1 Introduction

Finite element analysis (FEA) is ~~highly~~ effective at ~~calculating~~ computing stress intensity factors (SIFs) for cracks in ~~arbitrary~~ complex geometries. However, despite its accuracy, FEA has notable limitations. One significant drawback is its demanding computational requirements, as well as the need for a skilled analyst to achieve reliable results. These factors ~~can~~ lead to ~~substantial~~ added time and resource investments, especially ~~when numerous calculations are required~~, ~~such as those~~ for fatigue and uncertainty assessment. To address these challenges, ~~many~~ engineers in industry ~~often~~ rely on handbook solutions. These solutions offer a practical way to estimate SIFs for ~~common~~, ~~simplified~~ idealized and parameterized crack scenarios, and, when applied correctly, they can yield ~~very~~ accurate results. ~~Additionally, they entail significantly reduced computational costs compared to FEA~~.

The Raju Newman equations are widely used set of handbook ~~equations that provide predictions for SIFs under~~ SIF solutions for various geometries, and loading conditions. ~~These handbook solutions are user-friendly, given their straightforward polynomial forms~~. However, their ease of use sometimes leads to their application in situations that do not align well with the assumptions and limitations of the original idealized model, resulting in inaccurate predictions. This research ~~specifically~~ focuses on the ~~case of a~~ RN solution for semi-elliptical surface crack in a finite plate subjected to mode-I tension. ~~Raju and Newman have an equation designed for this particular crack scenario.~~ Since ~~the introduction of the Raju-Newman equations for a semi-elliptical surface cracks~~ its introduction alternative models have been proposed that allow for more complex loading conditions such as the models created by [1, 2]. The model created by Wang and Lambert allows for linear stress functions [1]. Pommier, et al. created a model that could predict SIFs under polynomial stress fields [2].

~~To prevent the improper use of SIF equations, it is crucial to build a more extensive database of SIF solutions~~. Machine learning (ML) can play a pivotal role in automating the creation of these SIF solutions. ~~ML enables the development of models for more complex crack scenarios, going beyond the scope of current handbook solutions, which primarily cover idealized cases~~. Notably, ML has been successfully employed to produce ~~highly~~ accurate SIF solutions, as documented in studies such as in [3, 4, 5, 6, 7]. The advantage of ML over other techniques, like those found in [8, 2, 1], is its ability to be trained relatively quickly, ~~resulting in very accurate models. This, in turn, facilitates the creation of more surrogate models. However~~, it's worth noting that many commonly used ML algorithms generate "black-box" models, ~~which~~ lack interpretability. In engineering, interpretability is crucial because engineers need to trust and explain their designs, which black-box models do not readily allow. This is why handbook solutions, such as those in [8], continue to be used since they inherently offer interpretability.

This paper specifically studies the case of a semi-elliptical surface crack in a finite plate subjected to mode-I tension, a common crack found in pressure vessels. An existing handbook solution by [8] covers this case. ~~However,~~ ~~more accurate~~ ML models have already been developed using ~~machine learning (ML)~~ ~~techniques~~ black box methods, as demonstrated by [5, 6, 7]. These ML models, utilizing methods like Gaussian processes regression and neural networks, which offer ~~higher~~ accuracy but produce less interpretable ~~"black-box" models~~. Notably, the ML models mentioned could predict SIFs at only a single point along the crack front, whereas the Raju-Newman equations can estimate SIFs across the entire crack length.

While ML models provide improved predictive capabilities, their lack of interpretability can be a drawback in certain contexts, especially when it comes to explaining and trusting the predicted SIF. The interpretability of the Raju and Newman equations gives engineers a clear understanding of how SIFs are derived, aiding decision-making and instilling confidence in the results. These equations systematically break down the crack case into subfunctions, each contributing differently to the final SIF predition. For instance, in the case of the semi-elliptical crack, they modify the analytical solution to the embedded ellipse with boundary correction functions, enhancing the inherent explainability of an analytical solution.

This research will utilize an interpretable machine learning code Bingo, developed by researchers at NASA and the University of Utah [9]. Bingo generates closed-form mathematical expressions to predict SIFs along the entire crack length, offering improved accuracy and simplicity compared to the Raju-Newman equations while preserving interpretability. Thanks to the closed-form nature of Bingo's models, they can be easily integrated into existing Linear Elastic Fracture Mechanics (LEFM) software applications like NASGRO, AFGROW, and SMART|DT [10, 11, 12]. ~~Closed form solutions can also be inverted to solve for applied stress for use in a constant SIF test.~~

1.1 Background

From linear elastic fracture mechanics (LEFM), the exact solution for SIF along an elliptical crack in an infinite volume is :

$$K_{ee} = \sigma \frac{\sqrt{\pi a}}{E} \left(\sin^2 \phi + \frac{a^2}{c^2} \cos^2 \phi \right)^{1/4} \quad \text{for } a \leq c, \tag{1}$$

where E is the complete elliptic integral of the second kind and σ is the far-field stress. The length of the minor axis of the ellipse is a and the length of the major axis is c ~~as shown in figure 1 [13]~~. For a semi-elliptical surface crack in a semi-infinite volume the values of a and c are no longer the minor and major axis, but the crack depth and the half-crack surface length. To account for the minor and major axes switching at $a/c = 1$ Raju and Newman ~~used~~ modified ~~equation~~ 1 to allow for all values of a/c to be used by using equation 2

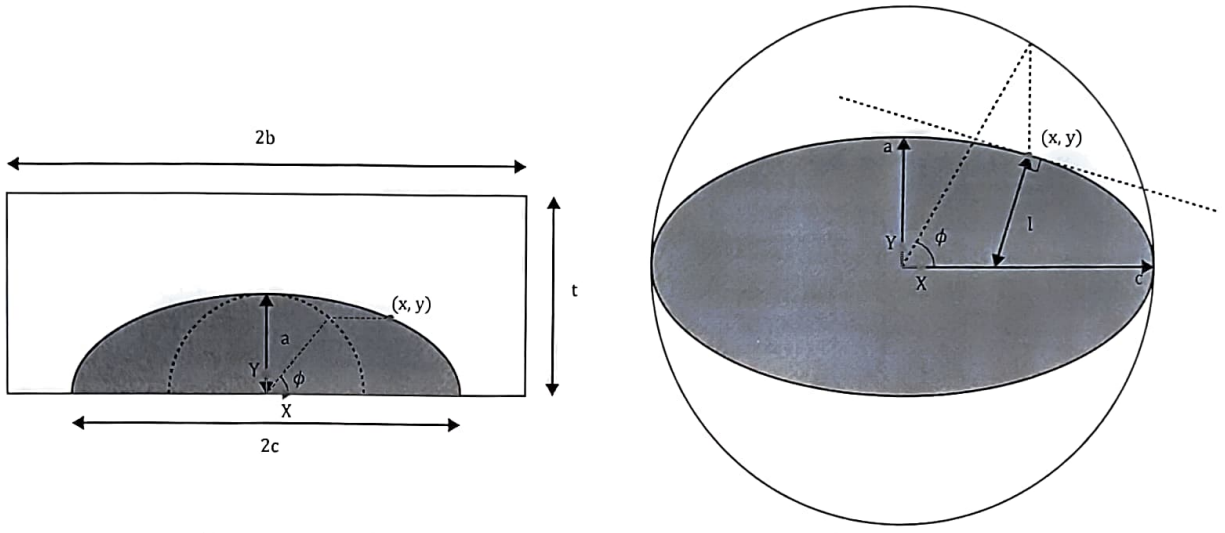
$$f_\phi = \begin{cases} \left[\left(\frac{a}{c}\right)^2 \cos^2 \phi + \sin^2 \phi \right]^{1/4} & \text{if } \frac{a}{c} \leq 1 \\ \left[\cos^2 \phi + \left(\frac{c}{a}\right)^2 \sin^2 \phi \right]^{1/4} & \text{if } \frac{a}{c} > 1 \end{cases} \tag{2}$$

which is piece-wise, resulting in equation 3

$$K_{ee} = \sigma \frac{\sqrt{\pi a}}{E} f_\phi. \tag{3}$$

~~Equation 3 is identical to equation 1 when $a/c \leq 1$.~~ Using equation 3 and multiplying by the correction factors f_w, M, and g Raju and Newman ~~found~~ leads to equation 4,

$$K = \frac{\sqrt{\pi a}}{E} f_\phi f_w M g, \tag{4}$$



(a) Crack dimensions

(b) Ellipse dimensions

Figure 1: (a) Crack parameters with a being the crack depth and $2c$ being the surface crack length. (b) ϕ is defined by the angle to the inscribed circle projected to the ellipse and l is defined as the distance perpendicular to the tangent line from the point of interest to the nearest axis.

which ~~is able to predict~~ SIFs along the ~~length~~ of semi-elliptical surface crack in a finite plate.

The finite width correction factor f_w ~~defined in equation 5.~~

$$f_w = \sqrt{\sec\left(\frac{\pi c}{2b}\sqrt{\frac{a}{t}}\right)}, \tag{5}$$

is a 3D modification to the equation developed in [14], which accounts for the finite width and thickness of the plate when $a/c = 1$ and $\phi = \pi/2$. The thickness of the plate is denoted as t and the half-width of the plate is denoted as b. The function M accounts for the aspect ratio effect in a finite thickness at $\phi = \pi/2$ and is given ~~by 6~~

$$M = M_1 + M_2\left(\frac{a}{t}\right)^2 + M_3\left(\frac{a}{t}\right)^4, \text{ where} \tag{6}$$

$$M_1 = \begin{cases} 1.13 - 0.09\left(\frac{a}{c}\right) & \text{if } \frac{a}{c} \leq 1 \\ \sqrt{\frac{c}{a}}\left(1 + 0.04\frac{c}{a}\right) & \text{if } \frac{a}{c} > 1 \end{cases} \tag{7}$$

$$M_2 = \begin{cases} -0.54 + \frac{0.89}{0.2+\left(\frac{a}{c}\right)} & \text{if } \frac{a}{c} \leq 1 \\ 0.2\left(\frac{c}{a}\right)^4 & \text{if } \frac{a}{c} > 1 \end{cases} \tag{8}$$

$$M_3 = \begin{cases} 0.5 - \frac{1}{0.65+\frac{a}{c}} + 14\left(1-\frac{a}{c}\right)^{24} & \text{if } \frac{a}{c} \leq 1 \\ -0.11\left(\frac{c}{a}\right)^4 & \text{if } \frac{a}{c} > 1 \end{cases} \tag{9}$$

The function g corrects for free surface effects and is denoted by ~~equation 10~~

$$g = \begin{cases} 1 + \left[0.1 + 0.35\left(\frac{a}{t}\right)^2\right](1-\sin\phi)^2 & \text{if } \frac{a}{c} \leq 1 \\ 1 + \left[0.1 + 0.35\left(\frac{c}{a}\right)\left(\frac{a}{t}\right)^2\right](1-\sin\phi) & \text{if } \frac{a}{c} > 1 \end{cases} \tag{10}$$

The function g is sinusoidal, having a value of 1 at $\phi = \pi/2$. By using this methodical approach of breaking down the problem into sub functions that each account for a different portion of the crack geometry being finite, allowed Raju and Newman to develop accurate equations that build upon the explainablity from the analytical solution of the embedded ellipse.

good! need to say this in the intro as a pro of the RN approach and an aspect you are combining with ML for the 'best of both worlds.'

2 Methods

An intro paragraph is needed to describe the big picture of generating training data, breaking down the subfunctions/training, to provide a map/guide for this section.

2.1 Computational Fracture Mechanics

Stress intensity factors (SIFs) were simulated using finite element analysis (FEA) using the ~~with the software tools~~ software FRANC3D and Abaqus [15, 16]. Abaqus served as the primary tool for creating ~~and solving the finite element mesh~~. FRANC3D was employed for tasks related to crack insertion and SIF computation. Abaqus' Python interface was used to generate the model geometries → the geometry, ~~mesh~~ initial mesh discretization, application of BCs, and solution.

~~For increased reliability in crack insertion using FRANC3D,~~ A local-global sub-modeling approach was employed. ~~This method~~ involved dividing the geometry into two components: the global model, which encompassed the boundary conditions, and the local model, ~~focused on~~ the region where the crack would be inserted. The local model was then loaded into FRANC3D, where the crack was ~~introduced~~ inserted using rings of hexagonal elements, with an inner ring of quarter-point elements surrounding the crack front, ~~allowing~~ for SIF ~~calculations~~ computations, ~~as shown in~~ Figure ??. ~~The remainder of~~ After crack insertion the local model was re-meshed, preserving the nodal locations ~~around~~ on the cut faces for ~~later reattachment to~~ coherency with the global model. The ~~recombined~~ complete model, with the inserted crack, was subsequently solved using ~~the~~ Abaqus ~~solver~~. ~~The flow of the finite element modeling process~~ This process is illustrated in Figure ??.

