



PERGAMON

Engineering Fracture Mechanics 70 (2003) 1259–1279

www.elsevier.com/locate/engfracmech

Engineering
Fracture
Mechanics

Fatigue life and crack path predictions in generic 2D structural components

A.C.O. Miranda ^a, M.A. Meggiolaro ^b, J.T.P. Castro ^{b,*}, L.F. Martha ^a,
T.N. Bittencourt ^c

^a Department of Civil Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rua Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil

^b Department of Mechanical Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rua Marquês de São Vicente 225, Rio de Janeiro, RJ 22453-900, Brazil

^c Department of Structural and Foundation Engineering, Polytechnic School at the University of São Paulo (EPUSP),
P.O. Box 61548, São Paulo, SP 05424-970, Brazil

Received 23 August 2001; received in revised form 18 May 2002; accepted 23 May 2002

Abstract

This paper proposes a reliable and cost-effective two-phase methodology to predict crack propagation life in generic two-dimensional (2D) structural components. First, the usually curved fatigue crack path and its stress-intensity factors are calculated at small crack increments in a specialized finite-element software, using automatic remeshing algorithms, special crack tip elements and appropriate crack increment criteria. Then, the computed stress-intensity factors are transferred to a powerful general-purpose fatigue-design program, which has been designed to predict both initiation and propagation fatigue lives by means of classical design methods. Particularly, its crack propagation module accepts any K_I expression and any crack growth rate model, considering sequence effects such as overload-induced crack retardation to deal with 1D and 2D crack propagation under variable amplitude loading. Non-trivial application examples compare the numerical simulation results with those measured in physical experiments.

© 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Crack propagation; Fatigue; Finite elements

1. Introduction

The prediction of fatigue crack propagation life under variable amplitude loading in intricate two-dimensional (2D) structural components is a challenging problem, which can be optimally solved by sequentially using the so-called global and local analysis procedures.

The (generally curved) crack path and its associated stress-intensity factors K_I and K_{II} can be conveniently calculated by a finite-element (FE) global discretization of the component, using appropriate crack

* Corresponding author. Tel.: +55-21-2511-5846; fax: +55-21-2294-9148.

E-mail address: jtcastro@mec.puc-rio.br (J.T.P. Castro).

tip elements, mesh regeneration schemes and crack increment criteria. However, this method is not computationally efficient under variable amplitude loading, since it requires remeshing procedures and FE recalculations of the *whole* structure's stress/strain field at each load event. Both tasks demand intensive and time-consuming numerical calculations. Moreover, the modeling of crack retardation effects increases the numerical burden and compromises even more the efficiency of this global approach.

On the other hand, the direct integration of the material crack growth rate model using an appropriate stress-intensity expression for the crack can be efficiently used to calculate the crack's increment at each load event, considering crack retardation effects if necessary. However, K_I solutions are simply not available for most real components, and the errors involved in using handbook expressions as an approximation increase as the real crack deviates from the tabulated one, making the accuracy of this local analysis questionable and its predictions unreliable in those cases.

Since the advantages of the global and local approaches are complementary, the crack propagation problem can be successfully divided into two steps. First, the crack path and its associated Mode-I stress-intensity factor $K_I(a)$ along crack length a are calculated, under *simple* loading, at small discrete steps using an appropriate FE software. Then, an analytical expression is fitted to the discrete $K_I(a)$ calculated values and exported to a local approach program. Finally, the actual variable amplitude loading is efficiently treated by the integration of the crack growth rate model, considering retardation effects if required.

The purpose of this paper is to describe the fundamentals of a system composed of two complementary programs designed to implement this two-step hybrid methodology, which was validated by means of crack growth experiments on modified four-point bending single-edge notch (SEN) and compact tension (CT) test specimens, in which holes were machined to curve the crack propagation path. This system has demonstrated that satisfactory fatigue life predictions under complex load in arbitrary 2D structural components can now be obtained even in PC environments. The following section describes the numerical procedures to compute stress-intensity factors in 2D geometries.

2. Numerical computation of stress-intensity factors

In 2D FE models, three methods can be chosen to compute the stress-intensity factors along the (generally curved) crack path:

- (i) the displacement correlation technique (DCT) [1];
- (ii) the potential energy release rate computed by means of a modified crack-closure (MCC) integral technique [2,3]; and
- (iii) the J -integral computed by means of the equivalent domain integral (EDI) together with a mode decomposition scheme [4,5].

Since Bittencourt et al. [6] showed that, for sufficiently refined FE meshes, all three methods predict essentially the same results, only the DCT method will be seen here. However, the other two methods also provide good results even for relatively coarse meshes (see Miranda et al. [7] for details).

In the DCT method [1], the displacements obtained from the FE analysis at specific locations are correlated with the analytical solutions expressed in terms of the stress-intensity factors. For quarter-point singular elements near the crack tip, the crack opening displacement δ is given by

$$\delta(r) = (4v_{j-1} - v_{j-2})\sqrt{\frac{r}{L}} = K_I \left(\frac{\kappa + 1}{\mu} \right) \sqrt{\frac{r}{2\pi}} \quad (1)$$

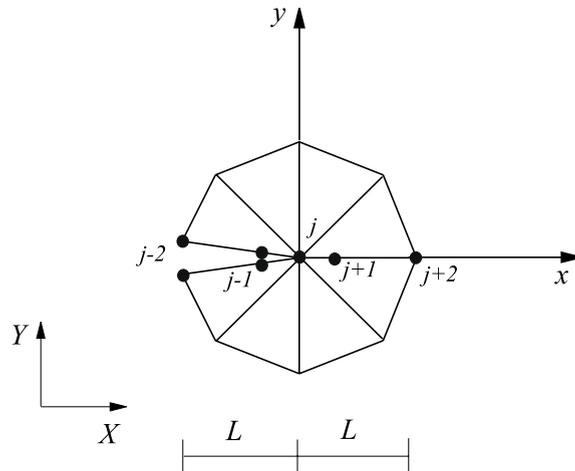


Fig. 1. Quarter-point elements at the crack tip.

where v_{j-1} and v_{j-2} are the relative displacements in the y -direction at the $j - 1$ and $j - 2$ nodes (see Fig. 1), L is the element's size, $\kappa = 3 - 4\nu$ in plane strain or $\kappa = (3 - \nu)/(1 + \nu)$ in plane stress, ν is the Poisson ratio, and μ is the shear modulus. The Mode-I, and analogously the Mode-II, stress-intensity factors can be evaluated from Eq. (1) by

$$K_I = \left(\frac{\mu}{\kappa + 1} \right) \sqrt{\frac{2\pi}{L}} (4v_{j-1} - v_{j-2}) \quad \text{and} \quad K_{II} = \left(\frac{\mu}{\kappa + 1} \right) \sqrt{\frac{2\pi}{L}} (4u_{j-1} - u_{j-2}) \quad (2)$$

where u_{j-1} and u_{j-2} are the relative displacements in the x direction at the $j - 1$ and $j - 2$ nodes (see Fig. 1).

