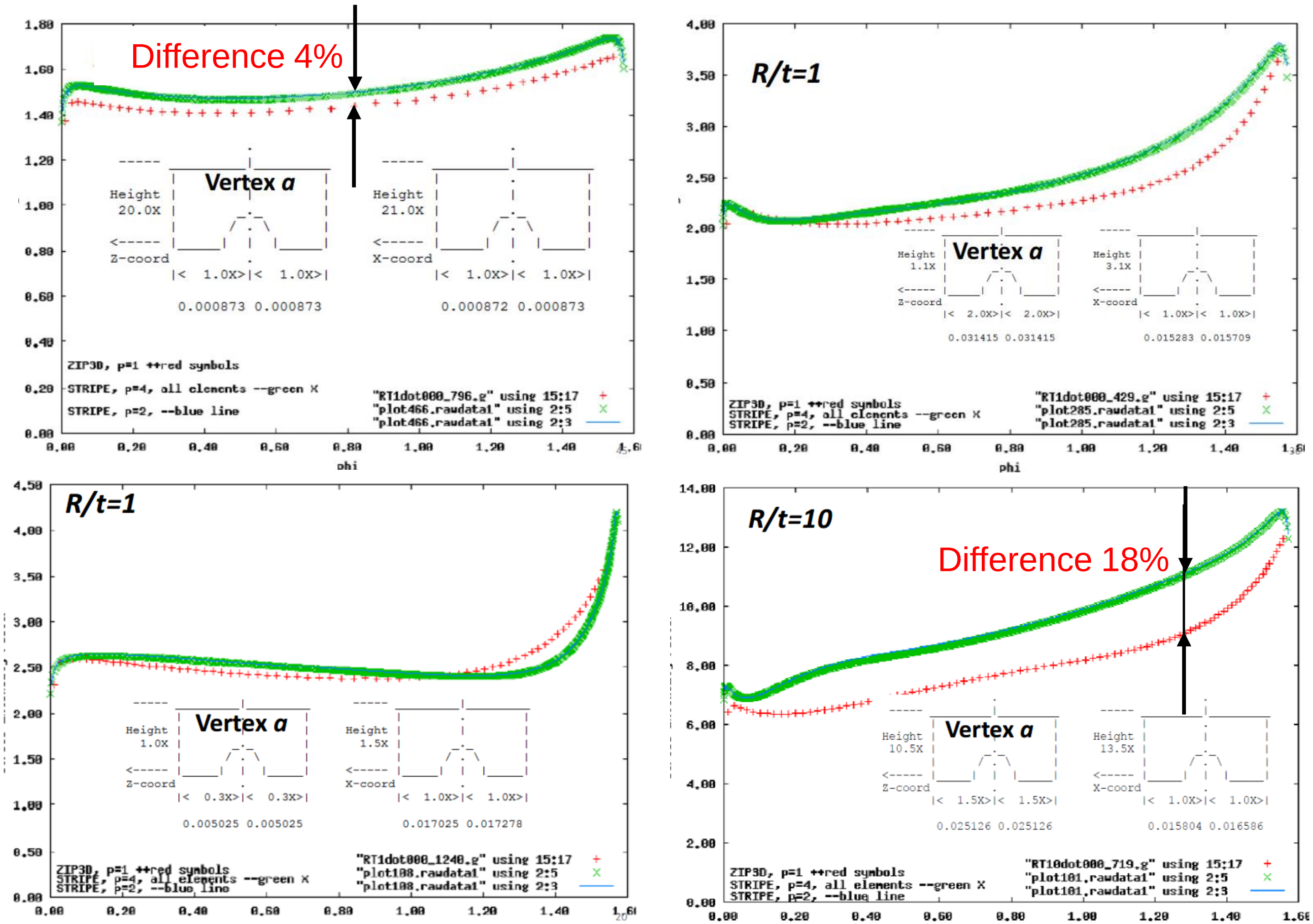


Mesh for *hp*-version of FEM. Blended function mapping is used to model geometry exact.

Comparison STRIPE/ZIP3D, Fawaz, Andersson 2014

++++++++ ZIP3D, *p*=1, *h*-version, VCE method for *K*-calculation.

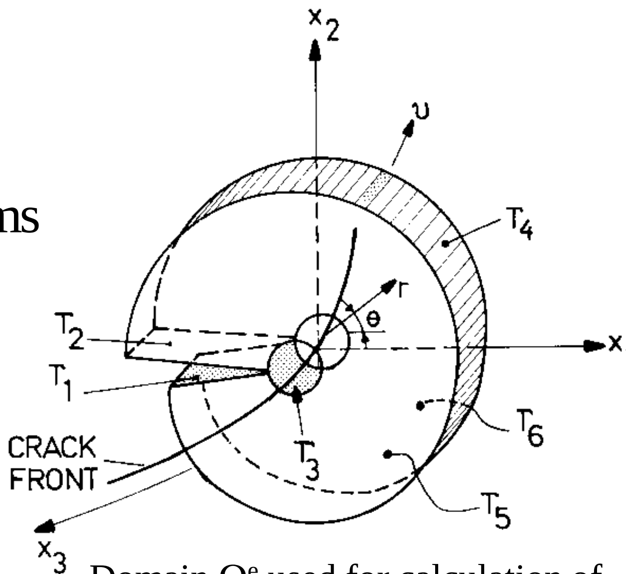
STRIPE, p=4, *hp*-mesh near crack front (not vertices), advanced extraction method for *K*-calculation.



Accurate Calculation of $K_I(\phi)$ at arbitrary $\phi=\phi^*, 0<\phi^*<\bar{\phi}$.

Displacements **u** at a point x_3 on the crack front:

$$\mathrm{u}(r,\theta,x_3)=\sum_{\alpha=I,II,III} K_\alpha(x_3) r^{1/2}\Psi_\alpha(\theta)+\text{smoother terms}$$



Domain Ω^e used for calculation of stress intensity factors

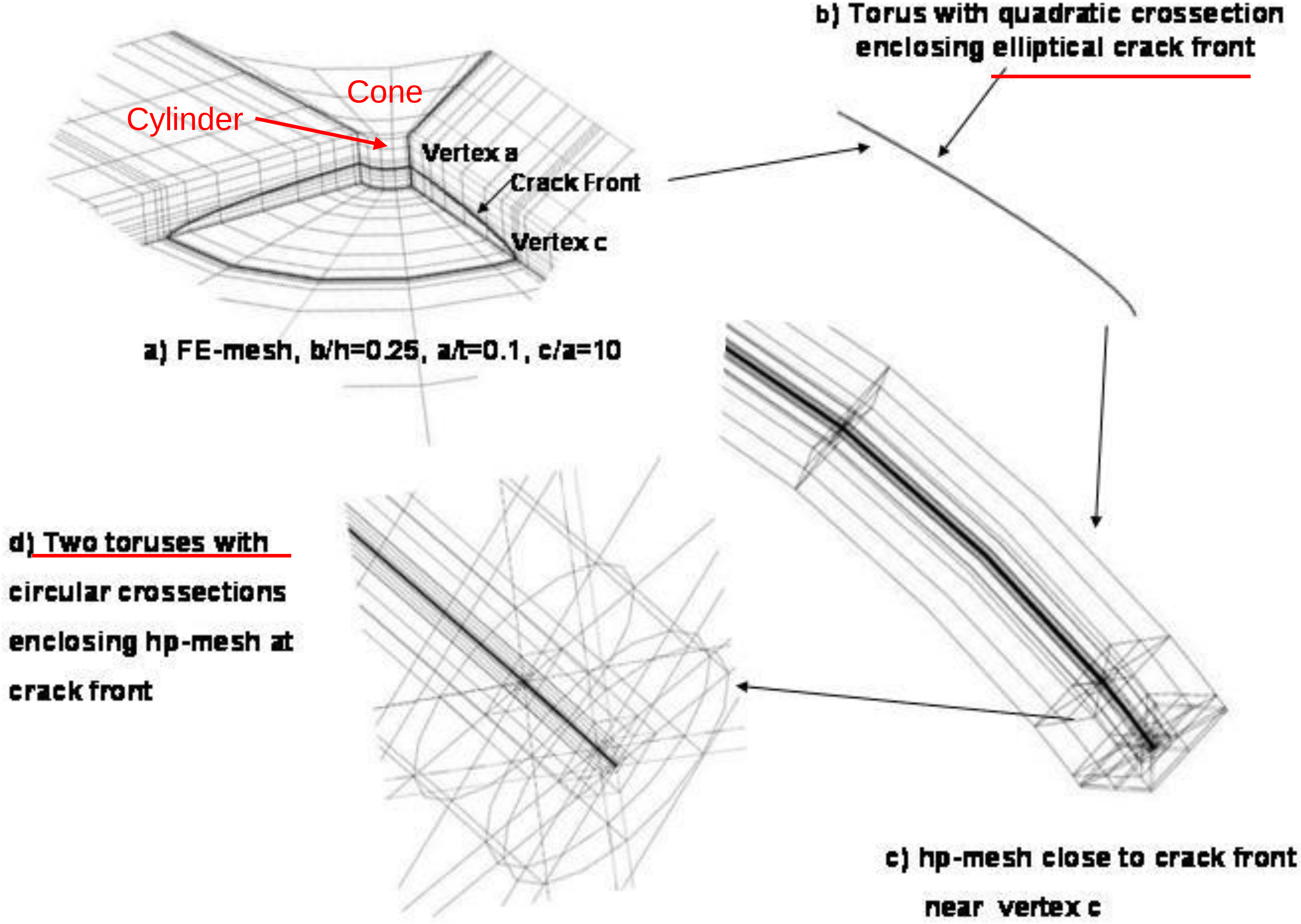
For smooth edges, the edge intensity functions K_α are analytic on open intervals $s_k \le x_3 \le s_{k+1}$. Hence, we approximate the edge intensity functions with the polynomials:

$$\bar{K}_\alpha(x_3)=\sum_{n=0}^{p}\bar{k}_{\alpha n}P_n(s), \qquad s=\frac{2(x_3-s_k)}{s_{k+1}-s_k}-1 \qquad \text{Eq. A}$$

Where $\bar{k}_{\alpha n}$ are unknown coefficients, p is the polynomial order of the finite element, and P_n the Legendre polynomials.

