Fatigue lifetime analysis of general 3D crack configurations using $\mathcal{H}$-matrix accelerated boundary element method

PhD student : <u>Adrien VET</u>[1,2]

Supervisors : Marc Bonnet[2], Luiz Faria[2], Raùl de Moura Pinho[1]

SAFRAN AIRCRAFT ENGINES

ENSTA IP PARIS

[1]Safran Aircraft Engines (Villaroche, France)

[2]POEMs, UMA, ENSTA (Palaiseau, France)

<u>Internal Mechanics Seminar at ENSTA – Paris : october 10th, 2025, Massy-Palaiseau</u>

Contents

I. Industriel context
II. General scheme for solving crack problem using fast BEM
III. Integration to fatigue lifespan analysis
IV. Numerical examples
V. Conclusion – PhD to be continued…

I. Industrial context

i. Introduction

ii. Established tools at Safran

iii. Goals of the PhD

iv. Link betwen lifespan and crack propagation

ARIZE Project – *Aeronautics Research and Industry new horiZons finite Elements software*

- Started in 2021
- Funded by the DGAC
- Aims to achieve the environmental objectives set by the European Commission and the French government through innovation
- Partnership between Safran, Onera, Armines, Mines Paris, and Transvalor

How does this thesis fit into this project?

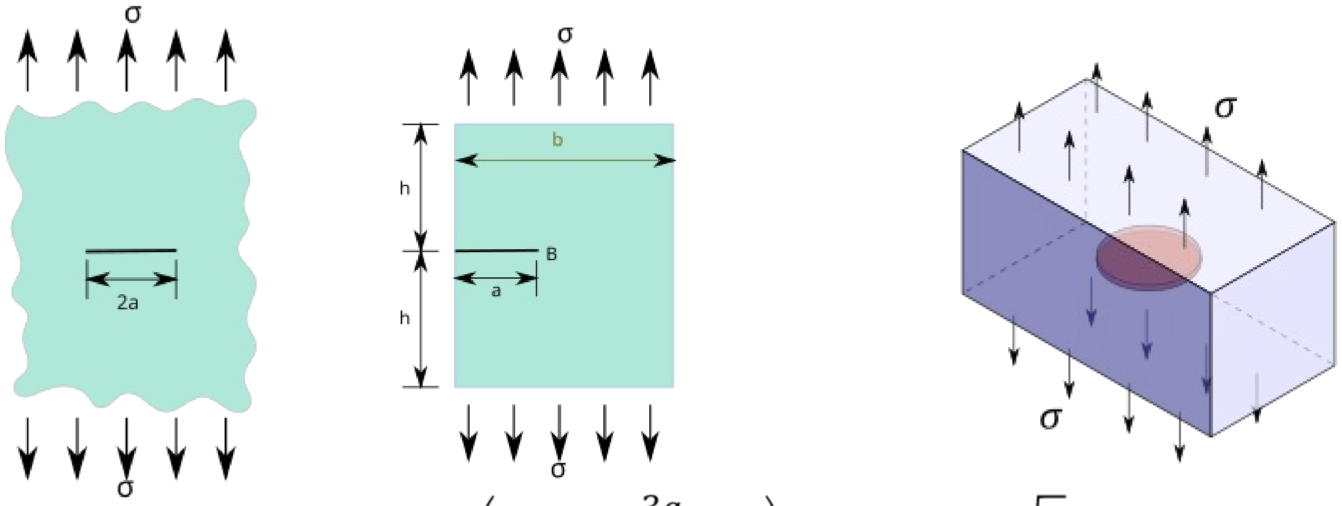
One focus of the ARIZE project involves **accelerating the computational resolution** of partial differential equations in mechanics, particularly in **fracture mechanics**. Safran's current methods are all based on **finite element methods**.

What crack problems models are used at SAE?

Existing Tools at SAE

1. Analytical tools
2. Semi-analytical tools
3. 3D crack problems with FEM
4. "2.5D" crack problems with FEM

1 Simple 1D, 2D, and 3D analytical solutions. Too canonical and therefore not applicable for industrial cases.



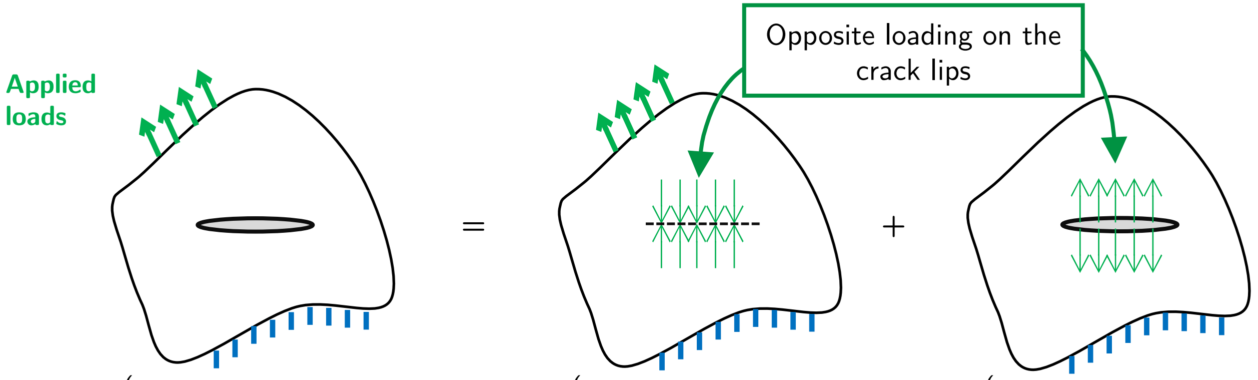
$K_I = \sigma\sqrt{\pi a}$ $\quad K_I = \sigma\sqrt{\pi a}\left(\dfrac{1+\frac{3a}{b}}{2\sqrt{\frac{\pi a}{b}}\left(1-\frac{a}{b}\right)^{3/2}}\right)$ $\quad K_I = \dfrac{\sigma\sqrt{\pi}}{E(k)}\sqrt{\dfrac{b}{a}}\left(\dfrac{a^4\sin^2\phi + b^4\cos^2\phi}{a^2\sin^2\phi + b^2\cos^2\phi}\right)^{1/4}$

What crack problems models are used at SAE?

Existing Tools at SAE

1. Analytical tools
2. Semi-analytical tools
3. 3D crack problems with FEM
4. "2.5D" crack problems with FEM

2 Bueckner's superposition principle combined with weight functions.
Disadvantages: over simplified geometry and does not take into account the impact of the presence of the crack on the structure's relaxation.



$$K_I(a) = K_I(b) + K_I(c)$$
$$K_I(a) = 0 + K_I(c)$$
$$K_I(a) = K_I(c)$$

$$\boxed{K_I(\boldsymbol{Q}) = \iint_{\text{Crack surface}} \underbrace{\boldsymbol{W_{QQ'}}(\boldsymbol{Q},\boldsymbol{Q'})}_{\text{Weight function}} .\sigma(\boldsymbol{Q'}).dS_{Q'}}$$

What crack problems models are used at SAE?

Existing Tools at SAE

1. Analytical tools
2. Semi-analytical tools
3. 3D crack problems with FEM
4. "2.5D" crack problems with FEM

3. Classical finite element method. Disadvantage: very long set up (up to several months), as well as for the calculation of crack propagation (up to a few weeks).

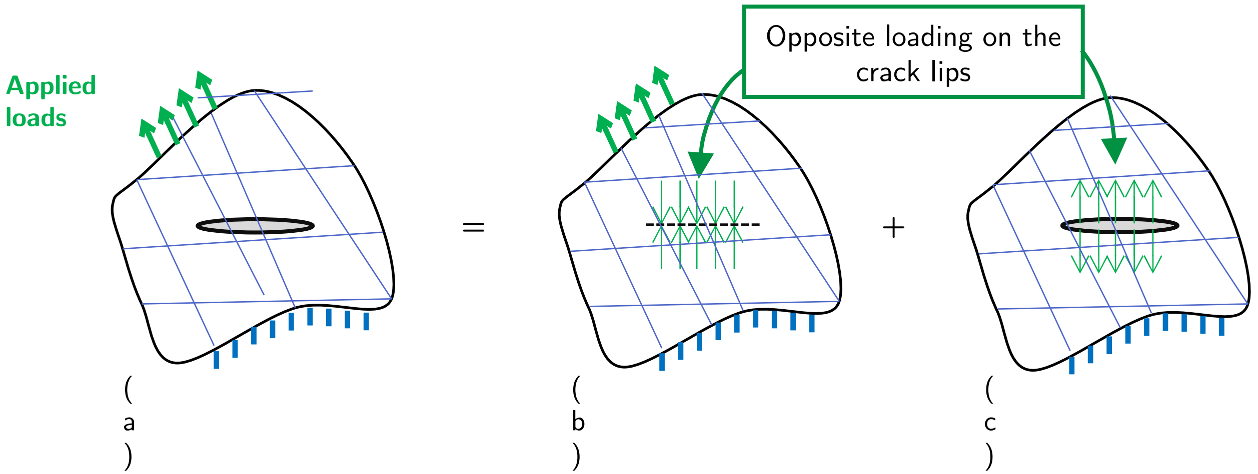
Supprimée pour confidentialité

What crack problems models are used at SAE?

Existing Tools at SAE

1. Analytical tools
2. Semi-analytical tools
3. 3D crack problems with FEM
4. "2.5D" crack problems with FEM

4 = 2 + 3. Use of the superposition principle as in 2 and finite element method as in 3.



The SIF is computed by **post-processing** FEM solution

And **my thesis**?

Existing Tools at SAE

1. Analytical tools
2. Semi-analytical tools
3. 3D crack problems with FEM
4. "2.5D" crack problems with FEM

My thesis: solving 3D crack problems (as 3 and 4) but using the **BEM** accepting several hypothesis.

Hypothesis of the framework

- LEFM (Linear Elastic Fracture Mechanics)
- Isotropic, homogeneous material

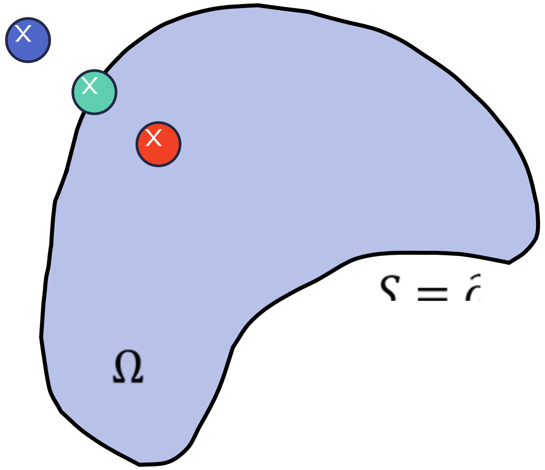
Final goal : speed up computation time for certain industrial study cases under the asumptions mentionned above.

II. General scheme for solving crack problem using fast BEM

i. Recall : Boundary Integral Equation in Elastostatics
ii. Displacement Discontinuity Method
iii. Numerical resolution by the Boundary Element Method
iv. Computation of (singular) integrals
v. Focus on enhanced Guiggiani's algorithm
vi. $\mathscr{H}-$ matrices compression

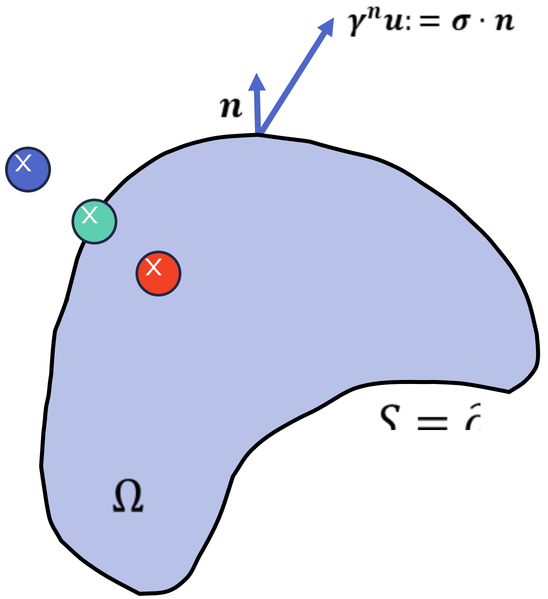
Boundary integral equation for elastic solids

$$\mathcal{S}\boldsymbol{t} - \mathcal{D}\boldsymbol{u} = \begin{cases} \boldsymbol{u} & \text{in } \Omega \\ \frac{1}{2}\boldsymbol{u} & \text{over } S \\ \boldsymbol{0} & \text{ow.} \end{cases} \qquad \mathcal{D}^*\boldsymbol{t} - \mathcal{H}\boldsymbol{u} = \begin{cases} \boldsymbol{t} & \text{in } \Omega \\ \frac{1}{2}\boldsymbol{t} & \text{over } S \\ \boldsymbol{0} & \text{ow.} \end{cases}$$



Boundary integral equation for elastic solids

$$\mathcal{S}\boldsymbol{t} - \mathcal{D}\boldsymbol{u} = \begin{cases} \boldsymbol{u} & \text{in } \Omega \\ \frac{1}{2}\boldsymbol{u}_{T[f](x\in S)=\int G(x,y)\cdot f(y)dT_y} & \\ \boldsymbol{0} & \text{ow.} \end{cases} \qquad \mathcal{D}^*\boldsymbol{t} - \mathcal{H}\boldsymbol{u} = \begin{cases} \boldsymbol{t} & \text{in } \Omega \\ \frac{1}{2}\boldsymbol{t} & \text{over } S \\ \boldsymbol{0} & \text{ow.} \end{cases}$$



$$\mathcal{S}\boldsymbol{\varphi}(\boldsymbol{x}) = \int_S \boldsymbol{G}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{\varphi}(\boldsymbol{y}) d\boldsymbol{T}_{\mathrm{y}}$$

$$\mathcal{D}\boldsymbol{\varphi}(\boldsymbol{x}) = \mathbf{p.v.} \int_S \left(\gamma_y^n \boldsymbol{G}(\boldsymbol{x},\boldsymbol{y})\right)^T \cdot \boldsymbol{f}(\boldsymbol{y}) d\boldsymbol{T}_{\mathrm{y}}$$

$$\mathcal{D}^*\boldsymbol{\varphi}(\boldsymbol{x}) = \mathbf{p.v.} \int_S \gamma_x^n \boldsymbol{G}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{f}(\boldsymbol{y}) d\boldsymbol{T}_{\mathrm{y}}$$

$$\mathcal{H}\boldsymbol{\varphi}(\boldsymbol{x}) = \mathbf{f.p.} \int_S \gamma_x^n \left(\gamma_y^n \boldsymbol{G}(\boldsymbol{x},\boldsymbol{y})\right)^T \cdot \boldsymbol{f}(\boldsymbol{y}) d\boldsymbol{T}_{\mathrm{y}}$$

Free space Green's fundamental solution:

$$\boldsymbol{G}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{16\pi\mu(1-\nu)r}\left\{(3-4\nu)\,\underline{\underline{\mathbf{I}}} + \hat{\boldsymbol{r}} \otimes \hat{\boldsymbol{r}}\right\}$$

Generalized normal derivative ("trace")

$$\gamma^n \boldsymbol{u} := \boldsymbol{\sigma} \cdot \boldsymbol{n} = (\lambda \mathrm{div}\boldsymbol{u})\boldsymbol{n} + \mu\left(\boldsymbol{\nabla u} + \boldsymbol{\nabla}^T \boldsymbol{u}\right) \cdot \boldsymbol{n}$$

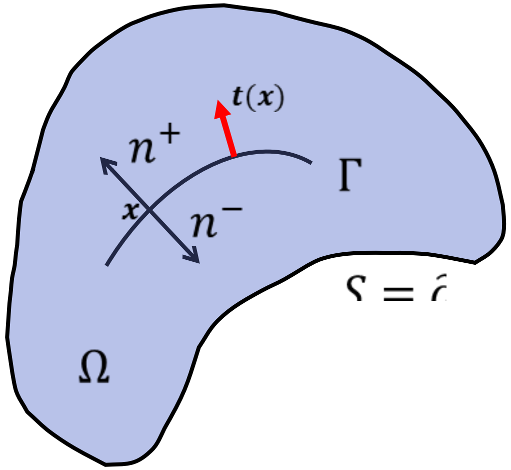
Equilibrium equation (without volumic forces) and Hooke's law :

$$div\boldsymbol{\sigma} = \vec{\boldsymbol{0}}, \qquad \sigma_{ij} = C_{ijk\ell}\, \partial u_k / \partial x_\ell$$

Extension to cracked solids...

Displacement discontinuity method

$$\begin{cases} \mathcal{H}_{\Gamma\Gamma}\boldsymbol{\phi}_\Gamma + \mathcal{D}^*_{\Gamma S}\boldsymbol{t}_S - \mathcal{H}_{\Gamma S}\boldsymbol{u}_S = -\boldsymbol{t} & \text{(on } \Gamma\text{)} \\ \mathcal{S}_{SS}\boldsymbol{t}_S - \mathcal{D}_{SS}\boldsymbol{u}_S + \mathcal{D}_{S\Gamma}\boldsymbol{\phi}_\Gamma = \frac{1}{2}\boldsymbol{u} & \text{(on S)} \end{cases}$$



... And its interior representation formulas.