3. Numerical computation of the crack increment direction

Fatigue cracks will attempt to propagate in Mode-I or to minimize the amount of Mode-II loading [6], curving their paths if necessary to avoid rubbing their faces. To simulate this behavior in 2D FE analysis, the most used criteria for the computation of crack growth in linear-elastic regime are: (i) maximum circumferential stress ($\sigma_{\theta \max}$); (ii) maximum potential energy release rate ($\mathcal{G}_{\theta \max}$); and (iii) minimum strain energy density ($S_{\theta \min}$).

Concerning the first criterion, Erdogan and Sih [8] considered that the crack extension should occur in the direction that maximizes the circumferential stress in the region close to the crack tip. On the second criterion, Hussain et al. [9] have suggested that the crack extension occurs in the direction that causes the maximum fracturing energy release rate. And, in relation to the third criterion, Sih [10] assumed that the crack growth direction is determined by the minimum strain energy density value near the crack tip. Bitencourt et al. [6] have shown that, if the crack orientation is allowed to change in automatic fracture simulation, the three criteria provide basically the same numerical results. Since the maximum circumferential stress criterion is the simplest, presenting a closed form solution, it is briefly described below.

The stresses at the crack tip for Modes I and II are given by summing up the stress fields generated by each mode

$$\sigma_r = \frac{1}{\sqrt{2\pi r}} \cos(\theta/2) \left\{ K_I [1 + \sin^2(\theta/2)] + \frac{3}{2} K_{II} \sin \theta - 2K_{II} \tan(\theta/2) \right\} \quad (3)$$

$$\sigma_{\theta} = \frac{1}{\sqrt{2\pi r}} \cos(\theta/2) \left[K_{\text{I}} \cos^2(\theta/2) - \frac{3}{2} K_{\text{II}} \sin \theta \right] \quad (4)$$

$$\tau_{r\theta} = \frac{1}{\sqrt{2\pi r}} \frac{\cos(\theta/2)}{2} [K_{\text{I}} \sin \theta + K_{\text{II}}(3 \cos \theta - 1)] \quad (5)$$

where σ_r is the normal stress component in the radial direction, σ_{θ} is the normal stress component in the tangential direction and $\tau_{r\theta}$ is the shear stress component. These expressions are valid both for plane stress and plane strain. The maximum circumferential stress criterion assumes that crack extension begins on a plane perpendicular to the direction in which σ_{θ} is maximum (thus $\tau_{r\theta} = 0$). Monotonic extension shall occur when $\sigma_{\theta \max}$ reaches a critical value corresponding to a property of the material (K_{IC} for Mode-I). Using $\tau_{r\theta} = 0$, Eqs. (4) and (5) have a trivial solution $\theta = \pm\pi$ for $\cos(\theta/2) = 0$, and a non-trivial solution otherwise

$$K_{\text{I}} \sin \theta + K_{\text{II}}(3 \cos \theta - 1) = 0 \quad (6)$$

This equation implies that for the pure Mode-I case $K_{\text{II}} = 0$, $K_{\text{I}} \sin \theta = 0$ and $\theta = 0^\circ$, and that for the pure Mode-II case $K_{\text{I}} = 0$, $K_{\text{II}}(3 \cos \theta - 1) = 0$ and $\theta = \pm 70.5^\circ$. These are the extreme θ values of the crack propagation angles. The mixed-mode intermediary values are found by solving Eq. (6) for θ , resulting in

$$\theta = 2 \arctan \left(\frac{1}{4} \frac{K_{\text{I}}}{K_{\text{II}}} \pm \frac{1}{4} \sqrt{\left(\frac{K_{\text{I}}}{K_{\text{II}}} \right)^2 + 8} \right) \quad (7)$$

where the θ sign is the opposite of the K_{II} sign.

4. Finite-element mesh generation

The computational models described above have been implemented in a program called Quebra2D (which means 2D fracture in Portuguese) [11,12], an interactive graphics program for simulating 2D fracture processes. The crack representation scheme used in Quebra2D is based on the discrete approach. In this sense, the program is similar to well-known 2D simulators, such as Franc2D [6,13] for example. The discrete approach has been widely employed to simulate a great variety of fracture problems, ranging from aerospace components [6] to concrete gravity dams [14]. However, Quebra2D brings some improvements with respect to its ancestors. It performs adaptive FE analyses, for example, and its graphical interfaces are much more flexible and portable. Moreover, the adaptive analyses are coupled with modern and efficient automatic remeshing schemes. This section concisely describes the automatic FE mesh-generation algorithm specifically devised for this program.

The meshing algorithm works both for regions without cracks and for regions with one or multiple cracks, which may be either embedded or surface breaking. This is an adaptation of an algorithm previously proposed for generating unstructured meshes for arbitrarily shaped 3D regions [15]. The 2D algorithm has been designed to meet four specific requirements, as follows.

First, the algorithm should produce well-shaped elements, avoiding elements with poor aspect ratio. While it does not guarantee bounds on element-aspect ratios, care is taken at each step to generate the best possible shaped elements. Empirical observations show that the algorithm is largely successful in meeting this requirement [7].

Second, the generated mesh should conform to an existing discretization on the region boundary. This is important to simulate crack growth, because it allows local remeshing in a region near a growing crack. The

algorithm, however, is not restricted to small regions near cracks and is relatively fast. In the examples shown in this work, the entire mesh was regenerated at each crack propagation step in a time interval of the order of ten seconds using a Pentium 650 MHz PC with 128 MB of RAM. Another advantage of this strategy is that the boundary curves are discretized independently from the model's domain, thus resulting in a more regular boundary discretization.

Third, the algorithm should shift smoothly between regions with elements of highly varying size, because in crack analysis it is not uncommon for the elements near the crack tip to be two orders of magnitude smaller than the other elements.

Fourth, the algorithm should have specific capabilities for modeling cracks, which are usually idealized without volume. That is, the surfaces representing the two sides of a crack face are distinct, but geometrically coincident. This means that nodes on opposite sides of crack faces may have identical coordinates, and the algorithm must be able to discriminate between the nodes and to select the one on the proper crack side.

The proposed algorithm incorporates well-known meshing procedures [16–22] and introduces some original steps. It includes an advancing front technique along with a quadtree procedure to develop local guidelines for the generated elements' size, taking special care to generate elements with the best possible shape during the advancing front. To enhance the quality of the mesh element's shape, an a posteriori local mesh improvement procedure is used.

Another innovation of the algorithm presented here is the generation of internal nodes simultaneously with the elements. Some authors, e.g. Rassinoux [22], use a quadtree/octree procedure to generate internal nodes prior to element generation. The current algorithm also uses a quadtree, but only as a node-spacing function. This approach tends to give a better control over the generated mesh quality and to decrease the amount of heuristic cleaning-up procedures. Moreover, it specifically handles discontinuities in the domain or boundary of the model, such as the evolving crack examples that will be shown below.

The input data is a polygonal description of the boundary of the region to be meshed, given by lists of nodes defined by their coordinates and of boundary segments (or edges) defined by their node connectivities. This type of input can represent geometries of any shape, including holes or cracks. From the boundary segments, a background auxiliary quadtree structure is created to control the sizes of the FE generated by the advancing front technique. The given boundary edges form the initial front that advances as the algorithm progresses. At each step of this meshing procedure, a new triangle is generated for each front base edge. The front advances replacing the base edge with new triangle edges. Consequently, the domain region is contracted, possibly into several regions. The process stops when all the contracted regions result in single triangles. The overall algorithm is presented next.