When ~~calculating~~ computing SIFs, integral methods are ~~preferred over~~ are generally more accurate than displacement methods, such as displacement correlation. ~~Integral methods assess the energy of the cracked system, providing greater accuracy~~. The J-integral, developed by [17], is a commonly used method for SIF ~~calculation~~ computation. However, it has a limitation, it cannot separate the SIFs into the three cracking modes, except for very simplified crack geometries, as noted by [18].

merge with prev. paragraph

To address this limitation, the M-integral offers a solution by using a combination of two SIFs: one representing the true SIF and the other an arbitrary solution. By employing this approach, it becomes possible to extract each SIF corresponding to the three cracking modes, as explained by [18]. The M-integral is defined as follows:

This isn't enough to explain what is meant. You should either (1) add detail or (2) delete these and rely on referencing [18] for more details.

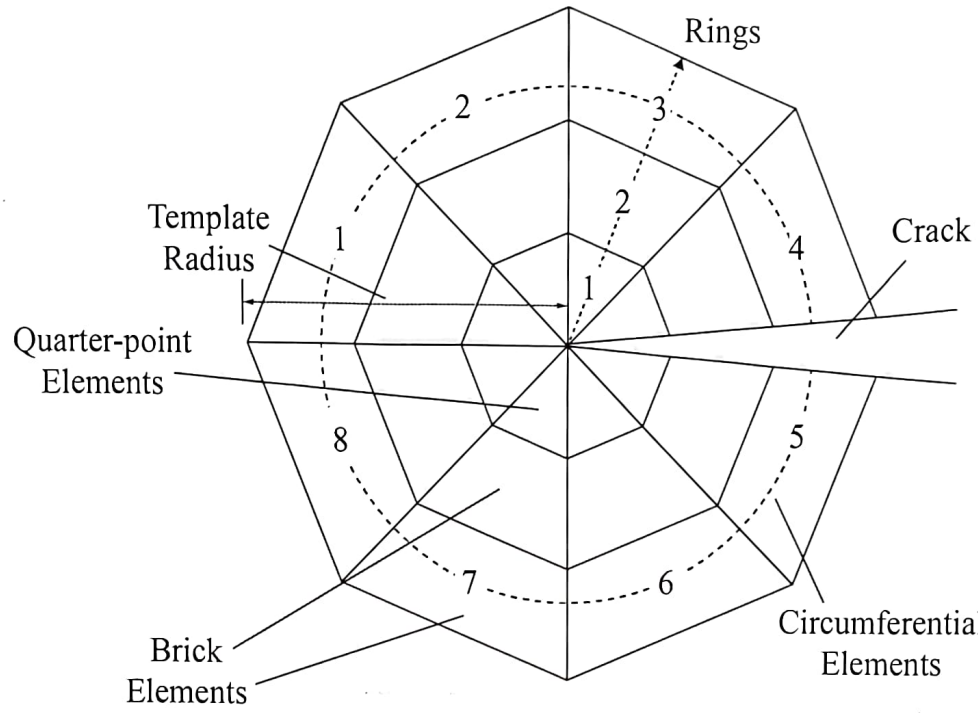


Figure 2: FRANC3D crack mesh containing 8 circumferential elements and 3 rings. The template radius is the radius of an inscribed circle

If you decide to nix the explanation of M (previous comment) then you can also delete this equation.

$$M^{(1,2)} = \int_{\Gamma} \left(W^{(1,2)} n_1 - T_i^{(1)} \frac{\partial u_i^{(2)}}{\partial x_1} - T_i^{(2)} \frac{\partial u_i^{(1)}}{\partial x_1} \right) \mathrm{d}s, \tag{11}$$

Here, $T_i = \sigma_{ij} n_j$ represents the traction vector, $W = 1/2 \sigma_{ij} \epsilon_{ij}$ stands for the strain energy density, and the superscripts 1 and 2 correspond to the two superposed solutions.

2.2 Genetic Programming Based Symbolic Regression

Symbolic Regression (SR) is a machine learning technique aimed at discovering closed-form analytical expressions that ~~accurately fit~~ model training data. Presently, the most effective optimization approach for SR is genetic programming (GP) [19]. The implementation of genetic programming-based symbolic regression (GPSR) employed in this research is an open-source Python package, ~~known as~~ Bingo [9]. Bingo enables ~~the~~ learning ~~of~~ real-valued mathematical expressions represented as acyclic graphs (Agraphs) enabling adaptive adjustments throughout training. The equation's complexity is determined by the number of nodes in the Agraph, figure ?? shows an Agraph for the equation $C_0 X_0 + C_1 X_1$, that has a complexity of 7, corresponding to the 7 nodes in the Agraph. The training begins with an initial assortment of randomly generated equations, varying in complexity. In each generation, equations undergo randomized ~~modifications~~ evolutions, employing a combination of crossover and mutation. Crossover entails ~~selecting~~ exchanging nodes from two ~~equations~~ Agraphs ~~and exchanging these nodes~~ them along with their associated branches and leaves, generating two ~~new~~ evolved equations. Mutation, ~~on the other hand,~~ introduces random changes to nodes in the Agraph as shown in

and known physics [ref. Karl, Hongsup]

The GP (not AGraph) "enables" this.

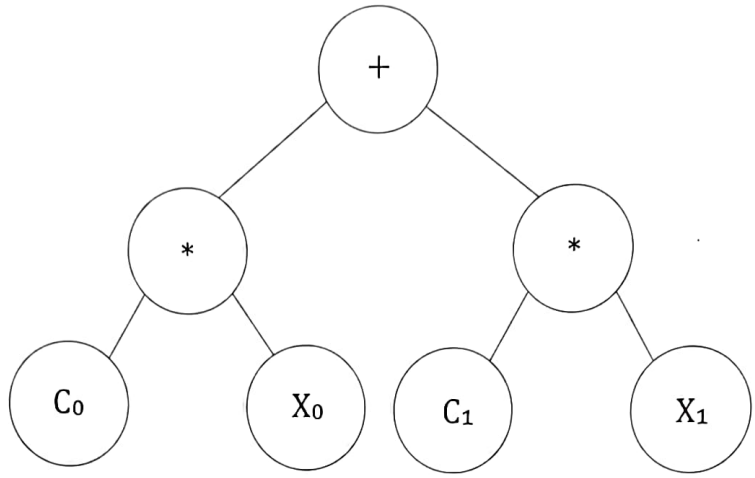


Figure 3: Example Agraph for for equation; $C_0X_0 + C_1X_1$, where the complexity is the sum of the nodes in this case 7.

← EFM will define formatting here.

Fig. figure 4 [20]. After crossover and mutation has occurred the equations are ranked based on a defined fitness criterion (function e.g., MAE, RMSE, etc.), a user-set function crucial that heavily influences the algorithm's speed and results. The best equations (most fit) are then used in the next generation where the cycle continues until a fitness threshold or maximum number of generations has been satisfied.

Symbolic Regression (SR) is a machine learning technique designed to discover closed-form analytical expressions that accurately match training data. Currently, the most effective optimization approach for SR is genetic programming (GP) [19]. In this research, genetic programming-based symbolic regression (GPSR) is implemented using an open-source Python package called Bingo [9]. Bingo allows for the learning of real-valued mathematical expressions represented as acyclic graphs (Agraphs), which enable adaptive adjustments during training.

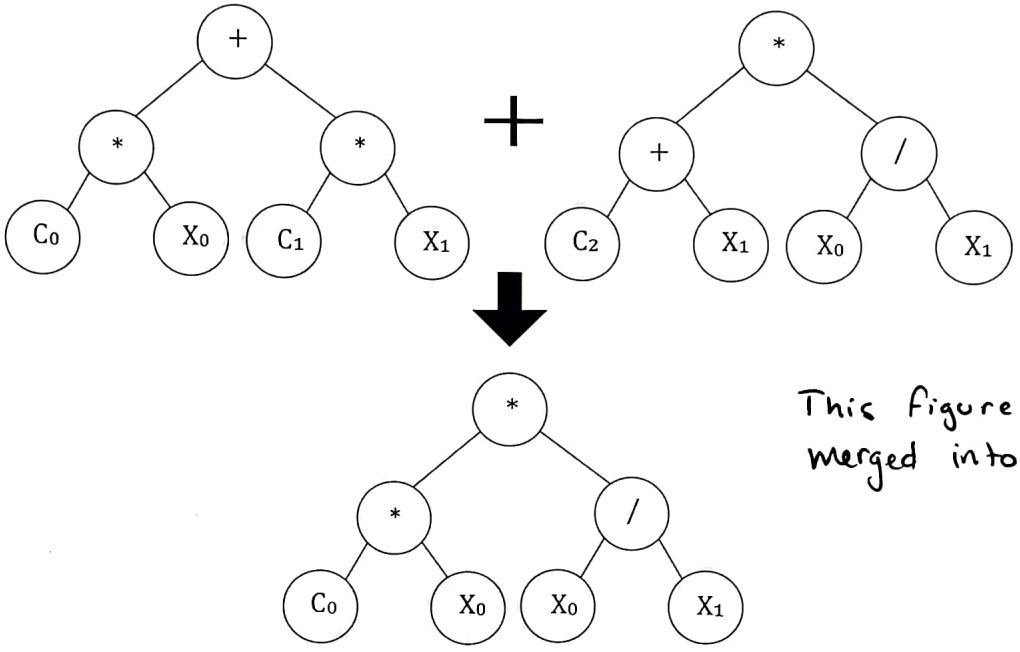
Already said in previous paragraph

The complexity of an equation is determined by the number of nodes in the Agraph. For instance, Figure ?? illustrates an Agraph for the equation $C_0X_0 + C_1X_1$, which has a complexity of 7, corresponding to the 7 nodes in the Agraph. The training process starts with an initial set of randomly generated equations that vary in complexity. In each generation, equations undergo random modifications, using a combination of crossover and mutation.

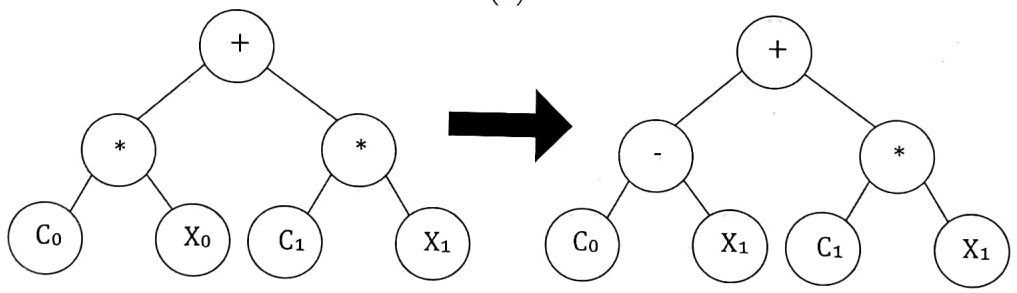
Crossover involves selecting nodes from the Agraphs of two equations and exchanging these nodes, along with their associated branches and leaves, to generate two new equations. Mutation, on the other hand, introduces random changes to nodes in the Agraph, as depicted in Figure 4 [20]. After crossover and mutation, the equations are ranked based on a predefined fitness criterion, a user-defined function that significantly impacts the algorithm's speed and results. The best equations are then carried forward to the next generation, and this cycle continues until a fitness threshold or maximum number of generations is reached.

The best performing equations, according to the fitness metric (at various complexity values) from the final population are taken and presented as a Pareto front figure ?? where both the complexity and fitness of each equation can be visualized. Because GPSR is an interpretable ML method, interpretability of the chosen equation is often important. Equations with high complexity have better fitness values; however, they can lack interpretability. Equations with low complexity tend often have reduced

from the engineer's perspective.

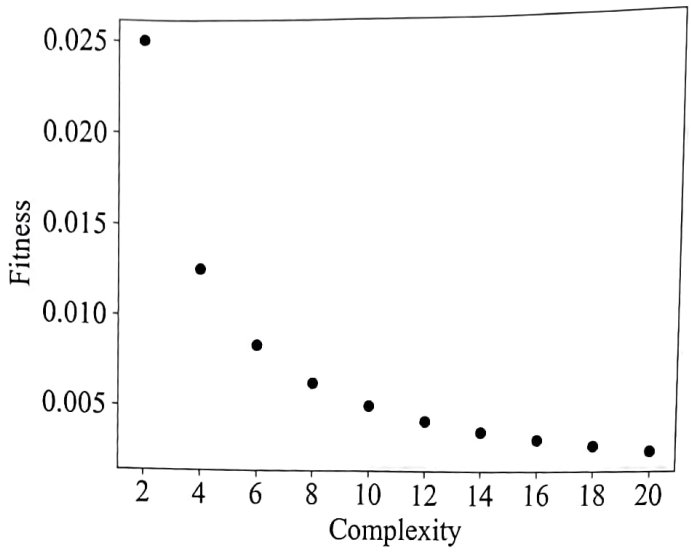


(a)



(b)

Figure 4: (a) crossover, where two parent graphs combine features to create a child graph. (b) where part of a graph is randomly changed creating a new graph.