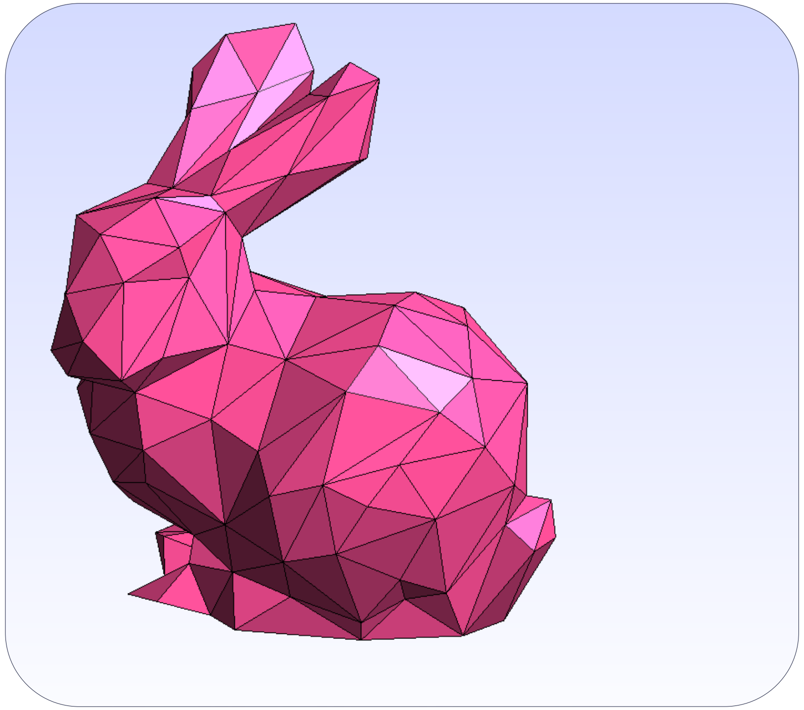
$$\underbrace{\boldsymbol{\phi} = \boldsymbol{u}^+ - \boldsymbol{u}^-}_{\text{Crack Opening Displacement}}$$

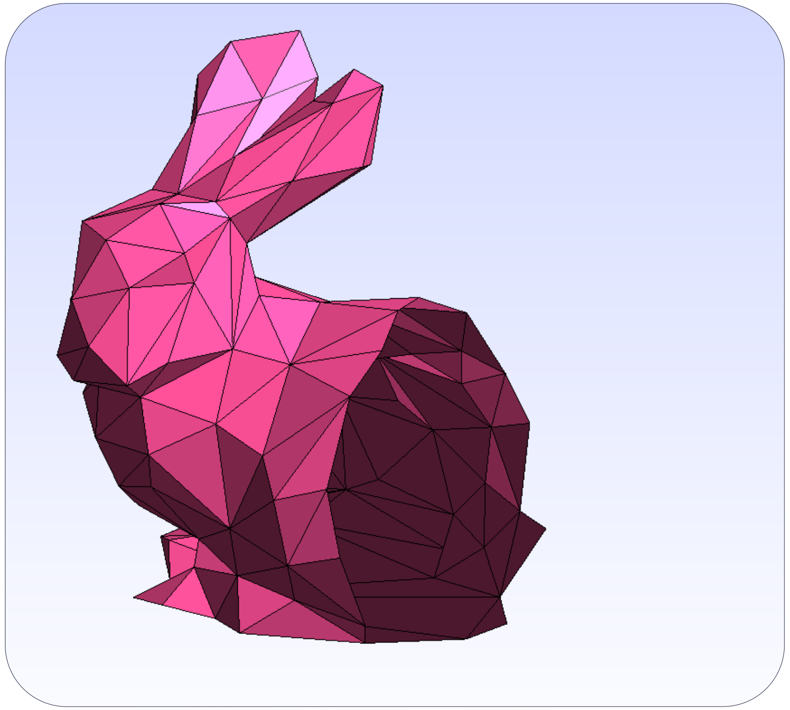
Interior representation

$$\left.\begin{aligned} \boldsymbol{u}_\Omega &= \mathcal{D}_{\Omega\Gamma}\boldsymbol{\phi}_\Gamma + \mathcal{D}_{\Omega\Gamma}\boldsymbol{u}_S \\ \boldsymbol{\sigma}_\Omega &= \boldsymbol{C} :: \left(\boldsymbol{\epsilon}(\mathcal{D}_{\Omega\Gamma}\boldsymbol{\phi}_\Gamma) + \boldsymbol{\epsilon}(\mathcal{D}_{\Omega\Gamma}\boldsymbol{u}_S)\right) \end{aligned}\right\} \text{(on } \Omega\text{)}$$

Main caracteristics of the BEM

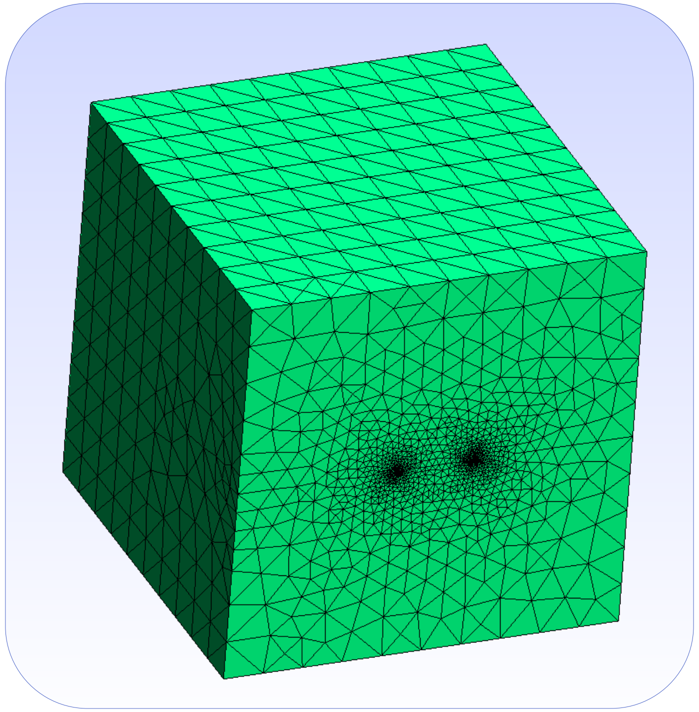
- **Only the boundary** (3D surface – including the crack) is meshed : purely 2D éléments in 3D framework
- We describe each 2D element by a reference element and its polynomial interpolant (**like in the FEM**)
- The **numerical difficulty** occurs when integrating singular kernels
- **Nyström method** : the quadrature nodes encode the unknown discrete values

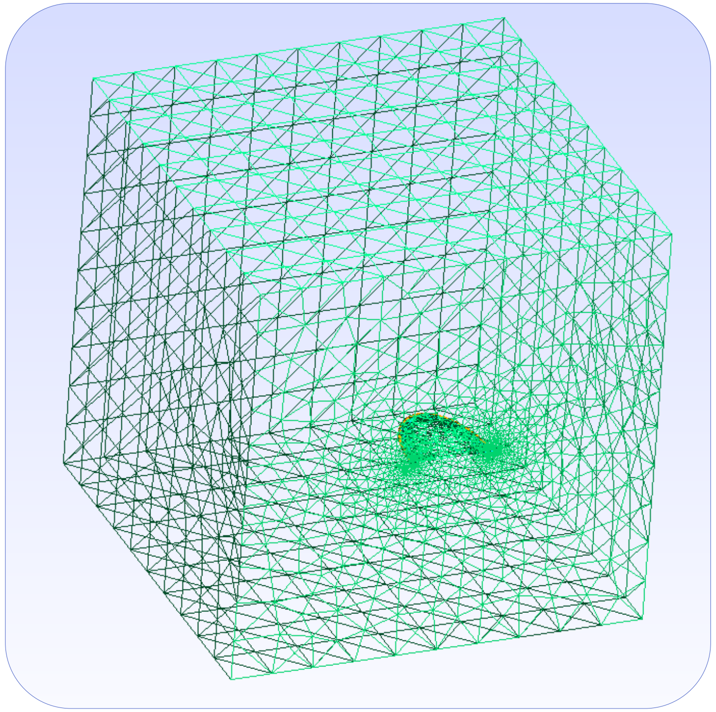




Main caracteristics of the BEM

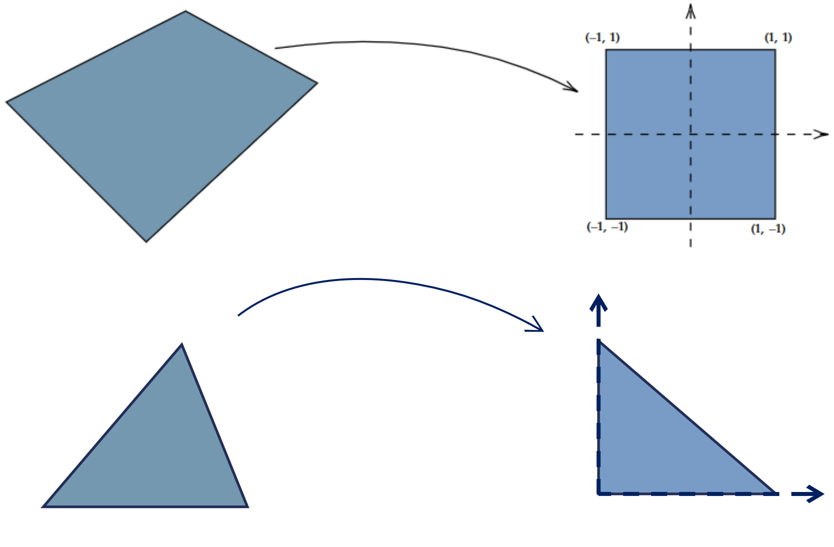
- **Only the boundary** (3D surface – including the crack) is meshed : purely 2D éléments in 3D framework
- We describe each 2D element by a reference element and its polynomial interpolant (**like in the FEM**)
- The **numerical difficulty** occurs when integrating singular kernels
- **Nyström method** : the quadrature nodes encode the unknown discrete values





Physical element

Reference element



Elementary integration on a reference element τ_e

1) $\boldsymbol{x}$ is far enough from τ_e (**regular integration** – Gauss-Legendre quadrature)
2) $\boldsymbol{x}$ is on τ_e (**singular integration**) → Enhanced Guiggiani's algorithm
3) Intermediate case : $\boldsymbol{x}$ is close to τ_e (**nearly-singular integration**) but not in

(un)known function
singular kernel

$$\mathcal{K}[\boldsymbol{\varphi}](\boldsymbol{x}) = \int_{\tau_e} \mathbf{K}(\boldsymbol{x},\boldsymbol{y}) \cdot \boldsymbol{\varphi}(\boldsymbol{y})\mathrm{dS}_{\mathbf{y}}$$

source point
integration element

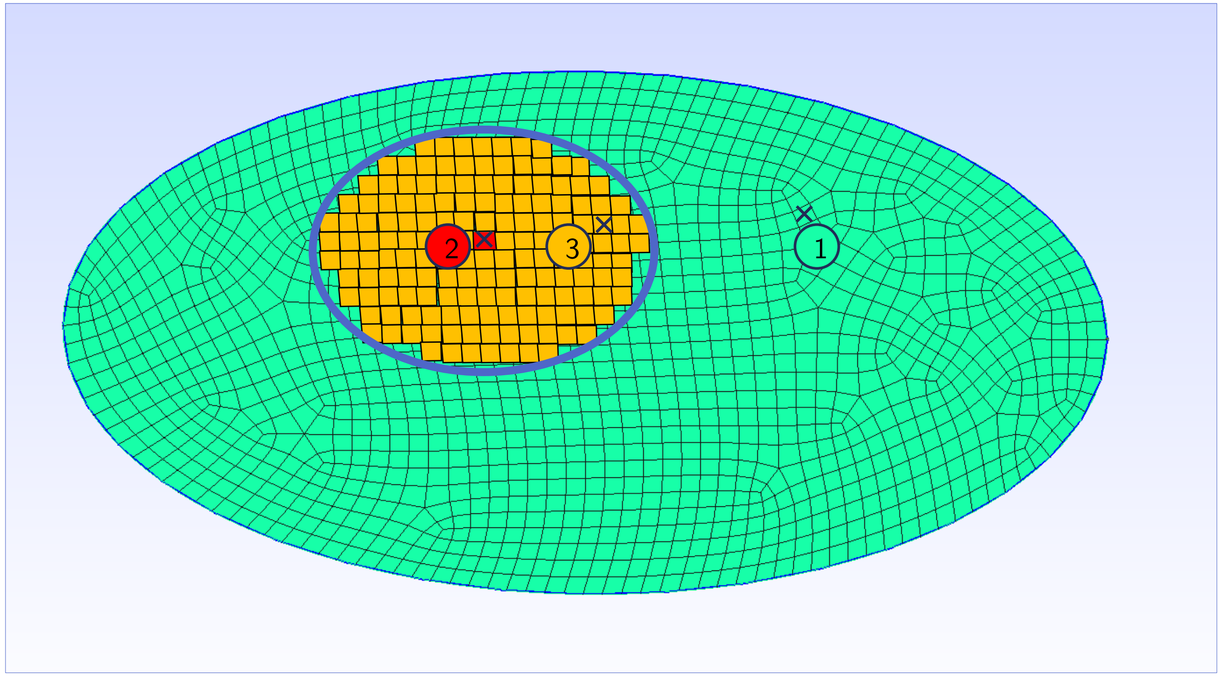


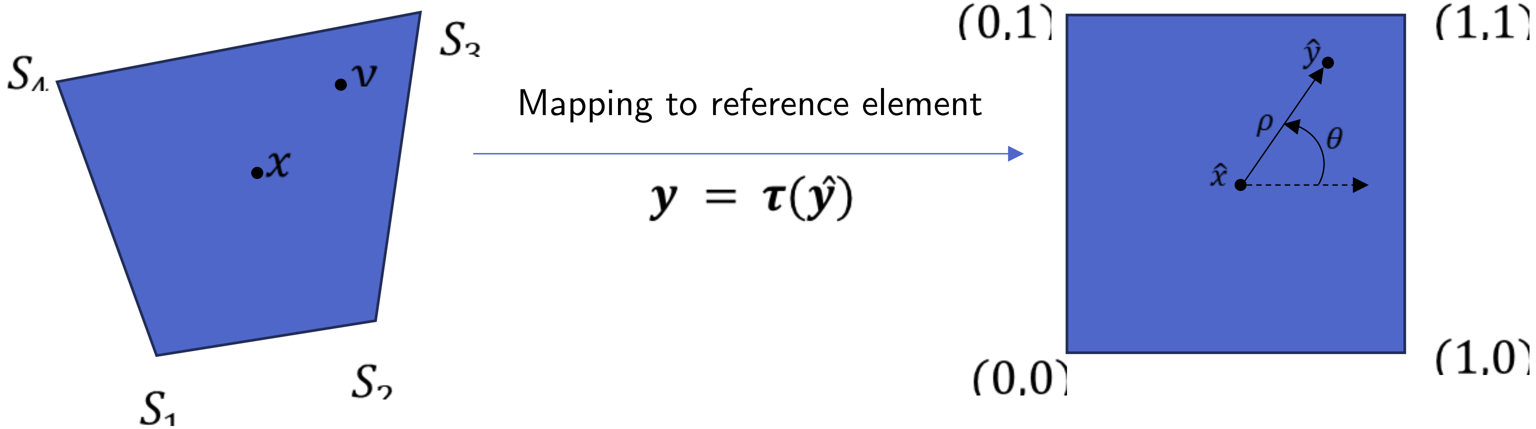
Illustration on an elliptic crack surface

- Integration element
- Source point
- Singular case
- Nearly-singular case
- Regular case

Guiggiani (1992) direct algorithm for computing singular elementary integrals

Core idea

$$K(\hat{x},\hat{y}) := \rho J(\hat{y}) N(\hat{y}) K(x,y) = \underbrace{\left\{K(\rho,\theta) - \frac{K_{-2}(\theta)}{\rho^2} - \frac{K_{-1}(\theta)}{\rho}\right\}}_{\text{Regularized part} \,=\, \mathcal{O}(1)} + \underbrace{\left\{\frac{K_{-2}(\theta)}{\rho^2} + \frac{K_{-1}(\theta)}{\rho}\right\}}_{\text{Remainder, analytically handled}}$$



$$dS_y = J(\hat{y})\rho d\rho d\theta \qquad\qquad dS_y = d\hat{y}_1 d\hat{y}_2 = \rho d\rho d\theta$$

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Laurent coefficient	Laplace $\Delta u + f = 0$
Original hypersingular kernel	$\frac{1}{4\pi r^3}\left\{\left(\underline{\underline{I}} - 3\hat{r}\otimes\hat{r}\right)\cdot \boldsymbol{n}(\boldsymbol{x})\right\}\cdot \boldsymbol{n}(\boldsymbol{y})$
$K_{-2}(\theta)$	$\frac{J(\eta)N(\eta)}{4\pi\lVert\boldsymbol{J}(\boldsymbol{\eta})\cdot\boldsymbol{u}(\boldsymbol{\theta})\rVert}$
$K_{-1}(\theta)$	$\frac{-1}{4\pi\left\lVert\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right\rVert^3}\Bigg\{\frac{3\left(\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right)\cdot\underline{\underline{\nabla}}\left(\underline{\nabla}\tau\right)(\eta):\left(\mathbf{u}(\theta)\otimes\mathbf{u}(\theta)\right)}{2\left\lVert\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right\rVert^2}J(\eta)N(\eta) + \frac{3\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)}{\left\lVert\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right\rVert^2}\Bigg\{\Bigg\{\Bigg(\Big(J(\eta)\mathbf{n}(\eta)\frac{1}{2}\underline{\underline{\nabla}}\left(\underline{\nabla}\tau\right)(\eta):\left(\mathbf{u}(\theta)\otimes\mathbf{u}(\theta)\right)\Big)\cdot\mathbf{n}(\eta) + \left(\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right)\cdot\left(\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right)\Bigg) - \underline{\nabla}(J\mathbf{n})(\eta)\cdot\mathbf{u}(\theta)\Bigg\}N(\eta) - J(\eta)\mathbf{n}(\eta)\left(\underline{\nabla}N(\eta)\cdot\mathbf{u}(\theta)\right)\Bigg\}\Bigg\}$