By applying the reciprocity theorem, Equation A gives *exponentially fast convergence*, with increasing polynomial order p to the exact K-values.

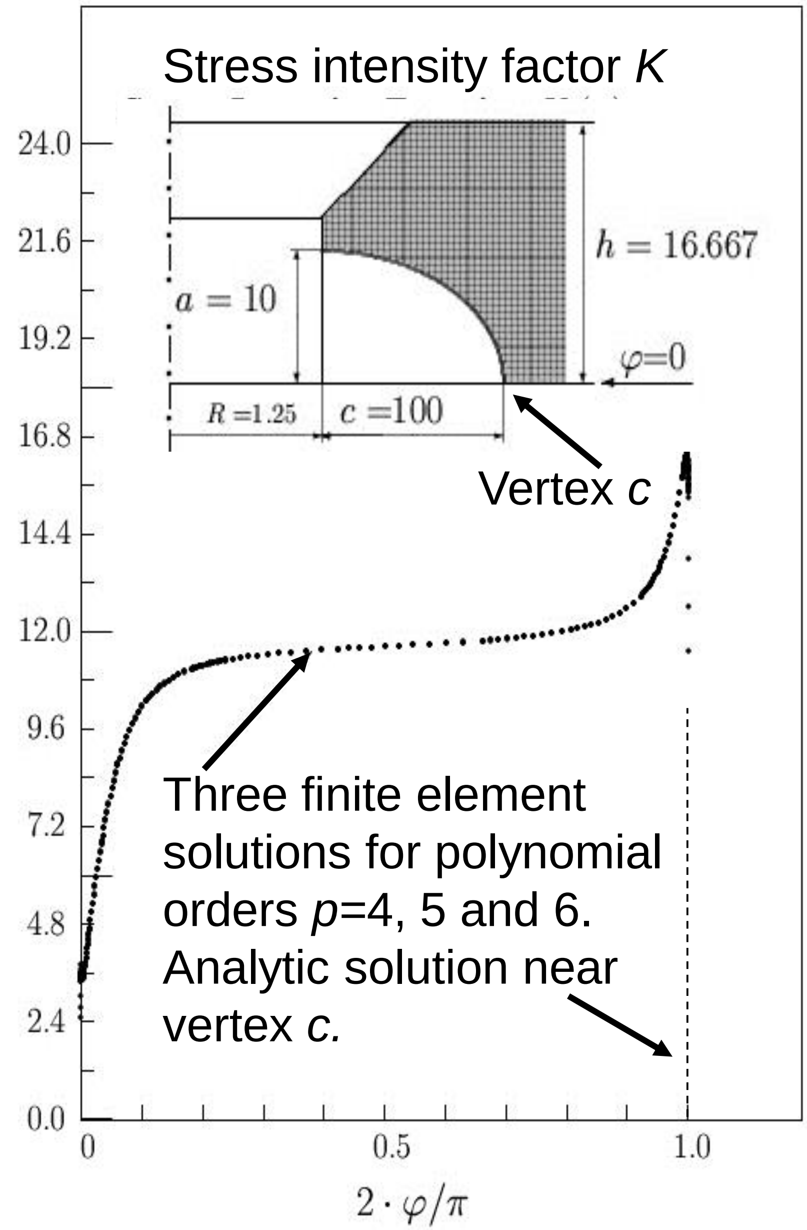
Meshes for the *hp*-version of FEM (Analytic geometry)



STRIPE: Calculation of Edge and Vertex stress intensity factors. Mathematical theory.

Vertices
-Babuska, Petersdorff, Andersson
SIAM J Num Anal, Vol 31, 1994, pp 1265-1288

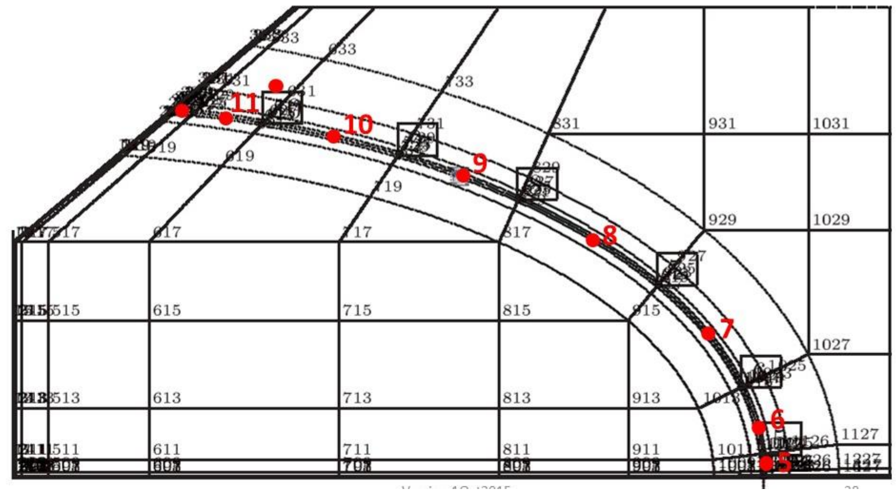
Edges and Vertices
-Andersson, Babuska, Falk
ICAS-90-4.9.2, 1990, p 1730-1746



BABEL values of *K(θ)* versus direct calculation using the definition of *K(θ)*.

Table shows:

(KI-stripe - $\sigma\sqrt{2\pi r}$)/KI-stripe



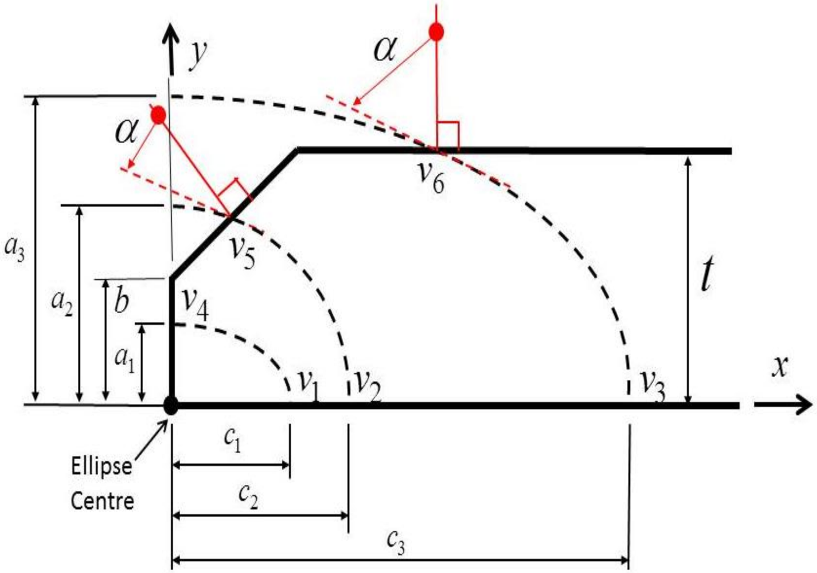
R/t=1.18, a/t=0.8, a=50, c=100,Tension, *p*=6

r ↓ θ →	DOM 4	DOM 5	DOM 6	DOM 7	DOM 8	DOM 9	DOM10	DOM11	DOM 12	DOM 13
0.10000000	-0.014	-0.000	-0.003	-0.002	-0.002	-0.001	-0.001	-0.001	-0.001	-0.040
0.05000000	-0.008	-0.001	-0.001	-0.000	-0.000	-0.000	-0.000	-0.000	-0.001	-0.016
0.02000000	-0.005	-0.001	-0.001	-0.002	-0.002	-0.002	-0.001	-0.001	-0.002	-0.004
0.01000000	-0.001	-0.000	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.002
0.00500000	-0.002	-0.001	-0.001	-0.000	-0.000	-0.000	-0.000	-0.001	-0.001	-0.003
0.00200000	-0.000	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001
0.00100000	-0.001	-0.001	-0.002	-0.000	-0.000	-0.001	-0.002	-0.002	-0.002	-0.002
KI-Stripe	18.78	18.78	17.89	17.81	19.30	21.58	25.02	30.91	38.68	47.34