← identify/annotate

Figure 5: Example Pareto front where the horizontal axis is the model complexity and the vertical axis is the model fitness. Each point on the graph represents a different model in the population.

Lacking additional criteria, a

to be easily interpreteble at the expense of decreased fitness. ~~The~~ user must decide what equation is the best for the application. Typically, there is a point at which increasing the complexity yields little gain on fitness. ~~this optimal value can be seen in~~ Figure (??), which serves as a ~~tradeoff~~ balance between accuracy and complexity.

3 Numerical Experiments

3.1 Training Data Generation

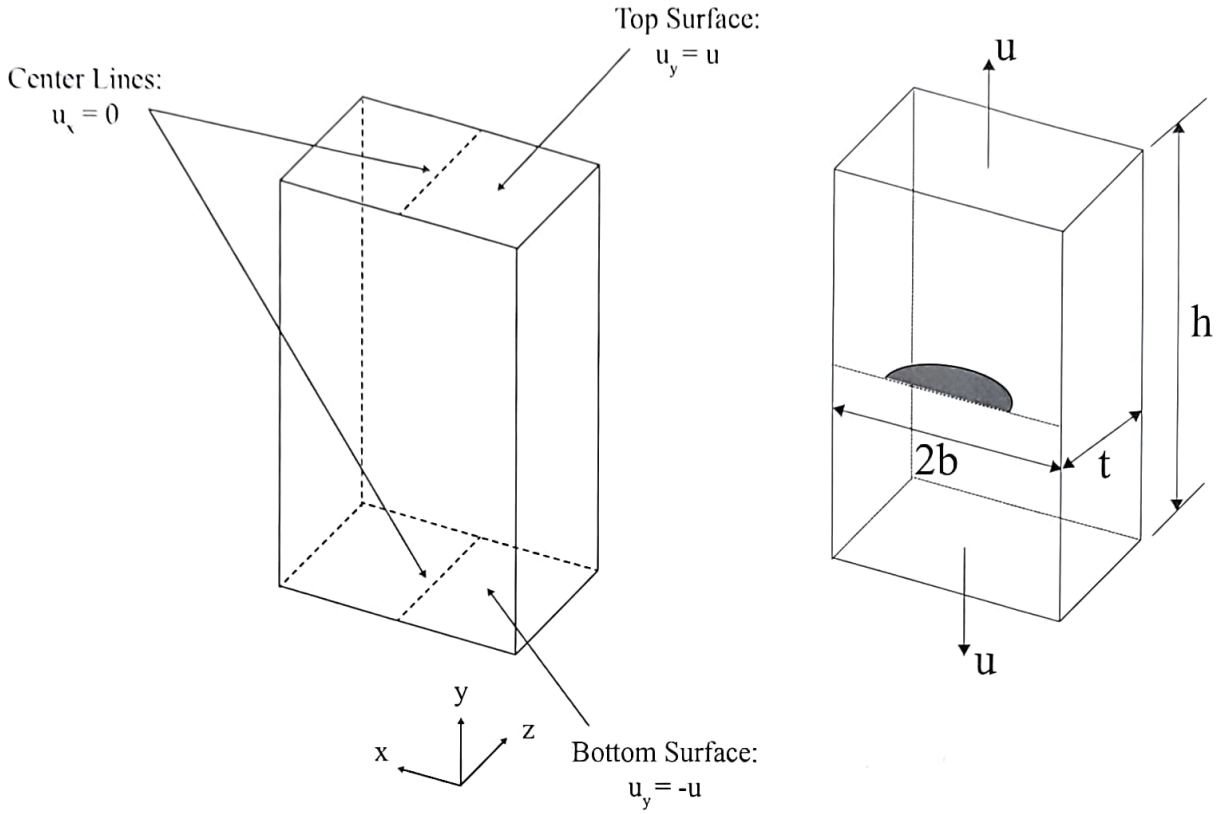
describe

The finite element models were designed to be as consistent as possible with the work presented in [8], while using modern ~~computational tools~~ software, ~~Using similar boundary conditions~~ enabling (~~BCs) allows for the~~ comparison between GPSR and the equations created in [8]. The cracked finite element models are entirely defined by the variables a, c, b, t, h, and u shown in figure 6. The far field stress, σ, is extracted from the reaction forces at the top and bottom surfaces. The variables a and c are crack parameters and the variables b, t, and h define the plate geometry, ~~of the plate~~.

In [8], Raju and Newman neglected effects due to finite height by assuming the height of the plate to be large relative to the thickness of the plate. To be consistent with this assumption, a constant value of $h/t = 64$ was chosen to remove any effects due to the height of the plate as shown in figure 7a. ~~As stated previously the BCs of the models were made to match the models from [8] without overly constraining the model as depicted in figure 6.~~ A constant displacement of $u = 0.001$ units was applied on the top and bottom surfaces in the direction of the surface normals. Additionally the center-lines of the top and bottom surfaces were held fixed in the x-direction to removed rigid body motion. U_2 constraint?

The crack mesh created by FRANC3D is shown in (??). The mesh contains a ring of quarter

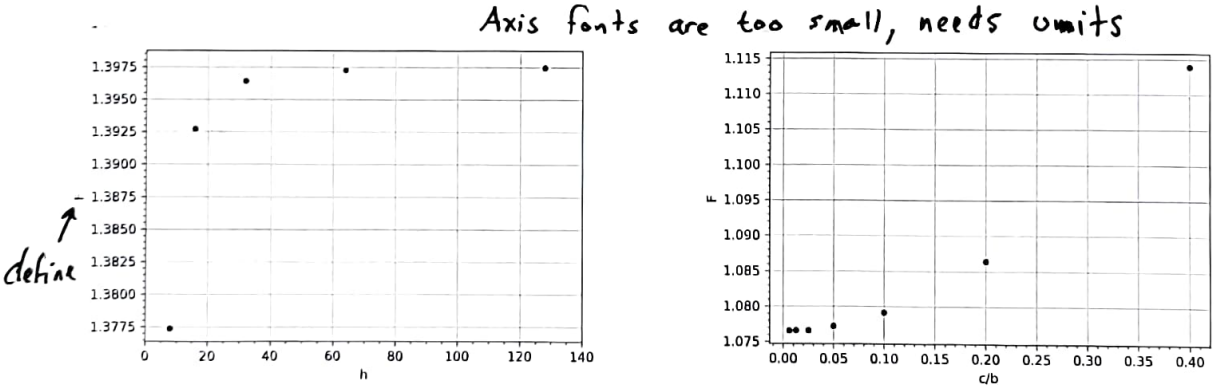
This application of BCs most accurately reproduced the conditions imposed in the quarter-symmetry model of Raju and Newman [8].



(a) Applied boundary conditions

(b) Model geometry

Figure 6: (a) Top and bottom surfaces with displacement in $+y$ and $-y$ respectively. The center lines of the top and bottom faces are held at zero displacement in x. (b) Model geometry plate hight, h, plate width, $2b$, and plate thickness, t.



(a) Convergence of h/t

(b) Convergence of c/b

Figure 7: (a) The convergence of h/t is plotted using the mean boundary correction factor. (b) The convergence of c/b is plotted using the mean boundary correction factor.

Feature	min	step	max
a/c	0.2	0.05	2
a/t	0.2	0.05	0.8
c/b	0.01	0.05	0.4
ϕ	0	$\pi/150$	π

Table 1: Max and min values for each of the features used when creating the cracked FE models

Thickness	1
h/t	64
Global seed	max(0.25, min(b/50, 0.5))
Local seed	0.25
Number of rings	8
Number of circumferential elements	14
Number crack front elements	~300

Table 2: Optimized model and mesh values

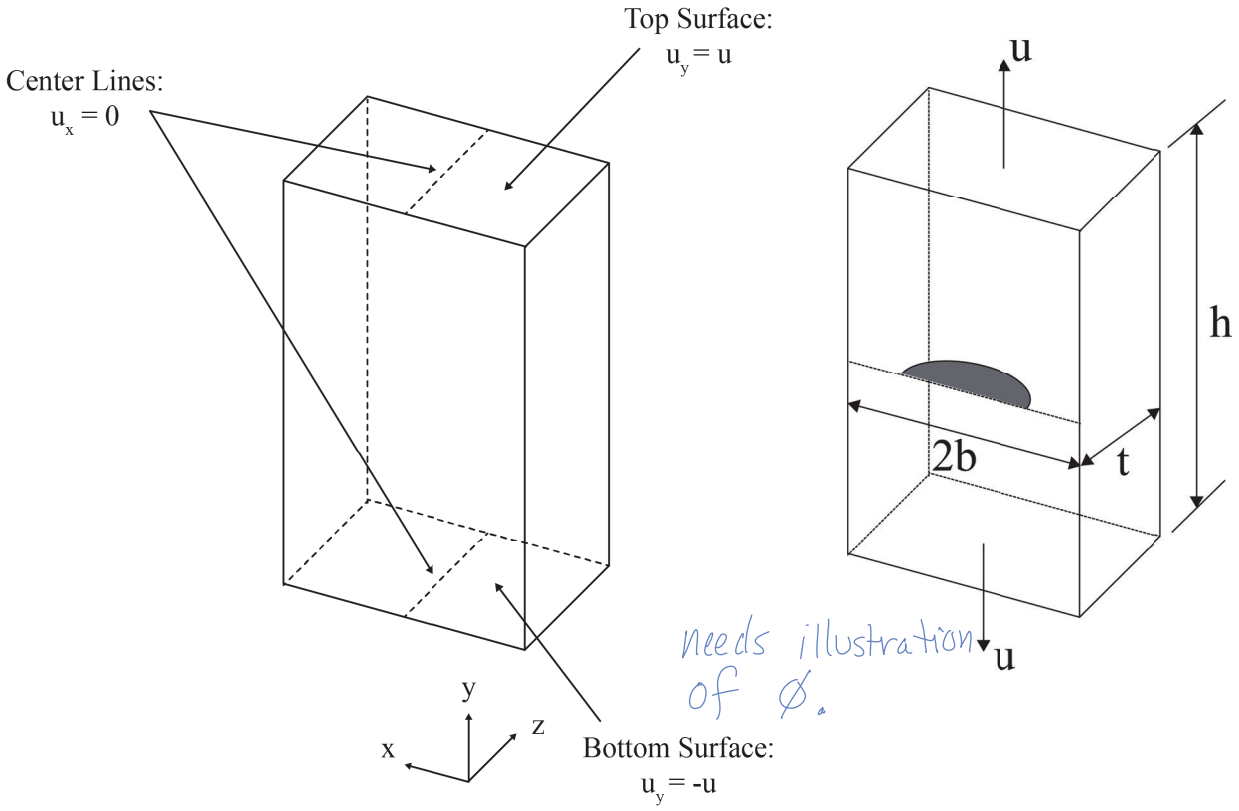
points elements around the crack front followed by rings of brick elements. The parameters for the crack mesh are the template radius which specifies the size of each element; the number of rings of elements around the crack; and the number of circumferential elements in each ring. Instead of using a fixed template radius, a number of elements along the entire crack was specified. This produced better results with varying crack aspect ratios. The best SIF values were computed with 300-600 elements along the crack, 8 rings of elements, and 14 circumferential elements per ring as shown in table 2. The average SIF value for each model is relatively insensitive to the crack mesh, however certain crack geometries display numerical noise for certain crack mesh sizes as shown in figure ??. The models that displayed the most numerical noise for the most crack sizes were the models with values close to the following: $a/c = 0.5$, $a/t = 0.6$, and $c/b = 0.2$. The crack mesh parameters were chosen based off of these models and then applied to the rest of the models. Before running the entire data-set, a random sample of 10% of the total models was computed to check for convergence. After convergence was confirmed on those models the entire data-set was computed.

The converged values are tabulated in table 2

After the converged model parameters were found, all 2500 of the models were run to calculate 300-450 SIFs per model. Figure ?? shows the calculated SIF values from a few of the models compared to the SIFs calculated from [21].