$\Bigg\}$ Direct approach, described in Guiggiani's original article

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Laurent coefficient	Laplace $\Delta u + f = 0$
Original hypersingular kernel	$\frac{1}{4\pi r^3}\left\{\left(\underline{\underline{I}} - 3\hat{r}\otimes\hat{r}\right)\cdot \boldsymbol{n}(\boldsymbol{x})\right\}\cdot \boldsymbol{n}(\boldsymbol{y})$
$K_{-2}(\theta)$	$\frac{J(\eta)N(\eta)}{4\pi\lVert \boldsymbol{J}(\boldsymbol{\eta})\cdot \boldsymbol{u}(\boldsymbol{\theta})\rVert}$
$K_{-1}(\theta)$	$\frac{-1}{4\pi\left\lVert\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right\rVert^3}\left\{\frac{3\left(\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right)\cdot\underline{\underline{\nabla}}\left(\underline{\nabla}\tau\right)(\eta):\left(\mathbf{u}(\theta)\otimes\mathbf{u}(\theta)\right)}{2\left\lVert\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right\rVert^2}J(\eta)N(\eta) + \frac{3\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)}{\left\lVert\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right\rVert^2}\left\{\left\{\left(\left(J(\eta)\mathbf{n}(\eta)\frac{1}{2}\underline{\underline{\nabla}}\left(\underline{\nabla}\tau\right)(\eta):\left(\mathbf{u}(\theta)\otimes\mathbf{u}(\theta)\right)\right)\cdot\mathbf{n}(\eta) + \left(\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right)\cdot\left(\underline{\nabla}\tau(\eta)\cdot\mathbf{u}(\theta)\right)\right) - \underline{\nabla}(J\mathbf{n})(\eta)\cdot\mathbf{u}(\theta)\right\}N(\eta) - J(\eta)\mathbf{n}(\eta)\left(\underline{\nabla}N(\eta)\cdot\mathbf{u}(\theta)\right)\right\}\right\}$

Direct approach, described in Guiggiani's original article

Hypersingular elastostatics kernel :

$$\gamma_{1,x}\left(\gamma_{1,y}\boldsymbol{G}(\mathbf{x},\mathbf{y})\right)^T = \frac{\mu}{4\pi(1-\nu)r^3}\Big[3\hat{\boldsymbol{r}}\cdot\boldsymbol{n}_y\Big\{(1-2\nu)\boldsymbol{n}_x\otimes\hat{\boldsymbol{r}} + \nu\Big((\hat{\boldsymbol{r}}\cdot\boldsymbol{n}_x)\mathbf{I} + \hat{\boldsymbol{r}}\otimes\boldsymbol{n}_x\Big) - 5(\hat{\boldsymbol{r}}\cdot\boldsymbol{n}_x)\hat{\boldsymbol{r}}\otimes\hat{\boldsymbol{r}}\Big\}$$
$$+ 3\nu\Big\{(\hat{\boldsymbol{r}}\cdot\boldsymbol{n}_x)\boldsymbol{n}_y\otimes\hat{\boldsymbol{r}} + (\boldsymbol{n}_x\cdot\boldsymbol{n}_y)\hat{\boldsymbol{r}}\otimes\hat{\boldsymbol{r}}\Big\} + (1-2\nu)\Big\{3(\hat{\boldsymbol{r}}\cdot\boldsymbol{n}_x)\hat{\boldsymbol{r}}\otimes\boldsymbol{n}_y + (\boldsymbol{n}_x\cdot\boldsymbol{n}_y)\mathbf{I}$$
$$+ \boldsymbol{n}_y\otimes\hat{\boldsymbol{r}}\Big\} - (1-4\nu)\boldsymbol{n}_x\otimes\boldsymbol{n}_y\Big] \tag{1.15}$$

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Improvement of the Guiggiani's direct approach from a numerical aspect

$$K_{-2}(\theta) = \lim_{\rho \to 0} K(\rho, \theta)\rho^2 \quad (1)$$

$$K_{-1}(\theta) = \lim_{\rho \to 0} \left\{ \rho K(\rho, \theta) - \frac{K_{-2}(\theta)}{\rho} \right\} \quad (2)$$

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Improvement of the Guiggiani's direct approach from a numerical aspect

$$K_{-2}(\theta) = \lim_{\rho \to 0} K(\rho, \theta)\rho^2 \quad (1)$$

$$K_{-1}(\theta) = \lim_{\rho \to 0} \left\{ \rho K(\rho, \theta) - \frac{K_{-2}(\theta)}{\rho} \right\} \quad (2)$$

Hybrid approach (direct / Richardson) for the Guiggiani's method :

1 $$K_{-2}(\theta) = \frac{1}{A^3(\theta)} \underbrace{\hat{K}(\hat{\boldsymbol{A}}(\theta))}_{\text{regular part of the kernel}} N(\eta)J(\eta), \qquad A(\theta) = \left\| \boldsymbol{D\tau}(\boldsymbol{\eta}) \cdot \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix} \right\|, \qquad \hat{\boldsymbol{A}}(\theta) = \boldsymbol{D\tau}(\boldsymbol{\eta}) \cdot \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix} / A(\theta)$$

Richardson extrapolation | Guiggiani algorithm

Main goal : get the expression of the Laurent coefficients $K_{-1}(\theta)$ and $K_{-2}(\theta)$

Improvement of the Guiggiani's direct approach from a numerical aspect

$$K_{-2}(\theta) = \lim_{\rho\to 0} K(\rho,\theta)\rho^2 \qquad (1)$$

$$K_{-1}(\theta) = \lim_{\rho\to 0}\left\{\rho K(\rho,\theta) - \frac{K_{-2}(\theta)}{\rho}\right\} \qquad (2)$$

Hybrid approach (direct / Richardson) for the Guiggiani's method :

1

$$K_{-2}(\theta) = \frac{1}{A^3(\theta)}\underbrace{\hat{K}(\hat{\boldsymbol{A}}(\theta))}_{\text{regular part of the kernel}}N(\eta)J(\eta), \qquad A(\theta) = \left\| \boldsymbol{D\tau}(\boldsymbol{\eta})\cdot\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}\right\|, \qquad \hat{\boldsymbol{A}}(\theta) = \boldsymbol{D\tau}(\boldsymbol{\eta})\cdot\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix}/A(\theta)$$

2

$$(\boldsymbol{a})\; K_{-1}(\theta) = \rho^{-1}[\rho^2 K(\rho,\theta) - K_{-2}(\theta)] + \underset{\rho\to 0}{\mathcal{O}}(\rho)$$

$$(\boldsymbol{b})\; K_{-1}(\theta) = (t\rho)^{-1}[(t\rho)^2 K(t\rho,\theta) - K_{-2}(\theta)] + \underset{\rho\to 0}{\mathcal{O}}(\rho)$$

same with $t\rho$ instead of ρ

$$(\boldsymbol{b}) - t(\boldsymbol{a}) \Longrightarrow \boxed{K_{-1}(\theta) = \underbrace{\frac{1}{1-t}\{t\rho(K(t\rho,\theta) - \rho K(\rho,\theta)) + (t-t^{-1})K_{-2}(\theta)\}}_{\tilde{K}_{-1}(\theta)} + \underset{\rho\to 0}{\mathcal{O}}(\rho^2)}$$

and so on...

Richardson extrapolation | Guiggiani algorithm

Problem

BEM matrices are **dense**, **non-symetrical** and **non-definite-positive**

	+	-
BEM	Surface mesh, very less dofs	Dense matrices, non-symetrical
FEM	Sparse matrices, symetrical and semi-definite-positive	Huge number of dofs

- **High storage** requirement $\mathcal{O}(n^2)$
- Long **assembling** $\mathcal{O}(n^2)$ and **solving**

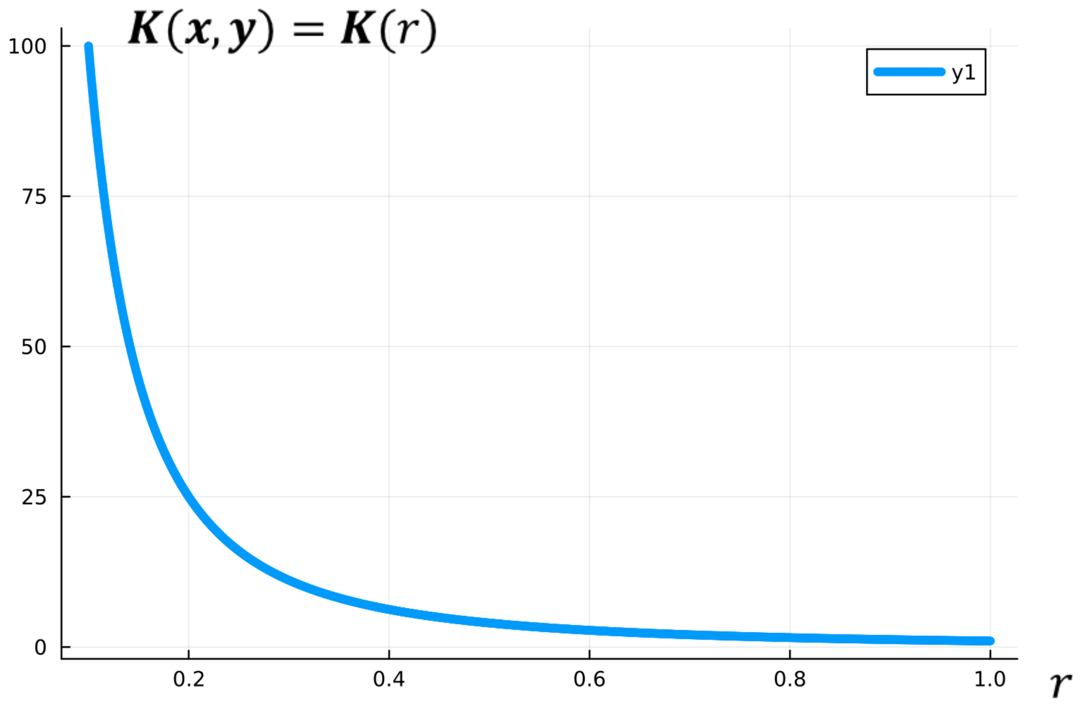
Asymptotically Smooth kernels → Clustering → PACA algorithm

Asymptotically Smooth kernels → Clustering → PACA algorithm

$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{y}) = \underset{r\to+\infty}{O}\left(\frac{1}{r^s}\right), \qquad s \in \{1,2,3\}$$

$$\forall \alpha,\beta \left|\frac{\partial^{|\alpha|+|\beta|}\boldsymbol{K}(\boldsymbol{x},\boldsymbol{y})}{\partial x_\alpha \partial y_\beta}\right| \leq (|\alpha|+|\beta|)C\|\boldsymbol{x}-\boldsymbol{y}\|^{-(|\alpha|+|\beta|+s)}$$