4.1. *Quadtree generation*

The first phase of the algorithm involves quadtree generation. To illustrate the required steps, a hypothetical example of a boundary input data is considered, as seen in Fig. 2a. The steps in this phase are as follows:

- Quadtree initialization based on given boundary edges (Fig. 2b). Each segment of the input boundary data is used to determine the local subdivision depth.
- Refinement to force a maximum cell size (Fig. 2c). The quadtree is refined to guarantee that no cell in its interior is larger than the largest cell at the boundary.
- Refinement to provide minimum size disparity for adjacent cells (Fig. 2d). This additional refinement forces only one level of tree depth between neighboring cells and provides a natural transition between regions of different degrees of mesh refinement.

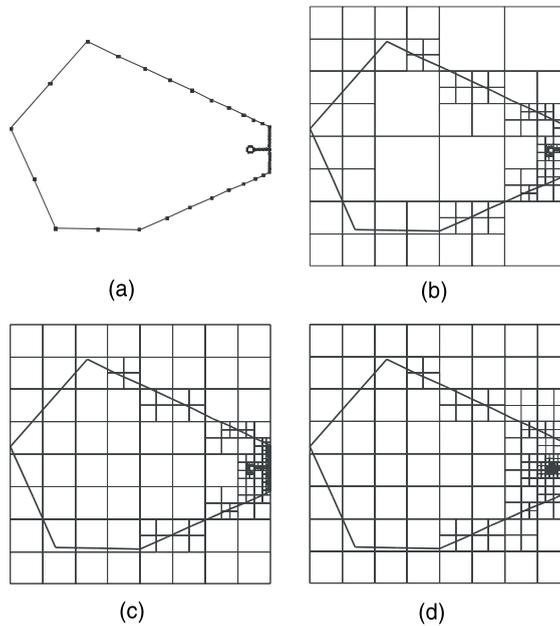


Fig. 2. Generation of background quadtree structure from a given boundary refinement.

4.2. Advancing front procedure

The advancing front method is divided into two phases. The first, called *geometry-based phase*, is based solely on the shape of the elements. In the second, called *topology-based phase*, the optimal element shape criteria are raised and the algorithm attempts to create valid triangles based only on topology, as in any advancing front method. This two-phase advancing front technique is certainly one of the advantages of the presented meshing scheme. The required steps for the advancing front procedure are as follows:

- Front initialization based on given boundary edges, where the current boundary edges are stored in two separated lists. The first is a list of active edges, which includes all boundary edges that have not been used in an attempt to generate well-shaped triangles (geometry-based phase). The other is a list of rejected edges, that is, the edges that failed in the element generation with well-shaped triangles (topology-based phase).
- Front contraction. In this step, for each base edge on the advancing front, it is necessary to:
 - Determine the optimal location $M1$ for a candidate node based on the local quadtree cell size (Fig. 3a).
 - Find all nearby nodes on the current front. The optimal point defines an optimal region where the vertex of the triangle to be generated is located. This region is a sector of a circle with center at the optimal point and a radius proportional to the quadtree cell size (see Fig. 3a).
 - Insert a new node at the optimal location $M1$ and generate an element using this node, if no nearby node is found inside the optimal region. Or, if more than one node is found in the region, rank them according to the angle that will be created with the base edge (Fig. 3b). Additional geometric checks are performed to ensure that the edges of the new element do not intersect any existing edge of the advancing front. If this is the case, then the element is rejected.
 - If there are two or more nodes with the same coordinates, as can occur in crack problems (see Fig. 3b), select the proper node using a simple test, based on the list of adjacent boundary edges of the

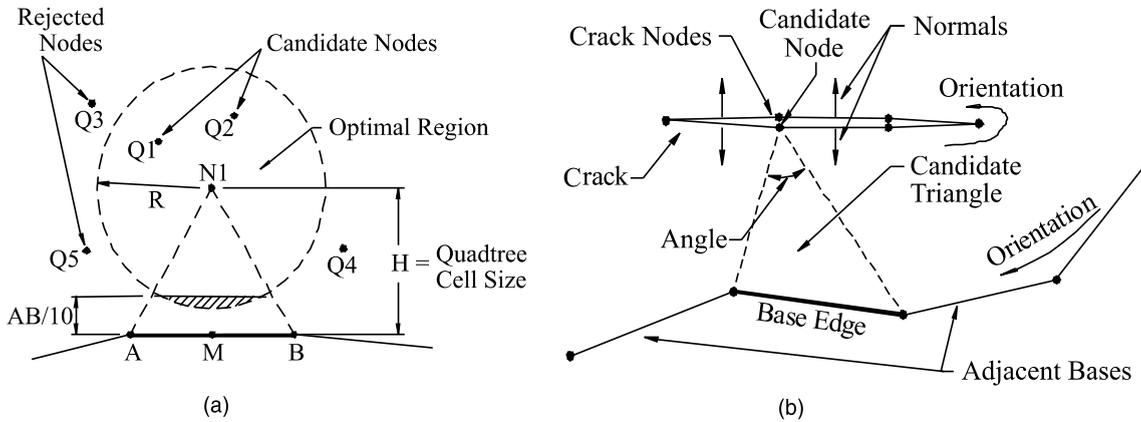


Fig. 3. Determination of a triangle in the advancing front procedure.

nodes on the advancing front. The normals to the crack curves adjacent to the selected nodes are used to perform this test, as illustrated in Fig. 3b, assuming that all crack curves are smooth (with no abrupt change of direction nor bifurcation).

- Update the list of active edges, once a valid triangle is generated for the current base edge.
- If the algorithm fails in forming a valid triangle for the current boundary base edge due to geometric bounds imposed by the current advancing front, remove it from the list of active edges, storing it in the separate list of rejected edges. The topology-based element generation phase starts when a boundary edge fails twice in trying to generate an optimal element. The list of rejected edges of the previous phase is transformed into a list of active edges. In this phase, any node close to the current base edge is selected. The node that has the best angle metric with respect to the base edge is chosen for the generation of the new triangle. This phase always generates a valid mesh (although not optimal), because it is always possible to triangulate a region that is defined by its boundary edges [23].

4.3. Local mesh improvement

The last phase of the algorithm is local mesh improvement. The required steps for this phase are as follows:

- Node relocation smoothing. The first procedure is a conventional nodal relocation smoothing technique [18,24] based on node coordinate averaging.
- Quality evaluation and local back-tracking with element deletion. The back-tracking procedure consists of deleting an element that is classified as poorly shaped and a group of elements in its vicinity. The classification of a “bad” triangle is based on a specified measure [7]. For each element of the generated mesh, this quality measure is evaluated. If the value of this metric is above a predefined limit value, then the element is classified as a poorly shaped element. A local polygon to be meshed is created by an algorithm that is defined through the following steps (see Fig. 4):
 - *Step 1:* Creation of an initial list of adjacent nodes. For each element that is adjacent to the poorly shaped element (the black element in Fig. 4), its nodes are stored in the list. An adjacent element is defined as an element that shares two nodes (an edge) with the “bad” element.
 - *Step 2:* Creation of an extended list of adjacent nodes. These additional nodes share more than one element with nodes in the initial Step 1 list. This step has been incorporated because it results in polygons with better shapes for reconstruction.