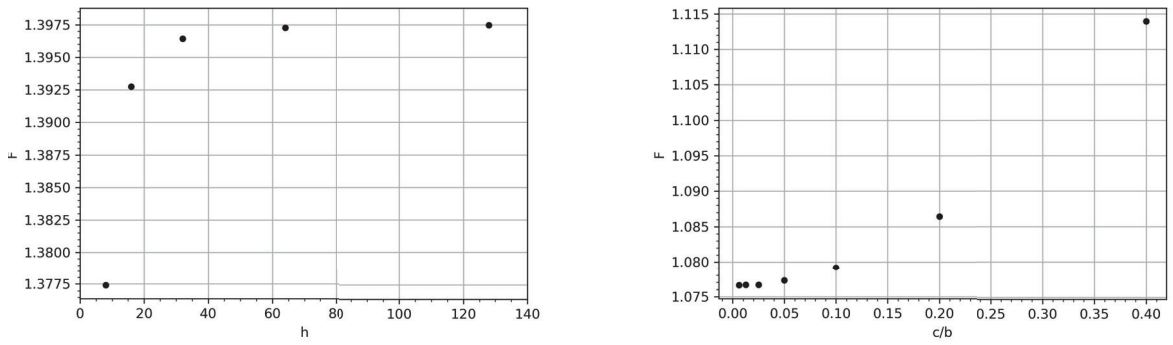
3.2 Training Data Subdivision

The 2500 models were broken up into train and validation data-sets, While a additional isolated set of 250 models were generated for testing to allow for better generalization error predictions. Instead of directly training on the SIF data extracted from FRANC3D this paper uses a mechanics based approach to simplify the training process which increased the explainablity of the final SIF model. The approach used in [8] will be used to break the



(a) Applied boundary conditions (b) Model geometry

Figure 6: (a) Top and bottom surfaces with displacement in $+y$ and $-y$ respectively. The center lines of the top and bottom faces are held 0 in x. (b) Model geometry plate hight: h, plate width: $2b$, and plate thickness: t.



(a) Convergence of h/t (b) Convergence of c/b

Figure 7: (a) The convergence of h/t is plotted using the mean boundary correction factor. (b) The convergence of c/b is plotted using the mean boundary correction factor.

Feature	min	step	max
a/c	0.2	0.05	2
a/t	0.2	0.05	0.8
c/b	0.01	0.05	0.4
ϕ	0	$\pi/150$	π

Table 1: Max and min values for each of the features used when creating the cracked FE models

Thickness	1
h/t	64
Global seed	max(0.25, min(b/50, 0.5))
Local seed	0.25
Number of rings	8
Number of circumferential elements	14
Number crack front elements	∼300

Table 2: Optimized model and mesh values

points elements around the crack front followed by rings of brick elements. The parameters for the crack mesh are the template radius which specifies the size of each element; the number of rings of elements around the crack; and the number of circumferential elements in each ring. Instead of using a fixed template radius a number of elements along the entire crack was specified. This produced better results with varying crack aspect ratios. The best SIF values were computed with 300-600 elements along the crack, 8 rings of elements, and 14 circumferential elements per ring as shown in table 2. The average SIF value for each model is relatively insensitive to the crack mesh, however certain crack geometries display numerical noise for certain crack mesh sizes as shown in figure **??**. The models that displayed the most numerical noise for the most crack sizes were the models with values close to the following: $a/c = 0.5$, $a/t = 0.6$, and $c/b = 0.2$. The crack mesh parameters were chosen based off of these models and then applied to the rest of the models. Before running the entire data-set, a random sample of 10% of the total models was computed to check for convergence. After convergence was confirmed on those models the entire data-set was computed.

The converged values are tabulated in table 2

After the converged model parameters were found all 2500 of the models were run to calculate 300-450 SIFs per model. Figure **??** shows the calculated SIF values from a few of the models compared to the SIFs calculated from [21].

3.2 Training Data Subdivision

The 2500 ~~models~~ were ~~broken up~~ into train and validation data-sets, ~~While a additional~~ isolated set of 250 models were generated for testing to allow for better generalization error ~~predictions~~. Instead of ~~directly~~ training on the SIF data ~~extracted from FRANC3D this paper uses~~ a mechanics-based approach to ~~simplify~~ the training process ~~which increased the~~ explainablity of the ~~final~~ SIF model. The approach used ~~in~~ [8] will be used to ~~break~~ the

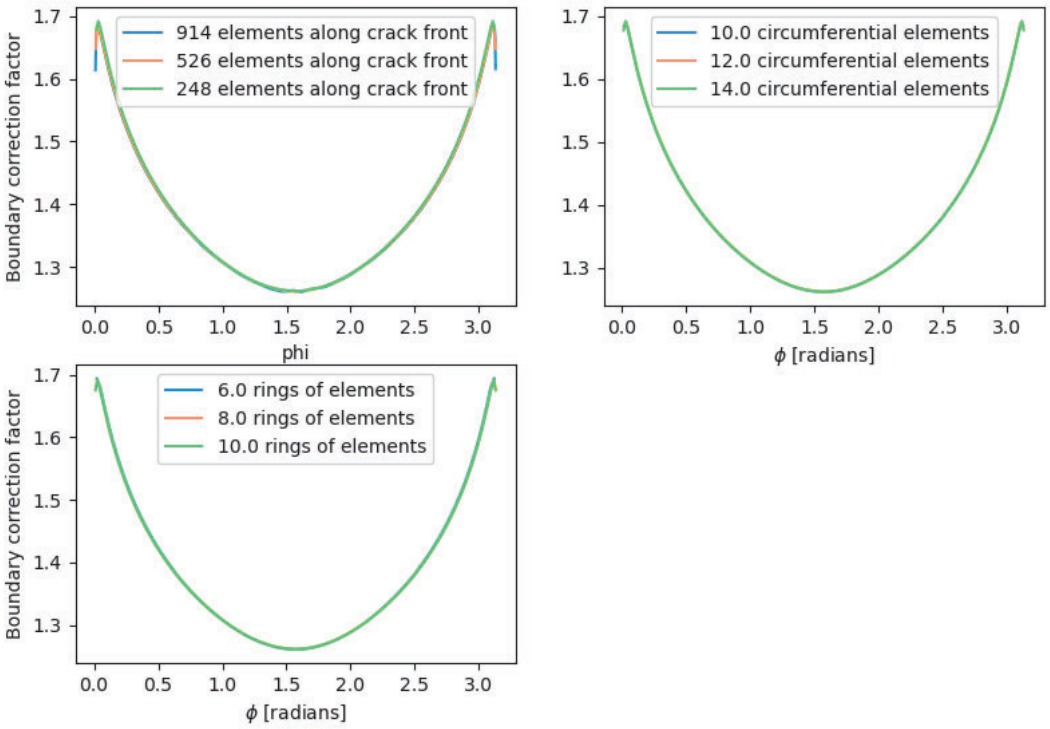


Figure 8: Mesh convergence for the crack mesh parameters

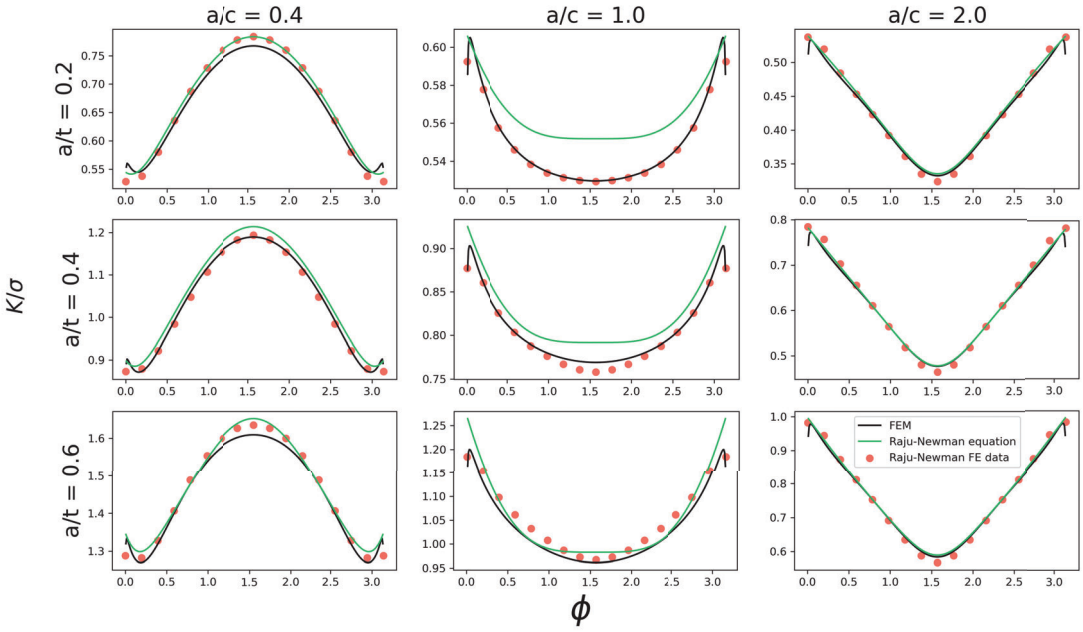


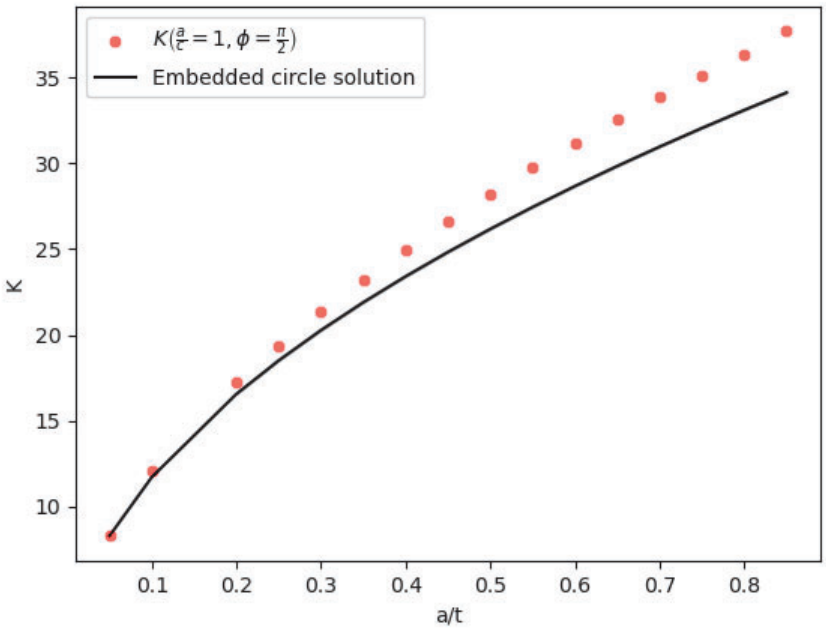
Figure 9: SIF values from a selection of models plotted with the Raju-Newman equation.

model for K into two parts: ~~the~~ known part and ~~the~~ learned part. The known part includes equation 1, which is the solution to the embedded ellipse in an infinite volume. The learned parts of ~~the equation~~ ~~is~~ the boundary correction factor ~~which is~~ comprised of ~~three sub-functions~~ f_w, M, and g, each accounting for a ~~different deviation from~~ the ~~infinite case~~. The finite width correction factor, f_w, accounts for the effect of finite width and thickness at the center of a semi-circular crack, ~~The~~ M ~~equation expands on f_w by~~ accounting for the aspect ratio of the crack. The two functions f_w and M together comprise the boundary correction factor ~~for the center of~~ a semi-elliptical crack in a finite plate. ~~Final the g equations takes~~ this correction at ~~the center and applies it to the rest of the~~ crack allowing for SIFs to be ~~predicted~~ along the ~~entire length of the~~ crack. GPSR is especially suited for this process as it finds closed form equations which can conform exactly to the known limits of each of the boundary correction terms significantly increasing interpretability.

Two modifications were made to the approach used in [8] to ~~make~~ the training process ~~easier and the resulting models more interpretable~~. ~~The first, is the method to~~ modify the ~~embedded ellipse equation to allow for $a/c > 1$. This was done~~ by using the function l from [13], ~~and modifying it for ellipses with $a/c \leq 1$ and $a/c > 1$ as seen in~~ equation 12

$$l = a\left(\frac{c}{\alpha}\right)^2 \sqrt{\left(\frac{a}{c}\right)^2 \cos^2\phi + \sin^2\phi}, \tag{12}$$

where a is the crack depth, c is half-crack surface length, and α is the length of the major axis of the ellipse. The function l is a ~~sort of pseudo-radius which is a~~ measure of the perpendicular distance from the tangent line at the point of interest to the nearest axis



← Figures should follow their text refs.

Figure 10: Comparison of K from the center of a surface crack and the solution to an embedded circle crack in an infinite volume as a function of a/t where c/b = 0.01, which has been determined to be close to 0 7 2 , Eqns 2 and 7-10,

seen in figure 1. The benefit of modifying the embedded ellipse equation by using equation 12 is that it removes the need for ~~any~~ piece-wise functions, ~~as the that is all handled by the variable a. This reduces the number of models that have to be trained from five as in [8] to three, one for each of f_w, M, and g.~~ such that a single continuous function can be used for f_w, M, and g. This approach reduces training time and lowers the complexity of the final K solution resulting in a more interpretable equation.