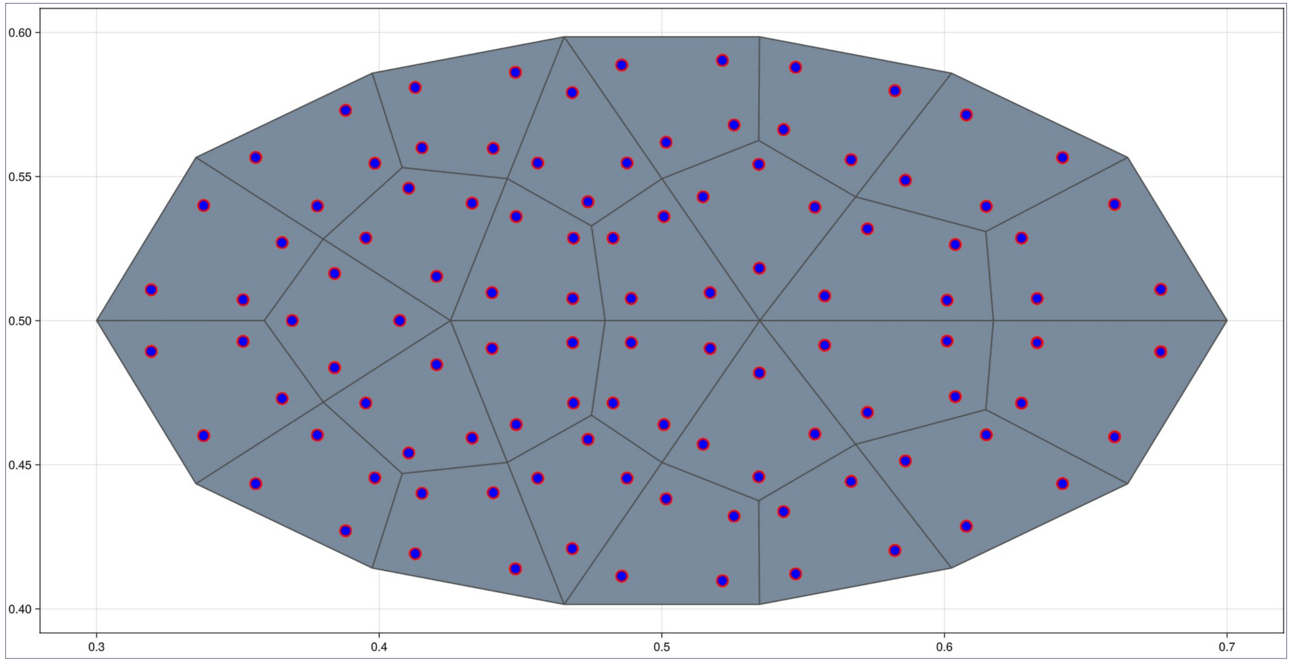
Smoothness condition



Asymptotically Smooth kernels

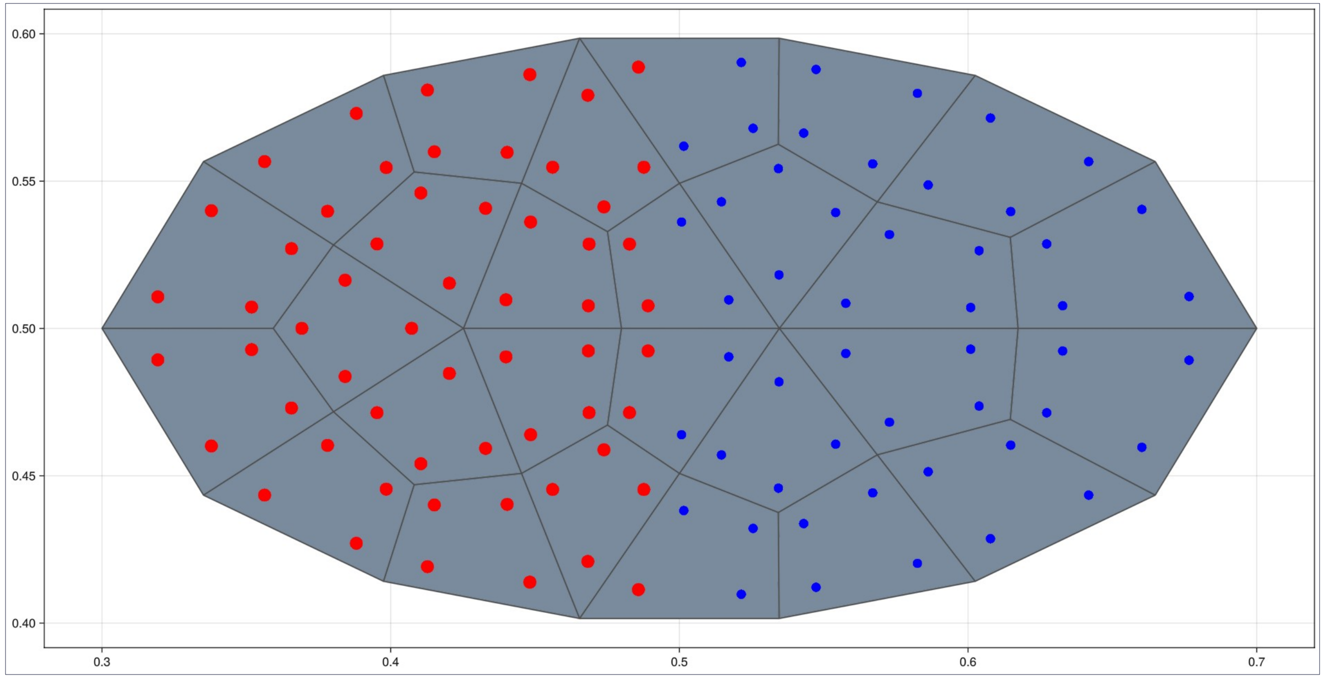
Clustering

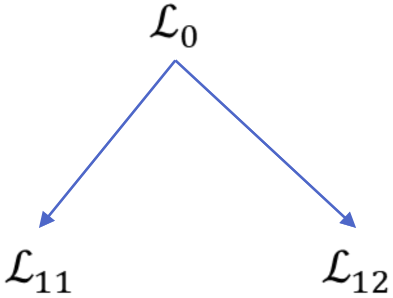
PACA algorithm



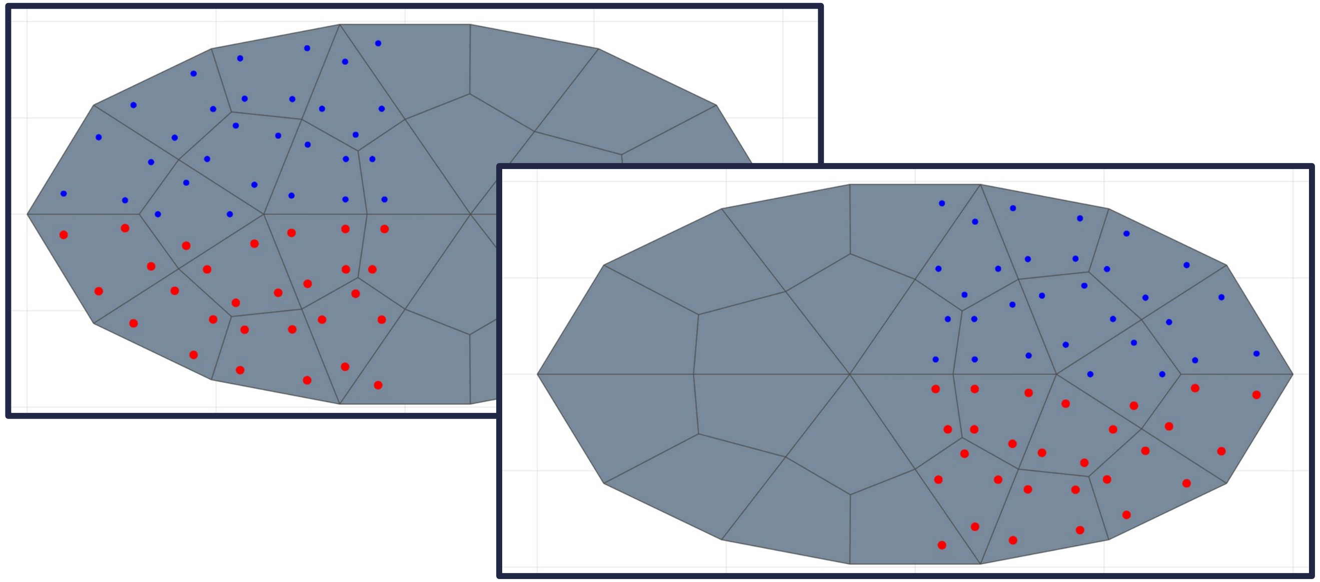
$\mathcal{L}_0$

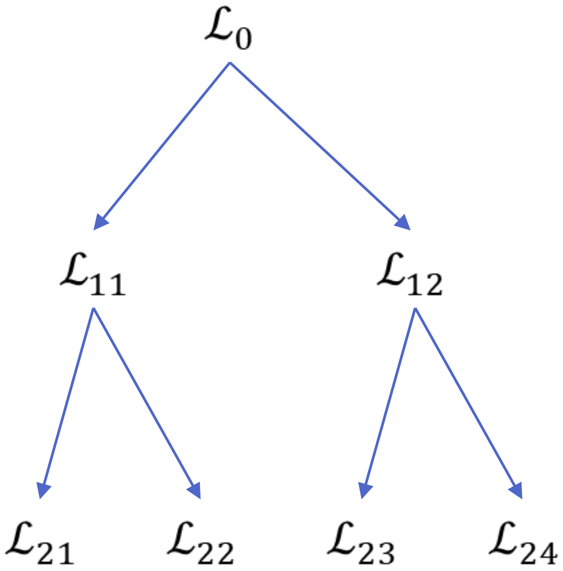
Asymptotically Smooth kernels → **Clustering** → PACA algorithm



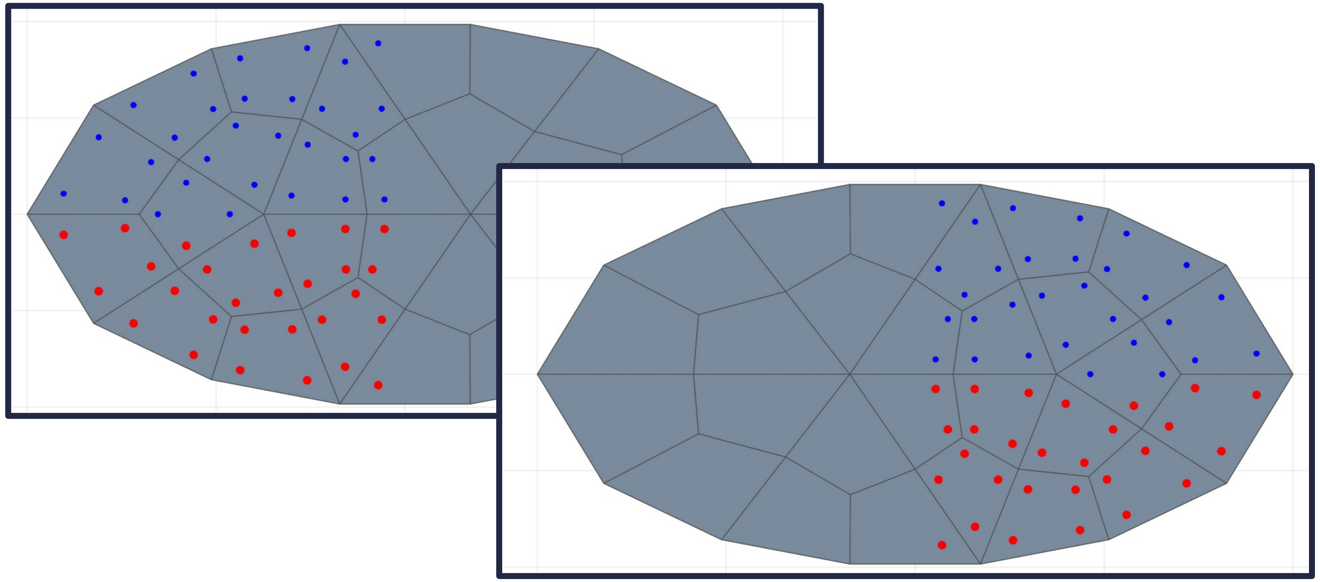


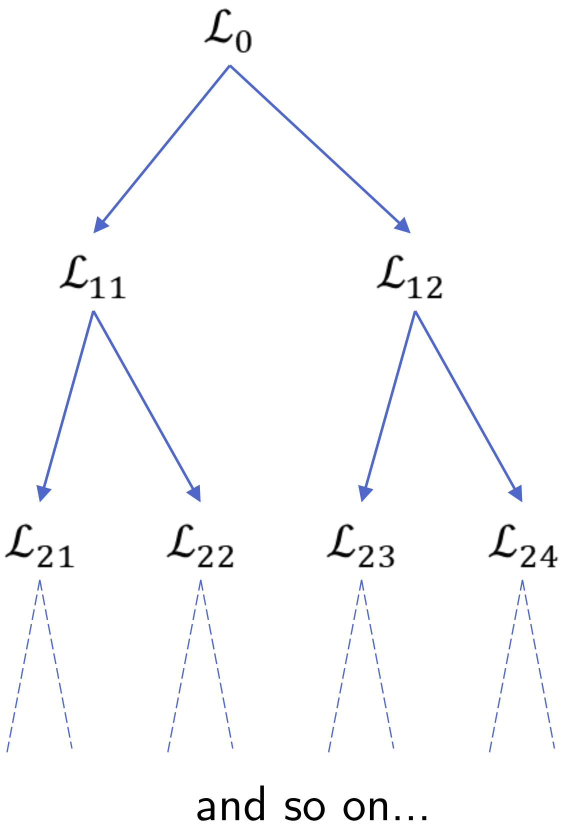
Asymptotically Smooth kernels → Clustering → PACA algorithm

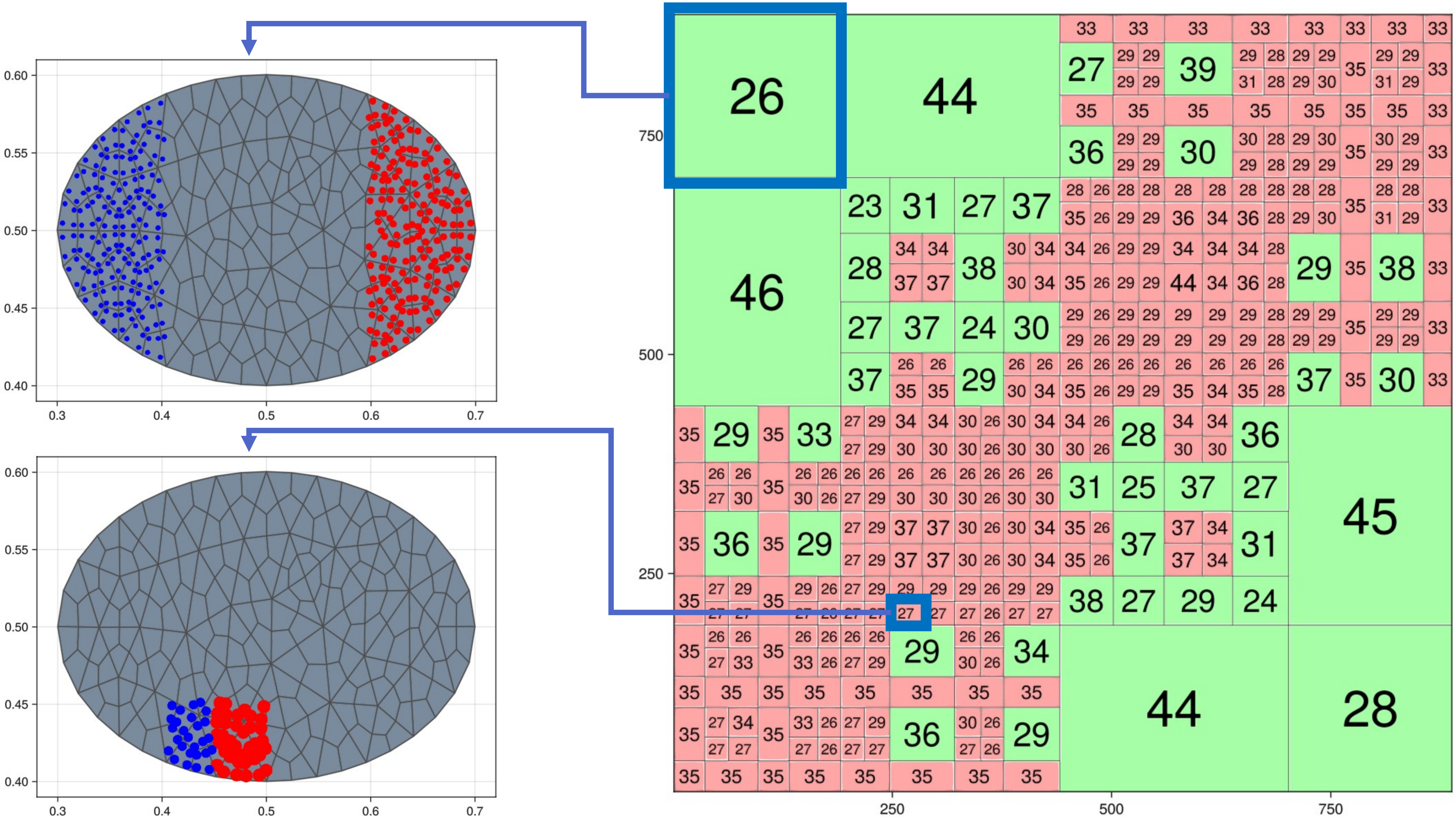




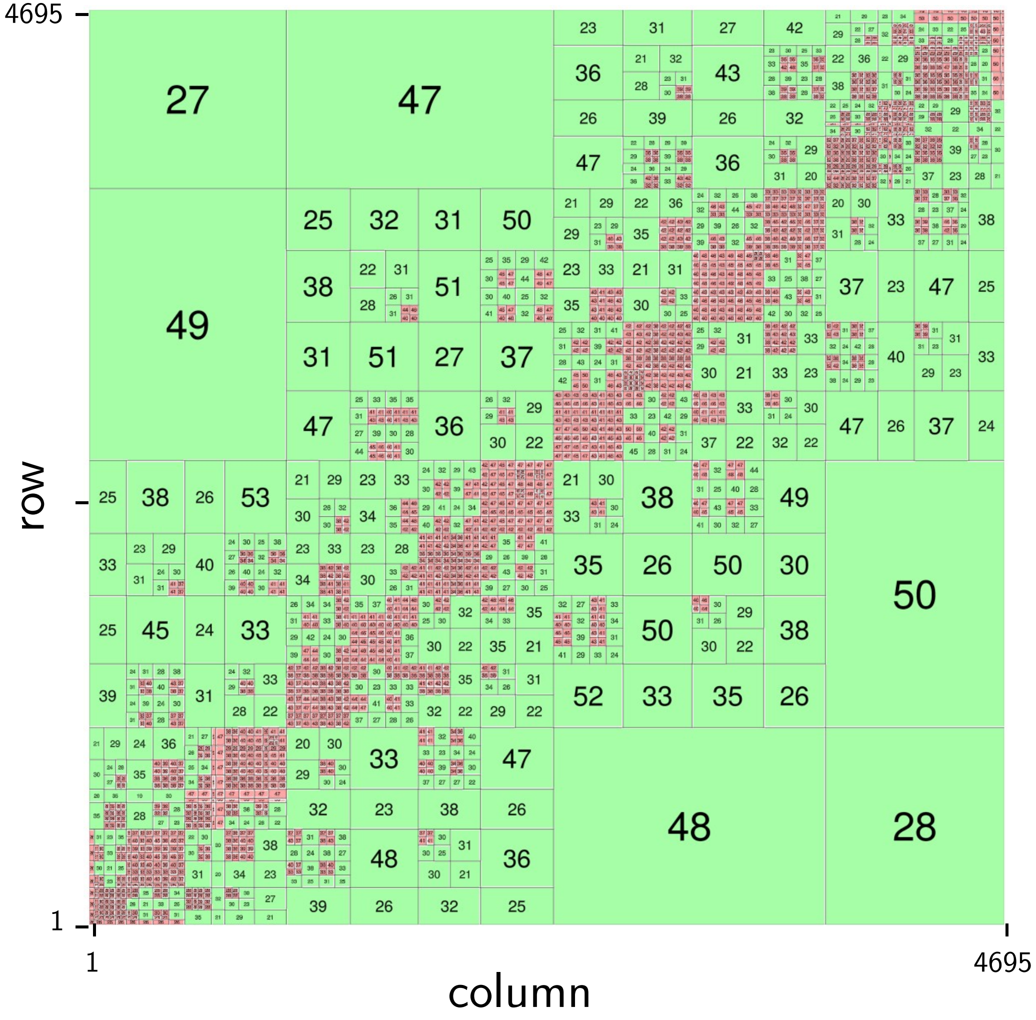
Asymptotically Smooth kernels → **Clustering** → PACA algorithm