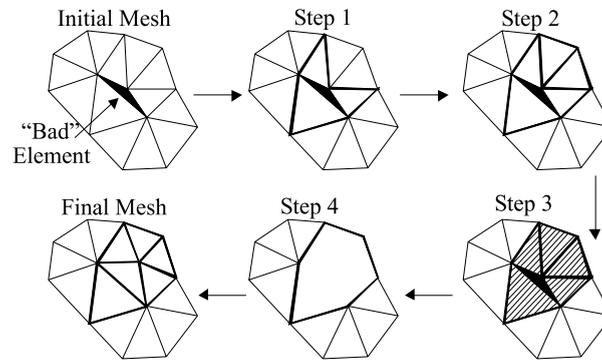


Fig. 4. Back-tracking procedure to remesh around a "bad" element.

- *Step 3*: Creation of a list of elements to be deleted. These elements are defined when three nodes are present in the extended list of adjacent nodes.
- *Step 4*: Creation of the local polygon to be remeshed. For each element to be deleted, each of its edges is inserted into the polygon boundary if it is not there yet, otherwise the edge is removed. After the creation of the local polygon, elements are generated by means of applying the previously described advancing front algorithm procedure.

The efficiency and reliability of the presented mesh-generation algorithm has been attested in a previous work [7]. An example of crack propagation FE simulation in a model with a relatively complex geometry, using the presented algorithm, will be discussed in Section 6.

The current algorithm may be optionally used in an adaptive mesh-generation scheme that is based on an a priori boundary refinement, such as the scheme devised by Paulino et al. [25]. In this case, the adaptive process first requires the analysis results from an initial FE mesh, usually rough, with the geometric description, boundary conditions and its attributes. Then a discretization of the domain's region boundary is performed, based on the geometric properties and characteristic sizes of the boundary elements (adjacent to the boundary curves), determined by the error estimation from the previous step of the FE analysis. From this discretization, a new mesh is generated using the algorithm described above with one minor improvement: as the quadtree structure is used to guide the size of the generated elements, an additional quadtree refinement is performed after the initial quadtree is generated. This additional refinement takes into account the characteristic element sizes that are determined by the error estimation analysis.

5. Modeling of fatigue crack growth under variable amplitude loading by the local approach

The modeling of fatigue crack propagation under variable amplitude loading by the *local* approach is discussed below. The loading complexity, whose amplitude can randomly vary in time, is unlimited. However, multiple-origin loads, which induce stresses whose principal directions vary significantly in time, are considered beyond the scope of this discussion. Sequence effects are considered, such as overload-induced crack retardation or arrest. Only Mode-I is discussed, since most fatigue cracks propagate perpendicularly to the maximum tensile stress.

The local approach is so called because it does *not* require the global solution of the stress/strain field in the entire structure. Its predictions are based on the direct integration of the material fatigue crack growth rate model, $da/dN = F(\Delta K, R, \Delta K_{th}, K_C, \dots)$, where ΔK is the stress-intensity range of the propagating crack, $R = K_{min}/K_{max}$ is a measure of the mean load, ΔK_{th} is the fatigue crack propagation threshold, and

K_C is fracture toughness. Therefore, neither the ΔK expression nor the crack propagation model should have their accuracy compromised in the name of mathematical simplicity when using this approach. In addition, most environmental effects can be dealt with an appropriate da/dN model.

In practice, fatigue crack growth calculations under variable amplitude loading require the aid of appropriate automation software, since the numerical effort to integrate the ΔK expressions can be quite heavy. There is a number of automation software available for that purpose, such as MSC.Fatigue, from MSC.Software, or NASGRO, the crack growth program developed by NASA [26,27]. The latter, for instance, calculates crack propagation under variable amplitude loading for several specimen's geometries, considering load interaction effects if necessary. However, this software restricts many of the user options, such as limiting the choice of the da/dN expression to Forman–Newman's equation, with load interaction models restricted to variations of the Willenborg, strip-yield, and constant closure models [27].

A powerful program named ViDa (which means "life" in Portuguese, but also stands for *V*isual *D*amagemeter) has been developed to automate the fatigue design routines by all local methods, giving the user total control over the calculation procedures [28,29]. This software can calculate fatigue crack propagation and also crack initiation using the $S-N$, the IIW (for welded structures) and the $\epsilon-N$ methods. Its crack growth routines include several load interaction models and over thirty da/dN models, but it accepts any other by means of its equation interpreter, reflecting its open philosophy. In particular, crack growth predictions using ViDa have been successfully validated through a direct comparison with similar calculations performed using NASA's NASGRO software.

In the following, first the main features of the ViDa software are concisely described and then the cycle-by-cycle integration method and the modeling of load sequence effects are discussed.

5.1. The ViDa software

The purpose of this software is to automate, in a friendly environment, all calculations required to predict fatigue life under variable amplitude loading by means of the local approach. It runs on PCs under Windows 95/NT or better operating systems, including all necessary tools to perform the predictions, such as an intuitive and friendly graphical interface in six languages; intelligent databases for stress concentration and intensity factors, crack propagation models, material properties, and more; traditional and sequential rain-flow counters; graphical output for all computed results, including elastic–plastic hysteresis loops and 2D crack fronts; automatic adjustment of crack initiation and propagation experimental data; an equation interpreter, etc. Crack growth can be calculated considering any propagation model and any ΔK expression that can be typed in a BASIC syntax (making it an ideal companion to the Quebra2D software, which can be used to generate the $\Delta K(a)$ expression if it is not available in the database).

The software has safety features to automatically stop the calculations if, during any loading event, it detects that: (i) $K_{\max} = K_C$; (ii) the crack has reached its maximum specified size; (iii) the stress in the residual ligament reaches the rupture strength of the material S_U ; (iv) da/dN reaches 0.1 mm/cycle (for most engineering alloys, above this rate the problem is fracturing, not fatigue cracking); or else if (v) one of the borders of the piece is reached by the crack front, in the part-through crack propagation case (however, for some geometries, the software is able to model the transition from part-through to through cracks). Moreover, it informs the user when there is yielding in the residual ligament before the maximum specified crack size or number of load cycles is reached. This way, the computed values can be used with the guarantee that the validity limits of the mathematical models are never exceeded.

