A Framework for Automating the Parameter Determination of Crack Growth Models

A. Iliopoulos[1,2], J. G. Michopoulos[2], R. Jones[3,4], A. J. Kinloch[5], D. Peng[3,4]

ABSTRACT

Motivated by the need for an efficient fatigue crack growth prediction infrastructure for both legacy and novel materials, we have initiated the development of an automated computational framework capable of determining crack growth model parameters for an equationally defined model. As a first step in addressing this need, the present paper focuses on the Hartman-Schijve crack growth variant of the NASGRO equation by exploring and comparing various global optimization methods for parameter determination and evaluating their performance by using both synthetic and actual data. It also introduces the concept of the total least-squares minimization criterion within the context of crack growth modeling. The development of an open-source software library and an application implementing the approach are also described and are available for distribution to the technical community.

1. INTRODUCTION.

Since the introduction and use of the first generation of fatigue crack growth models [1], many more phenomenological and empirical ones have been developed and proliferated, as noted in [2], [3], [4]. One of the enduring challenges associated with utilizing these models has been the determination of the unknown parameters given a set of experimental data. Fatigue Crack Growth Model Parameter Identification (FCGMPI) has been traditionally based on a mix of semi-automated methods, custom systems, individual user experience, and proprietary software [5], [6]. The lack of open source software and automated processes to identify these parameters has motivated us to pursue the development of relevant methodologies and associated software.

The objectives of our effort are to: (a) develop a method for calculating the model parameters from experimental data that requires minimal to no user intervention; (b) provide computable crack-growth models that can be easily integrated into engineering workflows associated with qualification and certification; (c) establish the basis for a general, automated and rigorous approach to determining and using crack growth models for structural life predictions, and (d) provide the outcomes of this research in an open-source manner for widest dissemination and utilization purposes.

In developing methodologies that can satisfy these objectives, we realized that the fatigue crack growth and life prediction research could benefit from the availability of a framework that includes higher level

[1] Corresponding author, athanasios.iliopoulos@nrl.navy.mil

[2] Computational Multiphysics Systems Laboratory, Code 6394, Center for Materials Physics and Technology, US Naval Research Laboratory, Washington, DC 20375, United States.

[3] Centre of Expertise for Structural Mechanics, Department of Mechanical and Aerospace Engineering, Monash University, Clayton, Victoria, 3800, Australia.

[4] Faculty of Science, Engineering and Technology, Swinburne University of Technology, John Street, Hawthorn, VIC, 3122, Australia.

[5] Department of Mechanical Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, UK.

(graphical user interface based) and lower level (functions and benchmarking) facilities. The higher-level facilities can be appropriate for performing FCGMPI for engineering purposes. The lower-level facilities can be integrated into engineering and research workflows and tools for further dissemination. In addition, the lower-level facilities can be used to explore alternative crack growth rate models or allow engagement in other advanced uses.

The most widely used software that can perform FCGMPI is NASMAT [6], which is a module of the NASGRO software suite [7]. The particular software requires an iterative process, where the user selects parameters, and the software performs part of the parameter identification. The process requires the user to modify the provided values until a good fit is achieved [6]. A significant concern raised in [6] that relates to work in [5], [8], [9] and [10], is that the effects of the generally asymmetric nature of performing FCGMPI using the Ordinary Least Squares Method (oLSM). The authors of [6] propose a modification of the oLSM to adjust for the difference in magnitude due to the asymmetries in the data scales. In addition, they propose fixing, at regular intervals (along x- or y-axes), the number of representative data points to achieve a more uniform representation of each of the regions of the crack propagation curve. Although the method seems to perform fairly well for the presented data sets, it does not address the fundamental issue of the asymptotic nature of fatigue crack growth curves in Regions I and III, and only considers regular --as opposed to logarithmic distances-- that can address the issues raised in [6].

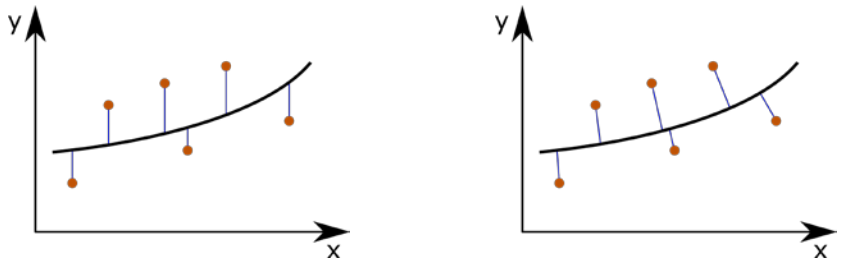
Therefore, and as the means to overcome difficulties emanating from the generally asymptotic nature of fatigue crack propagation models, we introduce the notion of the total least square optimality criterion. Instead of considering the vertical (parallel to the y-axis) deviation from the experimental data, the total least squares method considers the perpendicular to the curve deviation from the experimental data. Therefore, all the data are given the same treatment, whether they are located at the beginning (Region I), middle (Region II), or end (Region III) of the crack propagation curve.

In the past, we have used various approaches to determine the Hartman-Schijve model parameters. In these, several assumptions and constraints have been used in a manner that supports the communal understanding of the role of these parameters as material constants or not and their role in satisfying the similitude principle. However, in the present work, we have not employed any of these assumptions and constraints to focus on the numerical and computational aspects of the proposed effort. Accordingly, the determined values of the parameters in all the cases involving actual physical experiments are expected that they may be different from those in our previous publications.

The present work is divided into seven sections. First, we describe the role of objective functions in parameter determination and the advantages of total least squares when used in FCGMPI. In the same section, we derive and present the numerical aspects of implementing it. The subsequent section describes the various optimization methods we have explored, along with the addition of a custom method we developed particularly for FCGMPI. Comparisons using synthetic data follow. In the following section examples of applying the developed framework on experimental data follow. Finally, the paper closes with the conclusions and future plans.

2. OBJECTIVE FUNCTIONS IN PARAMETER DETERMINATION.

Performing parameter determination on a model from experimental data is usually performed with regression analysis and various curve fitting methods. The curve fitting problem can be described as:



(a) Ordinary Least Squares offsets (b) Total Least Squares offsets

Figure 1: (a) The objective function considers the vertical deviation (blue lines) of the model curve (black) from the experimental points (brown). (b) Total least squares offsets. The objective function considers the perpendicular deviation (blue lines) of the model curve (black) from the experimental points (brown).

Given a model $y = f(x; a_1, a_2, \dots)$ that relates the independent variable x with the dependent variable y, and given a list of N experimental pairs $\{x_i, y_i\}, i = 1 \dots N$, identify the values of the parameters $a_1, a_2, \dots$ such that the curve, optimally fits the experimental pairs in some quantitative sense called here the 'optimality criterion'.

Therefore, before presenting the actual optimization process that identifies the Hartman-Schijve parameters, the appropriate objective function needs to be defined. We will present two such objective functions (ordinary least squares and total least squares) and discuss the pros and cons of each.

2.1. Ordinary Least Squares

By far the most popular optimality criterion is based on ordinary least squares. In other words, one tries to find the values of the parameters such that the sum of squares

$$R^2 = \sum_{i=1}^{N} [y_i - f(x_i; a_1, a_2, \dots)]^2, \quad (1)$$

of vertical deviations from the curve is minimized. The deviations of ordinary least squares are shown in the example of Fig. 1a. Determining the parameters of the Hartman-Schijve model for the case of ordinary least squares is described in detail in [11] and [12].

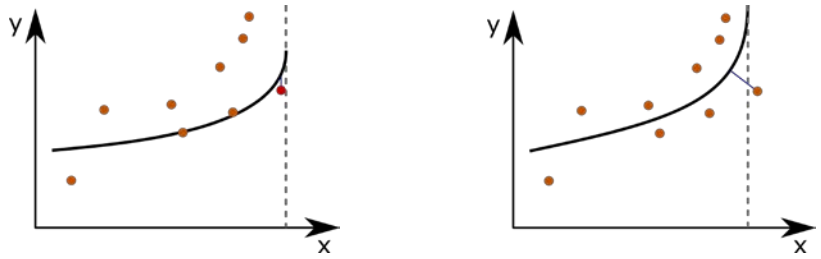
2.2. Total Least Squares

Another approach that is far less popular, is that of total least square [13], [14], [15]. In total least square, the objective function takes a form that accounts for the minimum distance of each experimental point to the curve (Fig. 1b).

The disadvantages of total least squares are (a) the computational cost and (b) the generally more complex derivation requirements of the objective function, that in the case of FCGMPI cannot be performed analytically and needs to be performed in a numerical manner. To address these issues for the curve fitting of the Hartman-Schijve model, we have developed a methodology that performs a numerical derivation and have implemented it in a high-performance C++ library.

One distinct advantage of total least squares is its ability to handle cases of asymptotic curves. This is particularly useful in the case of fatigue crack growth modeling, because the initial (stage I) and final (stage III) segments of the available models, exhibit asymptotic behavior along asymptotes parallel to the da/dN axis. The resulting effect of both types of curve fitting is illustrated in Fig. 2.

Fig. 2a, shows the case of ordinary least squares. In this case, if one aims at including all points for the regression process, then the asymptote will necessarily have to be outside the boundaries of the point



(a) Ordinary least squares (b) Total least squares

Figure 2: Fitting of asymptotic curves. The blue line represents the offset of the rightmost point from the fitted curve. (a) In ordinary least squares, all points need to be to the interior of the or considered outliers curve (red point). (b) In total least squares, the points can be outside the asymptote limit. In (a) the curve is necessarilly biased to allow for encompassing all points or is biased towards rejecting potentially valid ones.

set. Otherwise the included points need to be selected based on some subjective metric. In traditional fitting, where the engineer must manually adjust values and select points, the quality of the resulting fit depends heavily on his/her experience.

On the other hand, in the case of total least squares (Fig. 2b), the fit takes a much more natural form, considering even the points that may be outside the limits of the asymptote. Therefore, any outliers do not need to be excluded based on subjective criteria, but if desired can be selected based on regular probabilistic treatments that fall outside the scope of the present work.

2.2.1. Numerical computation of the sum of total squares

To calculate the sum of total squares of a point set $\{x_i, y_i\}, i = 1 \dots N$ from a function $f(x; a_j)$, with known parameters $a_j, j = 1 \dots M$, one needs to find the minimum distance of each point $\{x_i, y_i\}$, from the curve represented by $f(x; a_i)$. The problem can be defined as the following minimization problem:

$$\text{minimize } \mathrm{g}(x) = [x_i - x]^2 + \left[y_i - f(x_i; a_j)\right]^2. \tag{2}$$

The minimum of function $\mathrm{g}(x)$ of Eqn. 2 can be identified by solving the following equation:

$$h(x) = \frac{dg(x)}{dx} = 0. \tag{3}$$

Because for the case of crack growth curves Eqn. 3 will be non-linear, it is necessary to solve it by numerical methods. In particular, we selected the bisection method [16] for its stability and simplicity. We furthermore explored root finding based on the Newton-Raphson [17] and the Halley's [18] methods, but the non-convex nature of the function studied, rendered them less stable and in some cases non-converging.