The second modification is the formulation of f_w. While Raju and Newman [8] used a ~~3D modification of a~~ finite width correction factor for a 3d through crack from [14], this work ~~will create~~ uses a finite width correction factor ~~specifically~~ for the case of a semi-elliptical surface crack. ~~To be able to use the process outlined in [8] The f_w function was created as similarly as possible to the one use by Raju and Newman in order to be able to use their method of breaking down the equation. This means that~~ f_w ~~will~~ is still be a finite width and thickness correction when $\phi = \pi/2$ and $a/c = 1$. However, like Raju and Newman [8],

Merge with previous ← ~~When~~ $\phi = \pi/2$ and $a/c = 1$ the value of K converges to the embedded ellipse equation as a/t and c/b go to zero, ~~as seen in figure~~ (??), ~~This results in f_w being defined as equation 13.~~

By noting that when

$$f_w\left(\frac{c}{b}, \frac{a}{t}\right) = \frac{K\left(\frac{a}{c}=1, \phi=\frac{\pi}{2}\right)}{\sigma \frac{2}{\pi}\sqrt{\pi a}}, \tag{13}$$

~~resulting in f_w correcting for finite width and thickness effects~~. The final equation for K ~~will be in the form of equation 14~~ is then

$$K = \sigma \frac{\sqrt{\pi l}}{E} f_w M g, \tag{14}$$

where f_w, M, and g are the boundary correction functions learned ~~by~~ using Bingo and $\sigma\sqrt{\pi l}/E$ is the modified embedded ellipse solution.

~~The process of extracting the training data from the SIF data calculated from the FE models is outlined here.~~ A decomposition of training data will now follow the defined correction functions and their dependence on geometric variables as

$$K\left(\frac{a}{c}, \frac{a}{t}, \frac{c}{b}, \phi, \sigma\right) = K_{ee}\left(\frac{a}{c}, \sigma, \phi\right) f_w\left(\frac{a}{t}, \frac{c}{b}\right) M\left(\frac{a}{t}, \frac{a}{c}\right) g\left(\frac{a}{t}, \frac{a}{c}, \phi\right) \tag{15}$$

~~The process of calculating the f_w training data was previously shown in equation 13.~~ The next step is to extract the training data for M, which is computed similarly to f_w. Specifically, the SIF data from FEA simulations are taken at $c/b = 0$ and $\phi = \pi/2$ then normalized by Eqns 1 and 13 as evaluated at $\phi = \pi/2$ resulting equation 16 that is only a function of a/c and a/t.

$$M\left(\frac{a}{t}, \frac{a}{c}\right) = \frac{K_{FE}\left(\frac{c}{b} \to 0, \phi = \frac{\pi}{2}\right)}{f_w\left(\frac{c}{b} \to 0\right) K_{ee}\left(\phi = \frac{\pi}{2}\right)} \tag{16}$$

Why not = ?

The final step is to extract the training data for g. This is done by ~~taking SIFs from FEA at $c/b = 0$ and dividing by 1, equation 13 and equation 16~~ Normalizing by Eqns 1, 13, and 16 as

$$g\left(\frac{a}{t}, \frac{a}{c}, \phi\right) = \frac{K_{FE}\left(\frac{c}{b} \to 0\right)}{K_{ee} f_w\left(\frac{c}{b} \to 0\right) M} \tag{17}$$

Forcing g to only be defined at $c/b = 0$ matches the Raju-Newman equations ~~however it does not completely model K if done this way~~. Additional models were trained where g is allowed to be a function of c/b. for a more accurate SIF model.

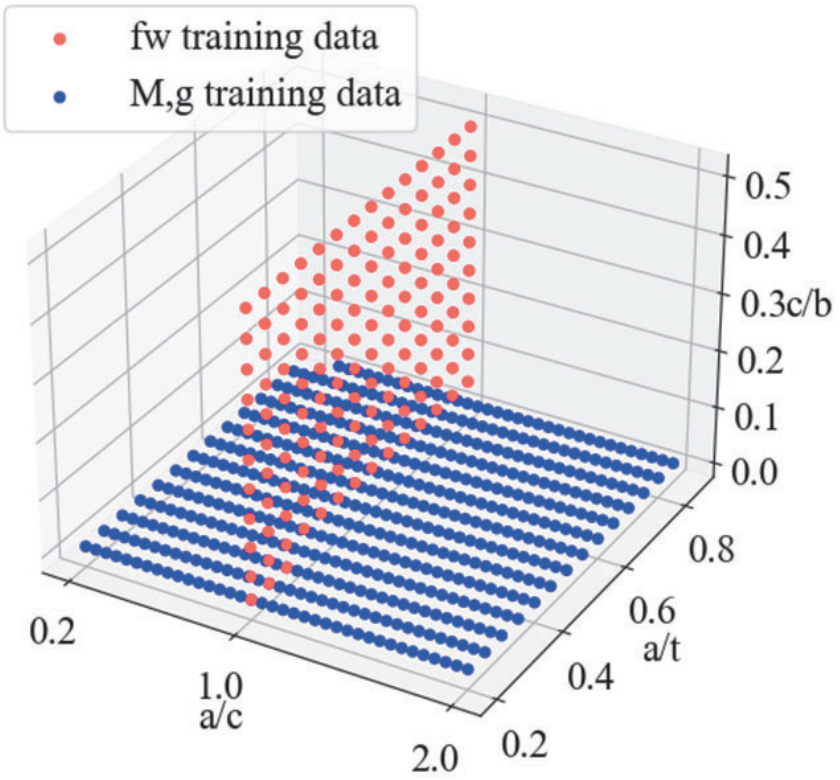
This section needs a description of the hyperparameters used and how those were determined.

3.3 Training Bingo Models

Due to the mechanics based ~~approach of~~ breaking down ~~the SIF solution into multiple sub-functions, only~~ a ~~small~~ reduced ~~portion~~ subdomain of the ~~full data~~ feature space is ~~required to train accurate models~~ used for training of training data in Eqs. 13, 16, and 17. The function f_w is trained on ~~the slice of the~~ data where $a/c = 1$ and $\phi = \pi/2$. The function M is trained on ~~the slice of the~~ data where $c/b = 0$ and $\phi = \pi/2$. ~~With~~ The function g ~~being~~ is trained on ~~the slice of the~~ data where $c/b = 0$. ~~Reducing the data that each model is trained on reduced the space that is searched by bingo allowing less complex models~~. Figure ?? ~~gives a visual representation of~~ shows this training data reduction. In this figure each point represents a ~~different~~ simulation FE ~~model~~, each having ~~many~~ SIF values ~~corresponding to each ϕ value along the crack~~ along the crack front, decomposing into the correction functions. One of the simplifications Raju and Newman used when ~~breaking up~~ decomposing the SIF ~~function~~, was ~~to assume~~ that ~~all of~~ the effects of c/b could be ~~predicted~~ modeled by the fw equation, ~~essentially~~ neglecting ~~any~~ interaction between ~~both~~ c/b and ϕ. ~~For the most part this is a good assumption as the models they created were very predictive~~. To understand the effect of this assumption ~~be thorough~~ an additional set of Bingo models was trained that allowed c/b to be used in ~~the~~ g ~~equation that would allow for the capture of any of these~~ to model c/b with β. interactions ~~involving c/b~~. These additional g models were trained using the entire ~~dataspace defined by the FE models~~. (not reduced) data set.

While this might be true, it should go in discussion or conclusion after supporting with results.

Explain why with an additional sentence.



subdomain used for training SIF correction functions.

Figure 11: ~~Visualization of the~~ Training data ~~reduction allowed by the mechanics based approach~~. Each point represents a ~~model in the training data-set~~ FE simulation. f_w was trained when $\phi = \pi/2$ and $a/c = 1$, M was trained when $\phi = \pi/2$ and $c/b = 0$, and g was trained when $c/b = 0$.

This paragraph needs a statement of the fitness function (e.g., MSE) used.

During the training process, two techniques were employed to ~~enhance the predictive ability~~ improve accuracy and interpretability of the models. The first of these techniques involved the use of custom fitness functions to enforce the known constraints of each ~~equation~~ model. ~~These fitness functions assess the suitability of each candidate equation at the end of each training generation. The use of custom fitness functions ensures that all equations identified by the bingo algorithm adhere to the known limits of each equation, thereby improving interpretability.~~ For instance, ~~in equation 13,~~ as the values of a/t and c/b approach zero, the value of f_w ~~converges to~~ must approach 1. A fitness function that calculates the loss when this condition is met and assigns an infinite loss when it is not met enforces the equation's limits. Similar limit constraints apply to M and g, such as $M(a/c = 1) = 1$ and $g(\phi = \pi/2) = 1$. The other method used to modify the training process was the ~~addition~~ introduction of seed functions in the initial population. ~~In the initial generation of the Bingo algorithm,~~ Instead of instantiating, a population entirely of randomly generated ~~equations~~ models ~~is created. Using~~ seeded ~~equations~~ functions allows the user to designate a fraction of these randomly generated ~~equations as pre-set equations~~ models using predefined forms. ~~This enables users to provide guidance to the algorithm.~~ For example, upon ~~visual examination~~ initial inspection of the training data for g, it was observed that $(1 - \sin\phi)$ ~~appeared to fit the data well.~~ modeled bounding conditions on g and $(1-\sin\phi)^{1/2}$ models the data across its domain. ~~Additional Bingo models were trained with this function seeded into the initial population. Other Bingo models were trained with only $1 - \sin\phi$ while other had no seeding at all.~~ While these seed functions are not required, their definition when known can improve performance. To assess this, we train bingo models with and without these seed functions.

Training Bingo on the functions f_w and M was relatively fast only requiring 1-2 days due to the small training data-set, which only contained one data point per model when $\phi = \pi/2$. Additionally, these functions only required the discovery of relationships between two variables, namely, c/b and a/t for f_w, and a/c and a/t for M. However, training the function g presented a different challenge due to the substantial increase in the number of data points as this function is now trained on every value of ϕ in each model. The Bingo algorithm tends to slow down as more data points are added. This results in a data-set comprising 300-400 data points per model, in contrast to the single data point per model for f_w and M. To expedite the training of ~~the~~ g ~~function~~, two strategies were employed. The first involved down-sampling the ϕ data to reduce it to only 20 data points per model. The second strategy made use of a Bingo feature called the fitness predictor island (FPI). FPI aims to identify the optimal subset of the data that best represents the entire data-set. The size of this subset is determined by a hyper-parameter that specifies the fraction the subset. Even with theses two methods of reducing the number of data-points Bingo still took 5-7 days to train the g models that did not contain c/b and up to 13 days for the models that did contain c/b.

Once ~~each~~ the training ~~data-set~~ has been ~~computed bingo can be trained to find~~ generated with the described splits and reductions all three ~~equations for~~ models, f_w, M, and g can be trained simultaneously. Multiple models were trained for each of the functions, due to the randomness of genetic programming this gave new equations that could be compared to one another to find the best combinations of f_w, M, and g that predict SIF. The training process is done by first taking the SIF data computed from the FE model and then extracting the f_w, M, and g training data-sets. Then bingo equations can be found for all three equations, these equation are then combined with the embedded ellipse solution from equation 14 to create a prediction for SIFS, this process is visualized in figure ??.

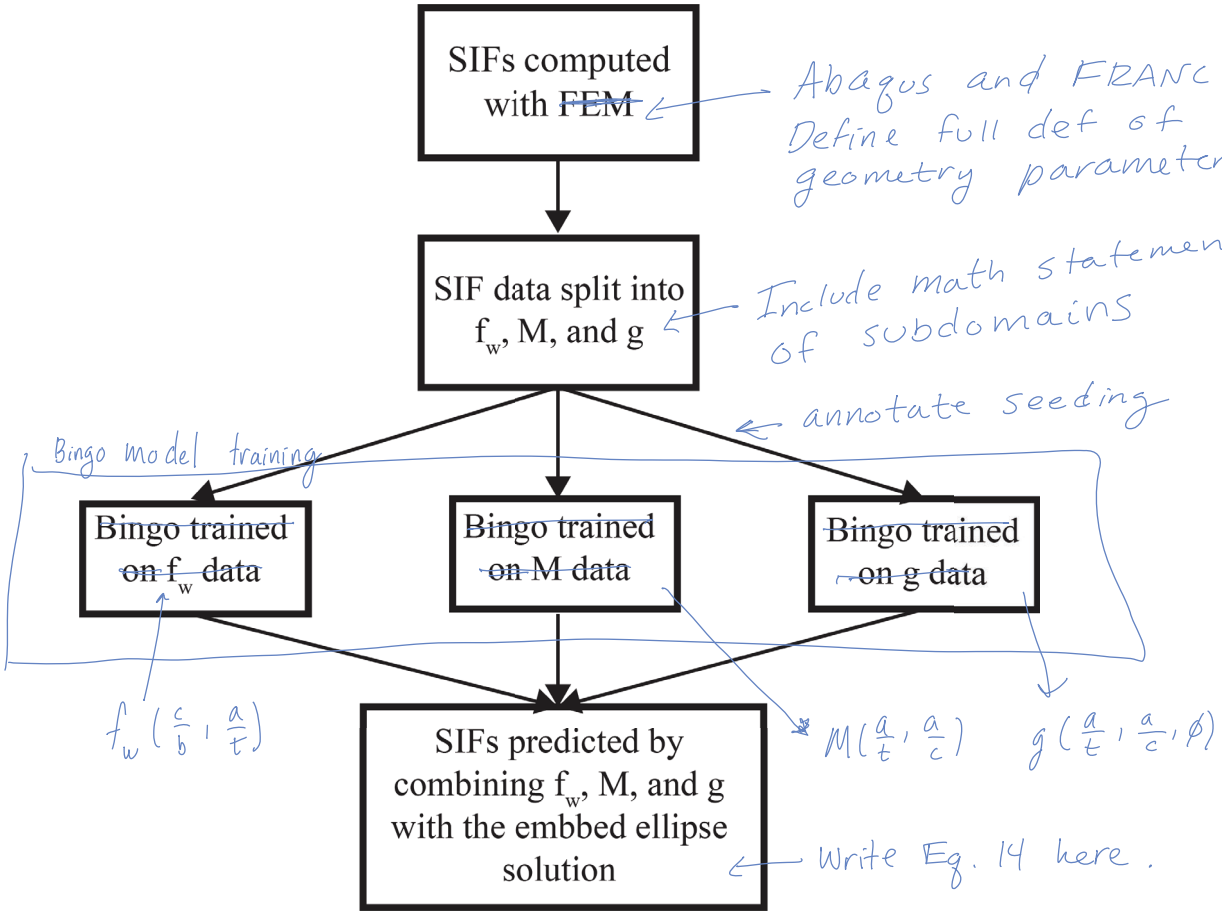


Figure 12: Flow chart illustrating the training process ~~of~~ for the three ~~sub-functions~~ models f_w, M, and g.