Verification

7. Behavior of $K(\theta)$ near crack front ends

Behavior of the stress intensity function $K_I(\phi)$ near a vertex v_i

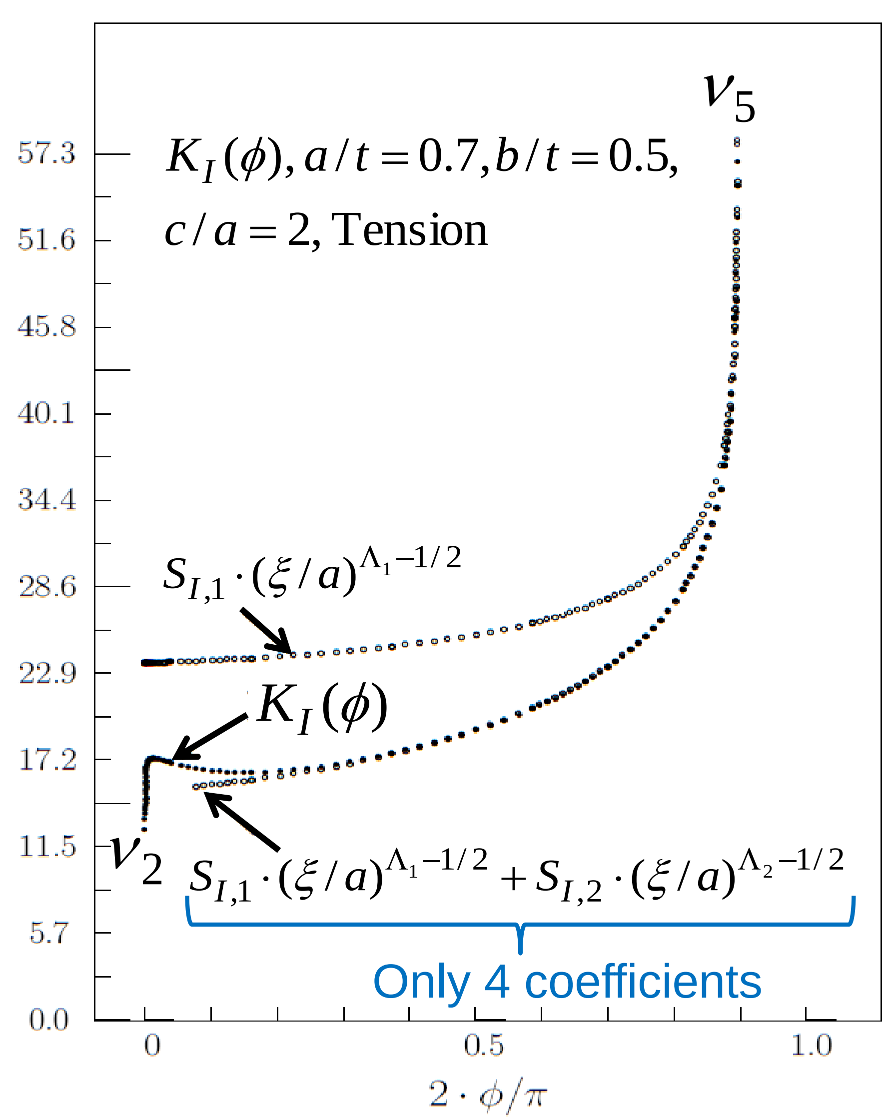


Displacements near a vertex

$$\begin{Bmatrix} u(\rho,\omega,\varphi) \\ v(\rho,\omega,\varphi) \\ w(\rho,\omega,\varphi) \end{Bmatrix} = \sum_{j=1}^{J} B^{(j)} \cdot \rho^{\Lambda_j} \cdot \begin{Bmatrix} \Theta_1^{(j)}(\omega,\varphi) \\ \Theta_2^{(j)}(\omega,\varphi) \\ \Theta_3^{(j)}(\omega,\varphi) \end{Bmatrix} + \ldots$$

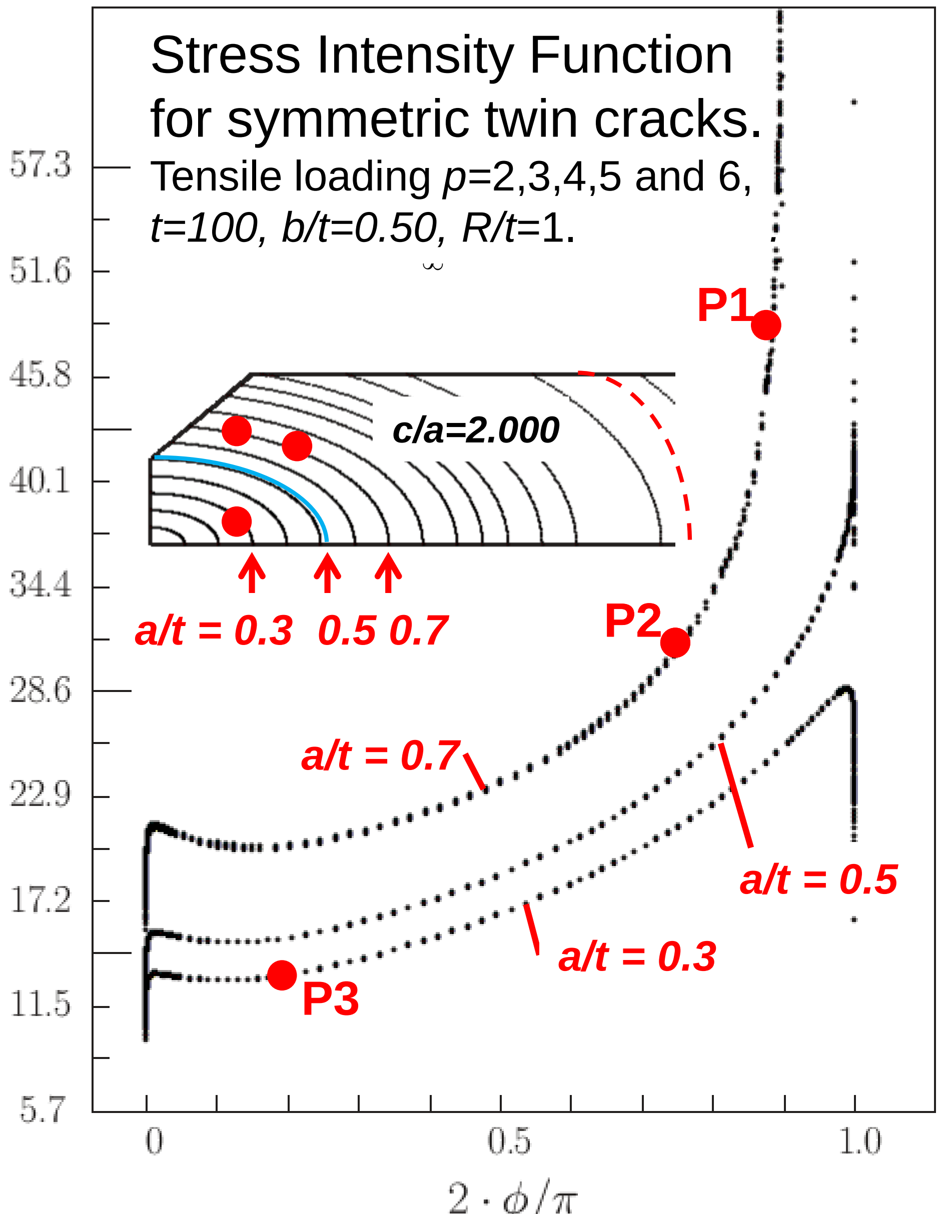
$K_I(\phi)$ near a vertex

$$S_{I,1} \cdot (\xi / a)^{\Lambda_1 - 1/2} + S_{I,2} \cdot (\xi / a)^{\Lambda_2 - 1/2}$$