After the minimum value of x, x_{min_i} has been computed for each data point $i = 1 \dots N$, the sum of total squares can be calculated by:

$$R^2 = \sum_{i=1}^{N} \mathrm{g}(x_{min_i}). \tag{4}$$

In the present paper we are using the Hartman-Schijve variant of the NASGRO crack growth equation, viz:

$$\frac{da}{dN} = D \left[\frac{\Delta K - \Delta K_{\text{thr}}}{\sqrt{1 - \frac{K_{\max}}{A}}}\right]^p = D \left[\frac{\Delta K - \Delta K_{\text{thr}}}{\sqrt{1 - \frac{\Delta K}{(1-R)A}}}\right]^p. \tag{5}$$

where a is the crack length/depth, N is the number of cycles, D is a material constant, p is a constant and is often approximately 2, A is the cyclic fracture toughness, K is the stress intensity factor and $\Delta K = (K_{\max} - K_{\min})$ is the range of the stress intensity factor seen in the cycle, and ΔK_{thr} is an effective lower-bound fatigue threshold value.

To stay consistent with the log-log representation of crack growth curves, the numerical operations are performed in the log-log space ([11], [12]). Therefore, the problem of identifying the total square for a fatigue crack growth data point $\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}$ to the Hartman- Schijve curve can be written:

$$\text{minimize } q(\Delta K) = [\log_{10}(\Delta K) - \log_{10}(\Delta K_i)]^2 + \left[s\left(\log_{10}\left(\left.\frac{da}{dN}\right|_i\right) - \log_{10}\left(\frac{da}{dN}(\Delta K)\right)\right)\right]^2. \tag{6}$$

The parameter s represents a normalization term. By introducing Eqn. 5 in Eqn. 6 and taking the derivative with respect to ΔK, we derive the following equation:

$$h(\Delta\text{K}) = \frac{\frac{2(\text{Log}[\Delta\text{K}]-\text{Log}[\Delta\text{K}_i])}{\Delta\text{K}} - \frac{ps^2(\text{Log}[\left.\frac{da}{dN}\right|_i]-\text{Log}[\text{D}(\frac{\Delta\text{K}-\Delta\text{K}_{\text{thr}}}{\sqrt{1+\frac{\Delta\text{K}}{A(-1+R)}}})^p])(2A(-1+R)+\Delta\text{K}+\Delta\text{K}_{\text{thr}})}{(A(-1+R)+\Delta\text{K})(\Delta\text{K}-\Delta\text{K}_{\text{thr}})}}{\text{Log}[10]^2}, \tag{7}$$

After solving the equation $h(\Delta\text{K}) = 0$ using algorithm 1, for all data pairs, it is elementary to calculate the total squares of a set of points $\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}, i = 1 \dots N$ from a known curve of the form of Eqn. 5, as shown in algorithm 2.

The parameter s is utilized to normalize the $\mathbb{R}^2$space upon which the total least squares evaluation will be performed. Without the particular normalization, the space will be considerably skewed because of the large difference between the magnitudes ΔK and $\frac{da}{dN}$ and the lack of scale invariance of the total least square method. Fortunately, such a parameter can be calculated by the use of the datapoints themselves to provide with an optimal aspect ratio between the ranges of the two parameters:

$$s = \frac{\log_{10}[\max(\Delta K_i)] - \log_{10}[\min(\Delta K_i)]}{\log_{10}\left[\max\left(\left.\frac{da}{dN}\right|_i\right)\right] - \log_{10}\left[\min\left(\left.\frac{da}{dN}\right|_i\right)\right]} \tag{8}$$

In the future we plan to investigate methods based naturally scale invariant approaches [19], [20], [21], [22], although because of the asymptotic nature of the crack propagation curves, we currently postulate these approaches may not be applicable.

Input:

Hartman-Schijve parameters($\Delta K_{\text{thr}}, A, p, D$).

R : Fatigue R ratio.

ΔK_i: ΔK of the ith data point.

$\left.\frac{da}{dN}\right|_i$: $\frac{da}{dN}$ of the ith data point.

β: Asymptote restriction offset ratio, default: 1×10^{-5}.

max_iter: maximum number of iterations in the optimization loop

Output:

Minimum distance of data point $\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}$ from curve $\frac{da}{dN}$ of Eqn. 5.

1. $\Delta K_{low} \leftarrow \Delta K_{\text{thr}}(1.0 + \beta)$
2. $\Delta K_{high} \leftarrow A(1.0 - R)(1.0 - \beta)$
3. $d_{low} \leftarrow \text{h}(\Delta K_{low})$
4. $d_{high} \leftarrow \text{h}(\Delta K_{high})$
5. If $d_{low} > 0$ & $d_{high} > 0$
    a. Return $\text{q}(\Delta K_{low})$
6. If $d_{low} < 0$ & $d_{high} < 0$
    a. Return $\text{q}(\Delta K_{high})$
7. If $d_{low} > 0$
    a. $x_+ \leftarrow \Delta K_{low}$
    b. $x_i \leftarrow \Delta K_{high}$

    Else
    a. $x_+ \leftarrow \Delta K_{high}$
    b. $x_- \leftarrow \Delta K_{low}$
8. $\text{tol} \leftarrow \left[\log_{10}(\Delta K_{high}) - \log_{10}(\Delta K_{low})\right] \times 1 \times 10^{-4}$
9. $\text{span} \leftarrow \log_{10}(x_+) - \log_{10}(x_-)$
10. $\text{niter} \leftarrow 0$
11. while $\text{niter} + + < \text{max\_iter}$ && $\text{span} > \text{tol}$
    a. $x_{half} = {}^{(x_+ + x_-)}/_2$
    b. $d_{half} = x_{half}$
    c. If $d_{half} > 0$
        i. $x_+ \leftarrow x_{half}$

    Else
        i. $x_- \leftarrow x_{half}$

    d. $\text{span} \leftarrow \log_{10}(x_+) - \log_{10}(x_-)$
12. Return $\text{q}\left({}^{(x_+ + x_-)}/_2\right)$

Algorithm 1: Calculation of minimum square distance of a data point $\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}$ from the Hartman-Schijve curve with known parameters in the log-log space.

Input:

Hartman-Schijve parameters($\Delta K_{\text{thr}}, A, p, D$).

R : Fatigue R ratio.

$\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}, i = 1 \dots N$ experimental data points.

β: Asymptote restriction offset ratio, default: 1×10^{-5}.

max_step: Min step in root finding loop, default: 1×10^{-3}.

max_iter: maximum number of iterations in the optimization loop

Output:

Total Squares (R^2).

1. $\mathrm{J}^2 = 0$
2. Foreach data point in the $\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}, i = 1 \dots N$ dataset
   a. $\mathrm{J}^2 = \mathrm{J}^2 + \mathbf{distance}$ // From Alg. 1
3. Return J^2

Algorithm 2: Calculation of the sum of total squares of a data point set $\left\{\Delta K_i, \left.\frac{da}{dN}\right|_i\right\}, i = 1 \dots N$ for the Hartman-Schijve curve with known parameters in the log-log space.

3. OPTIMIZATION METHODS FOR PARAMETER DETERMINATION.

For linear problems, the objective function of Eqn. 1 can be minimized by converting it to a linear system of equations. In non-trivial models though and for non-linear systems a non-linear optimizer is needed, that depending on the model can take a multitude of forms. Furthermore, because the objective function may have multiple minima (which is the case in the present work), one needs to employ a global optimizer that tries to minimize the objective function, by some sort of reasonable scanning of the parametric space. Likewise, a global optimizer needs to be employed for minimizing the objective function in the case of total least squares.

Several global optimization methods were investigated, aiming at identifying their performance in identifying the Hartman-Schijve equation parameters from the input data. The implementation of these algorithms can be found in [23]. In addition, a new method is introduced herein, that we term, Contracting Uniform Hyper-Rectangles Sampling for Optimization (CUHYSO) that as it will be demonstrated below seems to perform very well for the particular problems of fitting fatigue crack-growth data.

A description of the algorithms used in the comparison analysis of the present effort follows.

3.1. Locally biased DIviding RECTangles (DIRECT-L).

The DIRECT algorithm is a classical algorithm in global optimization methods [24]. The algorithm simultaneously searches the parametric space using all possible Lipschitz constants from zero to infinity. Therefore, it eliminates the need for specifying a Lipschitz constant. Here we are employing a variation of the DIRECT algorithm called locally biased DIRECT [25] that gives emphasis to local searches. The algorithm has been shown to be more appropriate in cases where there is a single global minimum and a few local minima.

3.2. Controlled Random Search with Local Mutation (CRS_LM).

The CRS algorithm [26], is a method based on higher-dimensional random sampling. It successively updates a list of minimum locations until a certain search stop criterion is satisfied. Here we are using a variation of the algorithm that adopts a local mutation strategy to better capture local minima [27], [28]. The local mutation is introduced in a global point generation scheme that uses linear interpolation.

3.3. Multi-Level Single-Linkage (MLSL) and MLSL with a Sobol Low-Discrepancy Sequence (MLSL_LDS).

The Multi-Level Single- Linkage (MLSL) method is a multi-start method. It performs several local optimizations by using a predefined local optimizer. The sampling is either random or low-discrepancy. We are using both the legacy variant (MLSL) and one that utilizes a Sobol low-discrepancy sequence [29] (MLSL LDS). The latter is used to reduce the negative effects of totally random sampling.

3.4. Improved Stochastic Ranking Evolution Strategy (ISRES).

The particular algorithm ([30], [31]) uses an evolution strategy that combines a mutation rule and an update rule (differential evolution). The method supports linear and non-linear equality and inequality constrains as well as regular bound constrains.

3.5. ESCH - evolutionary algorithm.

This is an evolutionary algorithm based on the work presented in [32], [33]. The method incorporates ideas found in [34] and [35]. The evolution algorithm is based on population evolution along generations with the candidate solutions on two distinct groups “parents” and “offsprings”. The population successively increases and decreases in size to imitate the average natural evolution of a species.

3.6. Minimization of multiextremal functions under nonconvex constraints (AGS).

The particular algorithm [36] uses the Hilbert curve to reduce the parametric space to a single parameter space. Then the algorithm divides the single parameter space into intervals. The search space is sampled based on posterior probabilities. The method generally does not work well for a parametric space of size over 5, because the 64-bit machine precision does not allow one to generate a tight Hilbert curve.

3.7. Contracting Uniform Hyper-Rectangle Sampling for Optimization (CUHYSO).

Uniform deterministic sampling is generally recognized both theoretically and in practice to offer more accurate results than random sequences [29]. In addition, the determinism provides with the same result for the same problem, while stochastic optimization methods do not. Stochastic optimization methods may result in the identification of different parameters for different executions of the same optimization problem. This can be confusing or outright unacceptable when reproducible minimization results are expected. To address this issue and certain unwanted behavior that we observed in the other algorithms, a new algorithm that uses uniform deterministic sampling was developed.