4 Results

When the Bingo algorithm has finished, it groups the best-performing equations in a Pareto front, where the x-axis is the complexity of each equation and the y-axis is the fitness of each equation. Models with a smaller complexity tend to be more interpretable. However, they will often have higher errors due to under-fitting. The Pareto front allows the models with a balance of fitness and complexity to be found. There are two ways that the final K model can be selected. The first is by selecting each model for f_w, M, and g individually and then combining them into the final model for K. The other way is to calculated all the possible combinations of the candidate f_w, M, and g. to create a new Pareto front for K. This Pareto front can then be used to select the model. both of these methods have The following equations were chosen to minimize the complexity while still providing at least as good fitness as the equations from [8].

Four equations were chosen for both of the model selection methods. This is done to create multiple comparisons points between GPSR and the Raju-Newman equations. To lower the number of equations presented for the individual selection process the same f_w (eqn 18) and M (eqn 19) equations are used for all four models with only the g equation changing.

$$f_w = \sqrt{\frac{\sqrt{\frac{a}{t} + \cos\left(\frac{a}{t}\right)}}{\cos\left(\frac{c}{b}\right)}} \qquad (18)$$

$$M = 1 + 0.06\left(\frac{a}{t}\right)^2\left(\frac{a}{c} - 1\right) + 0.28\frac{a}{t}\left(\frac{a}{c} - 1\right) + 0.0069\left(1 - \frac{c}{a}\right) \qquad (19)$$

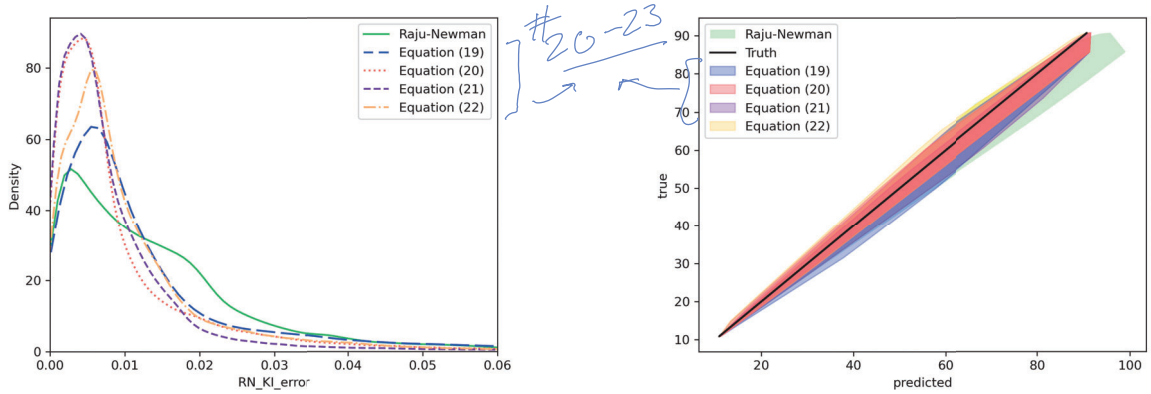
Using equation 20 for g creates a model for K that has similar error to the Raju-Newman equation. Equation 21 results in an a model with the same overall complexity as the Raju-Newman equation. The first model uses the data where g was trained at $\frac{c}{b} = 0$ and .

Many Bingo runs were done with different parameters. The candidate equations can be selected from the combined Pareto front of all the bingo runs.

Bingo equations are selected to compare with the Raju-Newman equations. First by matching the complexity of the equation and second by matching the error of the equations the equations f_w and M have a smaller effect on the value of KI than g, thus for consistency the same equations for f_w and M will be used for multiple models. Equation 20 is the g equation that produces a model with comparable error to the Raju-Newman equation. Equation 21 results in a model with the same complexity as the Raju-Newman equation. Equation 22 also matches the overall complexity; however, the g model was allowed to be trained on c/b. Equation 23 has a balance of error and complexity.

$$g = 1 + \frac{a}{t}\frac{1 - \sqrt{\sin(\phi)}}{\frac{a}{c} + 1} \qquad (20)$$

Include figures after text discussion.



#20-23

(a) ~~Distributions of error~~ (b) Parity plots

Figure 13: (a) Density plot for the errors of each of the equations. (b) Parity plot with each of the different equations.

Simplify expressions

$$g = 1 + 0.0316\frac{a}{c}\left((\sin(\phi) - 0.53)\left(\left(\frac{a}{c}\right)^{1}.54 - 1.18\frac{a}{t} + 1.35\right) - 2.6\right)\left(\sin^{\frac{a}{t}+0.175}(\phi) - 1\right) \frac{0.53\left(\frac{a}{c}\right)^{1}.54 - 0.625\frac{a}{t} + 3.32}{\left(0.739\left(\frac{a}{c}\right)^{1}.54 - 0.87\frac{a}{t} + 1\right)^{2}} \tag{21}$$

$$g = 1 + \frac{\left(0.78\left(0.66 - 1.37\sin\left(0.32\frac{a}{c}\frac{a}{t}\left(\frac{a}{t} - \frac{c}{b}\right) - \frac{a}{t} + 1.76\right)\right)(\sin(\phi) + 6.1) + 4.02\right)(\sin(\phi) - 1)}{\left(0.656 - 1.37\sin\left(0.32\frac{a}{c}\frac{a}{t}\left(\frac{a}{t} - \frac{c}{b}\right) - \frac{a}{t} + 1.76\right)\right)(\sin(\phi) + 6.1)} \tag{22}$$

$$g = 1 + \frac{0.4\frac{a}{c}\left(1 - \sin(\phi)^{\frac{a}{t}}\right)}{\left(\frac{a}{c}\right)^{2} - \frac{a}{t} + 1} \tag{23}$$

Figure ?? shows the Pareto fronts for all 3 equations. The distributions of the errors for each of the four models along with with Raju-Newman equation are shown in figure 13. The parity plots for each model is also shown in figure 13.

The Pareto front for the model selection using the Pareto front generated using all possible combinations of the models f_w, M, and g can be seen in figure ??. Equations [24-26] are the equations gathered from the Pareto front of all possible equations. The error distributions and parity plots for these models can be seen in figure 17.

$$\begin{cases} f_w = 0.13\frac{c}{b} + 0.13\frac{a}{t} + 1.0 \\ M = -0.223\frac{a}{t} + 1 + \frac{0.223\frac{a}{t}}{\frac{a}{c}\frac{a}{c}} \\ g = 0.554\frac{a}{t}\left(1 - \sin^{0.491\frac{a}{c}}(\phi)\right) \end{cases} \tag{24}$$

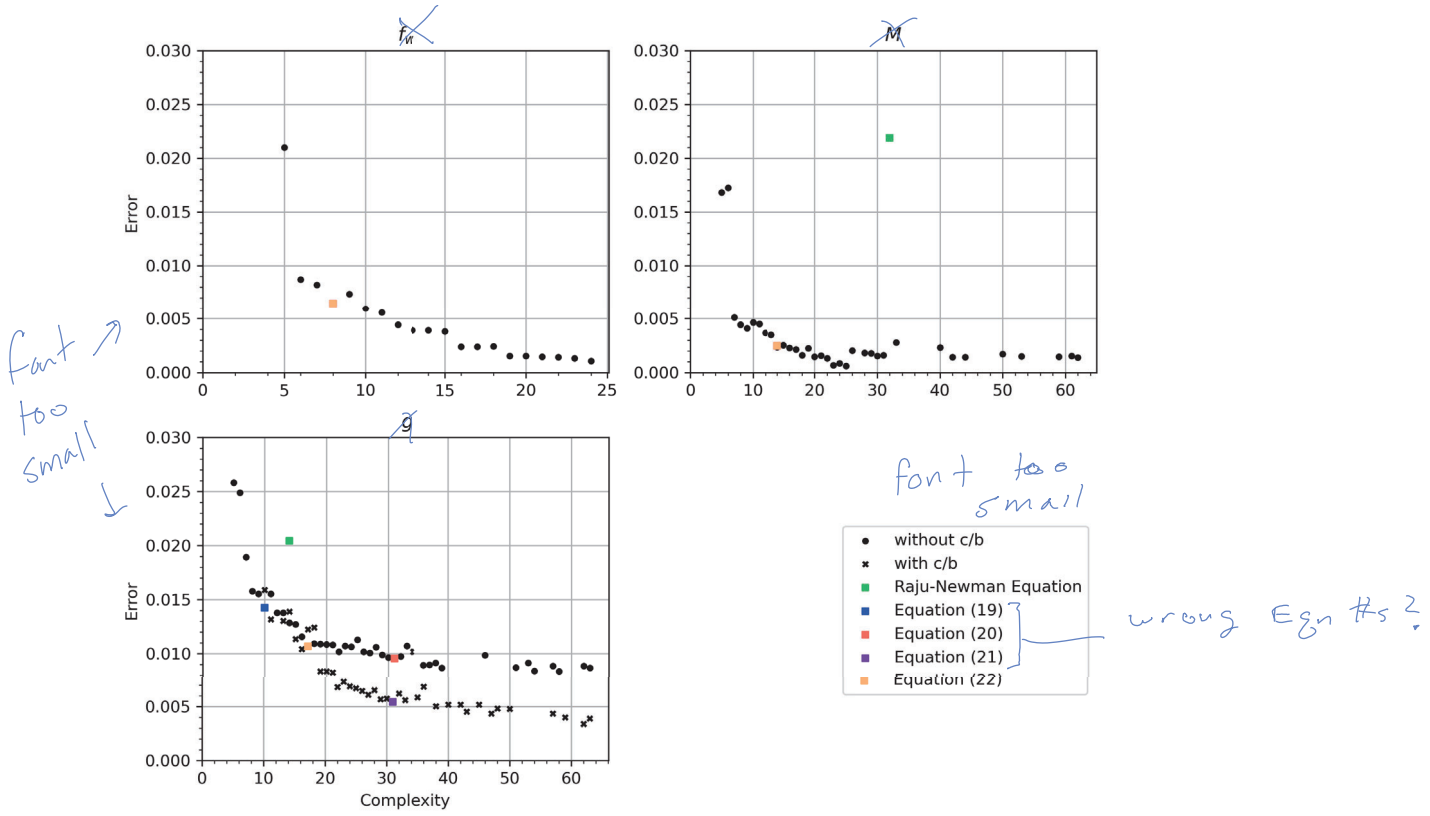
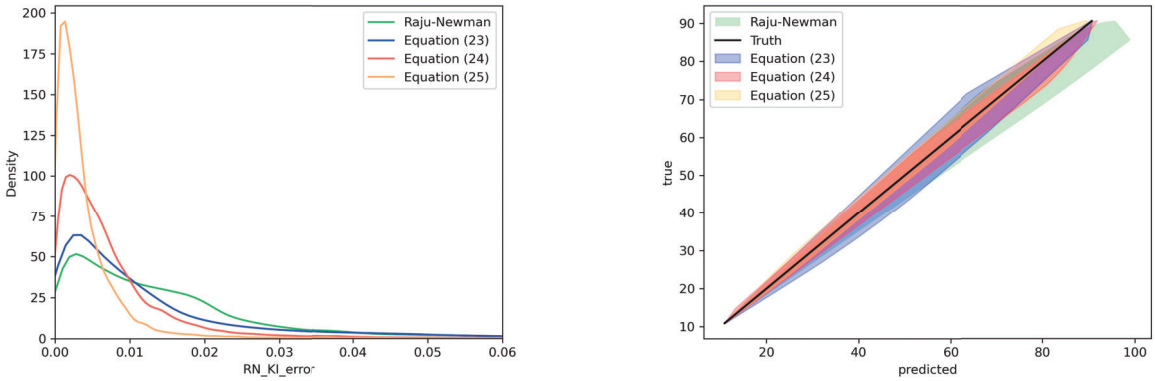


Figure 14: Pareto fronts for f_w, M, and g. f_w and M have the same equation for all four models. Both the Pareto fronts for g with and without c/b are shown



(a) Distributions of error

(b) Parity plots

Figure 15: (a) Density plot for the errors of each of the equations from the combined selection. (b) Parity plot with each of the different equations from the combined selection.