ξ is the distance to the vertex

Smooth and Singular behavior of $K(\theta)$



Point-wise convergence of $K_I(\phi_i)$

Point P1

p	K_I	Err %
2	48.18	0.75
3	47.91	1.31
4	48.50	0.10
5	48.55	0.02
6	48.55	Zero

Point P2

p	K_I	Err %
2	31.25	0.44
3	31.07	1.01
4	31.37	0.08
5	31.39	0.02
6	31.39	Zero

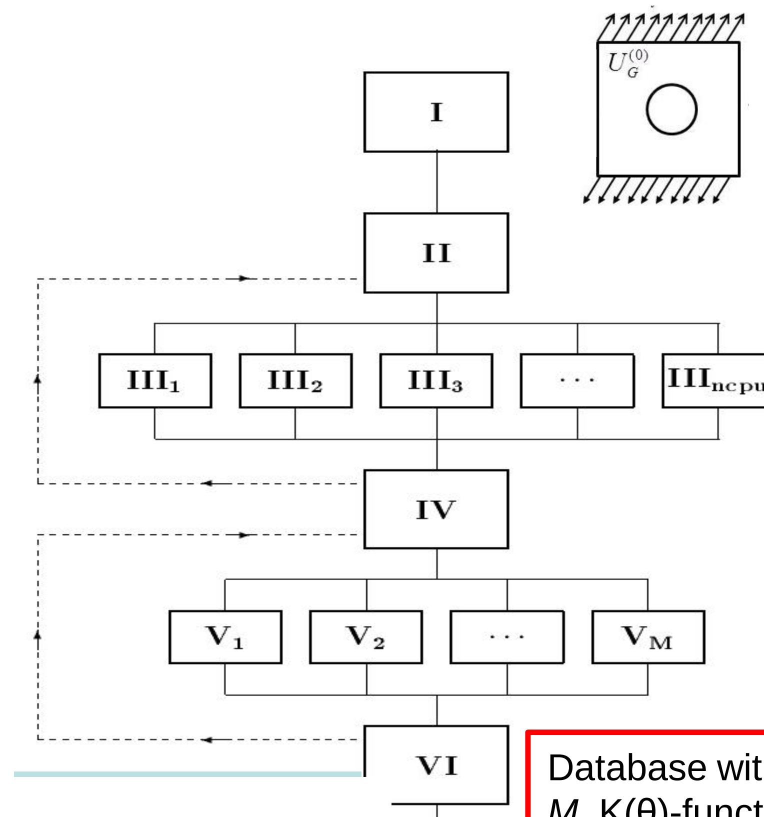
Point P3

p	K_I	Err %
2	12.87	0.40
3	12.84	0.14
4	12.81	0.07
5	12.82	0.02
6	12.82	Zero

8. A System for Multi-Level and Multi-Scale Analysis

BABEL

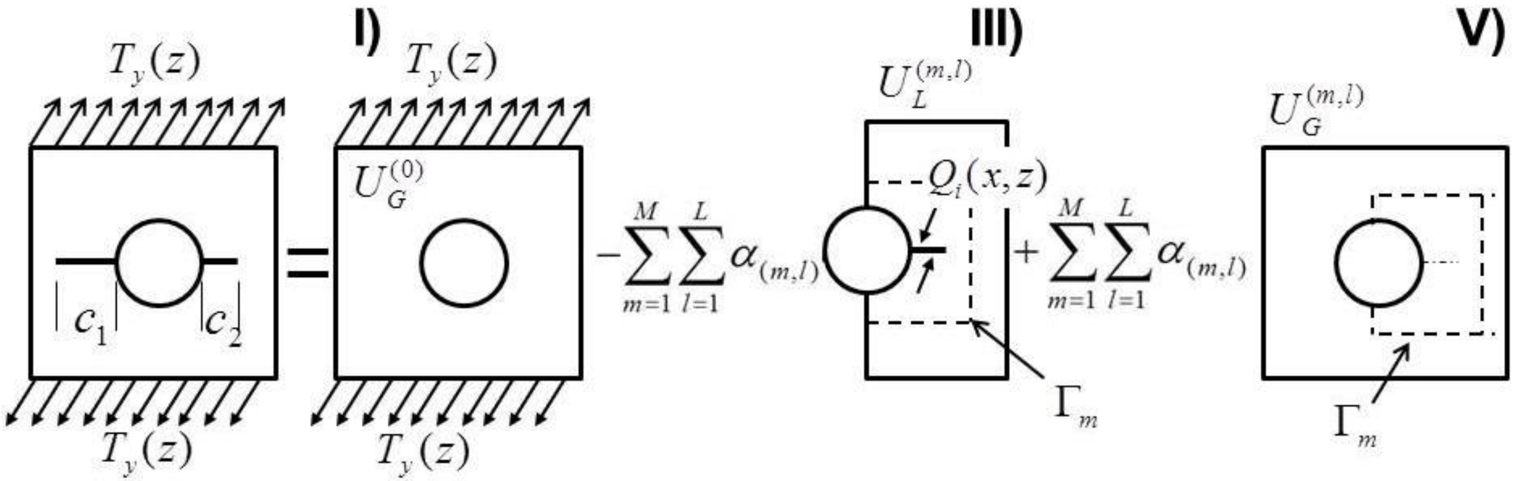
The Tower of BABEL, painting by Bruegel, the Year 1563



Database with many *M* K(θ)-functions

A mathematical splitting method for fast K_I -analysis

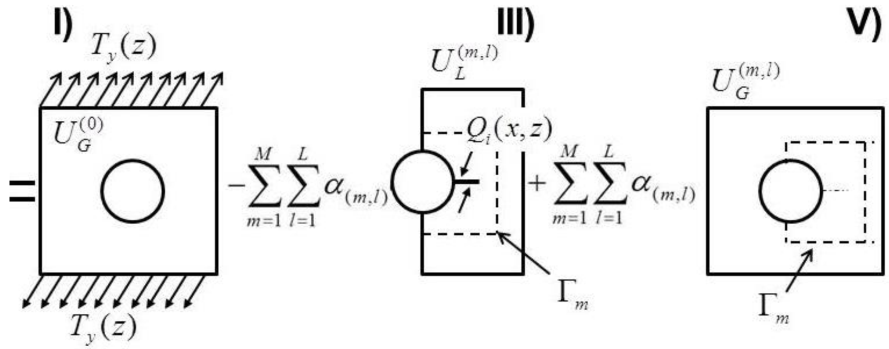
Babuska, Anderssson, SIAM J Scient. Comp. Vol 26, 2005



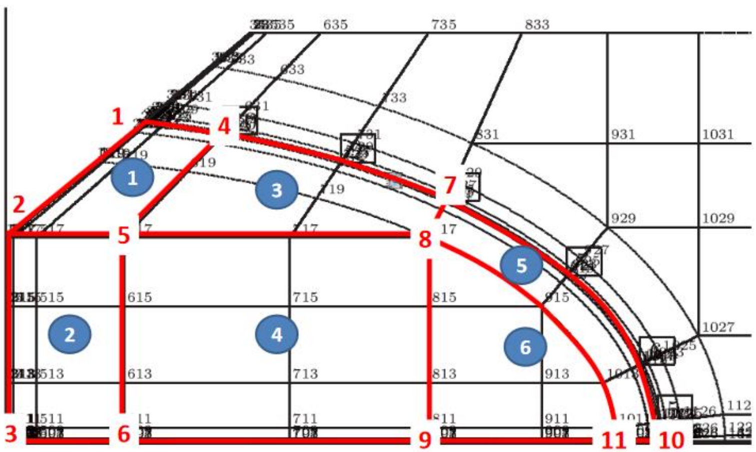
$$U = U_{global}^{(0)} + \sum_{m=1}^{M}\sum_{l=1}^{L}\alpha_{m,l}\cdot(U_{local}^{(m,l)} + U_{global}^{(m,l)})$$

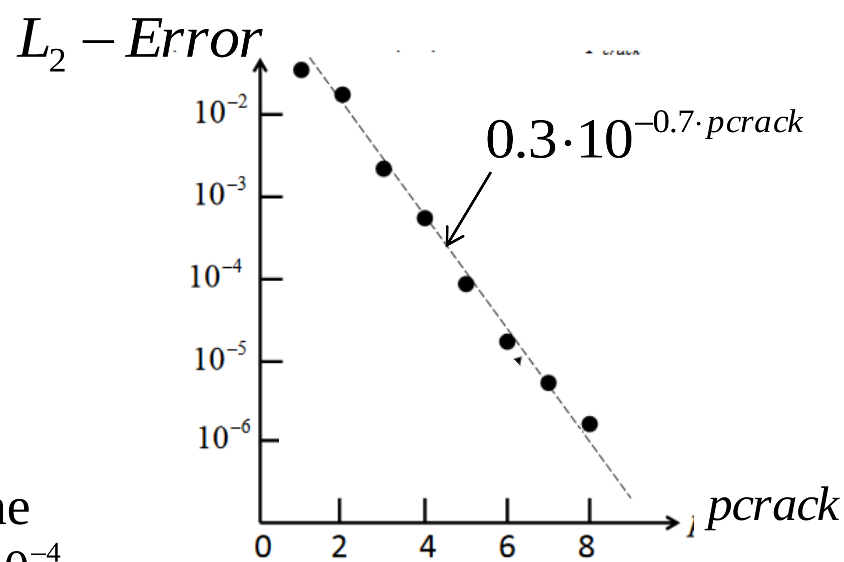
$$K_I^{(0)}(\phi), K_{II}^{(0)}(\phi), K_{III}^{(0)}(\phi) = \sum_{m=1}^{M}\sum_{l=1}^{L}\alpha_{m,l}\cdot(K_I^{(m,l)}, K_{II}^{(m,l)}, K_{III}^{(m,l)})$$

The unknown coefficients $\alpha_{m,l}$ are determined from the condition that stresses shall be zero on the two crack faces (in the least square sense).