The algorithm requires an input of bound constraints in the form of two lists, the number of subdivisions K in each direction and a contraction ratio $r > 1.0$. The lower bound can be represented by a vector $\boldsymbol{l} \in \mathbb{R}^N, \boldsymbol{l} = \{l_1, l_2, \dots, l_N\}^T$, and the upper bound by a vector, $\boldsymbol{u} \in \mathbb{R}^N, \boldsymbol{u} = \{u_1, u_2, \dots, u_N\}^T$ with N the number of parameters and $u_i \geq l_i$.

Subsequently, a uniform sampling list is constructed by subdividing the space between the bounds for each of the parameters. $\boldsymbol{s}_i = \{s_{ki}\}^T = \{s_{1i}, s_{2i}, \dots, s_{Ki}\}^T = \{l_i, l_i + \Delta_i, l_i + 2\Delta_i, \dots, l_i + (K-1)\Delta_i\}$, with

$\Delta_i = \frac{u_i - l_i}{K-1}$. The list is comprised of all possible combinations $\{s_{k_1 1}, s_{k_2 2}, \dots, s_{k_N N}\}^T$ with $k_i = 1 \dots K, i = 1 \dots N$ of the sampling points.

The objective function is then evaluated for each of the samples. For the values of the parameters $\boldsymbol{s}_m = \{s_{m1}, s_{m2}, \dots, s_{mN}\}^T$ where the objective function is minimum, the algorithm calculates a new bound by performing contraction assuming as $\boldsymbol{s}_m$ a higher-dimensional center point. The new bounds are calculated based on the following equations:

$$l_{i+1} = \max[l_i, s_{mi} - w_i], \tag{9}$$

$$u_{i+1} = \min[u_i, s_{mi} + w_i], \tag{10}$$

with $w_i = \frac{u_i - l_i}{2r}$. The $\max$ and $\min$ functions are used to ensure that the new bounds are always inside or on the previous step bounds.

To address the asymptotic behavior of the data, CUHYSO is implemented in a manner that ignores any choice of parameters that lead to non-finite ordinate results. This particular feature provides CUHYSO with a distinct advantage when used with a traditional ordinary least square optimality criterion based on scattered experimental data.

The described algorithm may be augmented with several features. For example, the number subdivisions may be specified per parameter or based on some predefined heuristic. Because our focus is on minimizing or eliminating user choices, we have decided to implement such features as we identify how they can be automated. These extensions will be the subject of future research efforts.

4. OPTIMIZATION METHODS COMPARISON.

To obtain insight in the optimization methods efficiency in identifying the Hartman-Schijve model parameters, a number of comparison tests were performed. The tests were based on synthetic data. For purposes of demonstration we are showing the results for the following set of model parameters:

$$D = 3.9 \times 10^{-10}, p = 2.29, \Delta K_{thr} = 3.04 \text{ MPa}\sqrt{m}, A = 116.81 \text{ MPa}\sqrt{m} \tag{11}$$

The synthetic data involved both smooth and perturbed datasets. We will first focus on the results on the smooth synthetic data.

Fig. 3 shows the convergence evolution to the values of the model parameters for the case of ordinary least squares with unperturbed data. It can be observed that some optimization methods could identify the reference parameters in a satisfactory manner while others performed a little worse. The plots for the crack propagation curves that use the identified values can be seen in Fig. 4, while Table 1 shows the identified parameters for each optimization method. It can be concluded that for the case of the unperturbed data perfectly satisfying the model, almost all optimization methods can identify the model parameters. However, the optimization method evaluation approach just described can be misleading because the unperturbed data do not reflect actual experimental situations.

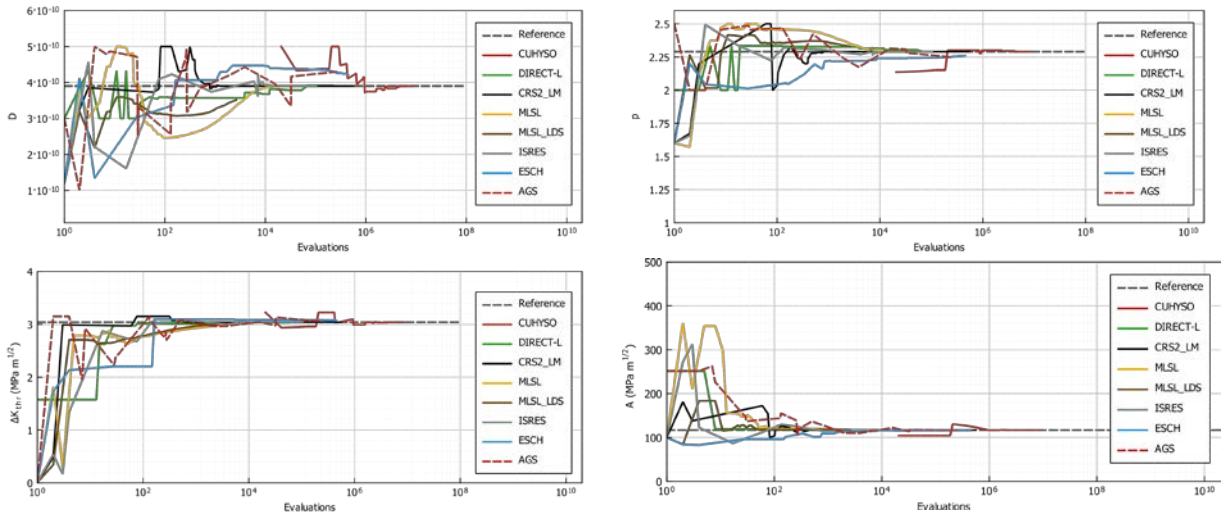


Figure 3: Convergence of the various global optimization methods to the reference Hartman-Schijve parameters by using an ordinary least squares optimality criterion for the case of the synthetic unperturbed data.

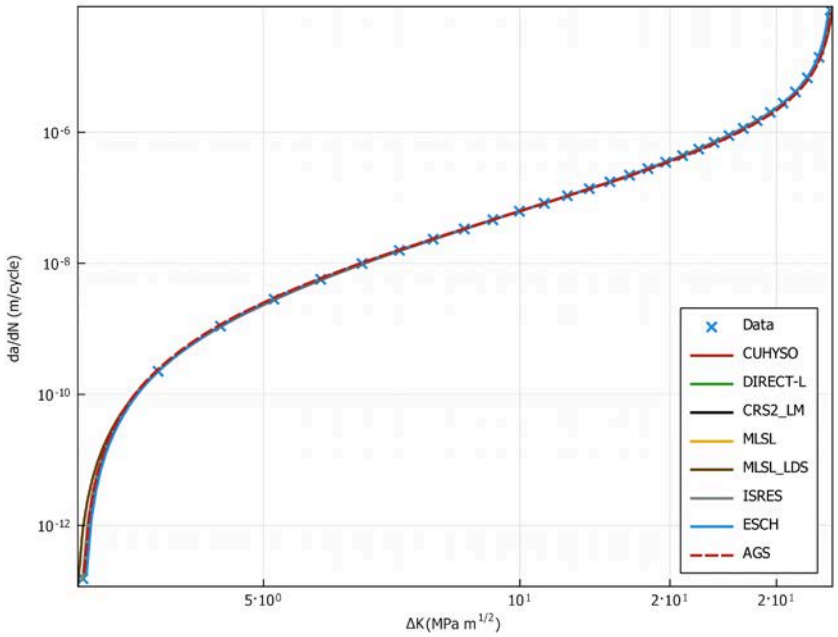


Figure 4: Evaluation of the Hartman- Schijve equation using the values of the identified parameters for the ordinary least square method for the synthetic unperturbed data

Table 1: Final values of the Hartman-Schijve model parameters as determined by the various methods using total least squares for the unperturbed synthetic data set. The results correspond to Figs. 3 and 4.

Method	Objective value @ min	D	p	ΔK_{thr} (MPa$\sqrt{m}$)	A (MPa$\sqrt{m}$)
Reference		3.9e-10	2.29	3.04	116.81
CUHYSO	<0.0001	3.9e-10	2.29	3.04	116.81
DIRECT-L	<0.0001	3.9e-10	2.29	3.04	116.89
CRS2	<0.0001	3.9e-10	2.29	3.04	116.81
MLSL	<0.0001	3.9e-10	2.29	3.04	116.83
MLSL LDS	<0.0001	3.5e-10	2.33	3.00	117.53
ISRES	<0.0001	3.9e-10	2.29	3.04	116.81
ESCH	<0.0001	4.2e-10	2.26	3.07	116.12
AGS	<0.0001	3.9e-10	2.29	3.04	116.81

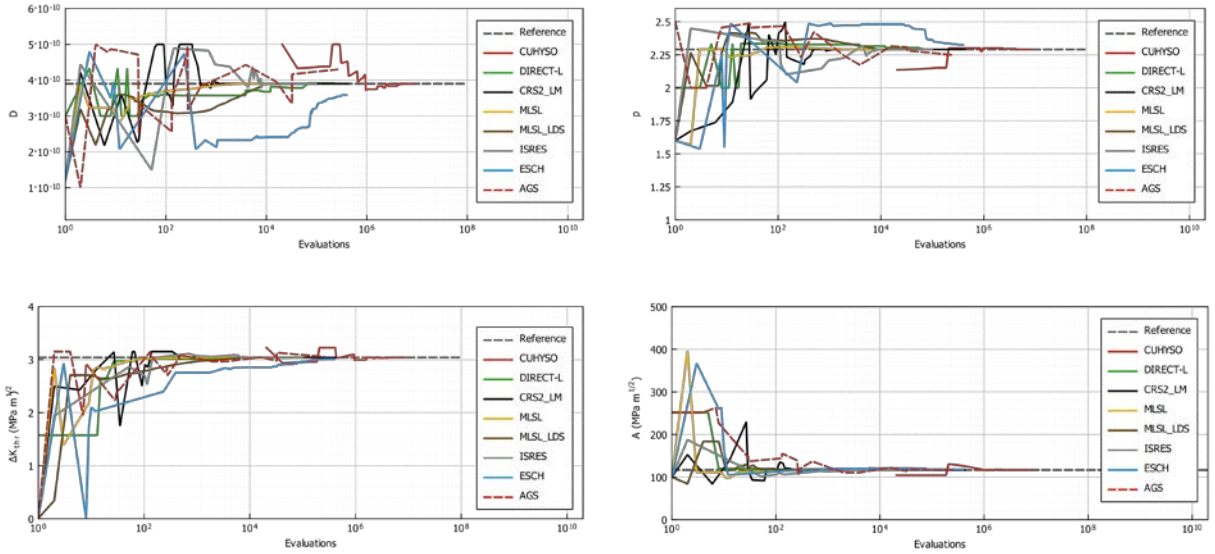


Figure 5: Convergence of the various global optimization methods to the reference Hartman-Schijve parameters by using a total least squares optimality criterion for the unperturbed synthetic data set.

Fig. 5 shows the convergence evolution of the values of the model parameters for the case of total least squares with unperturbed data. It can be observed that all the algorithms perform very well. Fig. 6 shows the evaluated curves for the identified parameters for each of the optimization methods. It can be shown that most methods provide excellent results. The final values of the identified parameters are shown in Table 2. From this table it can be observed that the final values produced from some of the optimization methods are not the as the ones (i.e. the reference values) used to generate the synthetic data, even if the fit appears very satisfactory. This is an expected outcome given that the model presented has a non-convex nature.