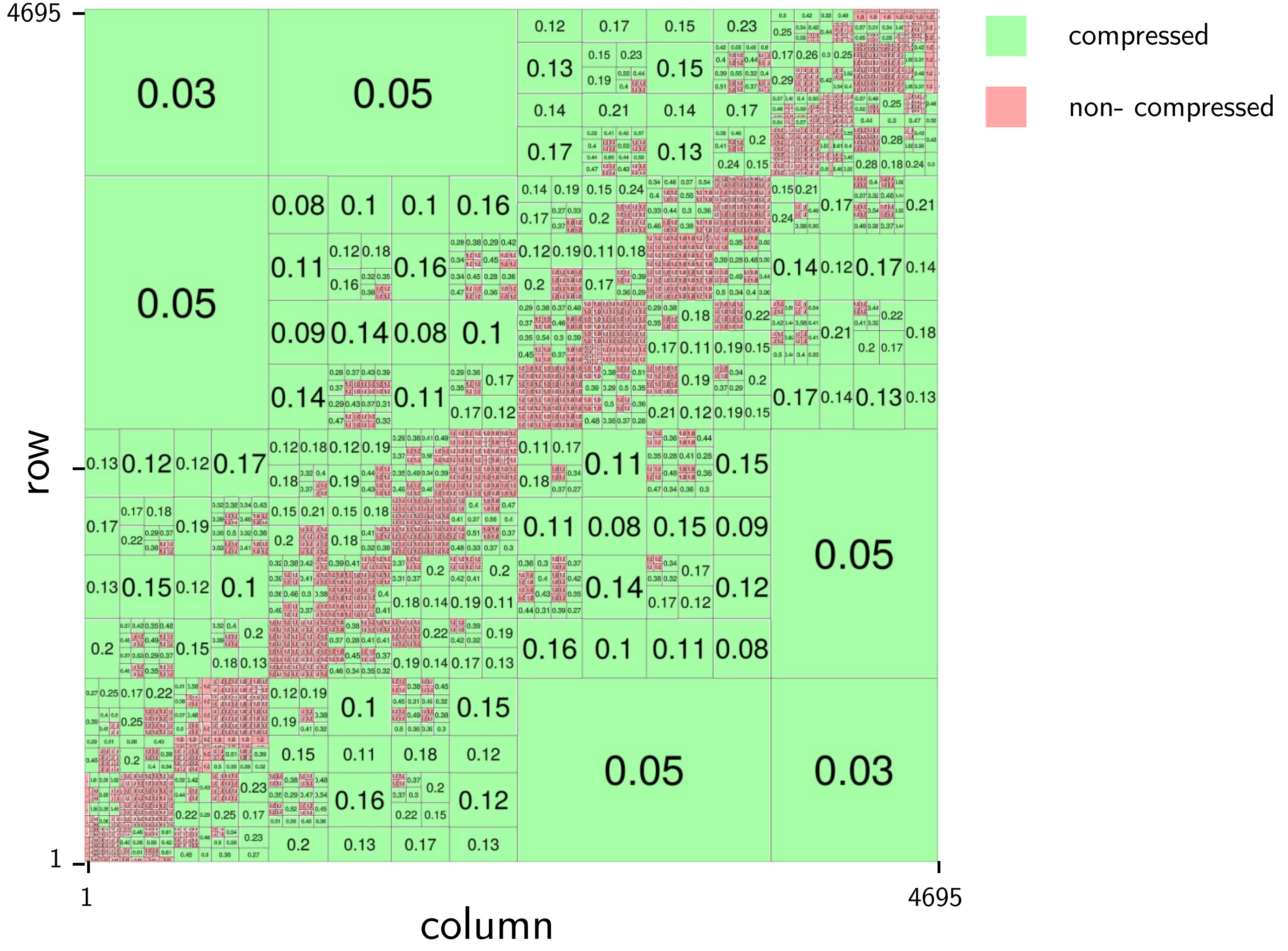



0.60
0.55
0.50
0.45
0.40
0.3
0.4
0.5
0.6
0.7
750
500
250
250
500
750

Absolute rank of all $\mathcal{H}$ −Matrices sub-blocks



Relative rank of all $\mathcal{H}$ −Matrices sub-blocks



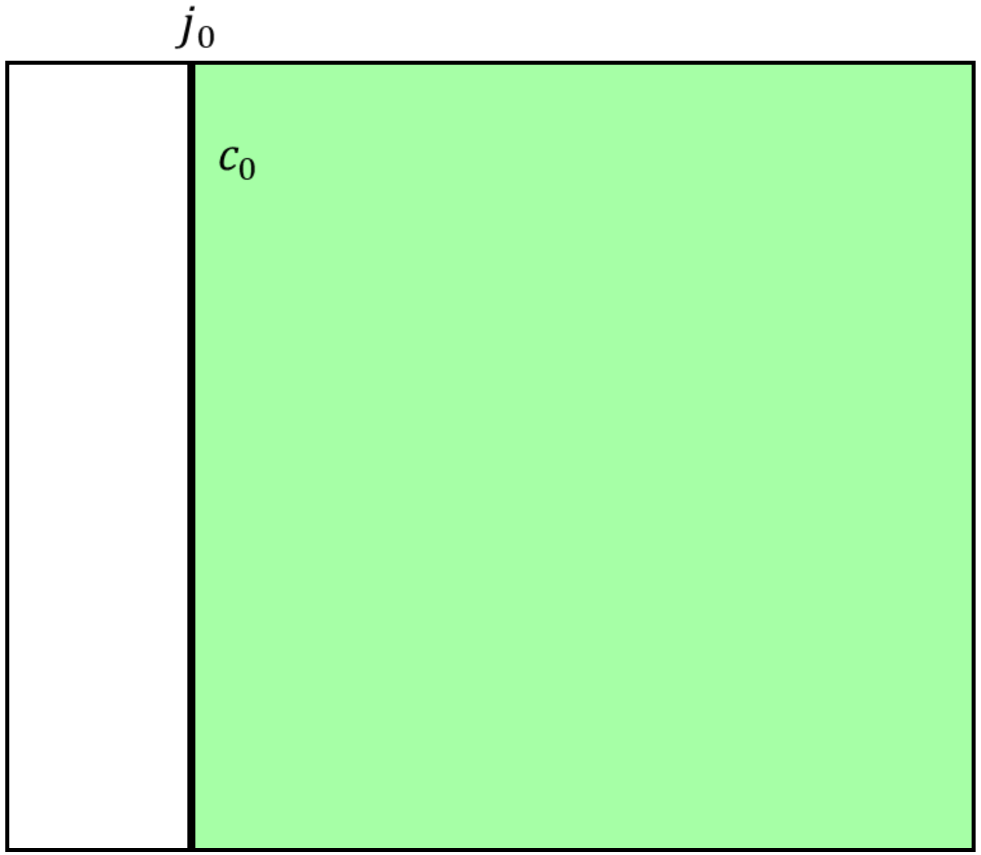
Asymptotically Smooth kernels

Clustering

PACA algorithm

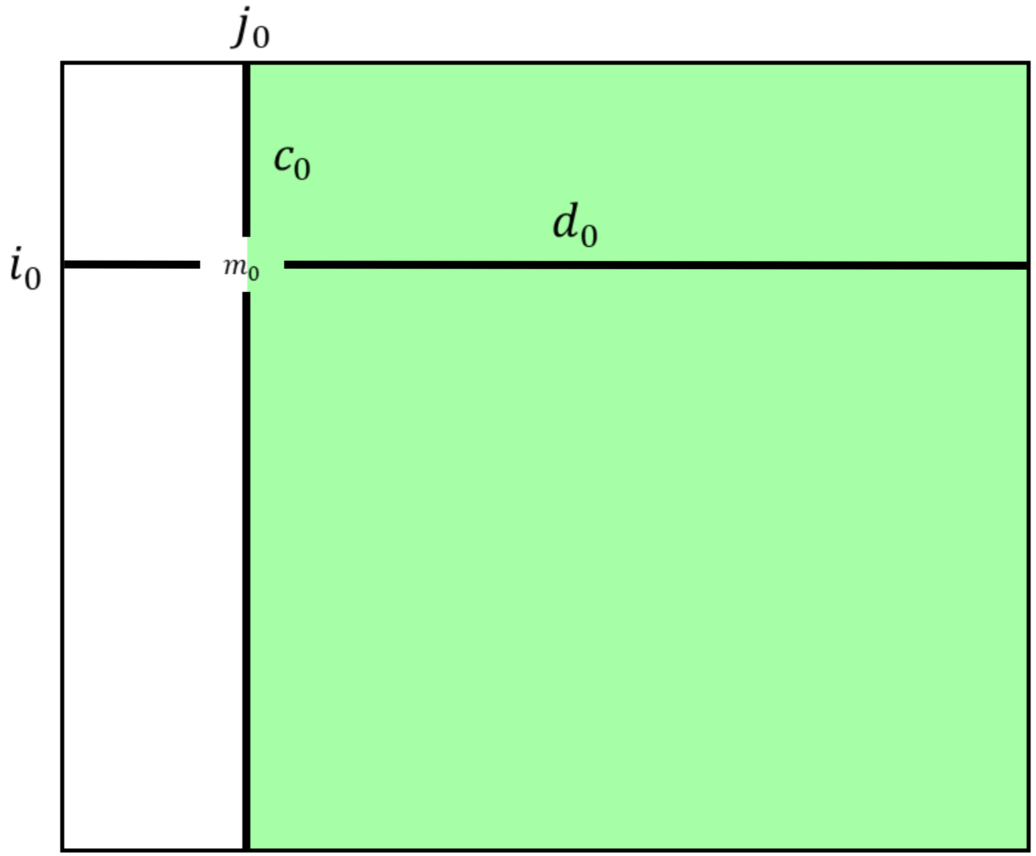
Admissible block B

Asymptotically Smooth kernels → Clustering → **PACA algorithm**



Admissible block B

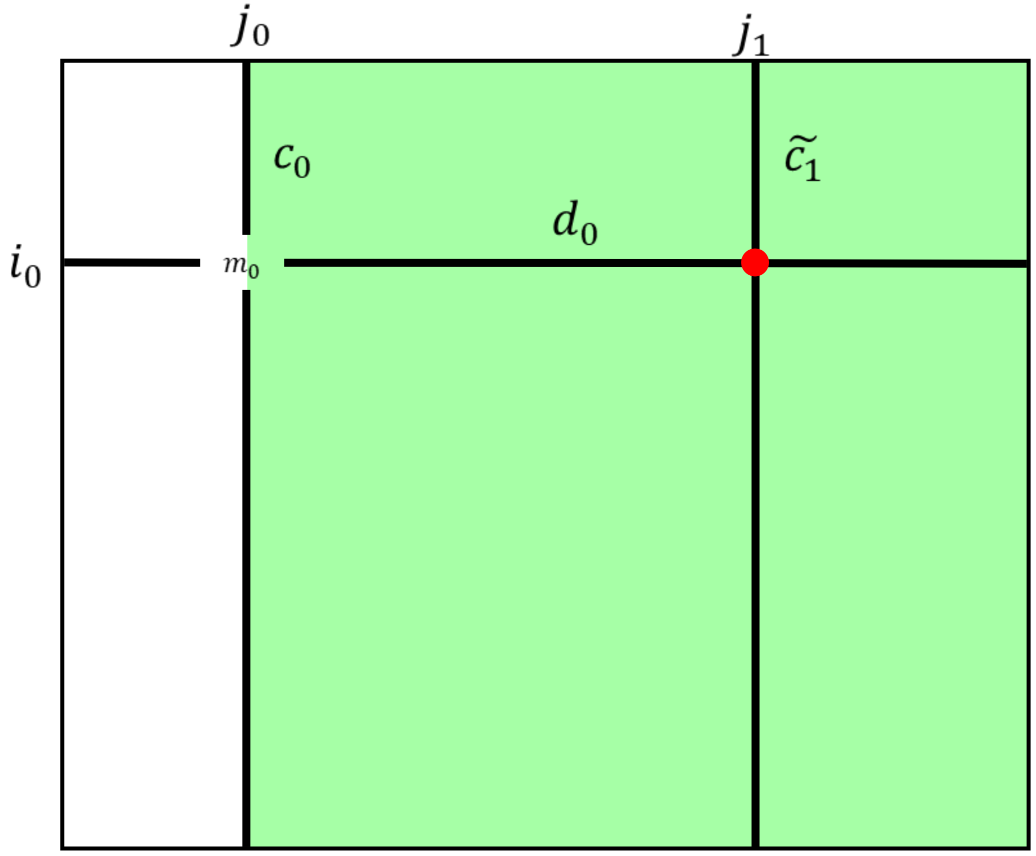
Asymptotically Smooth kernels → Clustering → PACA algorithm



Admissible block B

$$R_0 = \frac{1}{m_0} c_0 d_0^T \longrightarrow \text{rank } 1$$

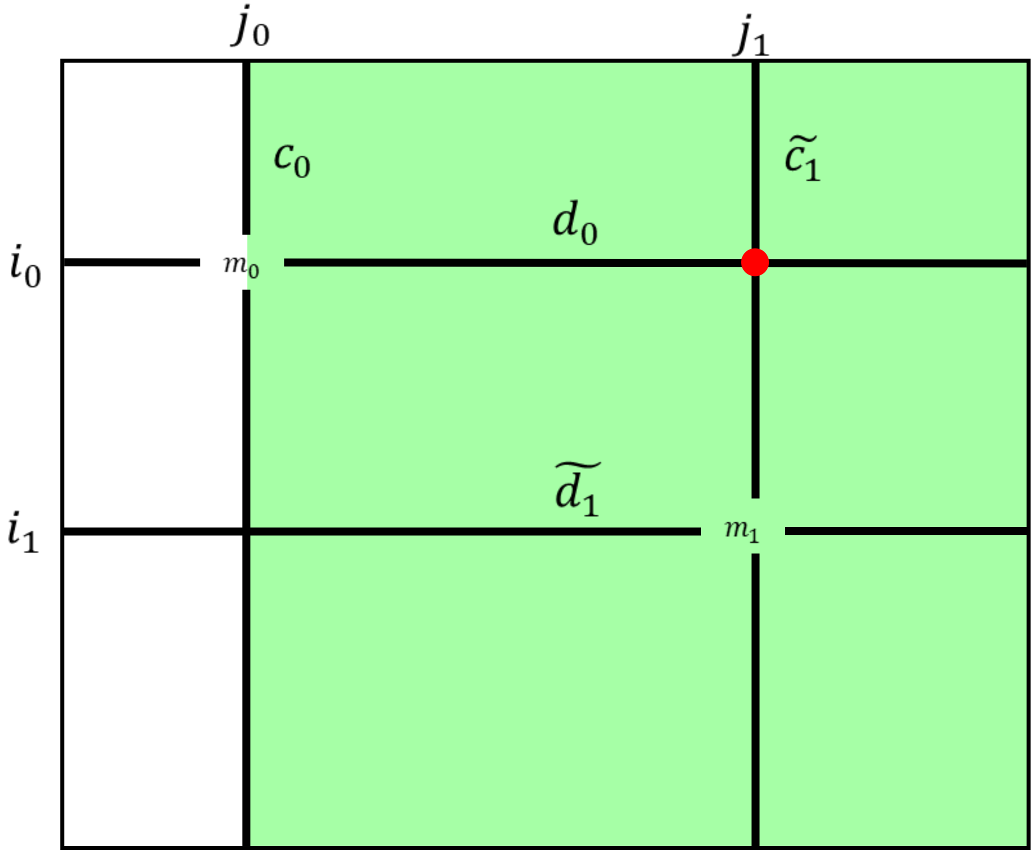
Asymptotically Smooth kernels → Clustering → PACA algorithm



Admissible block B

$$R_0 = \frac{1}{m_0} c_0 d_0^T \longrightarrow \text{rank 1}$$

Asymptotically Smooth kernels → Clustering → PACA algorithm



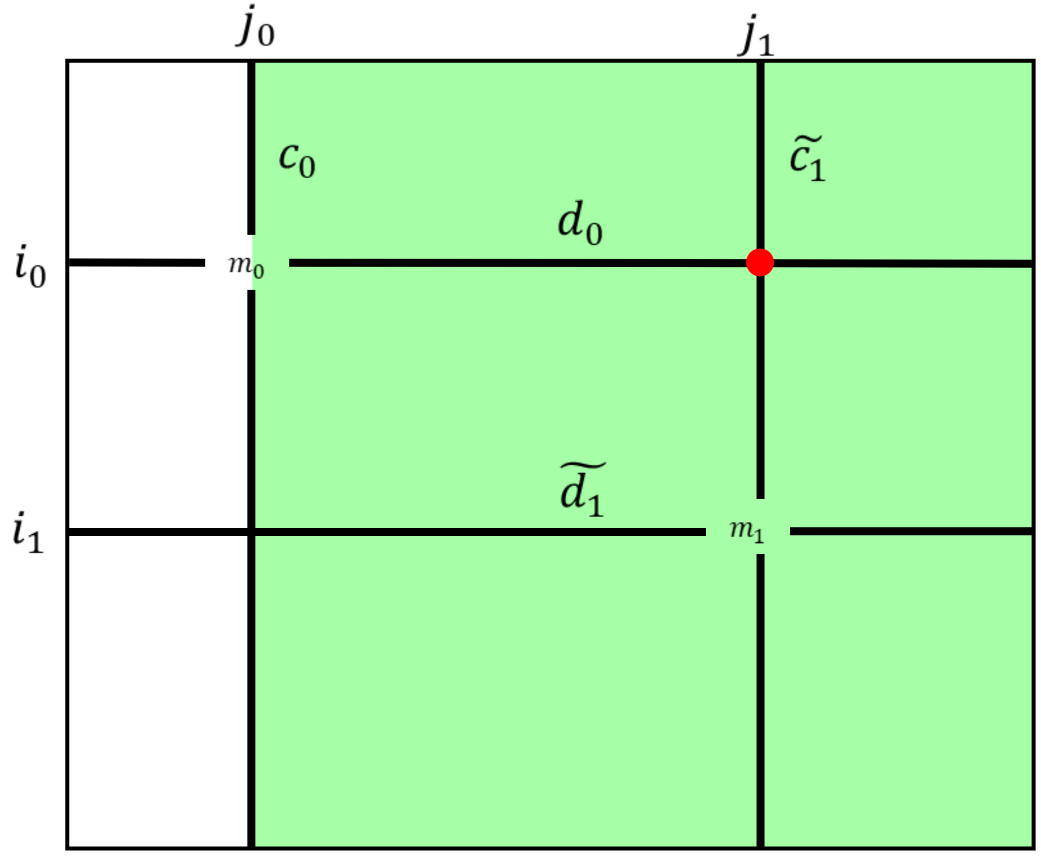
Admissible block B

$$R_0 = \frac{1}{m_0} c_0 d_0^T \longrightarrow \text{rank } 1$$

$$R_1 = \frac{1}{m_1} c_1 d_1^T \longrightarrow \text{rank } 1 \ (c_1 := \widetilde{c_1} - d_0[j_1]c_0)$$

$$(d_1 := \widetilde{d_1} - c_0[i_1]d_0)$$

Asymptotically Smooth kernels → Clustering → PACA algorithm



Admissible block B

$R_0 = \frac{1}{m_0} c_0 d_0^T$ → rank 1

$R_1 = \frac{1}{m_1} c_1 d_1^T$ → rank 1 $(c_1 := \widetilde{c_1} - d_0[j_1]c_0)$
$(d_1 := \widetilde{d_1} - c_0[i_1]d_0)$