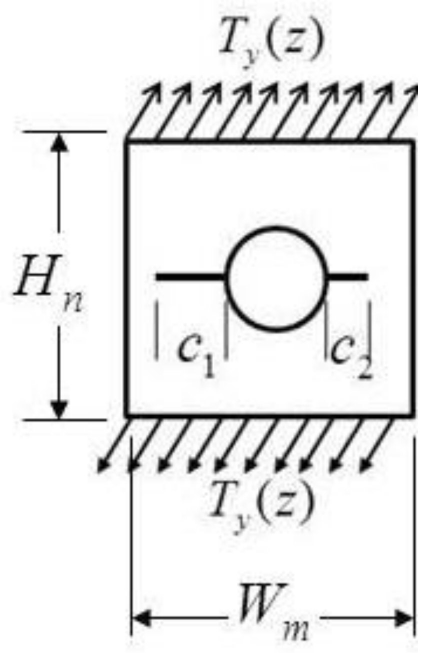
It is important to select the functions $Q_i(x,z)$, and *M* and *L* properly (easy to do)





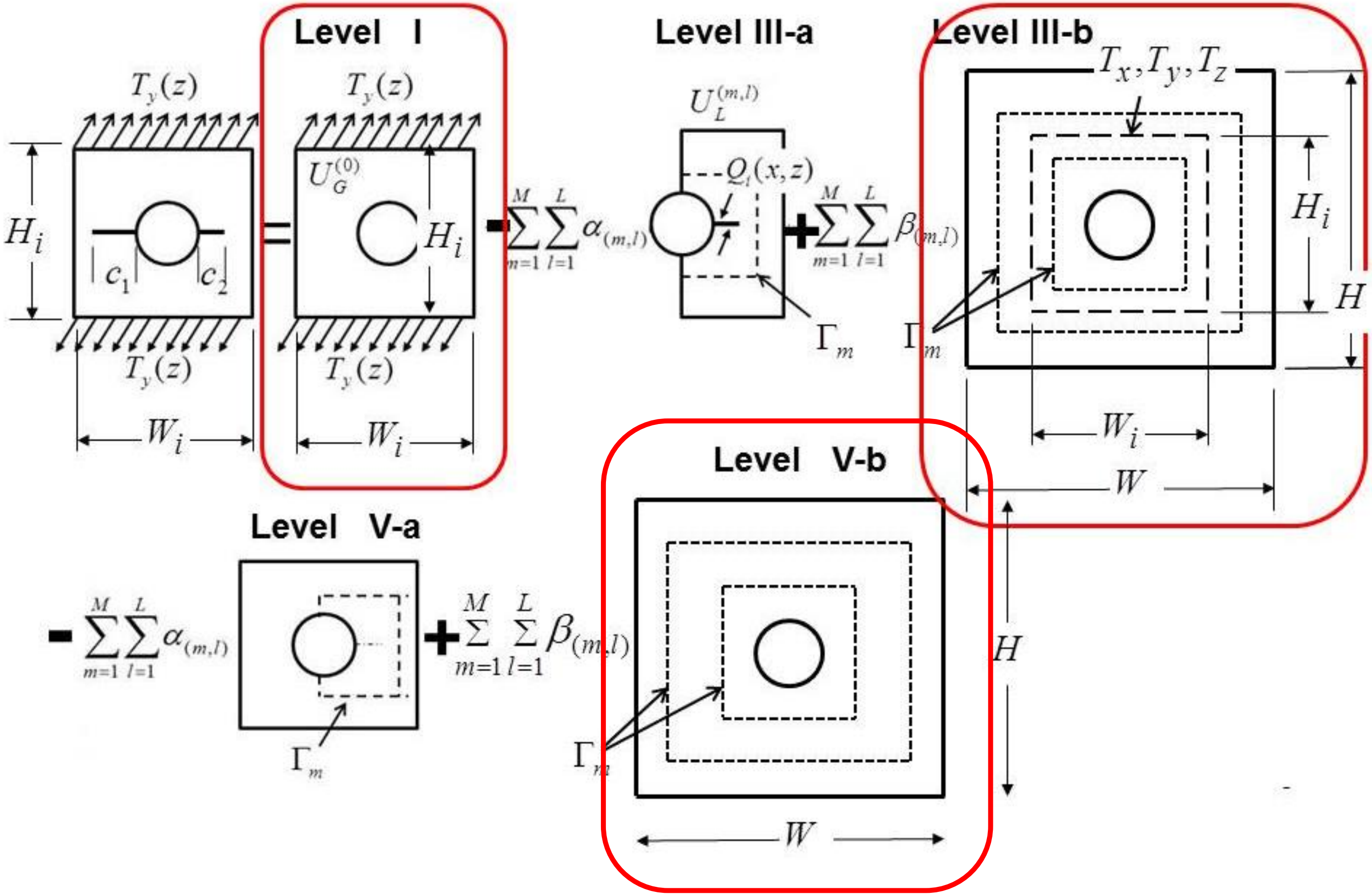
Mesh on crack face used to determine *M(p)* and *L(p)* for prescribed error 10^{-4}

9. Fast Calculation of $K(\theta)$ as Function of Plate Width W and Height H

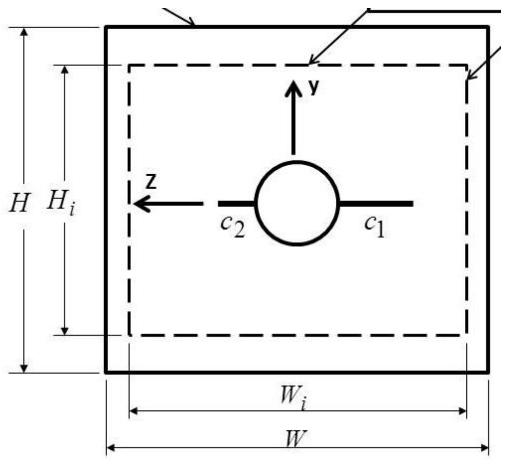


A mathematical splitting method for fast calculation of:

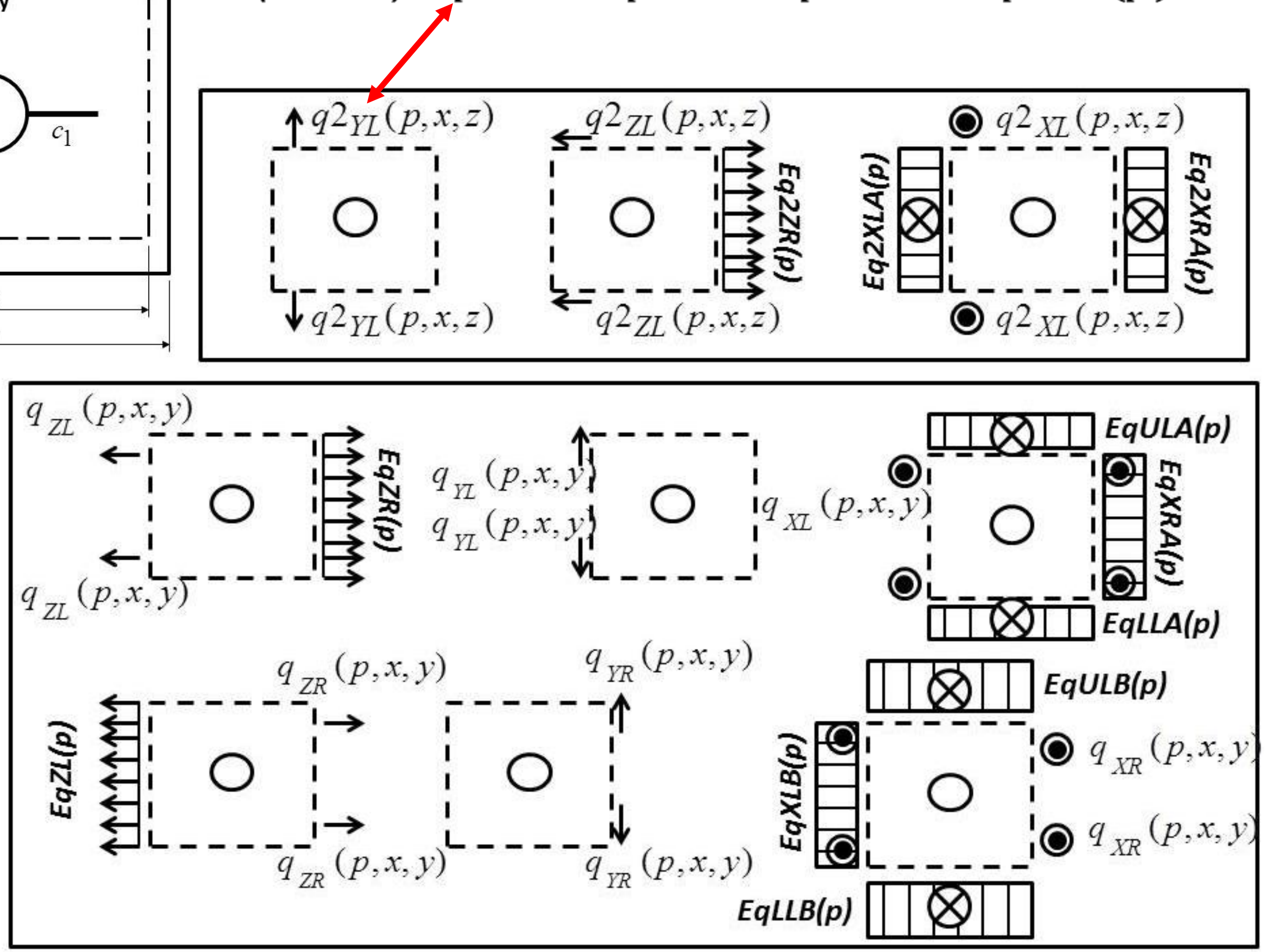
$$K_I(\phi, a_1/R, a_2/R, c_1/R, c_2/R, b/t, R/t, W/R, H/R)$$