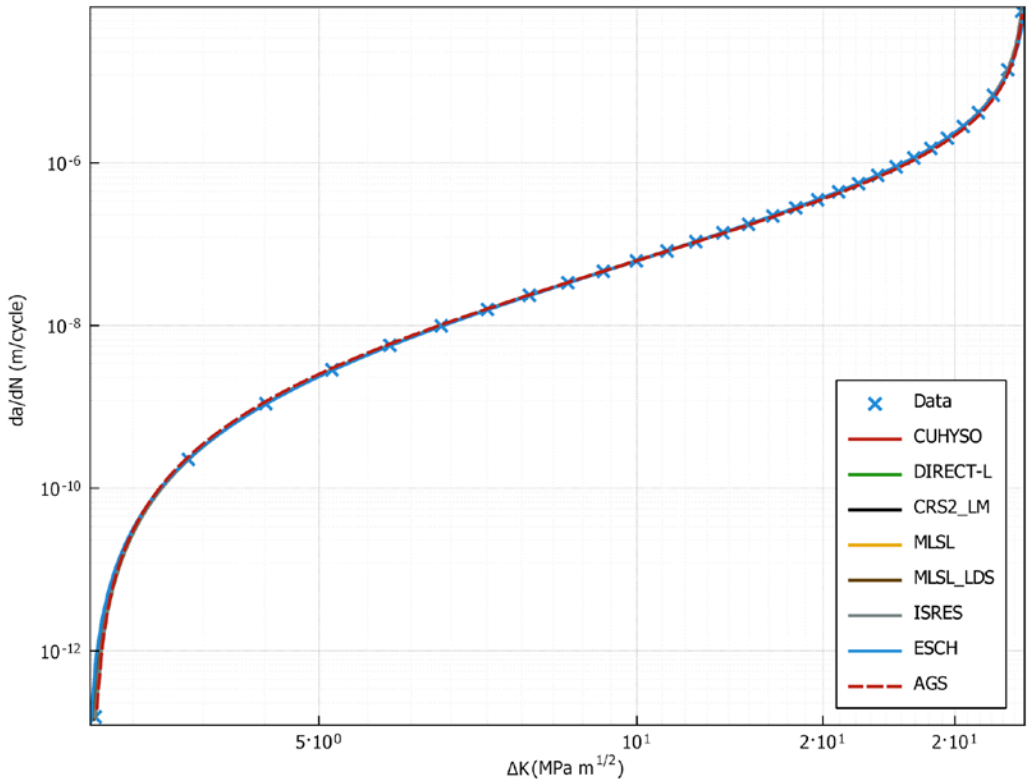


Figure 6: Evaluation of the Hartman- Schijve equation using the values of the identified parameters for the total least square method for the unperturbed case.

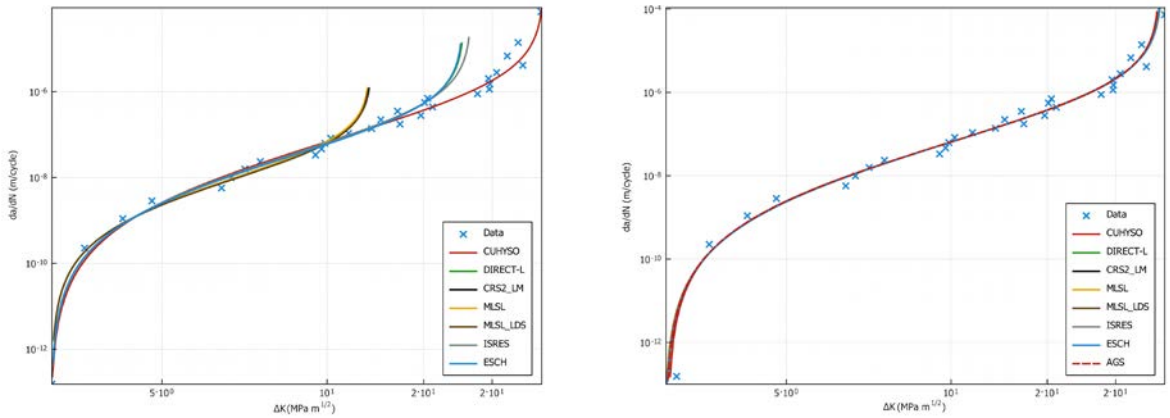
Although the previous benchmarks provide an initial estimate of the efficiency of the various optimization methods, a more realistic understanding can be obtained by performing the tests to give perturbed data which corresponds to a situation that relates better to actual experiments, particularly if multiple datasets are present for the same type of experiment. The perturbed data were generated by randomly offsetting the values of the dataset ΔK values at the log scale. The distribution used was uniform with a log range between -0.05 and 0.05.

The evaluated results for both the ordinary and total least squares evaluations are shown in Fig. 7, where the superiority of total least squares is evident. In the ordinary least squares, all of the solvers but CUHYSO fail to provide proper model identification. CUHYSO performs better than the others, because of the special provisions in handling nonfinite results. The main reason of the failure of the other optimizers is the unconditional rejection of points when the optimizers search at regions outside the asymptote limits of the curve.

However, the latter is not a problem for the total least square criterion and as shown in Fig. 7b, all the optimizers are able to compute very similar crack propagation curves. Furthermore, as shown in Table 4, all the optimizers manage to identify the reference value in a very satisfactory manner even at the presence of perturbation. Also, in Table 3 it can be seen that the same is not true for the ordinary least square criterion tests. In this case only CUHYSO finds parameter values that can be considered close to the reference values.

Table 2: Final values of the Hartman-Schijve model parameters as determined by the various methods for the case of using total least squares for the unperturbed synthetic data set. The results correspond to figures 5 and 6.

Method	Objective value @ min	D	p	ΔK_{thr} (MPa$\sqrt{m}$)	A (MPa$\sqrt{m}$)
Reference		3.9e-10	2.29	3.04	116.81
CUHYSO	<0.0001	3.9e-10	2.29	3.04	116.81
DIRECT-L	<0.0001	3.9e-10	2.29	3.04	116.89
CRS2	<0.0001	3.9e-10	2.29	3.04	116.81
MLSL	<0.0001	3.9e-10	2.29	3.04	116.81
MLSL LDS	<0.0001	3.8e-10	2.30	3.03	116.97
ISRES	<0.0001	3.9e-10	2.29	3.04	116.81
ESCH	<0.0001	3.6e-10	2.32	3.01	117.56
AGS	<0.0001	4.3e-10	2.25	3.05	116.92



(a) (b)

Figure 7: Evaluation of the Hartman- Schijve equation using the values of the identified parameters for the perturbed case. (a) Using the ordinary least squares criterion and (b) using the total least squares criterion

It should be pointed out that because some methods rely on random number generation, re-executing the identification programs may not yield the same results for these methods.

It should also be noted that CUHYSO is the only algorithm that currently supports parallelization and can take advantage of multi-core computer systems. This gives CUHYSO a considerable advantage and although it might take more iterations to converge, it completes the identification task much faster than other algorithms. Regardless, all of the methods are able to perform a satisfactory parameter identification in the first 10-20 seconds while their converged values are identified on an average after about 60 seconds on a regular desktop computer.

Table 3: Final values of the Hartman-Schijve model parameters as determined by the various methods for the case of using ordinary least squares for the perturbed data set. The results correspond to figure 7a.

Method	Objective value @ min	D	p	ΔK_{thr} (MPa$\sqrt{m}$)	A (MPa$\sqrt{m}$)
Reference		3.9e-10	2.29	3.04	116.81
CUHYSO	0.110135	4.90E-10	2.26	3.12	124.34
DIRECT-L	0.097735	5.00E-10	2.05	3.13	88.98
CRS2	0.101609	5.00E-10	1.74	3.14	60.27
MLSL	0.099688	5.00E-10	1.75	3.14	59.85
MLSL LDS	0.108303	5.00E-10	1.71	3.14	59.94
ISRES	0.110554	5.00E-10	2.1	3.13	91.61
ESCH	0.105475	5.00E-10	2.04	3.13	88.47
AGS	0.105475	4.80E-10	1.73	3.14	59.54

Table 4: Final values of the Hartman-Schijve model parameters as determined by the various methods for the case of using total least squares for the perturbed data set. The results correspond to figure 7b.

Method	Objective value @ min	D	p	ΔK_{thr} (MPa$\sqrt{m}$)	A (MPa$\sqrt{m}$)
Reference		3.9e-10	2.29	3.04	116.81
CUHYSO	0.0006	4e-10	2.30	3.04	120.11
DIRECT-L	0.0006	3.5e-10	2.35	2.99	121.40
CRS2	0.0006	3.5e-10	2.35	2.99	121.40
MLSL	0.0006	3.4e-10	2.36	2.98	121.64
MLSL LDS	0.0006	3.5e-10	2.36	2.99	121.53
ISRES	0.0006	3.5e-10	2.35	2.99	121.40
ESCH	0.0006	3.7e-10	2.33	3.01	120.87
AGS	0.0006	3.4e-10	2.37	2.99	123.12

5. SOFTWARE

To facilitate the widest dissemination possible of the proposed approach, a software package was architected and prototyped targeting a variety of use cases. These use cases include:

1. Automated parameter identification using experimental data. This facility is provided as an application with a Graphical User Interface (GUI). It is expected that it will fulfil end-user needs, including the needs for materials research and regular engineering practice.
2. Core library. The software core library is aimed at more customizable user cases. These use cases include development of custom software development or/and GUI applications.
3. Benchmark Tests. The tests include several benchmark facilities that can used to compare fitting approaches, optimization methods or develop additional crack growth models. The present work utilizes these tests to generate method comparisons presented earlier.
4. Language bindings for use from user's language of choice. This facility is at the planning stage and will include interface for the C language, and bindings for Python, Java, Matlab, Octave and other common languages and environments used in science and engineering.

The software was developed in modern C++ and requires a compiler supporting the ISO C++20 Standard [37]. The basic algorithmic components are in the public domain and therefore can be used and modified at will. The software's online repository is located at [38]. For MS-Windows, the software executables are distributed using a regular installer with the latest release located at [39].

5.1. Architecture

The adopted toolchain utilizes the CMake build system [40] and allows for architecting the software by separating it in three main components as planned for the target uses: a core library, a testing framework and an GUI application. The build system is computer architecture agnostic, although currently it has been tested only on Microsoft Windows. For some components, the toolchain automatically downloads and compiles library dependencies, reducing the setup time for development setups. The only external dependence is that of the Qt Software Development Kit (SDK) [41]. One component that has not been implemented yet is that of the bindings for other languages.

5.1.1. Core Library

The core library is based a single header library using templates for defining the numeric type [42]. The single header library has the advantage the it does not require separate compilation. Therefore, it allows for very simple dissemination, distribution and most importantly inclusion into other codes. Furthermore, the use of a generic numeric type allows the user to use arbitrary, multi or even mixed precision arithmetic to increase the numerical accuracy if desired.