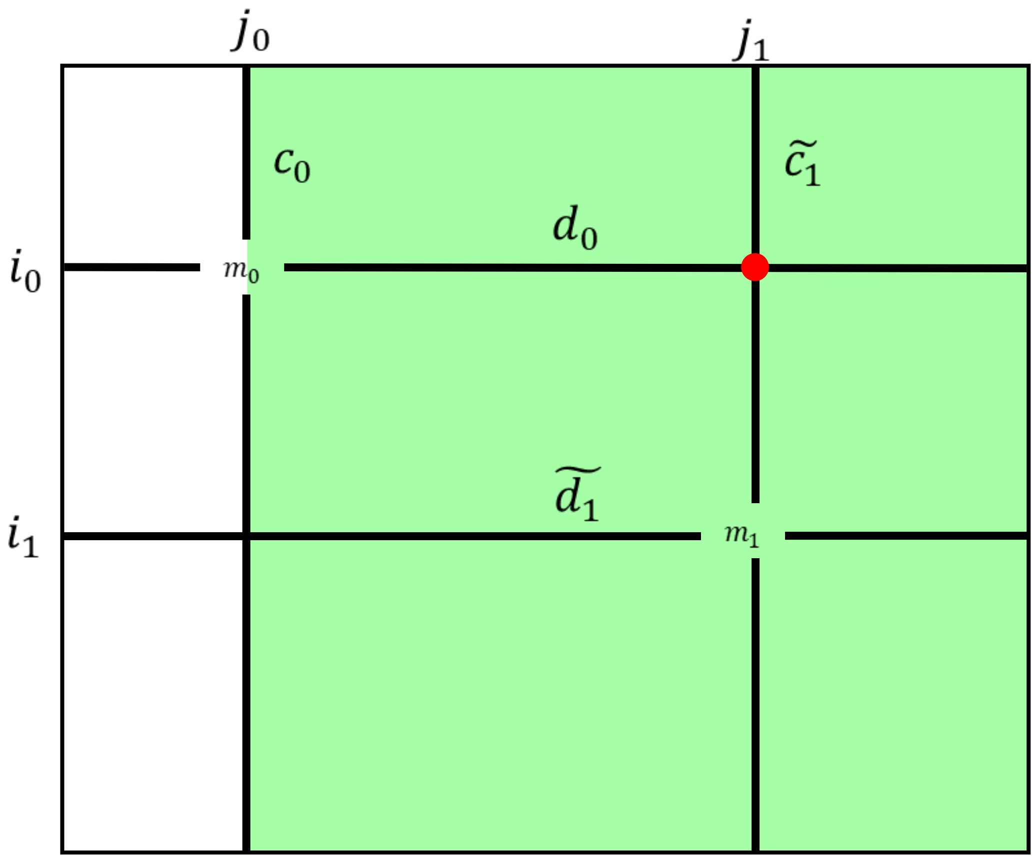
and so on…

$R_r = \frac{1}{m_r} c_r d_r^T$ → rank 1 $(c_r := \widetilde{c_r} - d_{r\text{-}1}[j_r]c_{r\text{-}1})$
$(d_r := \widetilde{d_r} - c_{r-1}[i_r]d_{r-1})$

Asymptotically Smooth kernels

Clustering

PACA algorithm



Admissible block B

$$R_0 = \frac{1}{m_0} c_0 d_0^T \longrightarrow \text{rank } 1$$

$$R_1 = \frac{1}{m_1} c_1 d_1^T \longrightarrow \text{rank } 1 \; (c_1 := \tilde{c}_1 - d_0[j_1] c_0) \; (d_1 := \tilde{d}_1 - c_0[i_1] d_0)$$

and so on...

$$R_r = \frac{1}{m_r} c_r d_r^T \longrightarrow \text{rank } 1 \; (c_r := \tilde{c}_r - d_{r\text{-}1}[j_r] c_{r\text{-}1}) \; (d_r := \tilde{d}_r - c_{r-1}[i_r] d_{r-1})$$

Error estimate

$$\epsilon_{PACA,k} = \frac{\|\boldsymbol{c}_k\|_F \cdot \|\boldsymbol{d}_k\|_F}{\|\boldsymbol{B}_k\|_F}$$

$$\boxed{B \approx B_r = \sum_{k=1}^{r} R_k \longrightarrow \text{rank r}}$$

Asymptotically Smooth kernels → Clustering → PACA algorithm

« **blind** » matrix
exploration & compression

Unassembled block $\xrightarrow{\text{input}}$ PACA $\xrightarrow{\text{output}}$ Rank r block $\mathrm{A} \cdot \mathrm{B}^{\mathrm{T}}$

↓

Doing that for each sub-block → Matrix-vector multipication algorithm → **GMRES**

III. Integration to lifespan analysis

i. Post-processing BEM results : computing the stress intensity factor

ii. Integration to the global pipeline at Safran for lifespan assessment

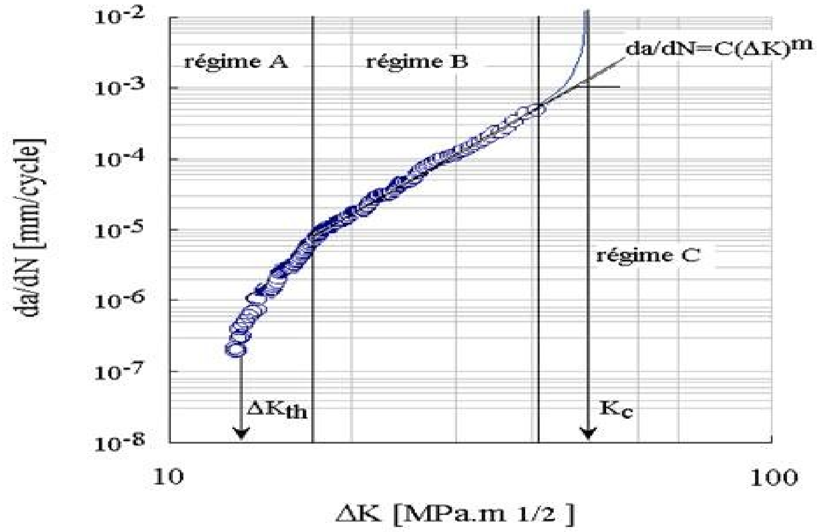
What does it mean to « **solve** » **a crack problem** at Safran ?

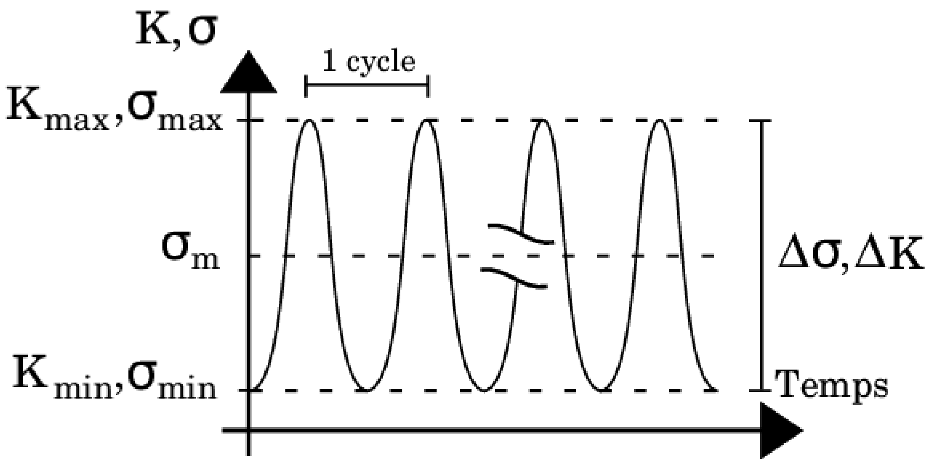
Irwin criterion (1957)

The piece breaks $\Leftrightarrow K > K_c$

Paris' law – *fatigue approach (1963)*

$$\frac{da}{dN} = C(\Delta K)^m$$





What does it mean to « **solve** » **a crack problem** at Safran ?

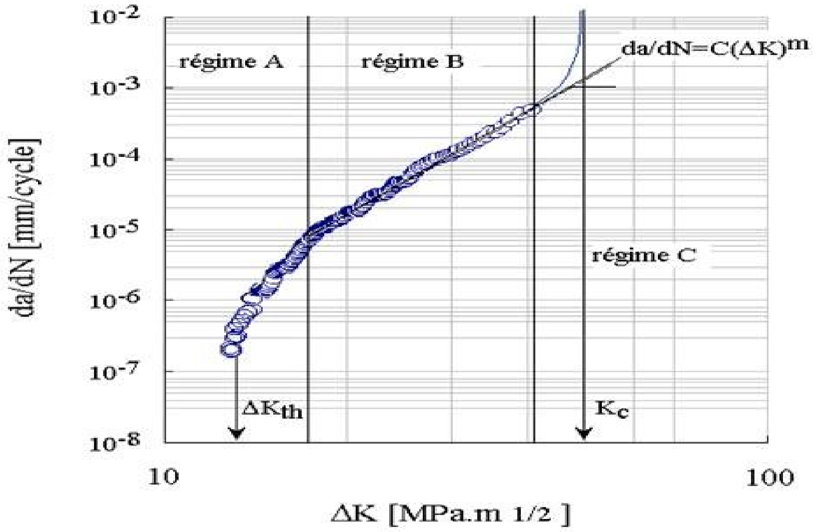
Irwin criterion (1957)

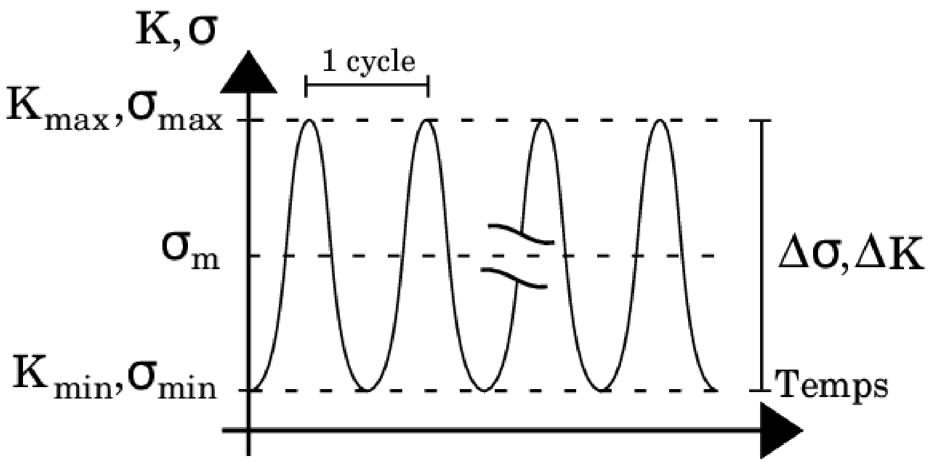
The piece breaks ⇔ $K > K_c$

Paris' law – *fatigue approach (1963)*

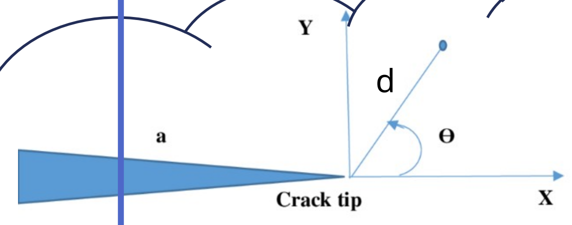
$$\frac{da}{dN} = C(\Delta K)^m$$

$N_{max} = ...$





Direct kinematic extrapolation



$$u(d) = K\sqrt{d}f(\theta) + \mathcal{O}(d)$$

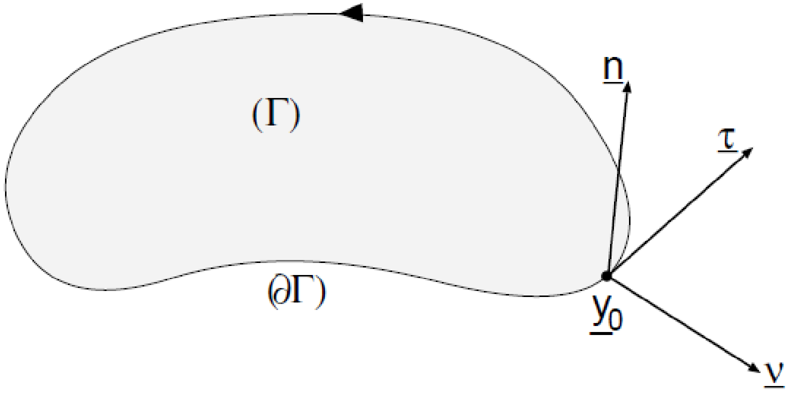
Williams' asymptotic expansion

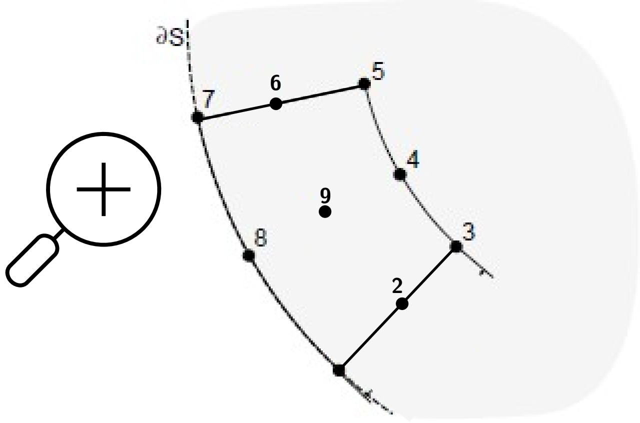
Theorical SIF

$$K = \lim_{d\to 0} \frac{\mu}{4(1-\nu)}\sqrt{\frac{2\pi}{d}}\phi_n$$

Numerical SIF

$$K = \frac{\mu}{4(1-\nu)}\sqrt{\frac{2\pi}{d_{6,9,2}}}\phi_n^{6,9,2}$$





Quarter-node éléments (*a.k.a* Barsoum elements)

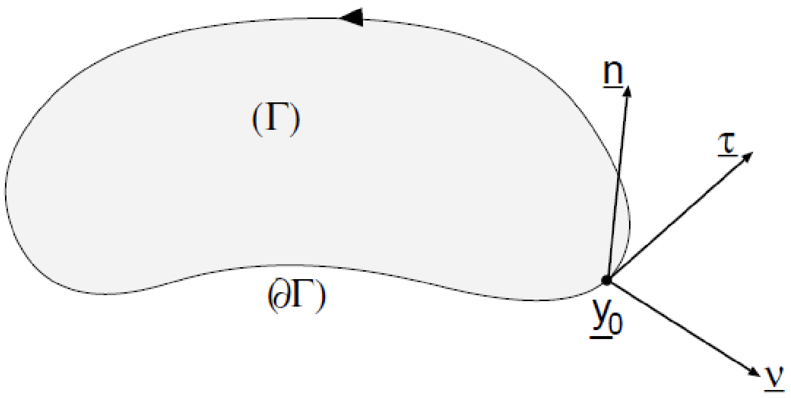
Theorical SIF

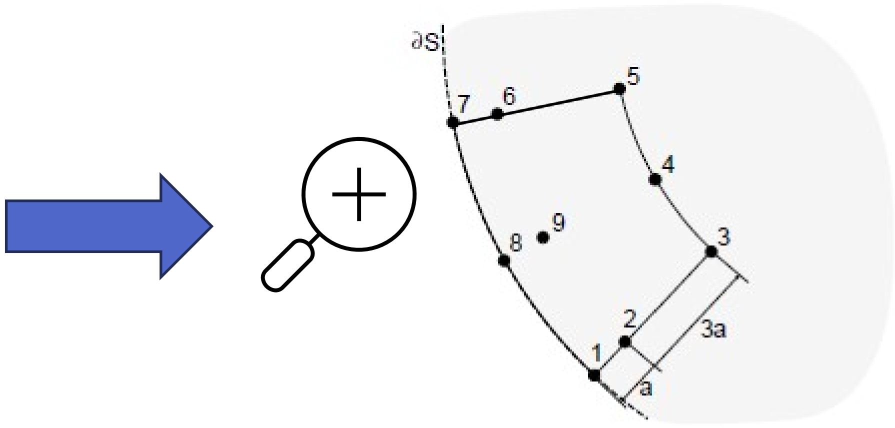
$$K = \lim_{d\to 0} \frac{\mu}{4(1-\nu)} \sqrt{\frac{2\pi}{d}} \phi_n$$

$$\phi(y) = \sqrt{\frac{d}{a}}\left(2\phi^2 - \frac{1}{2}\phi^3 + \sqrt{\frac{d}{a}}\left(\frac{1}{2}\phi^3 - \phi^2\right)\right)$$