Basic loading cases for determination of unknown tractions *T*

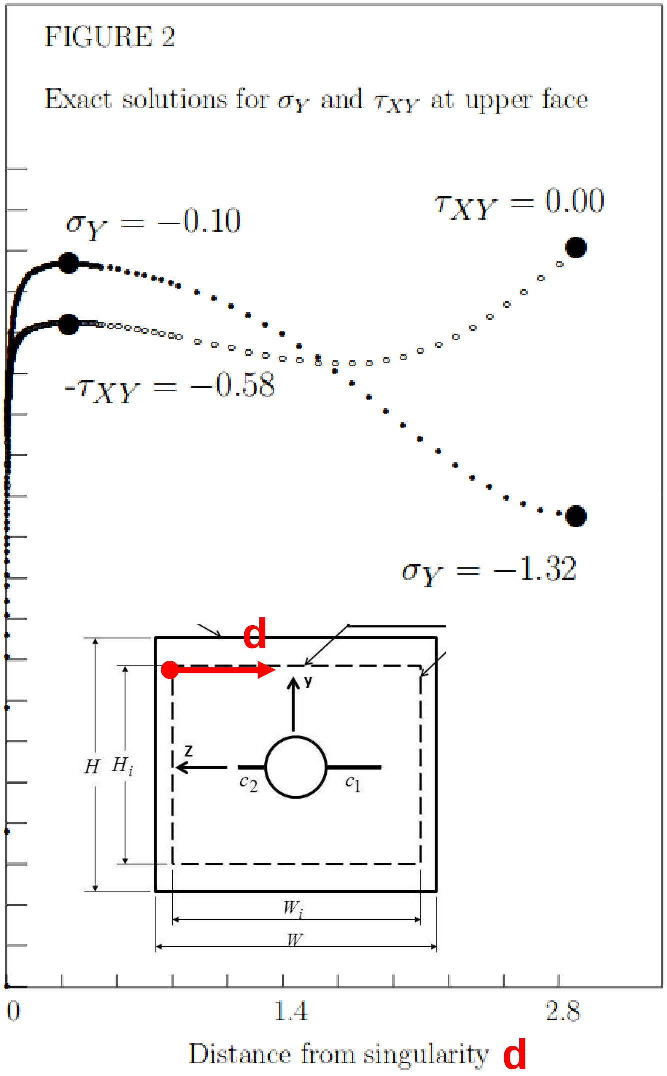


$$T(X,Y,Z) = q2YL + q2ZL + q2XL \ldots EqLLB(p)$$



The basis functions *q(X,Y,Z)* have to be selected so the represent the singular behavior near all 12 edges

The figure shows the exact load distribution and its singular behavior near one corner.



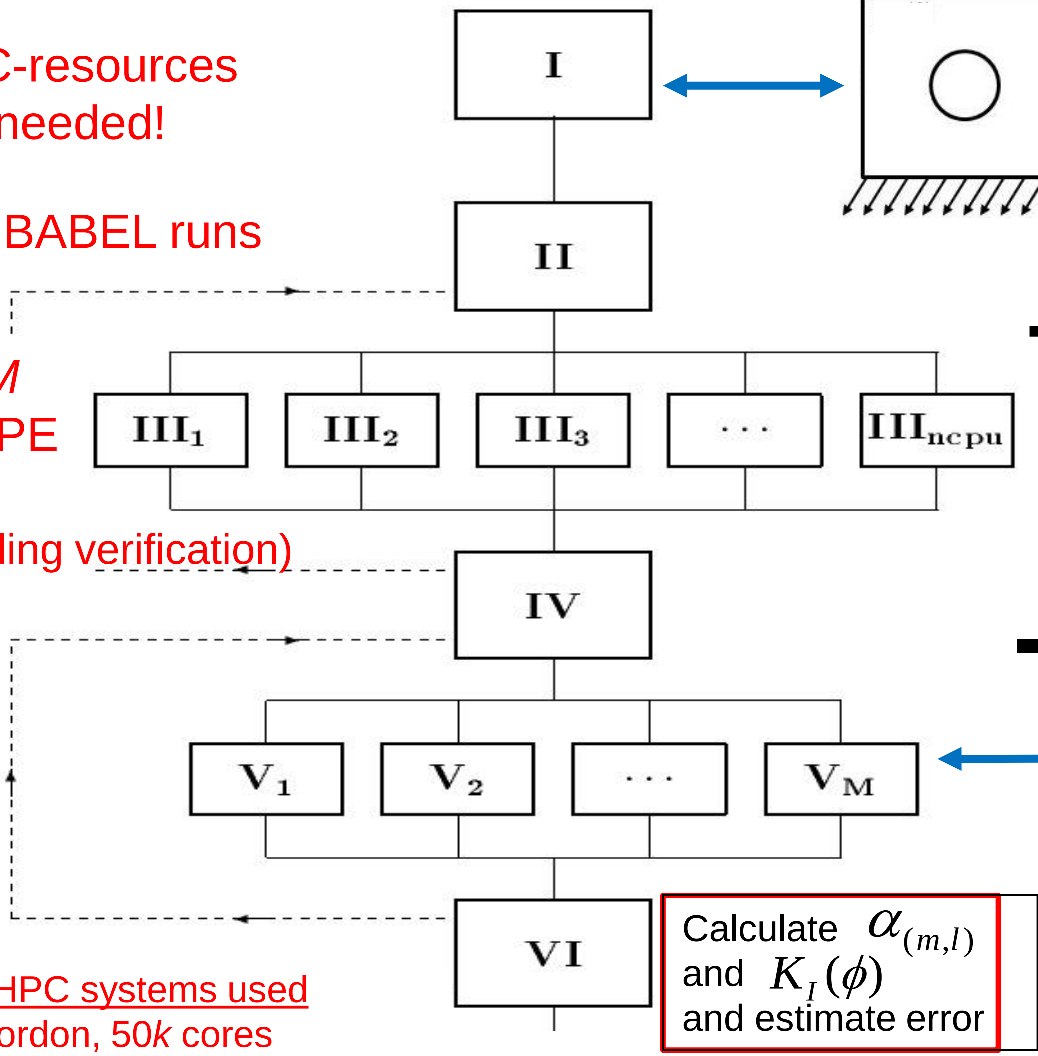
10. Large-Scale K(θ)-data generation using DoD HPC-Resources

A **HPC**-based System for generation of large *K*-Databases

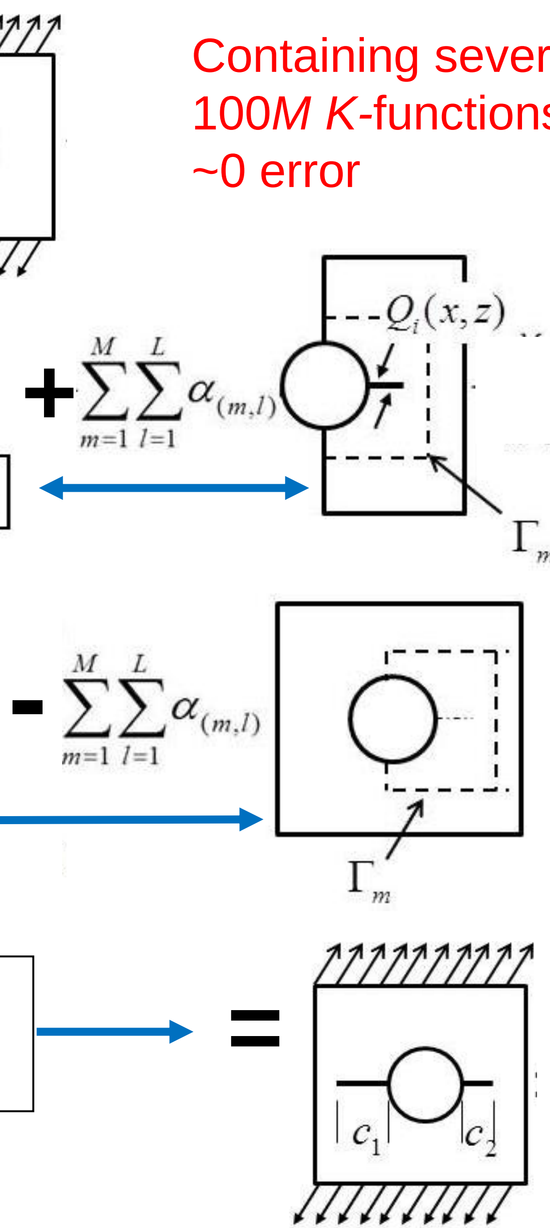
HPC-resources are needed!