5.2. The cycle-by-cycle integration method

This method associates to each load reversal the growth the crack would have if that 1/2 cycle was the only one to load the piece. Applying this assumption, it is easy to write a general expression for the

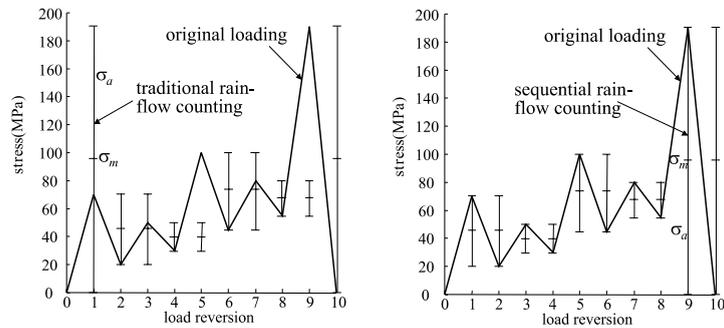


Fig. 5. Traditional rain-flow counting (anticipating the large load events) and sequential rain-flow counting (preserving most of the loading order).

incremental crack growth. Defining $da/dN = F(\Delta K, R, \Delta K_{th}, K_C, \dots)$, a_i and $\Delta\sigma_i$ as the crack length and stress range in the i th 1/2 cycle of the loading, and R_i as the mean load ratio, then the crack will grow by δa_i given by:

$$\delta a_i = \frac{1}{2}F(\Delta K(\Delta\sigma_i, a_i), R(\Delta\sigma_i, \sigma_{max_i}), \Delta K_{th}, K_C \dots) \quad (8)$$

The total growth of the crack is quantified by $\sum(\delta a_i)$. Therefore, the cycle-by-cycle method is similar in concept to the linear damage accumulation rule used in the SN and ϵN fatigue design methods. And, as in Miner's rule, it requires that *all* events that cause fatigue damage be recognized before the calculation, e.g. by rain-flow counting the loading. But, since it must be applied sequentially, load interaction effects can be recognized.

However, the traditional rain-flow counting algorithm alters the loading *order*, as shown in Fig. 5. This can cause serious problems in the predictions, since loading order effects in crack propagation are of *two* different natures: (i) *delayed* effects, which can retard or stop the subsequent crack growth due, for instance, to plasticity-induced Elber-type crack closure [30] or to crack tip bifurcation (these interaction effects among the loading cycles usually increase crack life and, if neglected, may induce excessively conservative predictions); and (ii) *instantaneous* fracture, which occurs in the first load peak where $K_{max} \geq K_C$, an event which must, of course, be precisely predicted.

Since the ViDa loading input can preserve the time *order* information, a *sequential* rain-flow counting option was introduced in that software [29]. With such technique, the effect of each large loading event is counted when it actually happens, and not *before* its occurrence, as in the traditional rain-flow method (see Fig. 5). Therefore, the *sequential* rain-flow option avoids the anticipation of overload-induced effects, which can cause *non-conservative* crack propagation life predictions: as $K(\sigma, a)$ usually grows with the crack, a given overload applied when the crack is large can be much more harmful than when the crack is small. Sequential rain-flow avoids most sequencing problems caused by the traditional method, and it is certainly an advisable option since it presents advantages over the original algorithm while maintaining its main features without increasing its difficulty.

5.3. Load interaction models

It is well known that load cycle interaction effects can be very important when predicting fatigue crack growth. There is a vast literature proving that tensile overloads can cause retardation or arrest of the subsequent crack growth, and that compressive underloads can sometimes affect the crack propagation rate [30–32].

Neglecting load interaction effects in fatigue life calculations can completely invalidate the predictions. In fact, only after considering overload-induced retardation effects can the life reached by real structural components be justified when modeling many practical problems. However, the generation of a universal algorithm to quantify these effects is particularly difficult, due to the number and the complexity of the mechanisms involved, such as: plasticity-induced crack closure; blunting and/or bifurcation of the crack tip; residual stress and/or strain fields; strain-hardening and/or strain-induced phase transformation; crack face roughness; and oxidation of crack faces. Besides, depending on the case, several of these mechanisms may act concomitantly or competitively, as functions of factors such as crack size, material microstructure, dominant stress-state, and environment.

On the other hand, the main characteristic of fatigue cracks is to propagate cutting a material that has already been deformed by the plastic zone that is always around their tips. Fatigue crack faces are embedded in an envelope of (plastic) residual strains and are consequently compressed when completely discharged, and open, alleviating in a progressive way, the (compressive) load transmitted through them.

According to Elber [33], the crack tip is stressed only after completely opening the crack at a stress-intensity K_{op} . Therefore, the greatest the K_{op} , the smaller would be the effective stress-intensity range $\Delta K_{eff} = K_{max} - K_{op}$, and this ΔK_{eff} instead of ΔK would be the fatigue crack propagation rate controlling parameter. Most load interaction models are, although indirectly, based on this idea. This implicates the supposition that the main retardation mechanism is caused by plasticity-induced crack closure: in these cases, the opening load should *increase* when the crack penetrates the plastic zone inflated by the overload, *reducing* ΔK_{eff} and stopping or delaying the crack, while the plastic zones associated with the loading are contained in the overload-induced plastic zone.

Several mathematical models have been developed to account for load interaction in crack propagation based on Elber's crack closure idea. In these methods, the retardation mechanism is only considered within the plastic zone situated in front of the crack tip. According to these procedures, a larger plastic zone Z_{ol} is created by means of an overload (see Fig. 6). When the overload is removed, an increased compressive stress state is set up in the volume of its plastic zone, reducing crack propagation under a smaller succeeding load cycle.

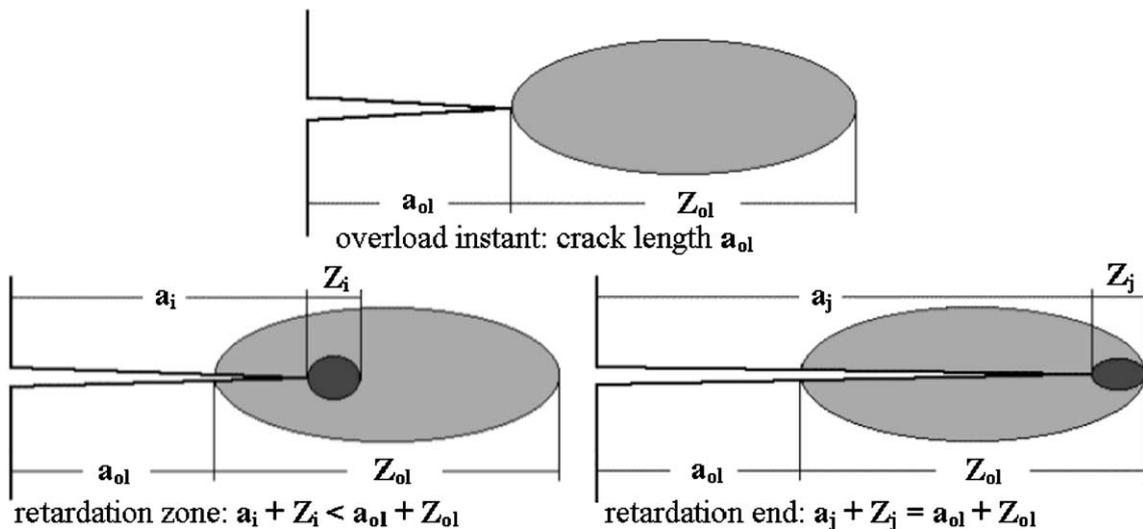


Fig. 6. Yield zone crack growth retardation region used by the Wheeler and Willenborg load interaction models.