The core library defines a basic data point type that encapsulates the values of ΔK and $\mathrm{d}a/\mathrm{d}N$ in a single pair. A test data structure describes a single test and encapsulates the test's R value and the list of data points for the test. Finally, a templated test set type-name is used to represent a generic list of multiple tests.

The main algorithm employs CUHYSO and it is called using the function named *fit*. The parameters of this function include the bounds of the global optimization, the number of subdivisions and the contraction ratio as well as callback functions to report progress.

5.1.2. Benchmark Tests

The software contains 3 main test folders that share a common architecture. The test code implements the comparisons that were presented previously in the paper and can be modified accordingly if evaluations of different methods are desired.

The tests are using the library found in [23], that implements the global optimizers, except for CUHYSO. Plotting and publication preparation graph export is provided by the cxxplot library [43].

5.1.3. Graphical User Interface

The graphical user interface uses the Qt SDK [41] library to provide graphical widgets. Qt is cross-platform and can be compiled for Windows, Linux and Mac OS X. The plotting functionality is provided by the QCustomPlot library [44].

The GUI uses the core library by making calls to the *fit* function. The execution is performed asynchronously on a different thread and the interface updates the graphical view to indicate the progress of the parameter identification. It should be noted that the CUHYSO optimizer is parallelized and takes advantage of the cores available in the system.

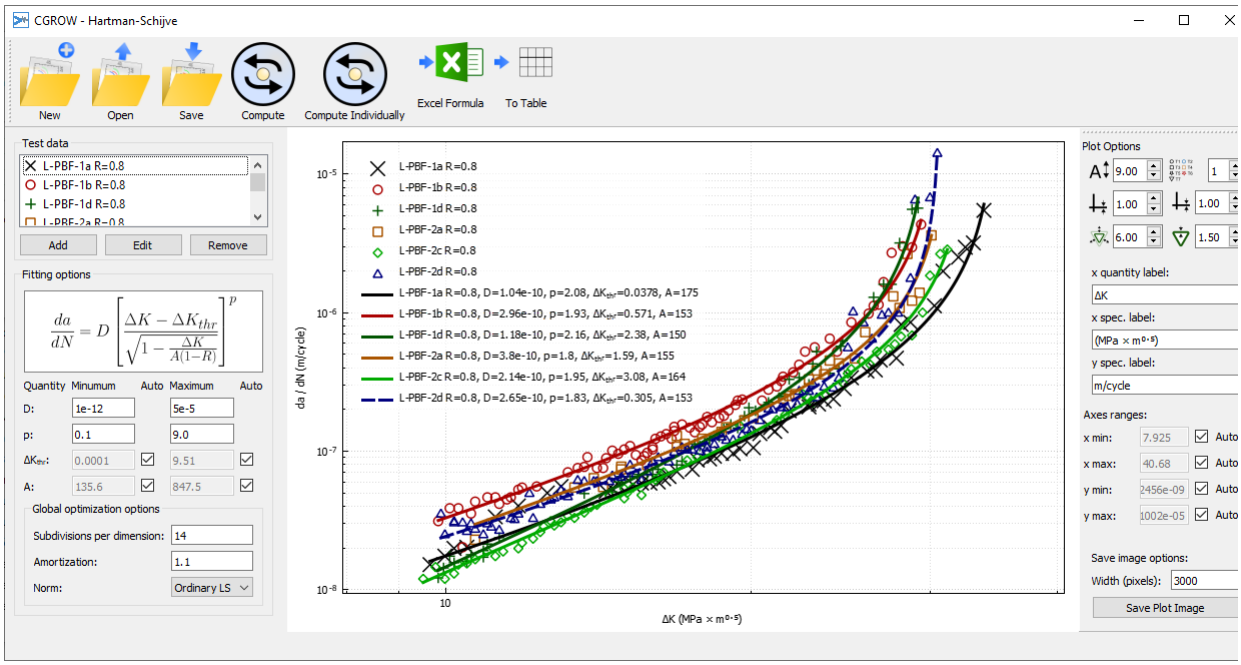


Figure 8: CGROW Graphical User Interface.

5.2. Graphical User Interface Description

The GUI of the application is shown in Fig. 8. Besides regular controls for creating, opening, and saving a project, the application provides a “Compute” button that starts the computation assuming a single master curve is desired to be computed for all datasets. A “Compute individually” button on the other hand, instructs the software to compute individual curves for each data set. An export button provides the functionality to exports the computed formula to MS Excel by copying the formula into the OS clipboard. A “To Table” button generates the model results into the clipboard for general use and reporting. A special dialog box provides the means to input the test data, that can be copy/pasted from and to regular spreadsheet applications like MS Excel and Libreoffice calc. The GUI application stores its data in a json format, that is both machine and human readable.

The user has the options to define the optimization limits for each of the four Hartman-Schijve parameters. Automatic limits can be selected for the ΔK_{thr} and A parameters. In the latter case the software attempts to identify reasonable limits based on the range of the experimental data points.

The CUHYSO global optimization options are provided in a dedicated entry box and include the number of subdivisions per dimension per optimization sampling sequence, the amortization (or contraction ratio) and the Norm (ordinary least squares or total least squares).

The application also provides plotting options that can be used to modify the size of the various elements, like line and marker thicknesses. There is also a save image option to store the current graph in a format appropriate for publication.

The GUI application includes several data sets that have been curated by the authors and can be found in the examples folder.

6. APPPLICATION ON EXPERIMENTAL DATA

6.1. Application on Additively Manufactured Ti-6Al-4V

A set of fifty-seven tests of Additively Manufactured Ti-6Al-4V were compiled from various sources, ([45], [46]) and were subsequently input in the CGROW software. The R-ratio for all the tests was 0.1. Default parameter ranges were used, and both the ordinary and total least squares methods were applied. It should be noted that both executions are using the CUHYSO optimizer. The software processes each data set and computes the model parameters for all the data sets automatically. Afterwards the data was separated in eight sets for clarity of visualization purposes and the results of the application of the two different methods are shown in Figs. 9 to 16.

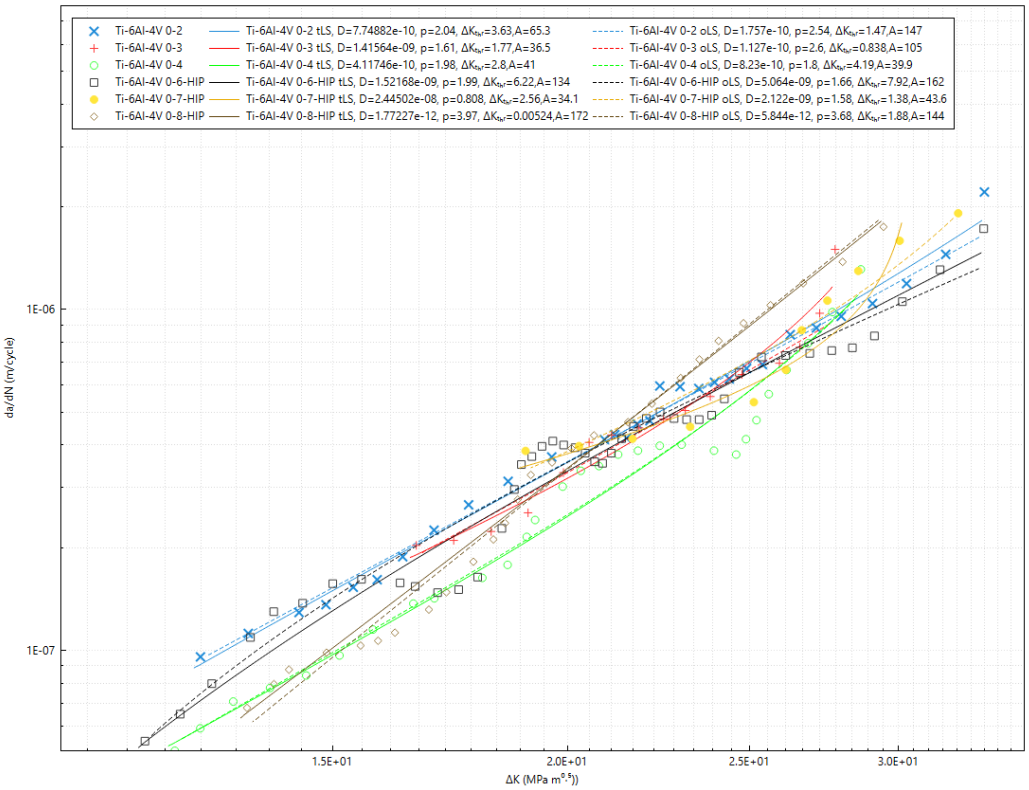


Figure 9: Results of processing of the first dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

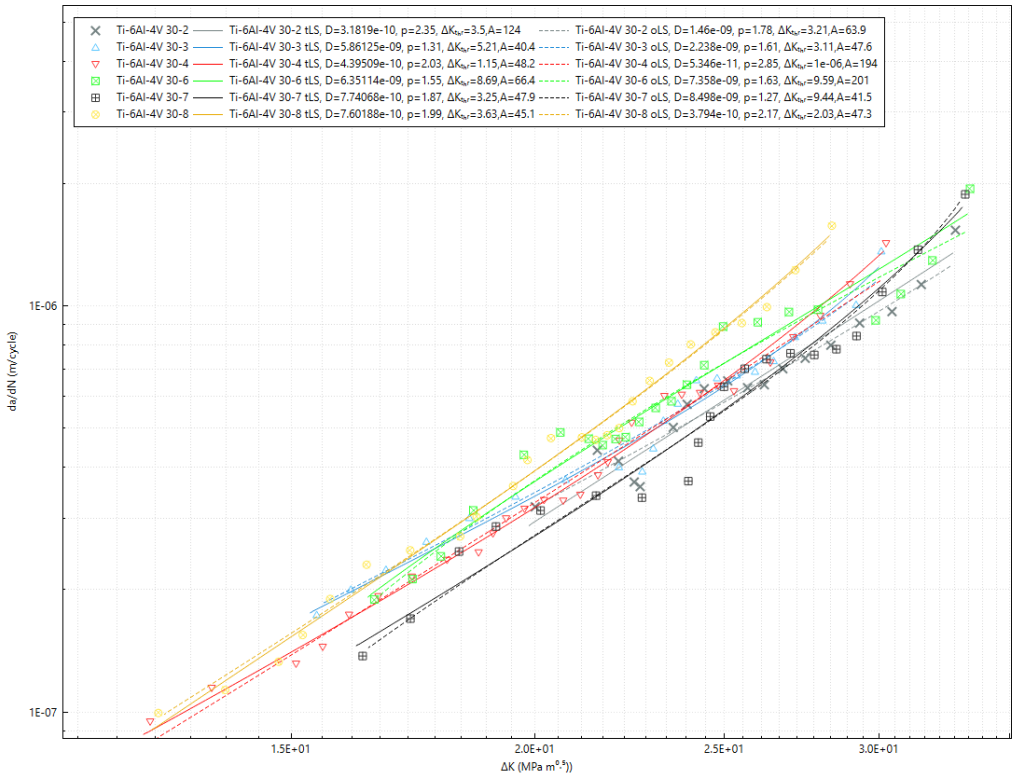


Figure 10: Results of processing of the second dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

From the graphs in these figures it can be observed that although both methods fit the data very well, the respective computed values of the parameters vary drastically between those computed with tLS and oLS. This observation may be interpreted as a strong argument corroborating the view that the values of the parameters correlate very loosely with material properties if one seeks to identify an optimal curve. For example, the value of A that can be used to infer the limiting or critical fatigue crack length, is usually over-estimated if compared to the last data point. Similarly, the value of ΔK_{thr} is underestimated. This observation relates to the inability of the particular model to properly represent the response both in terms of the actual physical response and also in terms of its utility as an engineering tool. The latter though can be remedied by additionally including the limiting response of A and ΔK_{thr} with the identified model parameters, or by following approaches similar to the ones discussed in [6]. The authors are in pursuit of investigating additional crack growth model types, including machine learning and data driven based ones, that can inherently include this information without the need to rely on a disjoint description of the various data components that are needed to relay the crack growth model information.