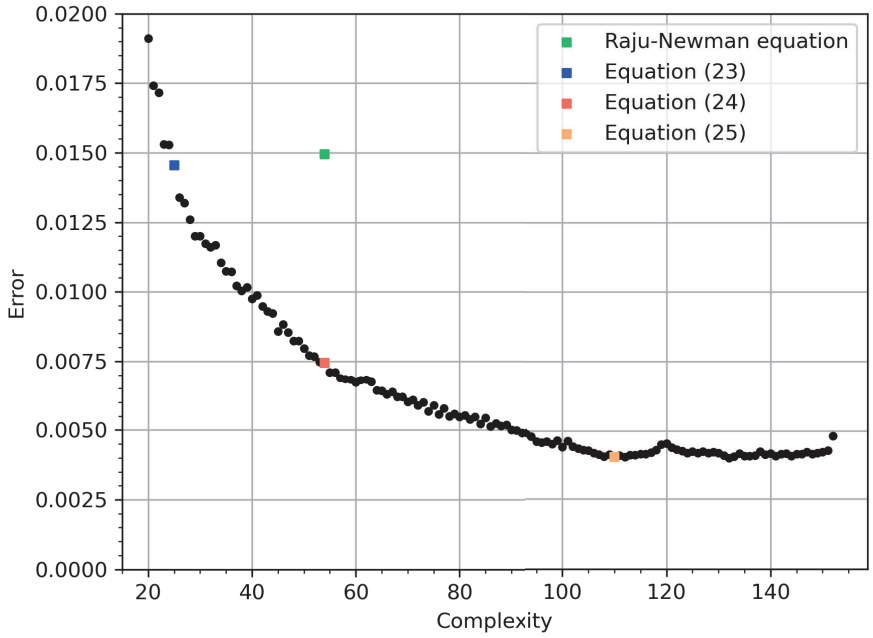


Figure 16: Pareto front for the models selected using all possible combinations.

If these equations are not interpeted or discussed, consider moving to an appendix.

$$\begin{cases} f_w = \frac{-63.94\frac{c}{b}+51.875\frac{a}{t}+38.855}{-63.94\frac{c}{b}+43.336\frac{a}{t}+38.855} \\ M = \frac{-0.028\frac{a}{c}^3+\frac{a}{c}^2\left(\frac{a}{c}+0.028\right)-0.644\frac{a}{c}\frac{a}{t}\left(\frac{a}{c}\frac{a}{t}-0.044\right)\left(-\frac{a}{c}\left(\frac{a}{t}-1.103\right)+0.044\right)+0.644\frac{a}{t}\left(\frac{a}{c}\frac{a}{t}-0.044\right)\left(-\frac{a}{c}\left(\frac{a}{t}-1.103\right)+0.044\right)}{\frac{a}{c}^3} \\ g = -\frac{0.127\frac{a}{c}\left(\sin\frac{a}{t}\left(\phi\right)-1\right)}{\left(\cos\left(\frac{c}{b}\right)-0.66\right)\left(\frac{a}{c}^{1.811}-1.066\frac{a}{t}+0.992\right)} \end{cases} \quad (25)$$

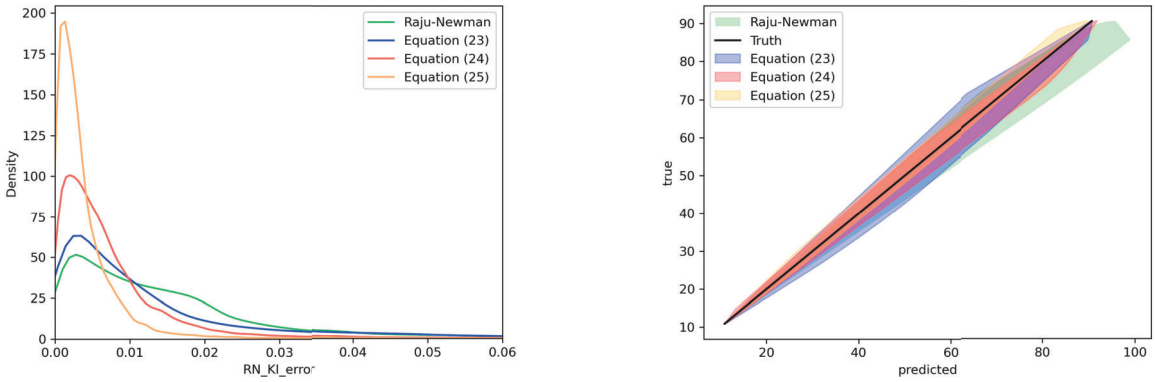
expand / simplify

$$\begin{cases} f_w = -\frac{a}{t}\left(\left(1.216\frac{c}{b}^2+0.11\right)\left(\frac{a}{t}-\sqrt{-\frac{c}{b}^2-0.751}\right)-0.119\right)+1.0 \\ M = \frac{a}{c}\frac{-0.037\frac{a}{c}-0.046\frac{a}{t}\left(\frac{a}{c}^2-15.413\right)\left(\frac{a}{t}-1.114\right)\left(1.919\frac{a}{t}-0.489\right)-0.06}{\frac{a}{c}+0.599} \\ g = -\frac{1.232\left(\frac{a}{c}\left(\frac{a}{c}+0.06\right)\left(0.103(1-\sin\left(\phi\right))^{\frac{3}{2}}+0.017\right)-0.36\left(\frac{a}{t}-0.187\right)\left(\frac{a}{c}\left(\left(\frac{a}{c}+0.06\right)\left(0.017\frac{a}{c}+0.033\frac{a}{t}-0.409\frac{c}{b}^2+0.061i\right)-0.646\right)\left(\frac{a}{c}+2\frac{a}{t}+3.67\right.\right.\right.}{\frac{a}{c}\left(\frac{a}{c}+0.06\right)} \end{cases} \quad (26)$$

Finally figure ?? shows all eight models on the same Pareto front.

5 Discussion

A benefit of GPSR is that it produces a population of equations. This allows the user to select the equation that best fits their needs. Typically the best equation will be one that is a balance of simplicity and accuracy. However, some use cases may require maximum



(a) Distributions of error (b) Parity plots

Figure 17: (a) Density plot for the errors of each of the equations from the combined selection. (b) Parity plot with each of the different equations from the combined selection.

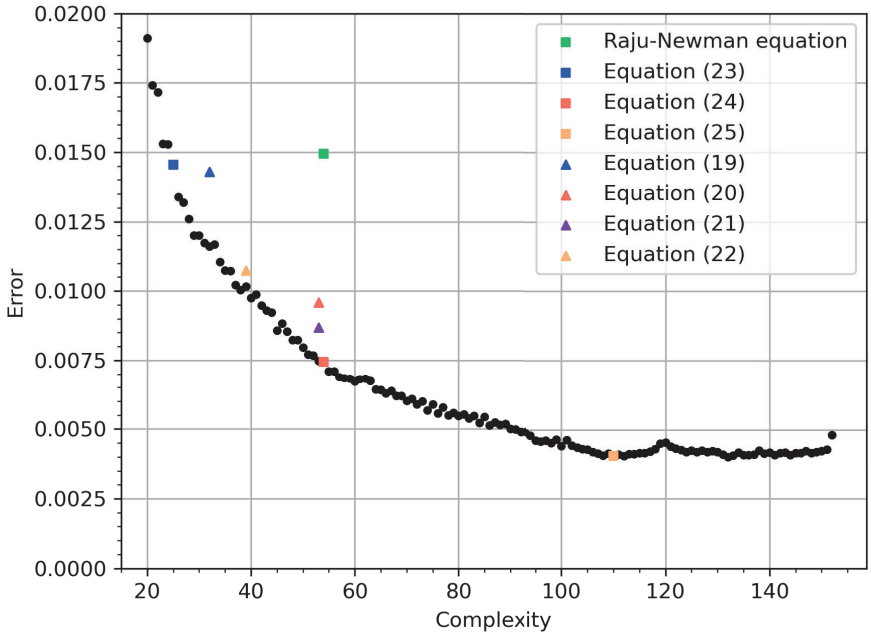


Figure 18: Pareto front with all models the triangle markers correspond to the models that were individually selected and the square markers are the models selected using all possible combinations.

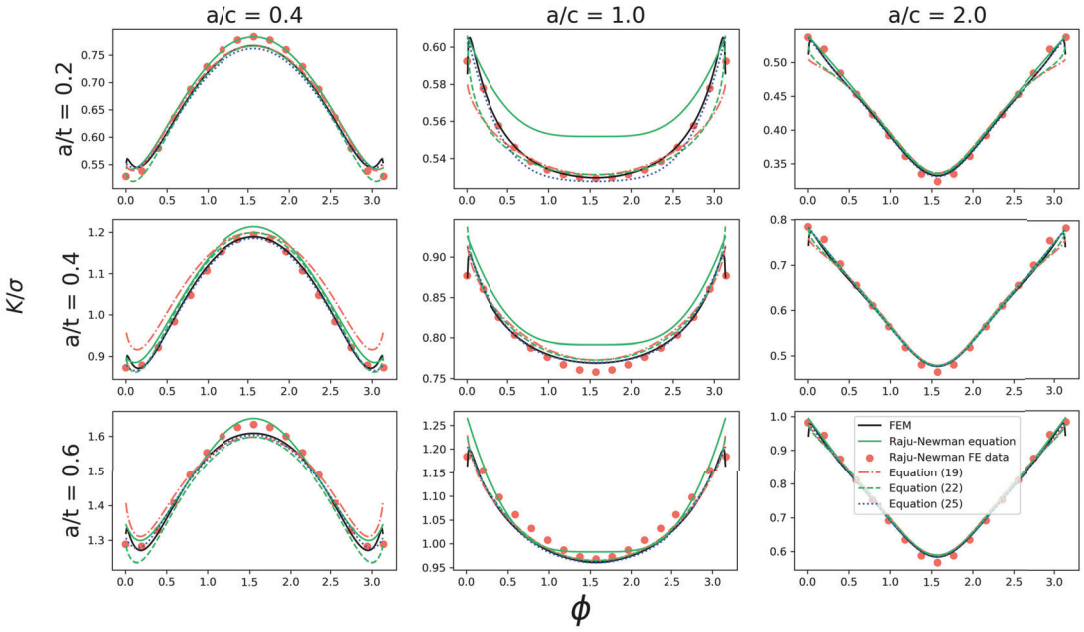


Figure 19: A sample of models comparing the SIFs collected from FRANC3D with the bingo equation, the Raju-Newman equation, and the Raju-Newman FE data.

Most text on this page is a repeat of a few pages back. Avoid writing the same thing twice by referencing previous sections with brief reminders, as needed.

~~accuracy or need very simple equations~~. A variety of models have been presented here that have differing accuracy and complexities. Two methods were used to create the final SIF equation from the f_w, m, and g equations. The first method was to use fixed f_w and m equations as the g equation has the largest effect on the overall SIF accuracy as this data-set is the most complex the equations extracted using this method are shown in figure ??. The second method was to use every possible combination of the f_w, m, and g equations created by bingo the extracted equations are shown in figure ??. To be able to compare with the Raju and Newman equation, models were selected that have similar error and similar complexity figure ?? compares all the bingo models with the Raju and Newman equation. From this result Bingo outperforms the Raju and Newman equation both in complexity and error. The models with similar complexity have much lower errors and the models with similar errors have much lower complexities.

all run-on sentences here.

as compared to what

When using the same method as Raju and Newman to break down K using f_w, M, and g, the effect of c/b is not fully captured. The finite width of the model is only accounted for in the function f_w, which is not a function of ϕ meaning that there is no coupling between ϕ and f_w. This results in what is shown in figure ?? where the g models that are allowed to use c/b outperform the models without c/b. This effect, however only occurs for models with complexity around 16 or higher. This is because the lower complexity models will under-fit in other aspects making the c/b effects much less important. From figure ?? it can be seen that after all the equations have been multiplied together with the equation for the embedded ellipse to produce SIF, that the difference between the models that allow g to use c/b and

specifically,

New discussion starts here.

the models that don't allow g to use c/b have almost identical error. This result means that the assumption of c/b being constant in ϕ is fairly accurate and does not result in decreased accuracy in the final SIF solution.