The detailed discussion of this complex phenomenology is considered beyond the scope of this work, but a revision of the phenomenological problem can be found in Broek's work [31]. A taxonomy of the load interaction models has been presented by Meggiolaro and Castro [32], including modifications proposed to better model effects such as crack arrest, crack acceleration due to compressive underloads, and small cracks. They classified these models in 4 categories: (i) da/dN models, such as the Wheeler model, which use empirical retardation functions to directly reduce the calculated crack propagation rate da/dN ; (ii) ΔK models, such as the modified Wheeler model, which use similar retardation functions to reduce the value of the stress-intensity factor range ΔK ; (iii) R_{eff} models, such as the Willenborg model, which introduce an effective stress ratio R_{eff} , calculated by reducing the maximum and minimum stress-intensity factors acting on the crack tip, however not necessarily changing the value of ΔK ; and (iv) K_{op} models, such as the strip-yield model, which use estimates of the opening stress-intensity factor K_{op} to directly account for Elber-type crack closure.

There are other retardation models, but none of those that can be implemented in a local-approach code has definitive advantages over the models classified above. This is no surprise, since single equations are too simplistic to model all the several mechanisms that can induce retardation effects. Therefore, in the same way that a curve da/dN vs. ΔK is experimentally measured, a propagation model should be fitted to experimental data to calibrate the retardation models, as recommended by Broek [31].

The numerical implementation of these retardation models in a cycle-by-cycle algorithm is not conceptually difficult, but it requires a considerable programming effort [32]. All load interaction models presented in [32] have been implemented in the ViDa software.

6. Experimental verification of the crack propagation modeling procedures in arbitrary 2D geometries

This section describes the testing procedures used to experimentally verify the modeling techniques proposed for predicting the curved crack fatigue propagation problem. For this purpose, two types of specimens were tested under constant amplitude fatigue loading with $R = 0.1$: a modified four-point bending SEN and a CT specimen, in which holes were machined to curve the crack propagation path (see Figs. 7 and 8, whose dimensions are in mm).

The tested material was a cold rolled SAE 1020 steel, with the analyzed weight percent composition: C 0.19, Mn 0.46, Si 0.14, Ni 0.052, Cr 0.045, Mo 0.007, Cu 0.11, Nb 0.002, Ti 0.002, Fe balance. The Young's modulus was $E = 205$ GPa, $S_Y = 285$ MPa the yield strength, $S_U = 491$ MPa the ultimate strength, and $RA = 53.7\%$ the area reduction. These properties were measured according to the ASTM E 8M-99 standard. The da/dN vs. ΔK data, also obtained under a stress ratio $R = 0.1$ and measured following ASTM E 647-99 procedures, was fitted by the modified Elber equation $da/dN = 4.5 \times 10^{-10} (\Delta K - \Delta K_{\text{th}})^{2.1}$, where $\Delta K_{\text{th}} = 11.6$ MPa $\sqrt{\text{m}}$ was the threshold stress-intensity range. The experimentally measured da/dN vs. ΔK curve is shown in Fig. 9.

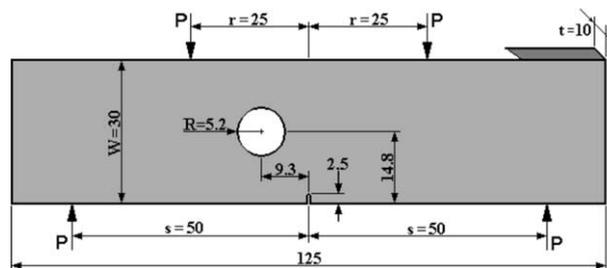


Fig. 7. Geometry of the modified SEN specimen (dimensions in mm).

In summary, the automatic calculation procedure used during the test specimens' design phase was: (i) the FE model of the holed specimen with the specified initial crack was solved to obtain its K_I and K_{II} stress-intensity factors and the corresponding propagation direction; (ii) the crack was incremented in the growth direction by the (small) specified step; (iii) after that, the model was remeshed to account for the new crack size; and (iv) the process was repeated until the required final crack size was reached.

Each calculation step, including automatic remeshing and FE solution for K_I , K_{II} and the incremental growth direction, required about 4 s on a PC. With an average of thirty increments to simulate the entire crack growth, the total calculation process took about 2 min. Therefore, it was indeed practical to implement the trial-and-error procedure used to optimize the design of the test specimens.

Although the curved crack path geometry is 2D, once it is known the crack itself can be described by its (1D) length a measured *along* the crack path. Hence, its K_I expression can be written as a function of a , $K_I(a) = \sigma\sqrt{\pi a}f(a/w)$. These discrete values of the geometry factors $f(a/w)$ calculated for each analyzed crack by Quebra2D were then exported to ViDa, where they were automatically fitted by appropriate continuous analytical functions.

Using these $K_I(a)$ expressions and the da/dN crack propagation curve measured under pure Mode-I loading, the load programs that would be applied during the tests of the hole-modified SEN and CT specimens were calculated to maintain a quasi-constant stress-intensity range around $\Delta K_I \approx 20$ MPa $\sqrt{\text{m}}$, with $R = K_{\min}/K_{\max} = 0.1$. These loading values induce a stage-II (Paris regime) fatigue crack growth in the 1020 steel da/dN vs. ΔK curve. The fatigue lives associated with the load programs were predicted in about 3 s using ViDa.

The experimental procedures used during the tests were very similar to those in the standard measurement of da/dN vs. ΔK curves. All the tests were run at 20 Hz frequency in a 250 kN computer-controlled servo-hydraulic testing machine. The loads were regularly adjusted to maintain the specified quasi-constant ΔK_I . The only major difference was the use of a digital camera and an image-analysis program to measure the crack size *and* path. This is a quite precise and economical option to automate these measurements, but its details are considered beyond the scope of this paper.

Before ending this section, it is worth mentioning that the design of these non-conventional test specimens, using Quebra2D to obtain the most interesting curved crack path and ViDa to predict their associated fatigue lives, is an interesting example of the practical potential of this virtual analysis methodology. These friendly software were easily and efficiently used to model the fatigue propagation of curved cracks in relatively complex 2D geometries, identifying and predicting their behavior in a very short time using inexpensive computational facilities. The actual behavior of the predictions is studied below.

6.1. Single-edge notch specimen

Cracks were fatigue propagated in SEN specimens with a hole slightly to the side of the starting notch line (created using a 0.3 mm jeweler's saw). Fig. 7 shows the dimensions of the specimen used to obtain the results described below, but several others were tested [34]. The final FE mesh automatically generated for predicting the propagation path is illustrated in Fig. 10. Note the density of this mesh around the crack path and, particularly, around its tip. The initial mesh had 1995 elements and 4185 nodes, and the final one had 2585 elements and 5467 nodes.

Fig. 11 compares the $f(a/w)$ expression calculated for this holed test specimen with the standard SEN expression obtained in the literature [31]. It can be observed that the hole has a significant influence in the $f(a/w)$ value.

Fig. 12 shows a picture of the real crack path after the test and the FE crack path prediction made *before* the test. This modeling is indeed quite satisfactory. Therefore, accurate K_I values calculated along this crack path could also be used to check the predicted fatigue life.