Numerical SIF

$$K^1 = \frac{\mu}{4(1-\nu)} \sqrt{2\pi} a(2\phi^2 - \phi^3)$$

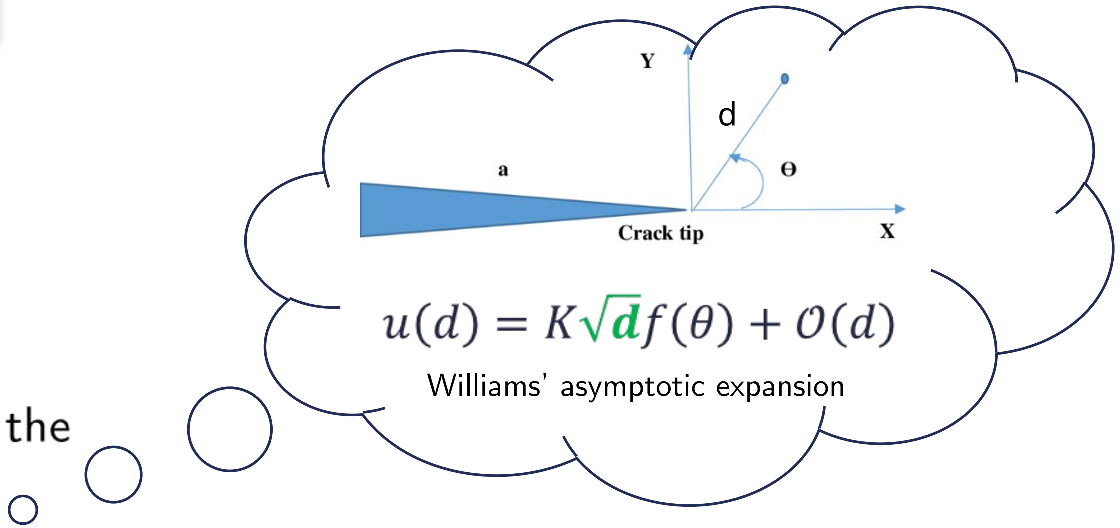




Weighting function

Change of variable : seeking *a priori* the COD ϕ as

$$\phi = w \cdot \psi$$



- w must be **asymptotically as the square root** of the crack front distance:

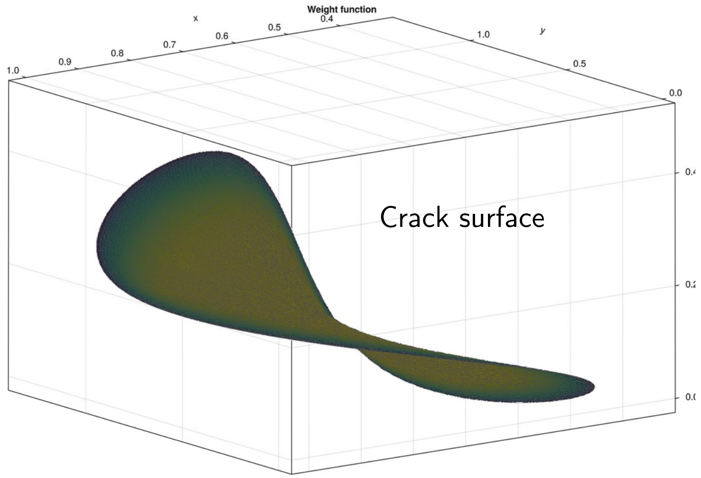
$$w \underset{y \to \text{crack front}}{\sim} \sqrt{d}$$

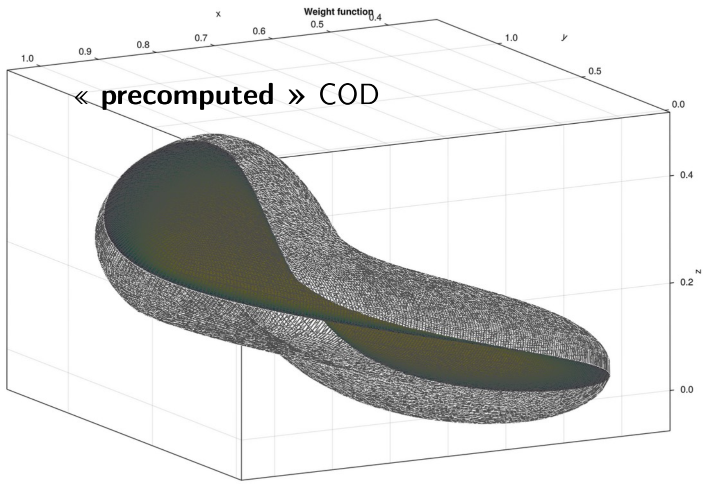
Consequences:

- New **weighted kernels**

$$K_w = w \cdot K$$

- **Better** SIF approximation

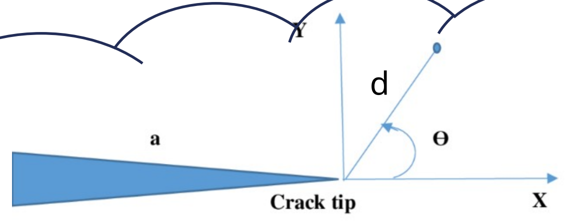




Weighting function

Change of variable : seeking *a priori* the COD ϕ as

$$\phi = w \cdot \psi$$



$$u(d) = K\sqrt{d}f(\theta) + \mathcal{O}(d)$$

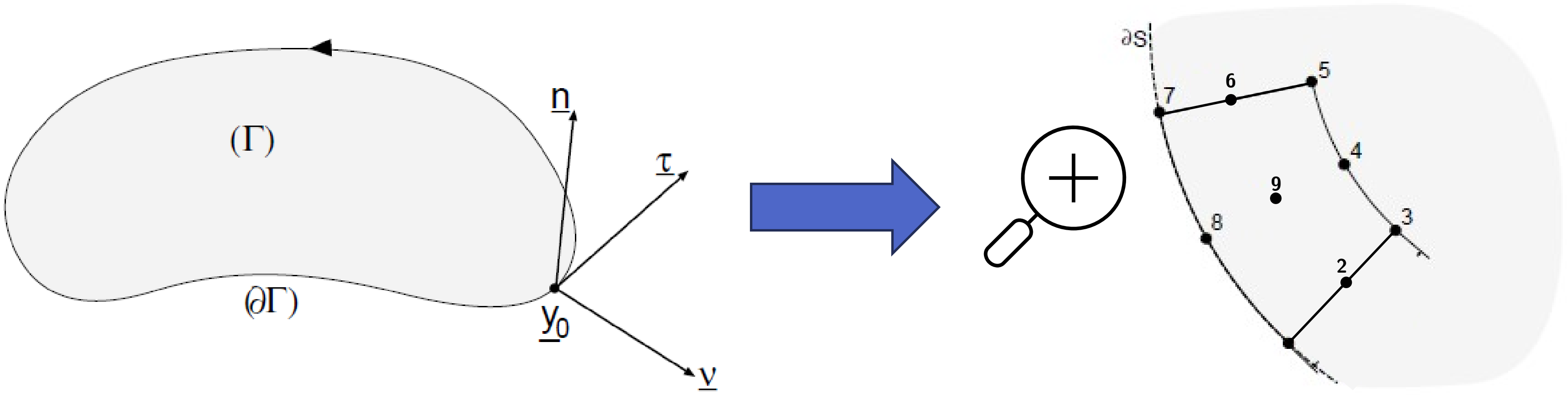
Williams' asymptotic expansion

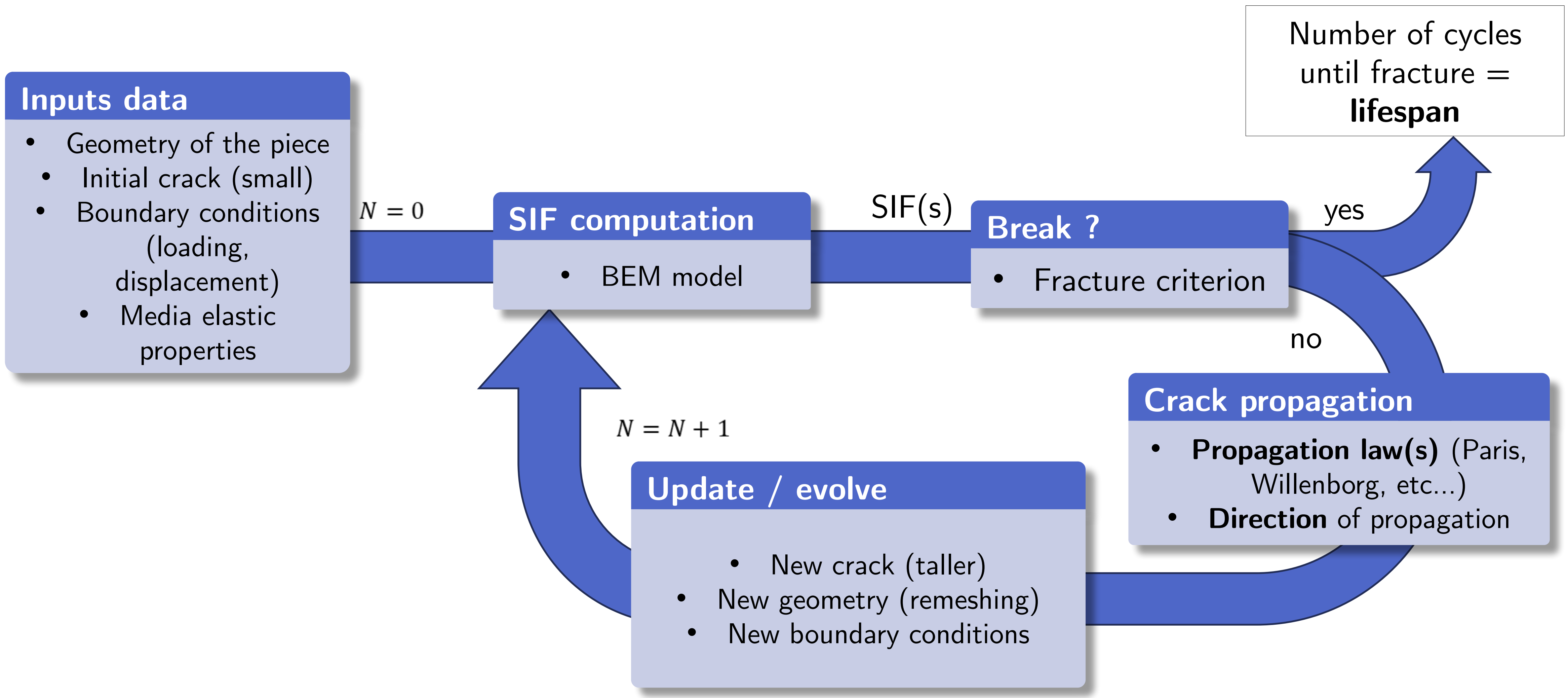
Theorical SIF

$$K = \lim_{d \to 0} \frac{\mu}{4(1-\nu)} \sqrt{\frac{2\pi}{d}} \phi_n$$

Numerical SIF

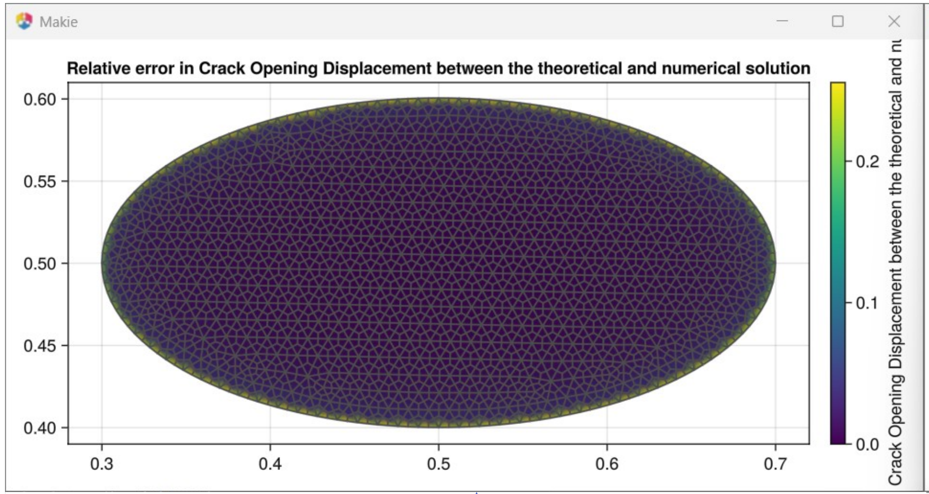
$$K^{6,9,2} = \frac{\mu}{4(1-\nu)} \sqrt{2\pi} \psi_n^{6,9,2}, \qquad \phi = w \cdot \psi$$

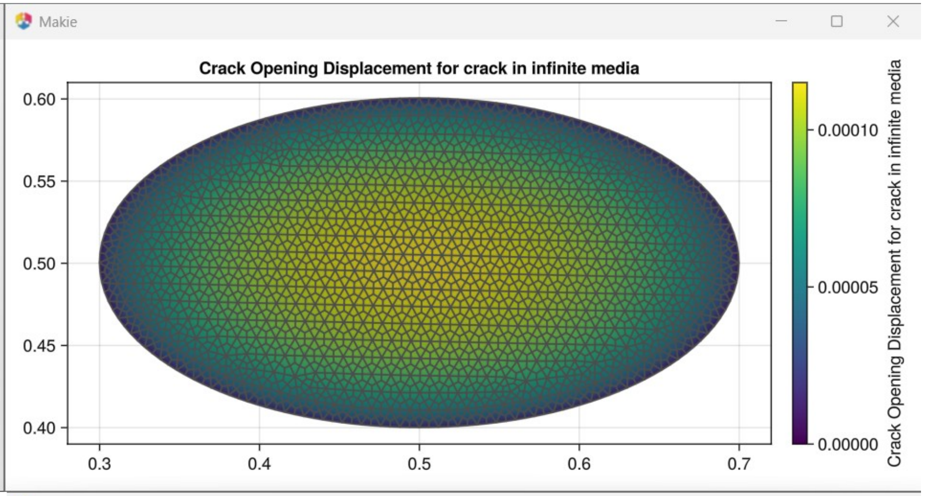



Number of cycles until fracture = **lifespan**
Inputs data
Geometry of the piece
Initial crack (small)
Boundary conditions (loading, displacement)
Media elastic properties
$N = 0$
SIF computation
BEM model
SIF(s)
Break ?
Fracture criterion
yes
no
Crack propagation
Propagation law(s) (Paris, Willenborg, etc...)
Direction of propagation
$N = N + 1$
Update / evolve
New crack (taller)
New geometry (remeshing)
New boundary conditions

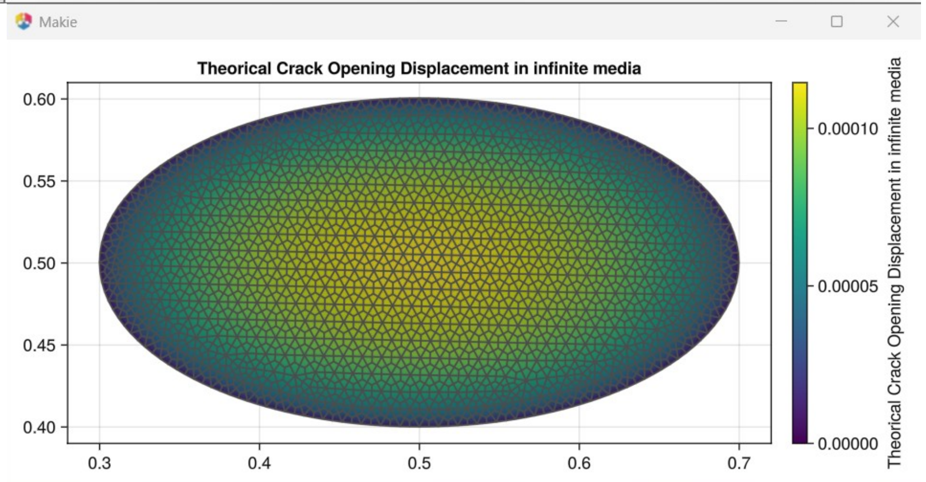
IV. Numerical examples / validation

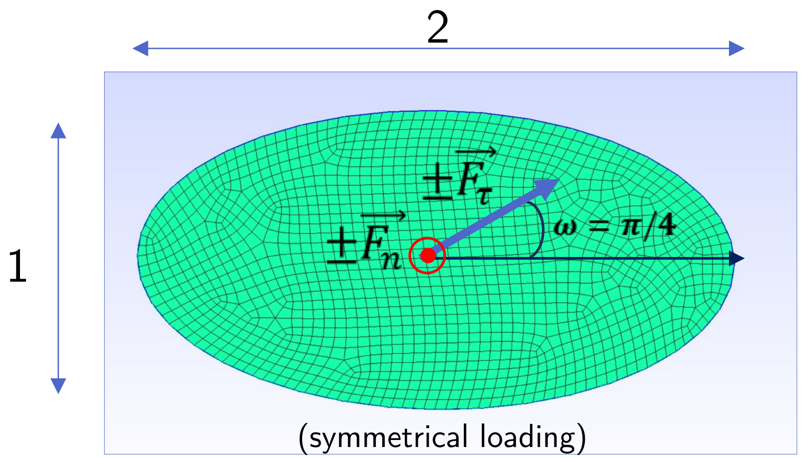
i. Elliptic crack in infinite media
ii. Elliptic crack in cube under mixed boundary conditions
iii. Quick presentation of the Julia library for solving 3D crack problems : `CrackFastBEM`