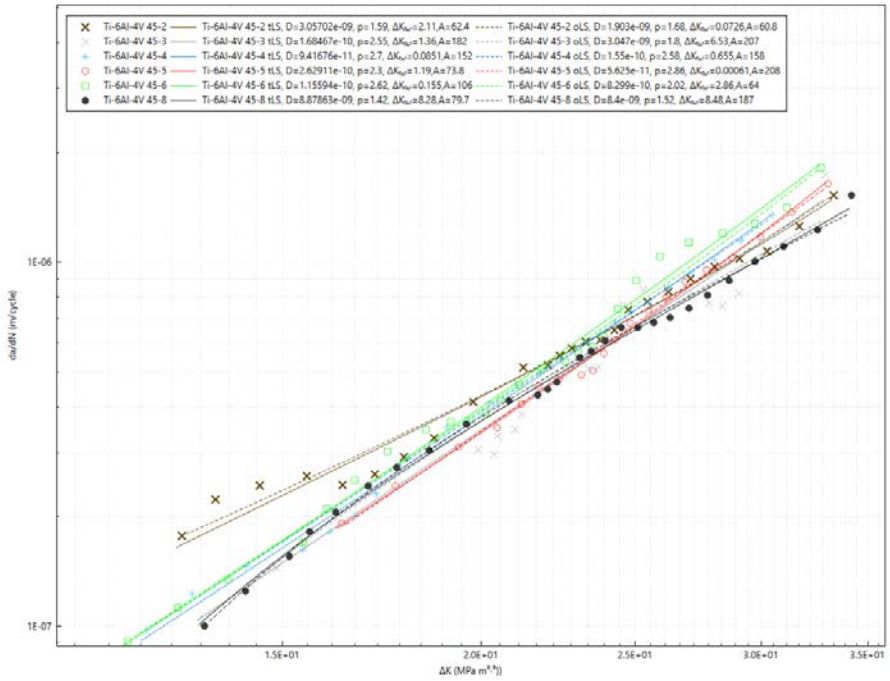


Figure 11: Results of processing of the third dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

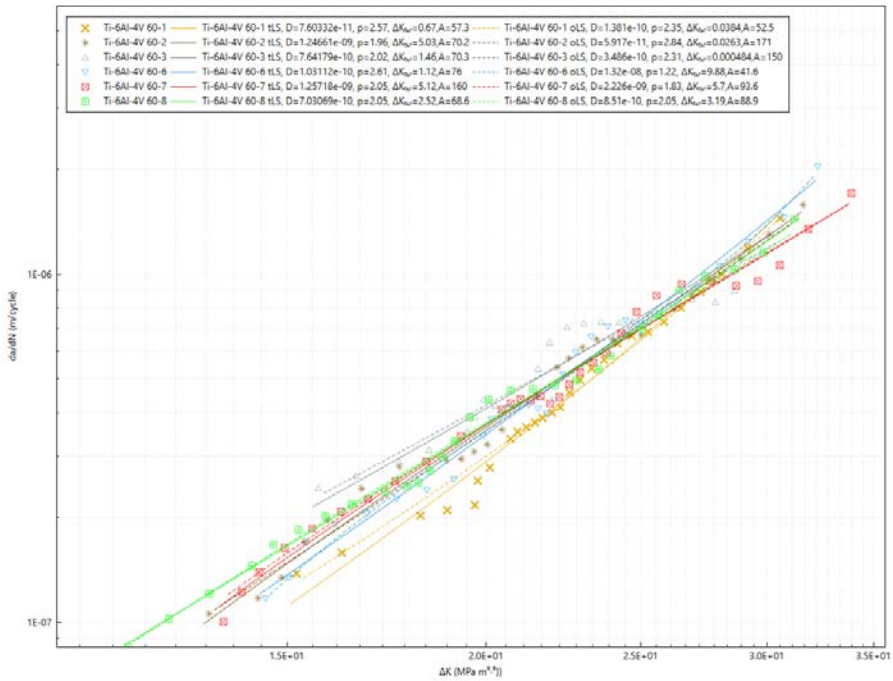


Figure 12: Results of processing of the fourth dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

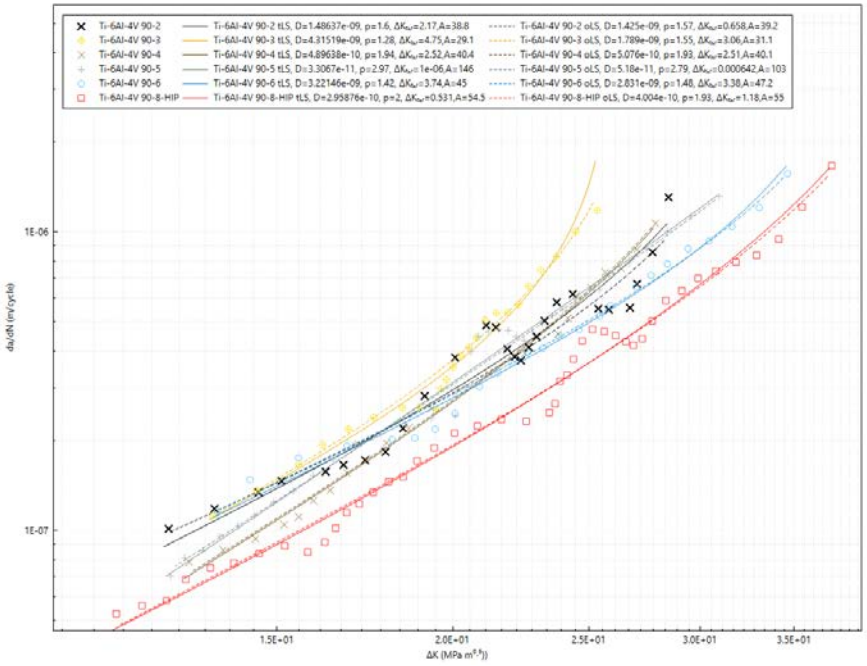


Figure 13: Results of processing of the fifth dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

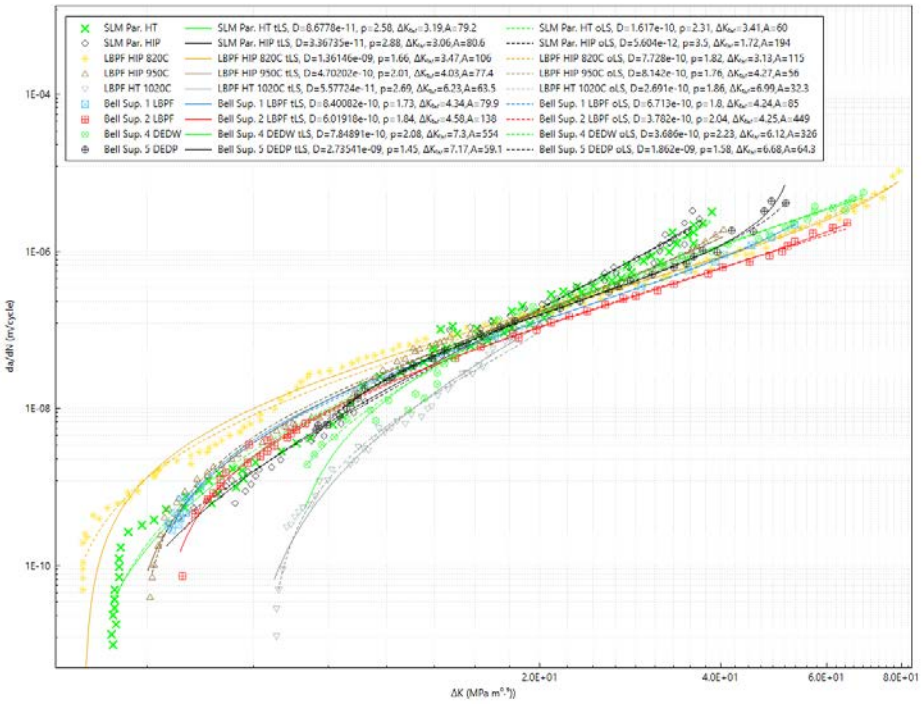


Figure 14: Results of processing of the sixth dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

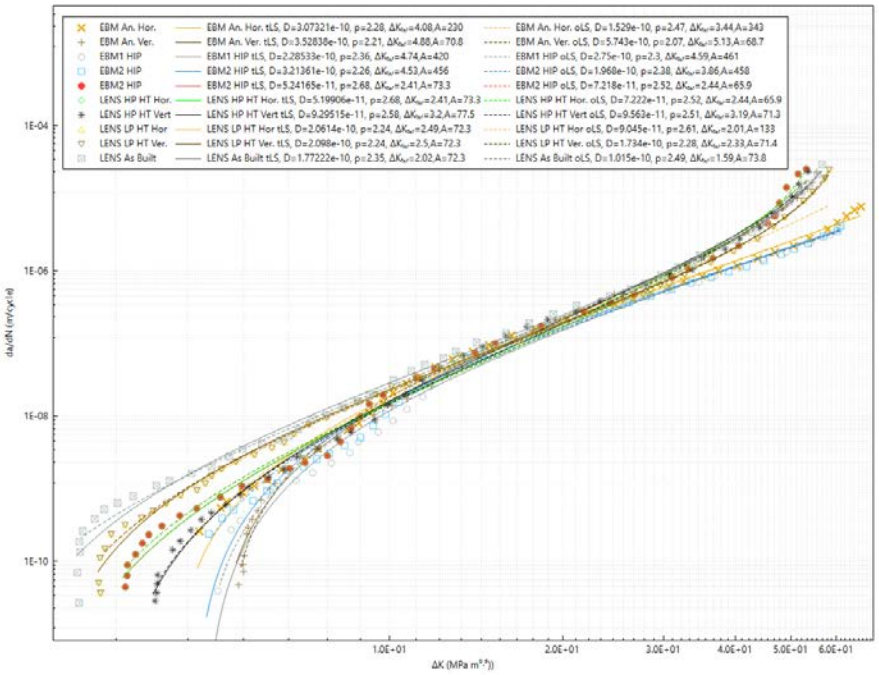


Figure 15: Results of processing of the seventh dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