~2*k* BABEL runs

~1/2*M* STRIPE Jobs (including verification)



Containing several 100*M* *K*-functions with ~0 error



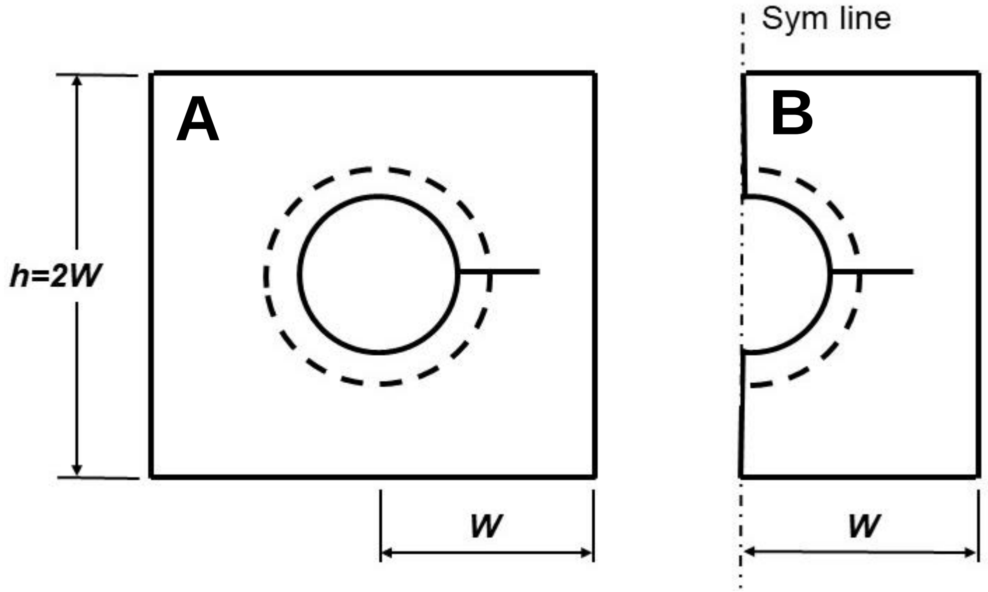
DoD HPC systems used

- Gordon, 50*k* cores
- Conrad, 50*k* cores
- Excalibur, 50*k* cores
- Onyx, 100*k* cores

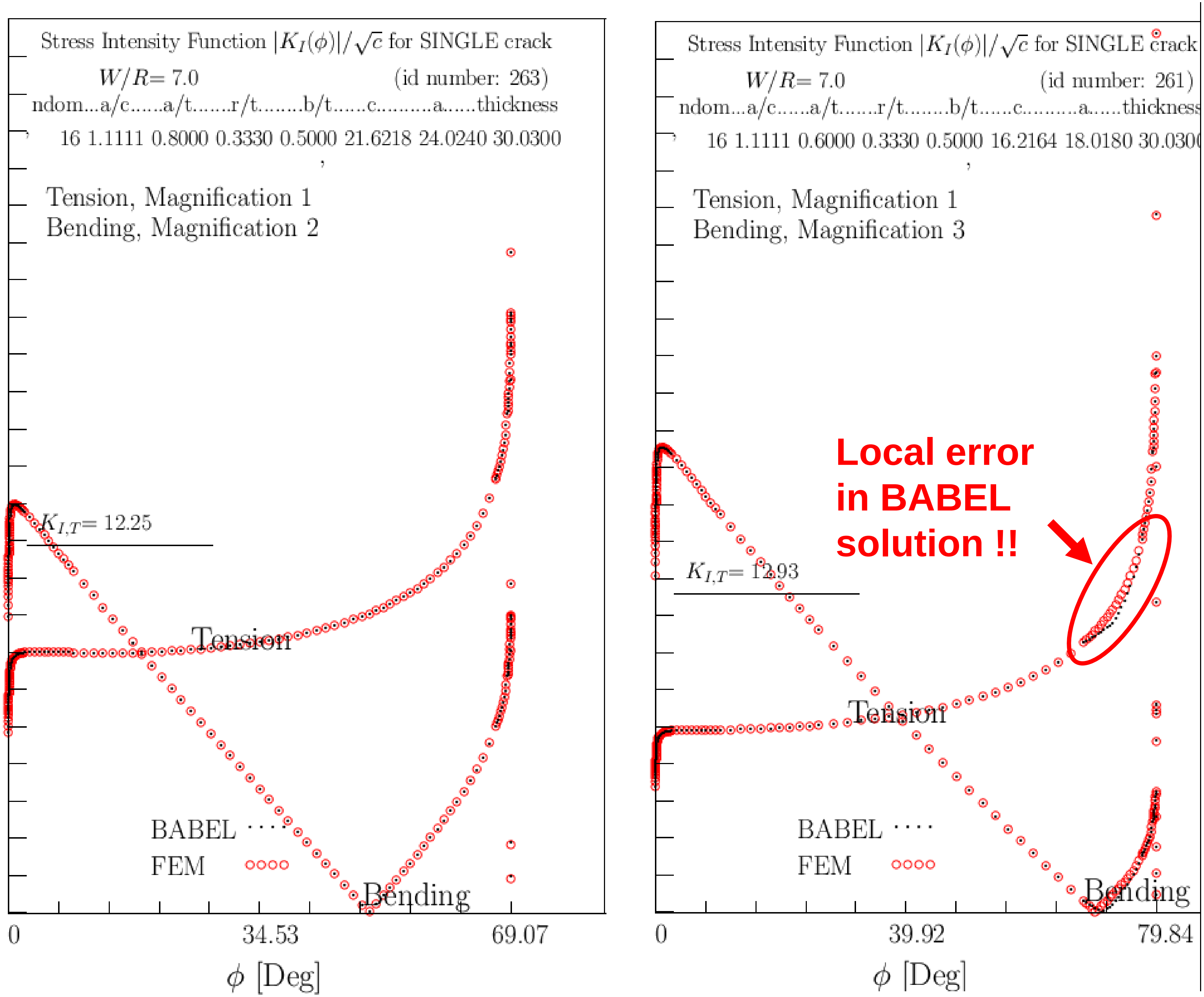
11. Reliability check of ~100*M* calculated stress intensity functions $K(\theta)$

Reliability check of calculated stress intensity functions *K(θ)*

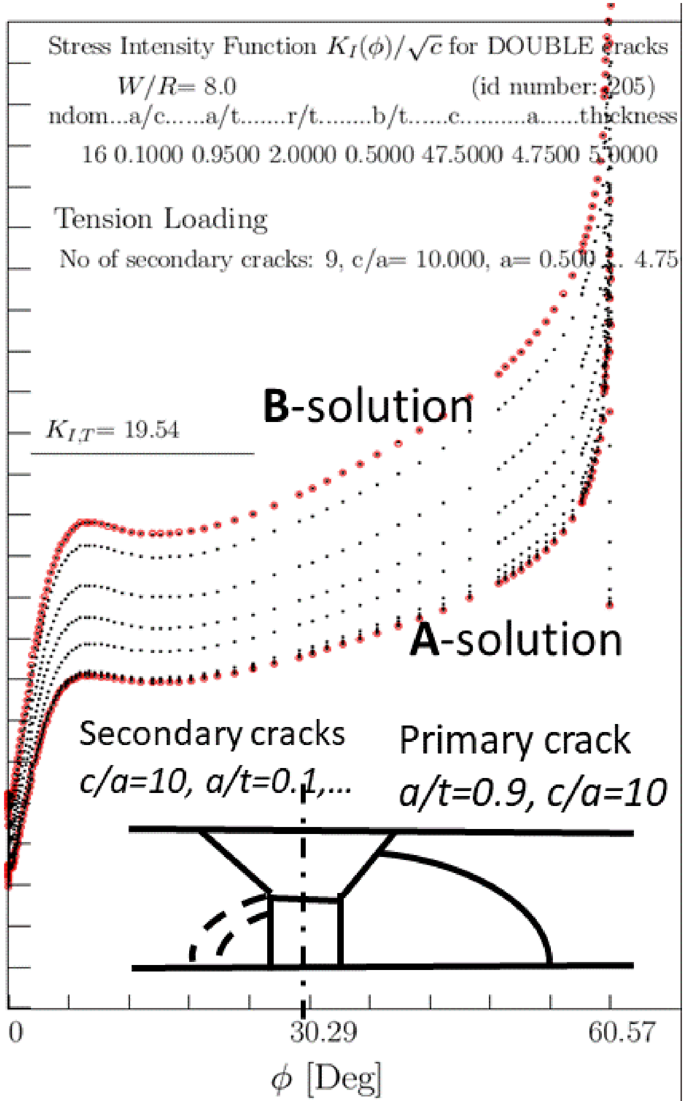
Two model problems ***A*** and ***B*** are used to check the single crack solution (*i.e.* the splitting scheme) and double crack solutions.