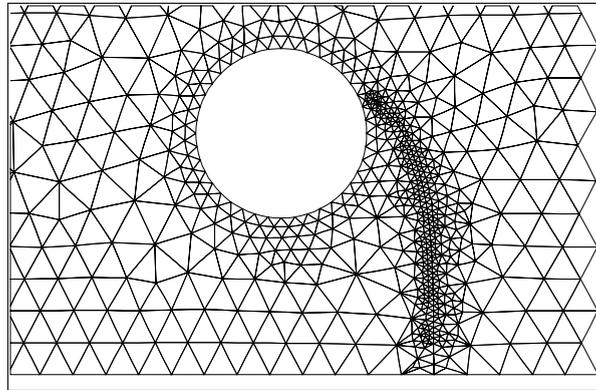


Fig. 10. FE mesh automatically generated for the modified SEN specimen.

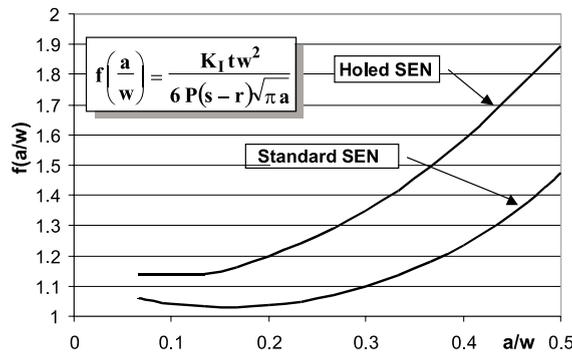


Fig. 11. Calculated $f(a/w)$ expression for the standard and the modified SEN specimen.

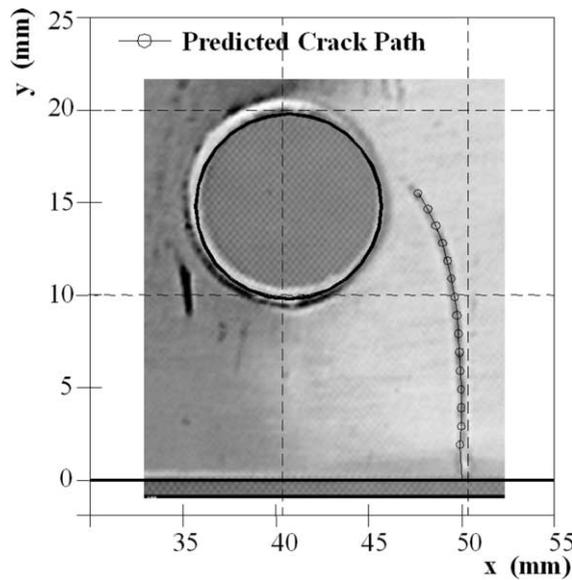


Fig. 12. Real and predicted crack paths for the modified SEN specimen.

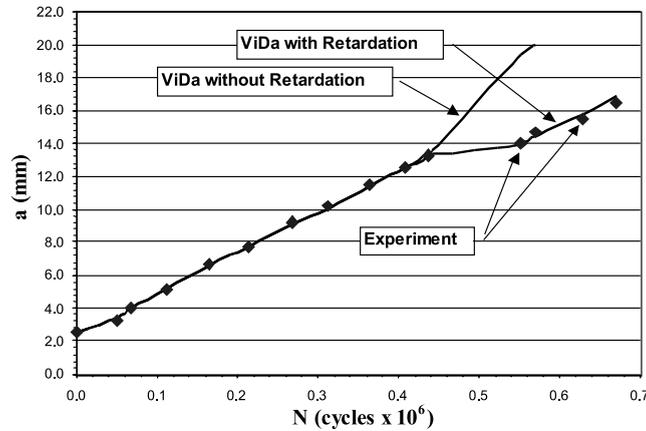


Fig. 13. Predicted and measured fatigue crack propagation behavior for the modified SEN specimen.

However, a significant deviation was observed when comparing the experimental data with the predicted life. They only match during part of test, as shown in Fig. 13, but, after about 440,000 load cycles, a clear deviation was present in the a vs. N graph. Since the initial match between the predicted and the measured crack behavior indicated that the integration of the material da/dN model using the calculated K_I values was correctly reproducing the expected crack growth rate, it could be concluded that this was a renewed indication that the modeling was obtaining satisfactory results and that the deviation could only be due to an unplanned accident during the test.

Indeed, a review of the loading history identified that an accidental 60% overload had occurred at around 440 k cycles during a load adjustment procedure, which was confirmed by the identification of a larger plastic zone observed on the crack path at that point. This single overload was then simulated in the ViDa software, using its crack retardation models which in principle were developed to predict *straight* crack load interaction effects. The idea was to check if such models could also be used to describe overload-induced retardation of curved cracks. The modified Wheeler model,

$$\Delta K_{\text{ret}}(a_i) = \Delta K(a_i) \left(\frac{Z_i}{Z_{\text{ol}} + a_{\text{ol}} - a_i} \right)^\gamma, \quad a_i + Z_i < a_{\text{ol}} + Z_{\text{ol}} \quad (9)$$

where $\Delta K_{\text{ret}}(a_i)$ is the reduced value of the stress-intensity factor after the overload at crack length a_i when the crack is under the overload-induced plastic zone influence (see Fig. 6), was used for this purpose. This model has quite interesting features [32], and was used with an exponent $\gamma = 1.43$ to simulate the curved crack growth retardation. This exponent was selected by personal experience, and finely adjusted to fit the crack growth data after the accidental overload. It is interesting to note that this simple procedure satisfactorily reproduced the whole crack growth behavior observed during the test of the holed SEN, as it is also shown in Fig. 13.

6.2. Compact tension specimens

Four modified CT specimens have been designed and tested. Each one had a 7 mm diameter hole positioned at a slightly different horizontal distance A and vertical distance B from the notch root, as shown in Fig. 8. This odd configuration was chosen because two non-trivial and unexpected crack growth behaviors had been predicted by the FE modeling of the holed CT specimens, depending on the hole position.

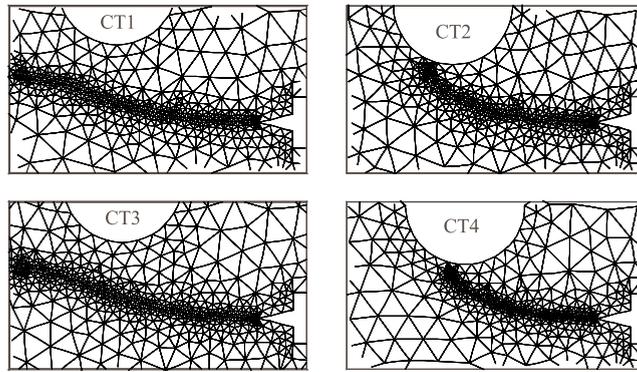


Fig. 14. FE meshes automatically generated for the modified CT specimens.

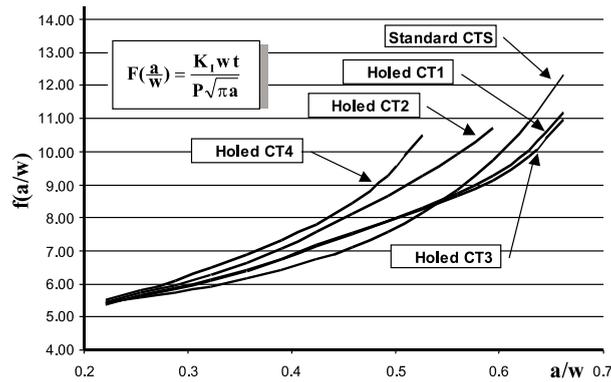


Fig. 15. Calculated $f(a/w)$ expressions for the standard and for the modified CT specimens.