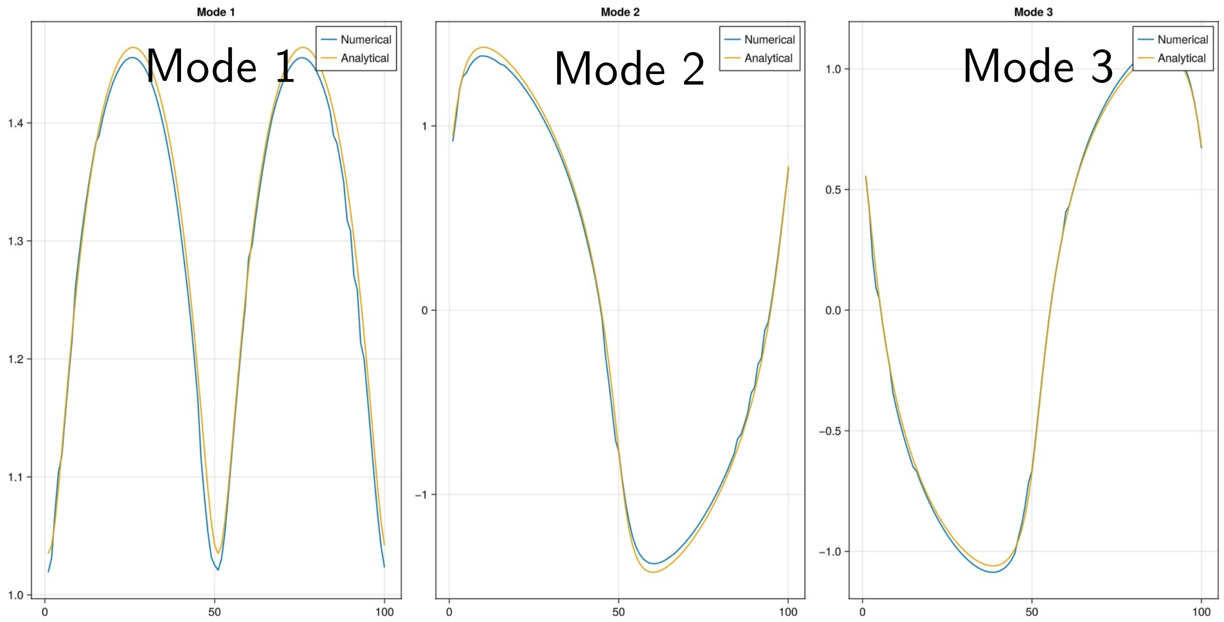


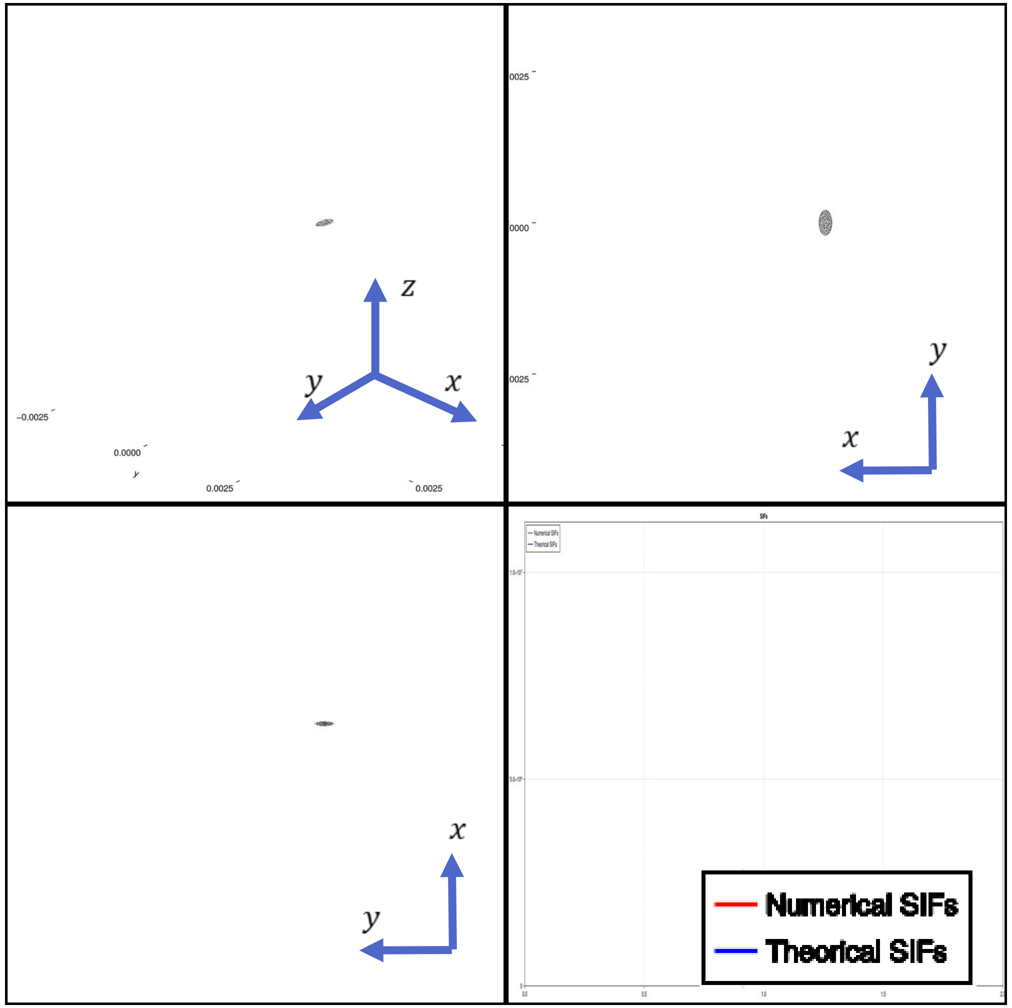


relative error

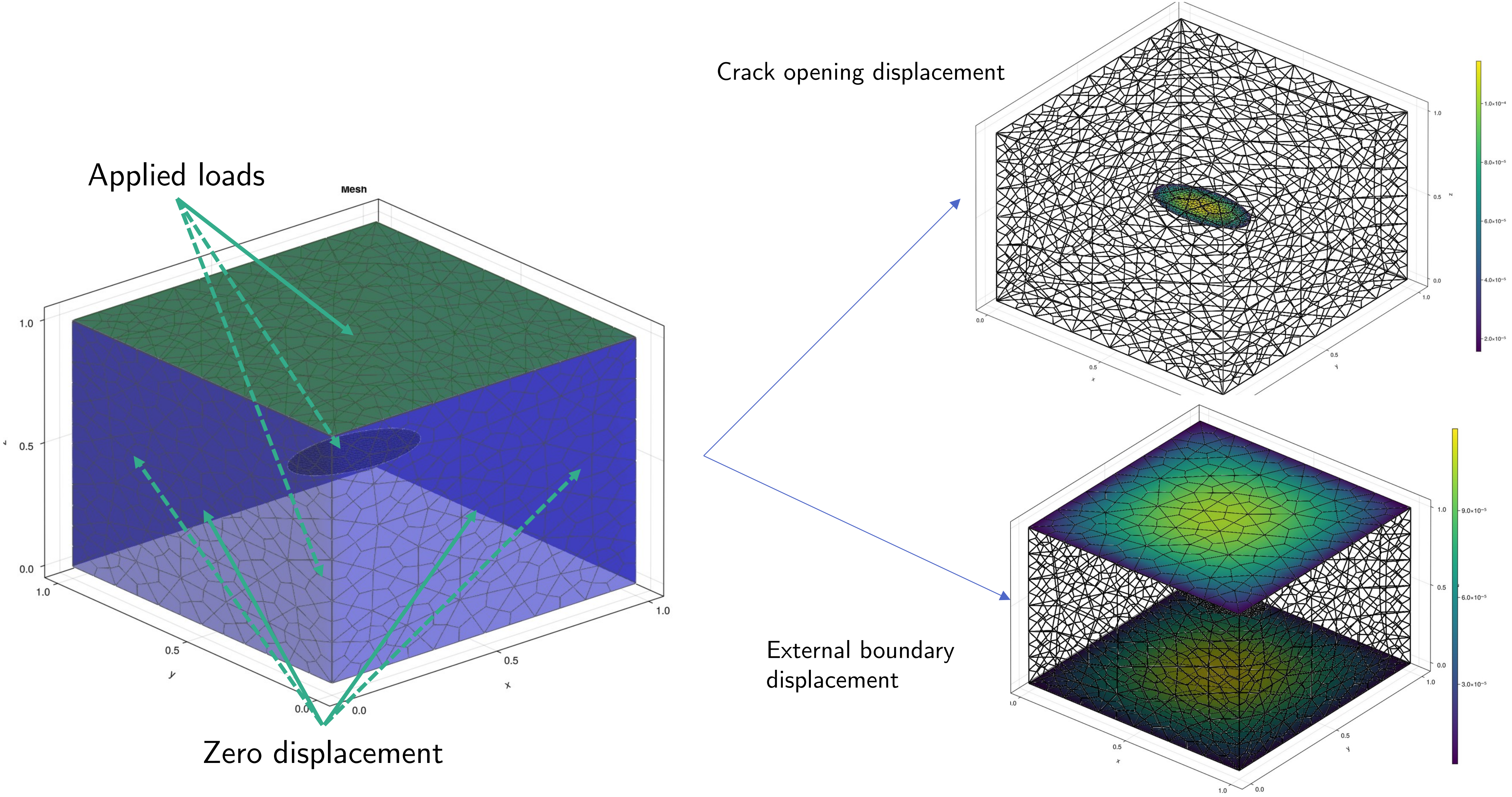




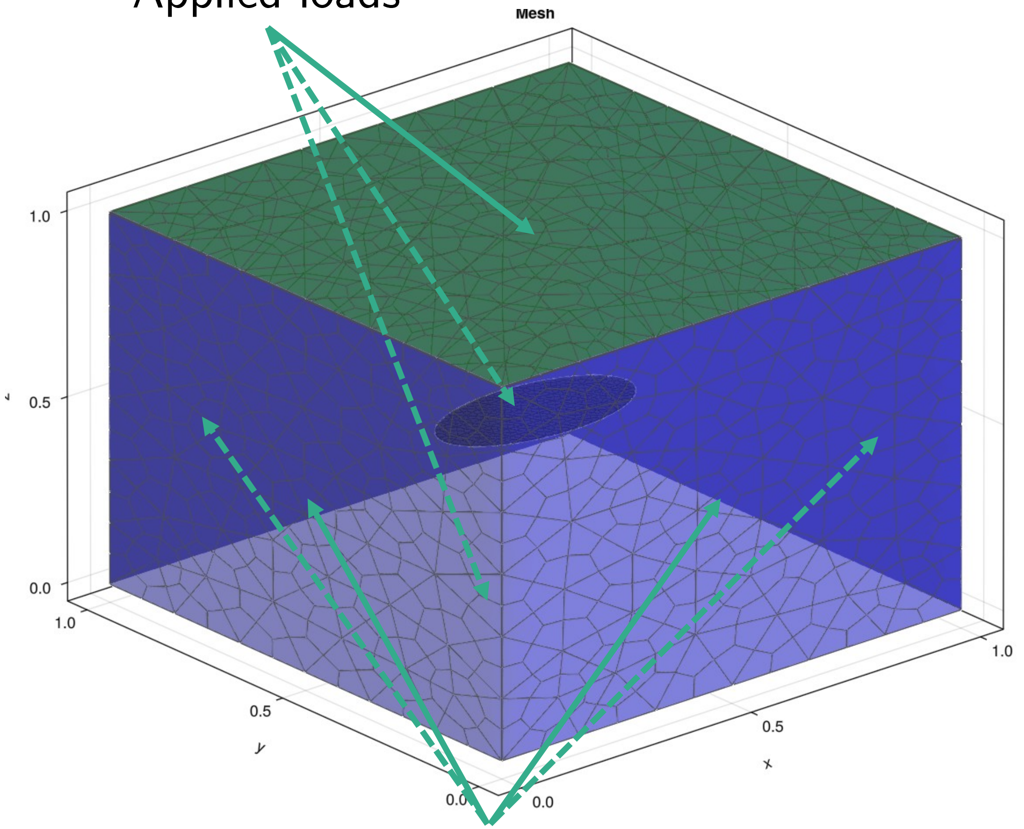




Last iteration = fatigue lifespan

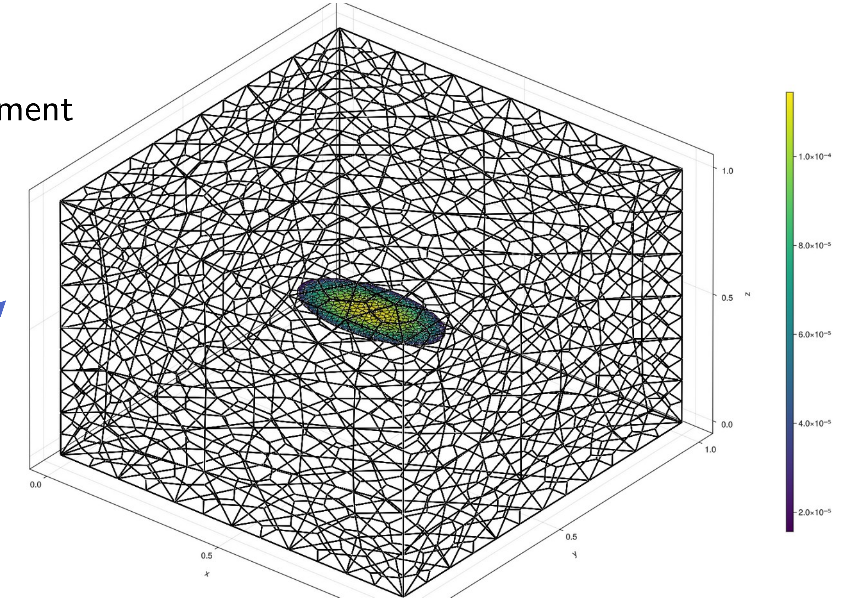

Applied loads
Mesh
Zero displacement
Crack opening displacement
External boundary displacement

Applied loads

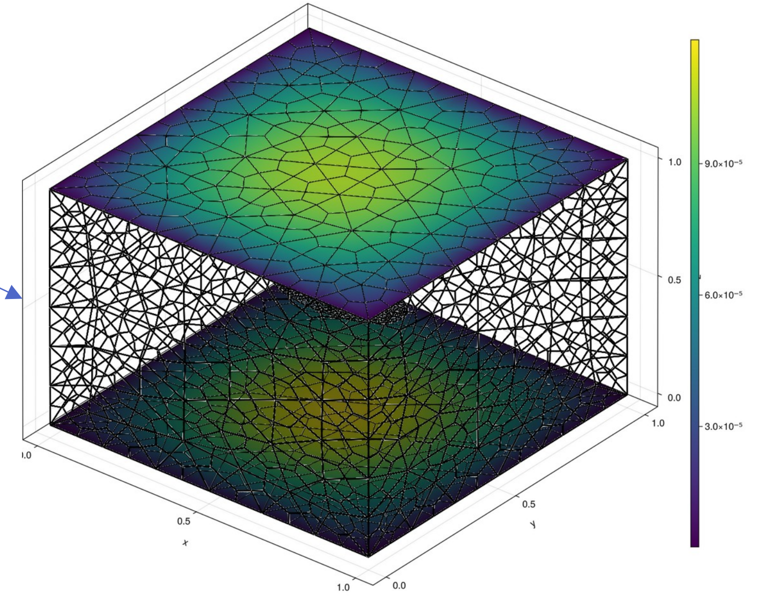


Zero displacement

Crack opening displacement



External boundary displacement



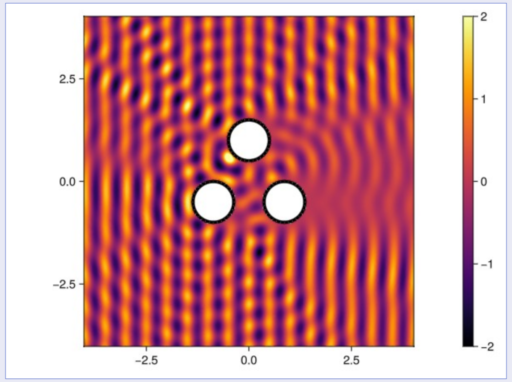
Library under development in julia

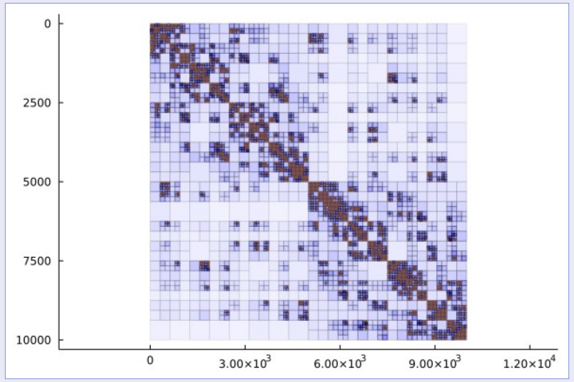
- Julia package for fatigue lifespan estimation for a 3D general crack configuration : **CrackFastBEM.jl**

Features : under development, but the goals are to deal with :

- 3D crack configurations herited from a BEM mesh
- Mixed or simple boundary conditions (Dirichlet + Neumann)
  - Crack in infinite solid
  - Crack in finite solid
  - Surface breaking cracks
- SIF computation : 1) naive approach, 2) with weighting function
- Compatible with the Bueckner's superposition principle
- « user-friendly » API

Inti.jl	**HMatrices.jl**
inti — Julia **library** for solving boundary and volume integral equations using Nyström discretization method	Julia **library** for assembling hierarchical matrices.





Luiz Faria, UMA, POEMS, ENSTA Paris

Forthcoming goals...

- FEM / BEM coupling
- Industrialization with Safran : FEM model → Coupling with BEM → Bueckner superposition
- Thermal gradient : thermal dilatation term
- T & Tz stress

Thank you for your attention