5.1 Model limitations

It was observed that for all of the trained bingo models the maximum errors occurred both at the surface of the crack where ϕ is close to 0 or π and when $a/c = 0.2$. At the surface of the crack where $\phi = [\pi, 0]$ the SIF values dive due to plane stress vs plane strain conditions. The bingo model is unable to fully capture this region due to lack of data at the surfaces, as there are only a handful of data-points at the surface for each model. It could be possible to pick up these surface effects by adding another correction factor that only accounts for the surfaces of the model. The reason for the highest errors occurring at $a/c = 0.2$ is because this is the lower bound of the training data, and there is also a sharp change in the SIF data in this region, and without information about the SIFs smaller than $a/c = 0.2$ in the training data Bingo fits this region of the data the worst. Adding smaller aspect ratios into the training data would likely lower the the maximum error of the model, however, $a/c = 0.2$ is already a quite small aspect ratio and rest of the domain with aspect ratios above $a/c = 0.2$ has very low error.

highest · does not · behavior · near · effects · at $a/c < 0.2$ would improve this behavior.

This section should have three paragraphs: 1) Why a comparison is being made 2) Compare accuracy 3) Compare runtime

5.2 Comparison with Black Box methods

Unclear what is being said here.

Comparing train, validation, and test errors shows that Bingo produces similar errors across all three data-sets indicating that it's generalization error is similar to the error values displayed. The black box methods, however, show very different errors depending on the data set, meaning that these models tend to over-fit the data. Bingo is able to produce the lowest errors without over-fitting the data. Comparing the evaluation times of each of the ML models shown in ?? shows that bingo and the Raju-Newman have similar evaluation times; however, the black-box models are slower with the random forest taking 3 seconds to evaluate the data set, and SVM taking 40 seconds. These results that Bingo and the Raju-Newman equation outperform the black box methods both in generalization error and evaluation time. This shows that the method of breaking the SIF equation down into multiple sub-functions results in a better model than can be created from just training on the entire data-set.

model

incomplete sentence

Comparing the results shown here from Bingo with other ML methods such as the ones used in [?]. Figure ?? shows Bingo and the Raju-Newman equation compared with two commonly used ML methods for a data-set of this size, support vector machine regression (SVM) and random forest regression (RFR). Using the optimal hyper-parameters from [?] and training directly on K results in the square markers on the plot. Training the models again using the mechanics bases approach used in this research, results in the circle points. From this figure, the error of the RFR is reduced by using the mechanics based approach, while the SVM produces worse error when using the mechanics based approach. One possible reason that the SVM produces a worse model when using the mechanics based approach, is that it cannot fit to the smaller data-sets, while Bingo and RFR are easily able to fit to fewer data-points. It is shown in [?] that as the data-points decrease to 100 that SVM performs

worse. Being able to produce accurate models with fewer data-points is a strength as having many training examples is rarely feasible.

Bingo outperforms SVM and RFR both in evaluation time and error, while also producing a closed for equation than can be explained. The Bingo models only need a small subset of the data to be able to predict SIF for the entire domain. Bingo does, however, have a downside in its long training time on the order of hours to days, while SVM and RFR can be trained very quickly on the order of seconds to minutes. Bingo takes more effort than SVM and RFR to create a model for this data-set, however, the final model created by Bingo outperforms SVM, RFR, and the Raju-Newman equation in all metrics.

6 Conclusion

Using knowledge of the mechanics of the problem with Bingo allows for closed for equations that accurately predict Stress intensity factors in a semi-elliptical surface crack. This methods is able to outperform the equations developed by Raju and Newman both in accuracy as well as Bingo allows for more simple equations. When compared with other machine learning techniques the bingo models show that they are able to compete in accuracy while having significantly lower evaluation times. One limitation of Bingo is that the training time is much longer in the range of 10s of hours vs just a few minutes for other black box machine learning methods.

The interpretable nature of the equations found by Bingo allows for their results to be more easily defensible resulting in better decision making. Because the Bingo models are closed for equations they can be easily implemented in place of other empirical equations such as the Raju-Newman equations, providing easily accessible accuracy increases, as opposed to black box models which would require much more work to have their models implemented in existing methods.

The work here shows that using knowledge about the mechanics of the problem being learned increases the accuracy of machine learning

GPSR has been shown to be able to create interpretable models that can predict stress intensity factors. By using knowledge of the mechanics of the problem to break the problem down allows for more accurate and intepretable models than what can be created by simply training on the data directly from finite elements.

By using the mechanics bases training method more accurate and interpretable models can be created using GPSR, than other typically more expressive black-box methods. GPSR models also have the benifit of increased evaluation time do to the closed form nature of the equations. The only downside of GPSR

Bingo was able to produce equations that outperform the Raju-Newman equations in both accuracy and complexity. By using the mechanics based approach of breaking down the problem into subfunctions that

Complete paragraphs are needed.

References

Incomplete refs are circled.

[1] X Wang and S B Lambert. Local weight functions for semi-elliptical surface cracks in finite thickness plates, 1995.

can't be in middle of refs

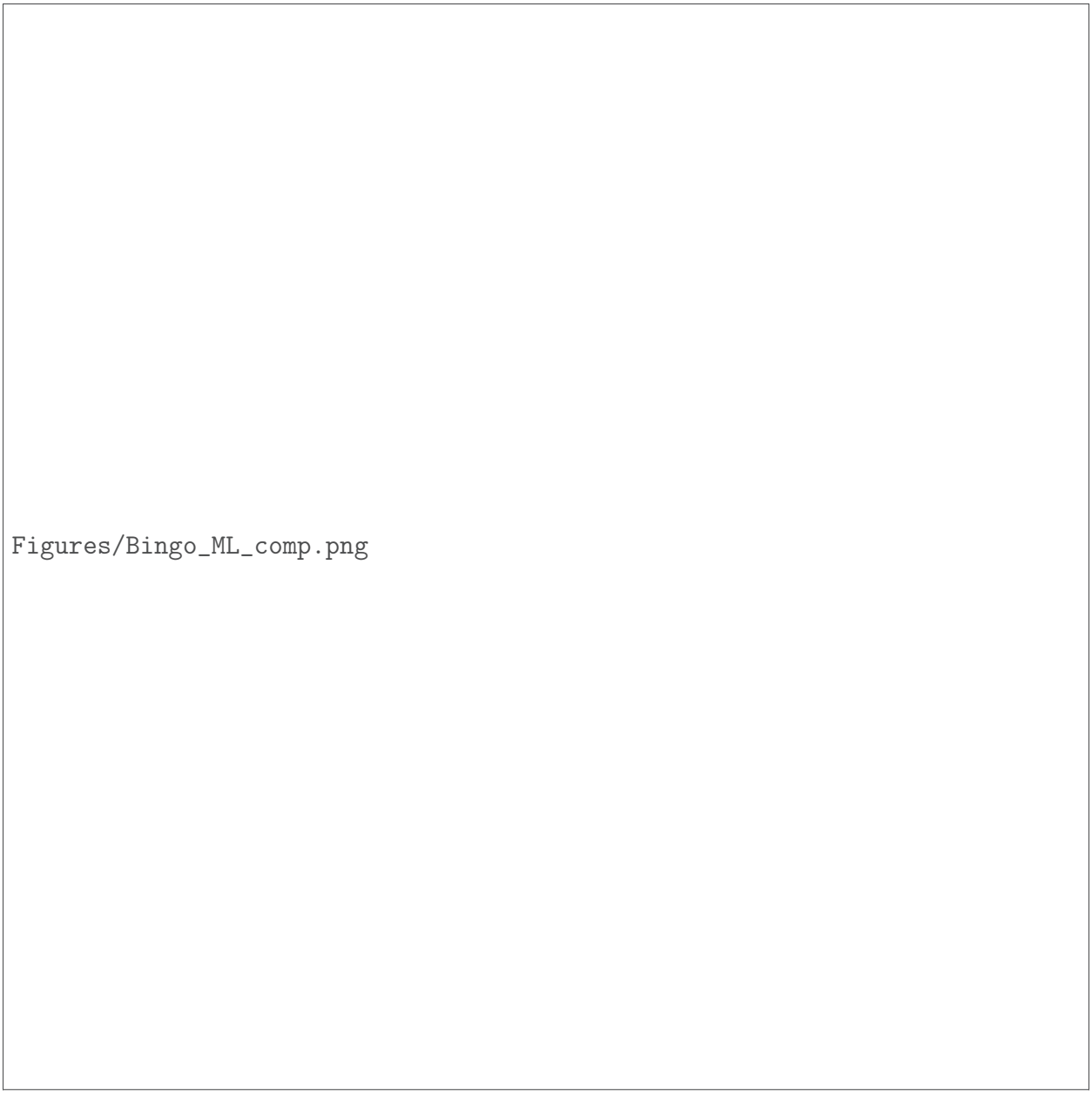


Figure 20: Plot comparing training times and errors between Bingo, Raju-Newman equation, RFR, and SVM. The circular markers were all trained using the mechanics based approach presented in this research, the square makers were trained directly on K. The evaluation times were measured on one core of an Intel i7-4790 with 32Gb DDR3 RAM at 1600 MHz on a data-set containing 150k data-points.

[2] S Pommier, C Sakae, and Y Murakami. An empirical stress intensity factor set of equations for a semi-elliptical crack in a semi-infinite body subjected to a polynomial stress distribution, 1999.

[3] Xinyu Zhang, Tingting Zhao, Yifan Liu, Qingqing Chen, Zhiyong Wang, and Zhihua Wang. A data-driven model for predicting the mixed-mode stress intensity factors of a crack in composites. *Engineering Fracture Mechanics*, 288, 8 2023.

[4] J. C. Sobotka and R. C. McClung. Stress-intensity factors solutions for straight through cracks in c-sections. *Engineering Fracture Mechanics*, 271, 8 2022.

[5] Arvind Keprate, R. M.Chandima Ratnayake, and Shankar Sankararaman. Comparison of various surrogate models to predict stress intensity factor of a crack propagating in offshore piping. *Journal of Offshore Mechanics and Arctic Engineering*, 139, 12 2017.

[6] Tongge Xu, Shuiting Ding, Huimin Zhou, and Guo Li. Machine learning-based efficient stress intensity factor calculation for aeroengine disk probabilistic risk assessment under polynomial stress fields. *Fatigue and Fracture of Engineering Materials and Structures*, 45:451–465, 2 2022.

[7] Mohamed El Amine Ben Seghier, Hermes Carvalho, Behrooz Keshtegar, José A.F.O. Correia, and Filippo Berto. Novel hybridized adaptive neuro-fuzzy inference system models based particle swarm optimization and genetic algorithms for accurate prediction of stress intensity factor. *Fatigue and Fracture of Engineering Materials and Structures*, 43:2653–2667, 11 2020.

[8] J. C. Newman and I. S. Raju. *Stress-intensity factor equations for cracks in three-dimensional finite bodies*, volume 4. 1983.

[9] David L. Randall, Tyler S. Townsend, Jacob D. Hochhalter, and Geoffrey F. Bomarito. Bingo: A customizable framework for symbolic regression with genetic programming. pages 2282–2288. Association for Computing Machinery, Inc, 7 2022.

[10] Southwest Research Institute. Nasgro.

[11] TODO. Todo.

[12] TODO. Smart|dt.

[13] Hiroshi Tada, Paul C. Paris, and George R. Irwin. *Stress Analysis of Cracks Handbook*. American Society of Mechanical Engineers (ASME), 3rd edition, 1985.

[14] J. E. Srawley W. F. Brown. *Plane Strain Crack Toughness Testing of High Strength Metallic Materials*. ASTM International, 1966.

[15] Fracture Analysis Consultants, Inc. Franc3d.

[16] Dassault Systèmes Simulia Corp. Abaqus.

[17] J R Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks, 1968.

[18] Leslie Banks-Sills, Itai Hershkovitz, Paul A. Wawrzynek, Rami Eliasi, and Anthony R. Ingraffea. Methods for calculating stress intensity factors in anisotropic materials: Part i - z = 0 is a symmetric plane. *Engineering Fracture Mechanics*, 72:2328–2358, 10 2005.

[19] F. O. de Franca, M. Virgolin, M. Kommenda, M. S. Majumder, M. Cranmer, G. Espada, L. Ingelse, A. Fonseca, M. Landajuela, B. Petersen, R. Glatt, N. Mundhenk, C. S. Lee, J. D. Hochhalter, D. L. Randall, P. Kamienny, H. Zhang, G. Dick, A. Simon, B. Burlacu, Jaan Kasak, Meera Machado, Casper Wilstrup, and W. G. La Cava. Interpretable symbolic regression for data science: Analysis of the 2022 competition. 4 2023.

[20] Michael Schmidt and Hod Lipson. *Comparison of Tree and Graph Encodings as a Function of Problem Complexity*. Association for Computing Machinery, 2007.

[21] Isadore Newman and I. S. Raju. Three dimensional finite-element analysis of finite-thickness fracture specimens, 1977.