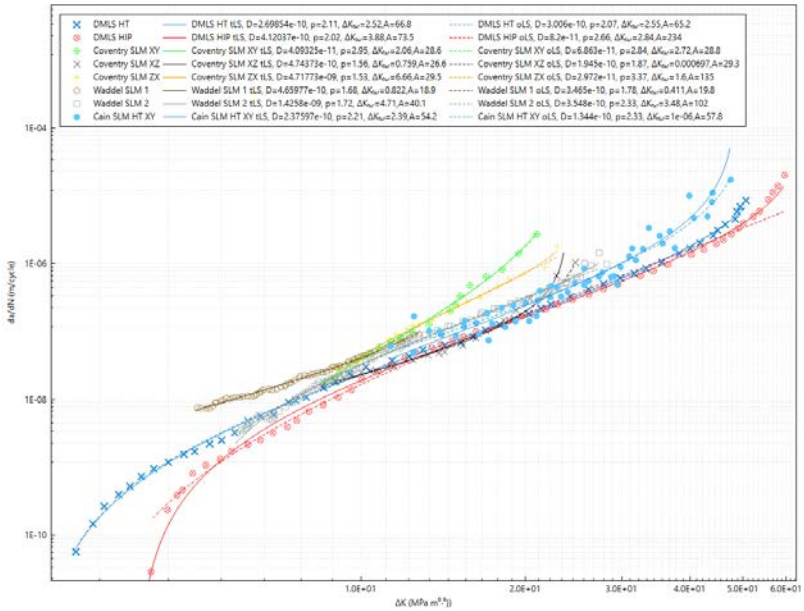


Figure 16: Results of processing of the eighth dataset of Ti-6Al-4V. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

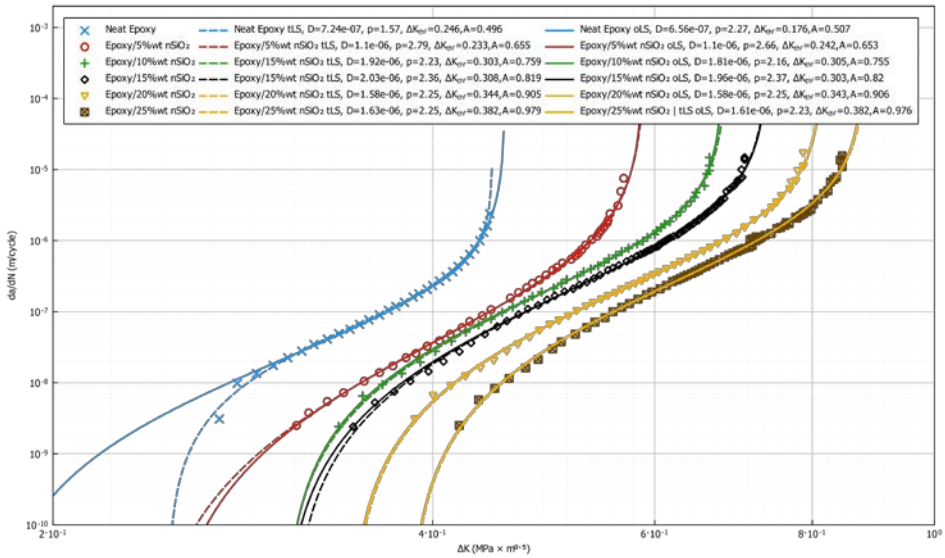


Figure 17: Epoxy with nano-SiO_2 results at various concentrations. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 5.

6.2. Application in nanocomposites consisting of nano-SiO_2 particles in an anhydride-cured epoxy polymeric matrix

This application example involves the results in [47], which examined crack growth in nanocomposites consisting of nano-SiO_2 particles in an anhydride-cured epoxy polymeric matrix. The epoxy polymers possessed a glass transition temperature of about 150°C. The spherical nano-SiO_2 particles were very well dispersed and possessed a mean diameter of 20 nm with a narrow range of particle distribution. The concentrations of the nano-SiO_2 fillers studied were 0, 5, 10, 15,20, and 25 wt.%.

The results of the analysis are shown in Fig. 17. It can be observed that the proposed algorithm computes the crack growth curves in a very satisfactory manner. It should be noted that the same default parameters were used as in the case of Ti-6Al-4V, demonstrating that the approach is robust and automated even if the materials have a vastly different crack growth response.

An interesting observation that can be made in Fig. 17, is that for the 'Neat Epoxy' case, the total Least Squares approach calculates a more reasonable curve that models the first and last stages of the experiment in a more intuitively correct manner than the oLS approach. This is an expected outcome as discussed in previous sections. It should be noted that we expect that the advantageous characteristics of the tLS method will be more prominent when models that provide higher flexibility in describing stages 1 and 3 are employed. The latter is a topic for future research direction.

6.3. Application in NASA experiments of the 'IM7-8552' CFRP composite

To demonstrate the flexibility of the proposed algorithm and software, we applied them to yet another dataset of 'IM7-8552' carbon-fibre reinforced plastic (CFRP) composite fatigue crack growth tests that

exhibit particularly high variability [48]. The particular dataset is provided in terms of the maximum energy-release rate, G_{max}. For the specific R ratio, the maximum energy release rate can be converted to an equivalent range of the energy release rate, $\Delta\sqrt{G}$ [49], and Eqn. 5 can be rewritten as:

$$\frac{da}{dN} = D \left[\frac{\Delta\sqrt{G} - \Delta\sqrt{G}_{\text{thr}}}{\sqrt{1 - \frac{\Delta\sqrt{G}}{(1-R)\sqrt{B}}}} \right]^p . \tag{12}$$

It can be readily observed that Eqns. are 5 and 12 are related 1 to 1 and therefore the implemented computational infrastructure can be utilized, as is, to identify the model parameters.

The results of the parameter identification can be seen in Fig. 18 where it can be observed that both methods identify models that represent the experimental data points very satisfactorily. The software was again used with its default parameters and even if it is used in a different metric (i.e. $\Delta\sqrt{G}$), the only interaction the user had to perform, was to input the data and instruct the software to perform the identification, further corroborating about the feasibility of an automated method for the identification of fatigue crack growth model parameters.

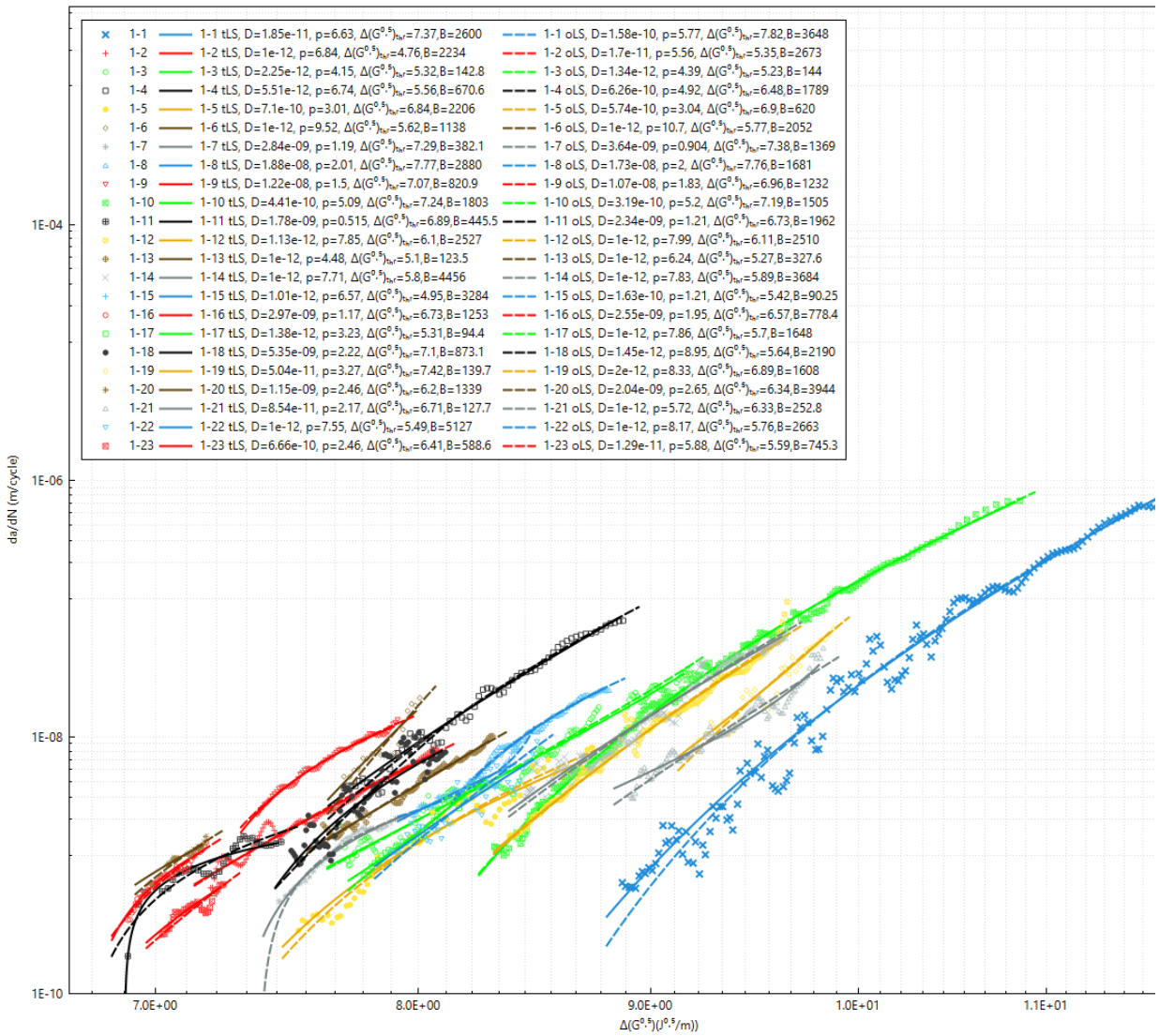


Figure 18: Experimental data and computed crack growth curves for IM7-8552. The graph represents the data points and the identified parameters of the total Least Squares approach (tLS) and of the ordinary Least Squares approach (oLS). The model parameters relate to Eqn. 12.

7. CONCLUSIONS AND PLANS.

In the present paper we presented a comparison between various global optimizers and two optimality criteria for the purpose of the automatic identification of fatigue crack propagation model parameters. It was shown that for crack propagation model identification, the use of optimality criteria based on total least squares (as opposed to the usual ordinary least squares) to avoid evaluating the evolution curves outside asymptotic limits is a necessity for usual optimizers. The use of such an optimality criterion results in very robust automatic identification of crack propagation model parameters by all the investigated algorithms. Alternatively, the global optimization algorithm CUHYSO, that we introduced in the present work can robustly handle both the ordinary least square and the total least square optimality criteria.

Furthermore, a software package that implements a library, a graphical user interface and the synthetic identification tests presented in this paper, was described. The software uses C++ and multi-threading to provide with high performance parameter identification and is available as open-source [38].