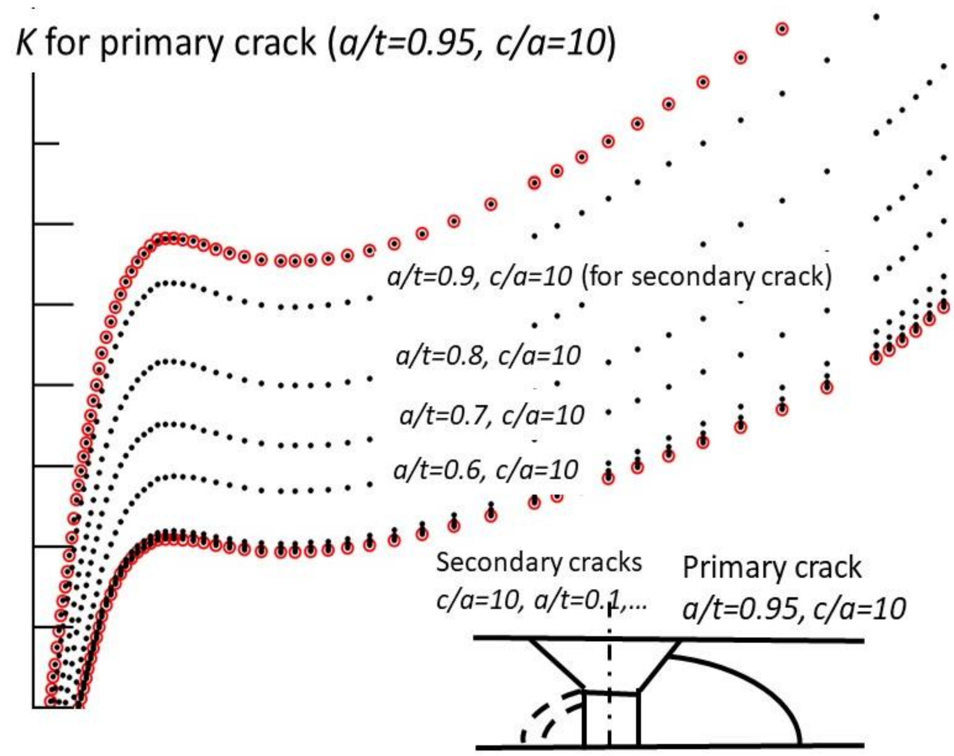


A plate with one crack, problem **A** or two cracks, problem **B** subject to bending, tension and pin loading.


Stress Intensity Function $|K_I(\phi)|/\sqrt{c}$ for SINGLE crack
W/R= 7.0
(id number: 263)
ndom...a/c......a/t.......r/t........b/t......c..........a......thickness
16 1.1111 0.8000 0.3330 0.5000 21.6218 24.0240 30.0300
Tension, Magnification 1
Bending, Magnification 2
$K_{I,T}$= 12.25
Tension
Bending
BABEL
FEM
0
34.53
69.07
ϕ [Deg]
Stress Intensity Function $|K_I(\phi)|/\sqrt{c}$ for SINGLE crack
W/R= 7.0
(id number: 261)
ndom...a/c......a/t.......r/t........b/t......c..........a......thickness
16 1.1111 0.6000 0.3330 0.5000 16.2164 18.0180 30.030
Tension, Magnification 1
Bending, Magnification 3
$K_{I,T}$= 12.93
Local error in BABEL solution !!
Tension
Bending
BABEL
FEM
0
39.92
79.84
ϕ [Deg]

Reliability check of twin crack solutions





Example: Reliability check of 34.6*M* *K*-functions 2019

8 R/t-values

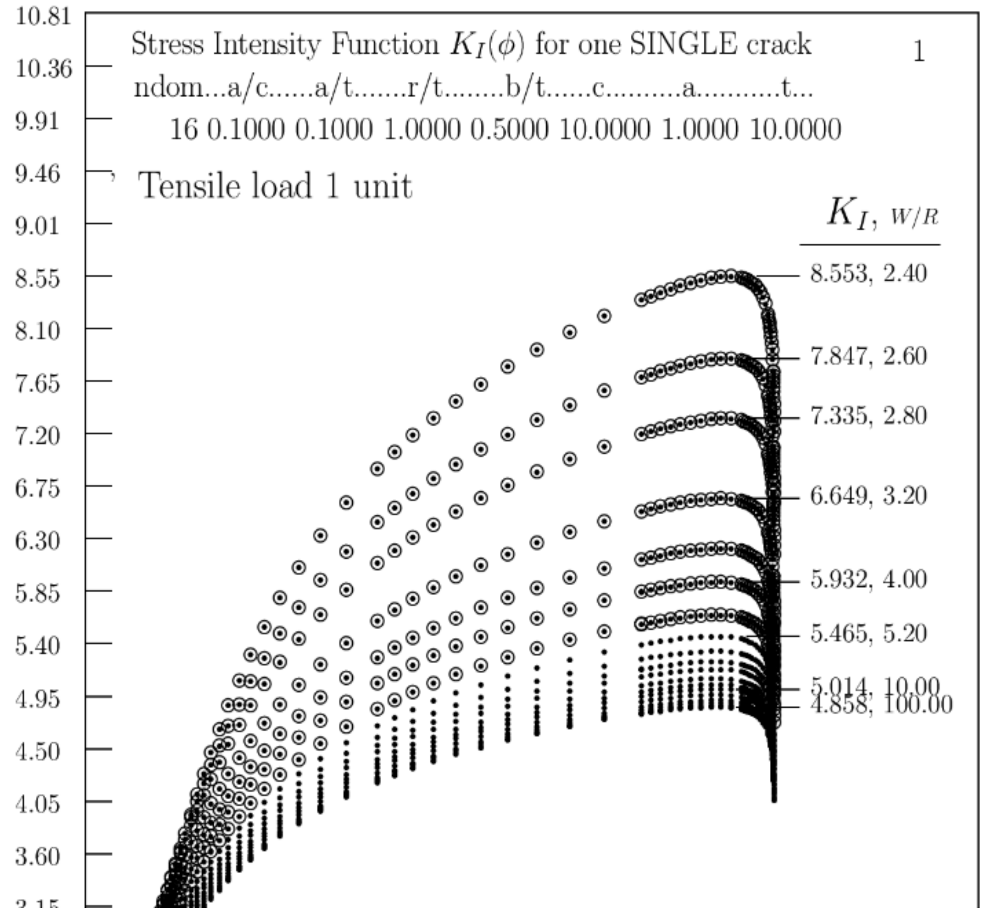
R/t	No of W/R databases	Number of K-functions	No of plots, single cracks	No of plots, twin cracks
0.2	14	3.4M	2.8k	2.8k
0.333	15	3.6M	3.0k	3.0k
0.5	16	3.8M	3.2k	3.2k
1.0	20	4.8M	4k	4k
1.5	20	4.8M	4k	4k
2.0	20	4.8M	4k	4k
3.0	20	4.8M	4k	4k
5.0	19	4.6M	3.8k	3.8k
SUM	134	**34.6M**	**28.8k**	**28.8k**

Checked > 60*k* plots

A few remarks on data storage

Very Much *K*-Data for Practical Use !

- We have several hundred million accurate $K(\theta)$-functions in the databases.
- Each $K(\theta)$- functions is initially stored as about 200 (x,y) –coordinates as shown to right.
- The derived databases must be drastically reduced in size before being distributed to AFGROW users.



Can this be done without any significant loss in accuracy in fatigue crack growth life predictions?

Answer: Yes, of course, but with how much? Factor 10^4, 10^5… ?

Compact Data Storage:

Previous works:

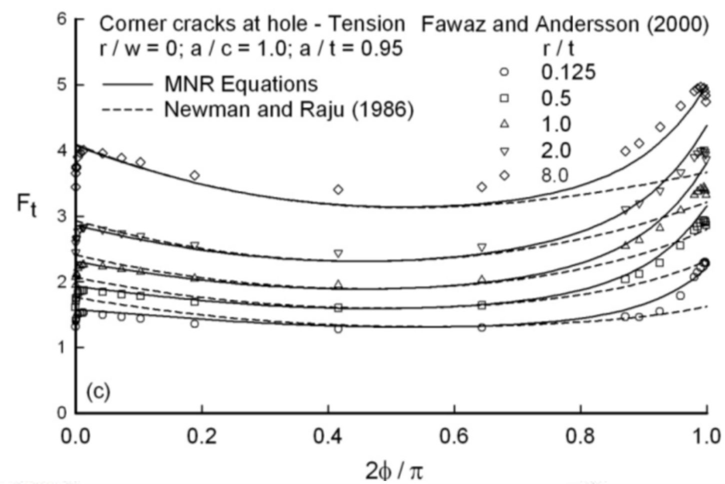
- AFGROW staff has successfully done that for smaller databases
- Newman&Raju type of equations, see recent paper by J. Newman, AIAA-SciTech2018-Raju,2018, 14 pp
- Attempts to train a Neural Network (M.S Ewing, S Fawaz, 2009)

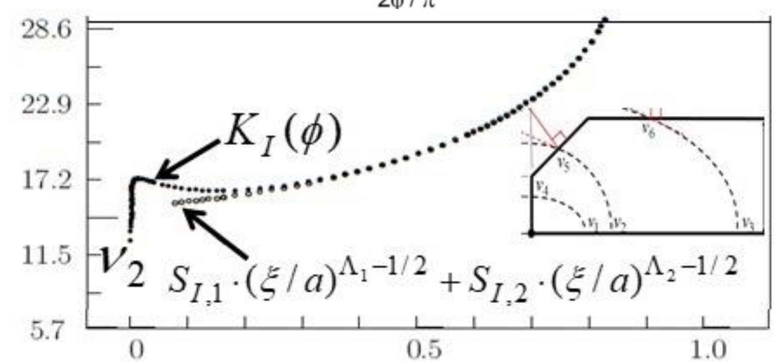
Present work:

- Asymptotic expansions

Way Forward?:

- Use deep learning strategies developed the last 5-10 years.





End

Acknowledgements

Part of this work was performed under subcontract to Sabreliner Aviation, LLC, under an USAF contract by the U.S. Air Force Academy's centre for aircraft structural life extension (CASTLE), FA7000-10-D-0014. This work could not have been accomplished without the support of the US DoD High Performance Computing Modernization Office.