By performing parameter identification in a range of materials that exhibit vastly different fatigue crack growth characteristics and also very high variability, we demonstrated that the proposed approach can be used in a fully automated manner, dramatically reducing the need for extensive user experience in identifying such curves. The latter characteristic paves the road for a common engineering tool shared among researchers, scientists and practitioners that provides objective rather than subjective means to represent fatigue crack growth models.

In the future we plan to implement several new features to the software. These include the addition of other crack propagation models as well as a multi-parameter identification method like the one presented in [11]. An important goal is the provision of crack propagation model evaluators for a variety of software applications and languages, including Python, C, MATLAB, Excel and others.

By removing the requirement for continuous manual adjustments in model identification, the developed method and software allows the user to investigate other advanced concepts. One such concept is that of uncertainty quantification, that in the presence of highly non-linear problems, presents various open challenges [50]. Another concept is the investigation of crack-propagation models that are based on grey and black box approaches, including various machine learning and data-driven methods.

REFERENCES

[1] P. Paris and F. Erdogan, "A Critical Analysis of Crack Propagation Laws.," *J. Basic Eng.,* vol. 85, no. 4, p. 528–533, 1963.

[2] W. Liang, F. Conle, T. Topper and S. Walbridge, "A review of effective-strain based and multi R-ratio crack propagation models and a comparison of simulated results using the two approaches," *International Journal of Fatigue,* vol. 142, p. 105920, 2021.

[3] S. Blasón, J. Correia, N. Apetre, A. Arcari, A. De Jesus, P. Moreira and A. Fernández-Canteli, "Proposal of a fatigue crack propagation model taking into account crack closure effects using a modified CCS crack growth model," in *Procedia Structural Integrity*, vol. 1, 2016, pp. 110-117.

[4] H. Ye, T. Wang, C. Wu, Z. Duan and C. Liu, "A comparative analysis of driving force models for fatigue crack propagation of CFRP-reinforced steel structure," *International Journal of Fatigue,* vol. 130, p. 105266, 2020.

[5] R. Forman, V. Shivakumar, J. Cardinal and L. &. M. P. Williams, "Fatigue crack growth database for damage tolerance analysis," 5DOT/FAA/AR-05/15,, Washington, DC, 2005.

[6] S. Kłysz and A. Leski, "Good Practice for Fatigue Crack Growth Curves Description," in *Applied Fracture Mechanics*, A. Belov, Ed., InTech, 2012.

[7] *NASGRO® Fracture Mechanics and Fatigue Crack Growth Analysis Software,* vol. 9.2, NASA-JSC and Southwest Research Institute, 2020.

[8] P. White, L. Molent and S. Barter, "Interpreting fatigue test results using a probabilistic fracture approach," *International Journal of Fatigue,* vol. 27, no. 7, pp. 752-767, 2005.

[9] F. Taheri, D. Trask and N. Pegg, "Experimental and analytical investigation of fatigue," *Marine Structures,* vol. 16, no. 1, pp. 69-91, 2003.

[10] X. Huang and W. &. L. J. Cui, "A model of fatigue crack growth under various," in *A model of fatigue crack growth under various*, Montreal, Canada, 2005.

[11] A. Iliopoulos, R. Jones, J. Michopoulos, N. Phan and R. K. Singh Raman, "Crack Growth in a Range of Additively Manufactured Aerospace Structural Materials," *Aerospace,* vol. 5, no. 4, p. 118, 2018.

[12] R. Jones, J. G. Michopoulos, A. Iliopoulos, R. Singh Raman, N. Phan and T. Nguyen, "Representing crack growth in additively manufactured Ti-6Al-4V," *International Journal of Fatigue,* vol. 116, pp. 610-622, 2018.

[13] G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," *Numer. Anal.,* vol. 17, no. 6, p. 883–893, 1980.

[14] M. Senning, "Process parameter identification: A total least square approach," in *Proceedings of the Third IFAC/IFIP Symposium, 5–8 October 1982*, Madrid,Spain, 1983.

[15] I. Markovsky and V. H. Sabine, "Overview of total least-squares methods," *Signal Processing,* vol. 87, no. 10, pp. 2283-2302, 2007.

[16] K. Sikorski, "Bisection is optimal," *Numerische Mathematik,* vol. 40, pp. 111-117, 1982.

[17] T. Yamamoto, "Historical Developments in Convergence Analysis for Newton's and Newton-like Methods," in *Numerical Analysis : Historical Developments in the 20th Century*, C. Brezinski and L. Wuytack, Eds., Elsevier, 2001, p. 241–263.

[18] G. Alefeld, "On the convergence of Halley's method," *American Mathematical Monthly,* vol. 88, no. 7, pp. 530-536, 1981.

[19] P. A. Samuelson, "A Note on Alternative Regressions," *Econometrica,* vol. 10, no. 1, pp. 80-83, 1942.

[20] D. I. Warton, I. J. Wright, D. S. Falster and M. Westoby, "Bivariate line-fitting methods for allometry," *Biological Reviews,* vol. 81, no. 2, pp. 259-291, 2008.

[21] N. Draper and H. Smith, Applied Regression Analysis, Wiley-Interscience, 1998.

[22] C. Tofallis, "Model Fitting for Multiple Variables by Minimising the Geometric Mean Deviation," in *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*, Dordrecht: Kluwer Academic Publ., 2002.

[23] S. G. Johnson, "The NLopt nonlinear-optimization package," 2021. [Online]. Available: http://github.com/stevengj/nlopt. [Accessed September 2021].

[24] D. Jones, C. Perttunen and B. Stuckman, "Lipschitzian Optimization without the Lipschitz Constant," *J. Optim. Theory Appl.,* vol. 79, pp. 157-181, 1993.

[25] J. M. Kelley and C. T. Gablonsky, "A locally-biased form of the DIRECT algorithm," *J. Global Optimization,* vol. 21 (1), pp. 27-37, 2001.

[26] W. L. Price, "A controlled random search procedure for global optimization," in *Towards Global Optimization 2*, L. C. W. D. a. G. P. Szego, Ed., Amsterdam, North-Holland Press, 1978, pp. 71-84.

[27] W. L. Price, "Global optimization by controlled random search," *J. Optim. Theory Appl.,* vol. 40, no. 3, pp. 333-348, 1983.

[28] P. Kaelo and M. M. Ali, "Some variants of the controlled random search algorithm for global optimization," *J. Optim. Theory Appl.,* vol. 130, no. 2, pp. 253-264, 2006.

[29] S. Kucherenko and Y. Sytsko, "Application of deterministic low-discrepancy sequences in global optimization," *Computational Optimization and Applications,* vol. 30, pp. 297-318, 2005.

[30] T. P. Runarsson and Y. X., "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evolutionary Computation,* vol. 4, no. 3, pp. 284-294, 2000.

[31] T. P. Runarsson and Y. X., "Search biases in constrained evolutionary optimization," *IEEE Trans. on Systems, Man, and Cybernetics Part C: Applications and Reviews,* vol. 35, no. 2, pp. 233-243, 2005.

[32] C. H. da Silva Santos, M. S. Gonçalves and H. E. Hernandez-Figueroa, "Designing Novel Photonic Devices by Bio-Inspired Computing," *IEEE Photonics Technology Letters,* vol. 22, no. 15, p. 1177–1179, 2010.

[33] C. H. da Silva Santos, Parallel and Bio-Inspired Computing Applied to Analyze Microwave and Photonic Metamaterial Strucutures (Ph.D. thesis), University of Campinas, 2010.

[34] H.-G. Beyer and H.-P. Schwefel, "Evolution Strategies: A Comprehensive Introduction," *Journal Natural Computing,* vol. 1, no. 1, pp. 3-52, 2002.

[35] I. Rechenberg, Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (Ph.D Thesis), Reprinted by Fromman-Holzboog (1973), 1971.

[36] R. G. Strongin and D. L. Markin, "Minimization of multiextremal functions under nonconvex constraints," *Cybernetics,* vol. 22, no. 4, pp. 486-493, 1986.

[37] *ISO/IEC 14882:2020 - Programming languages - C++,* Geneva: International Organization for Standardization, 2020.

[38] A. Iliopoulos and J. G. Michopoulos, "CGROW: A framework for fatigue crack growth model parameter determination," U.S. Naval Research Laboratory, 21 Sep 2021. [Online]. Available: https://github.com/USNavalResearchLaboratory/cgrow. [Accessed 21 Sep 2021].

[39] A. Iliopoulos and J. G. Michopoulos, "CGROW - Releases: A framework for fatigue crack growth model parameter determination," U.S. Naval Research Laboratory, 21 Sep 2021. [Online]. Available: https://github.com/USNavalResearchLaboratory/cgrow/releases. [Accessed 29 Sep. 2021].

[40] K. Martin and B. Hoffman, Mastering CMake: A Cross-Platform Build System, New York: Kitware, Inc., 2008.

[41] "Qt: Cross platform software development for embedded and desktop," The Qt Company, Sep 2021. [Online]. Available: https://www.qt.io/. [Accessed 29 Sep 2021].

[42] B. Stroustrup, The C++ Programming Language, Boston, MA: Addison-Wesley, 2013.

[43] A. Iliopoulos and J. G. Michopoulos, "CXXPLOT: A simple to use C++ 2D plotting library.," U.S. Naval Research Laboratory, 29 Sep 2021. [Online]. Available: https://github.com/USNavalResearchLaboratory/cxxplot. [Accessed 29 Sep 2021].

[44] E. Eichhammer, "Qt Plotting Widget QCustomPlot," Sep 2021. [Online]. Available: https://www.qcustomplot.com/. [Accessed Sep 29 2021].

[45] R. Jones, C. Rans, A. Iliopoulos, J. Michopoulos, N. Phan and D. Peng, "Modelling the Variability and the Anisotropic Behaviour of Crack Growth in SLM Ti6Al4V," *Materials,* vol. 14, no. 6, p. 1400, 2021.

[46] A. Iliopoulos, R. Jones, J. Michopoulos, N. Phan and C. Rans, "Further Studies into Crack Growth in Additively Manufactured Materials," *Materials,* vol. 13, no. 10, p. 2223, 2020.

[47] M. Kothmann, R. Zeiler, A. Rios de Anda, A. Bruckner and V. Altstädt, "Fatigue crack propagation behaviour of epoxy resins modified with silica-nanoparticles," *Polymer,* vol. 60, pp. 157-163, 2015.

[48] G. Murri, "Evaluation of Delamination Onset and Growth Characterization Methods under Mode I Fatigue Loading," NASA/TM-2013-217966, Langley Research Center, Hampton, Virginia, 2013.

[49] R. Jones, A. J. Kinloch, J. G. Michopoulos, A. J. Brunner and N. Phan, "Delamination growth in polymer-matrix fibre composites and the use of fracture mechanics data for material characterisation and life prediction," *Composite Structures,* vol. 180, pp. 316-333, 2017.

[50] K. W. Vugrin, L. P. Swiler, R. M. Roberts, N. J. S. Mack and S. P. Sullivan, "Confidence Region Estimation Techniques for Nonlinear Regression: Three Case Studies," Sandia National Laboratories, Albuquerque, 2005.