The predictions indicated that the fatigue crack was always attracted by the hole, but it could either curve its path and grow toward the hole or just be deflected by the hole and continue to propagate after missing it. Fig. 14 illustrates the prediction of these two quite different curved crack paths. The initial meshes in the FE models have about 1300 elements and 2300 nodes, and the final ones after the simulated crack propagation have about 2200 elements and 5500 nodes. The computed K_I values are presented and compared to the standard CTS values in Fig. 15.

To test the accuracy of the FE modeling, the transition point between the “sink in the hole” and the “miss the hole” crack growth behaviors was identified. Then, two borderline specimens were dimensioned: one with the hole just half a millimeter below that point and the other with the hole half a millimeter above it. Due to machining tolerances, the actual difference between the vertical position of the holes in specimens CT1 and CT2 turned out to be 1.2 mm instead. These specimens were then remodeled to predict the actual crack path. The measured and the predicted crack paths are compared in Fig. 16.

These results were so encouraging that two other specimens, CT3 and CT4, were built to recheck the experiments. This time, the vertical distance between their holes turned out to be 1.4 mm, and they were also FE remodeled to account for this deviation. However, the crack paths in these specimens were not as well predicted as they were in CT1 and CT2, as is also shown in Fig. 16. But, in fact, the predicted paths

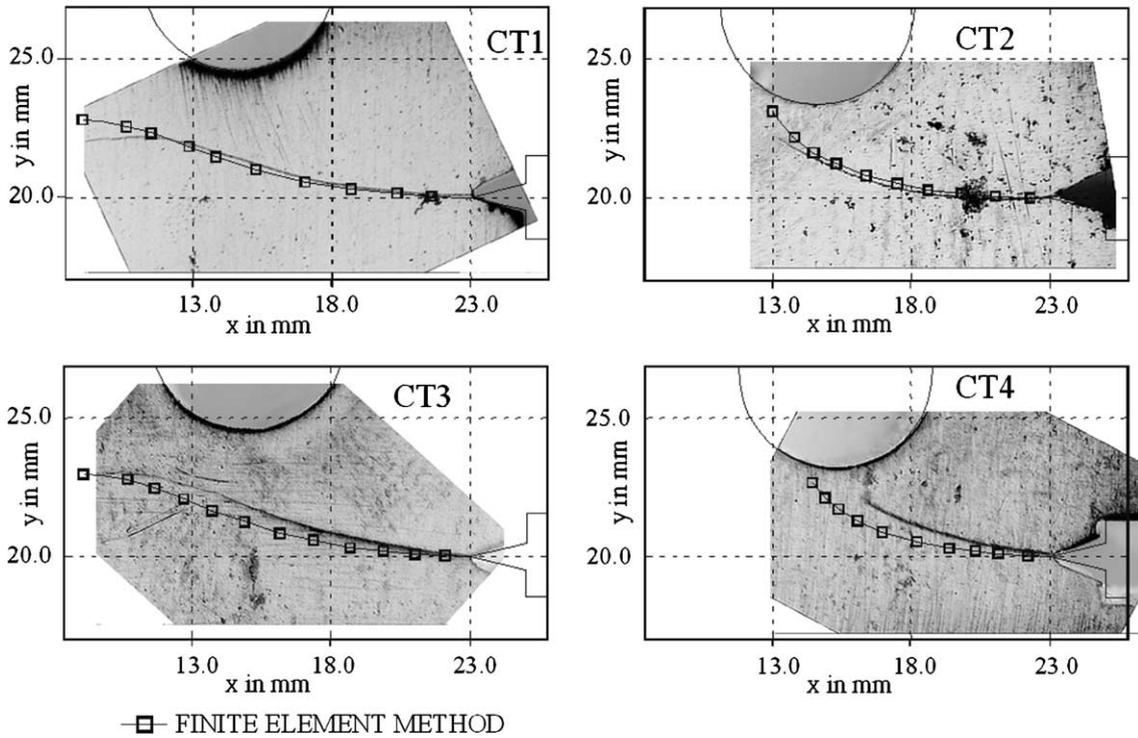


Fig. 16. Predicted and measured crack paths for the modified CT specimens.

were in between the measured ones in the two faces of the specimens, which was an indication that an unwanted transversal moment had also loaded them. Indeed, frictional problems have been found in the universal joint of the load train, which had to be substituted after those tests. However, since the average path between the two faces showed a good match with the FE predictions, it confirmed the predicted “sink in the hole” and the “miss the hole” crack growth behaviors. The predicted and measured fatigue lives are shown in Figs. 17–20, and they also show a quite good match.

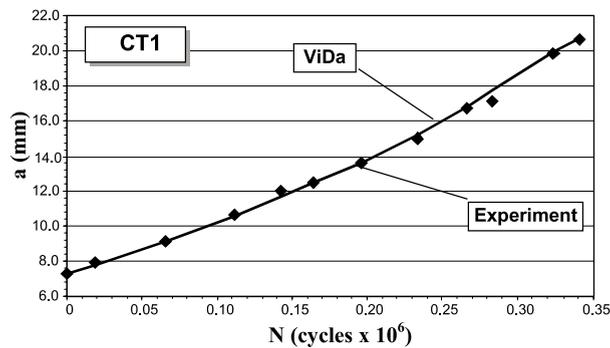


Fig. 17. Predicted and measured fatigue life for CT1.

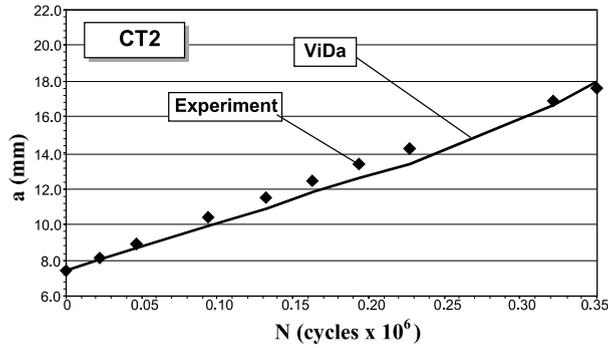


Fig. 18. Predicted and measured fatigue life for CT2.

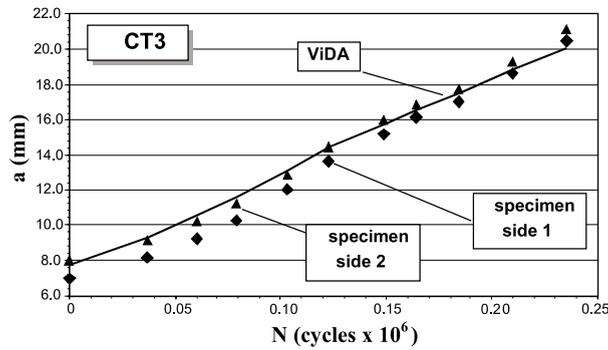


Fig. 19. Predicted and measured fatigue life for CT3.

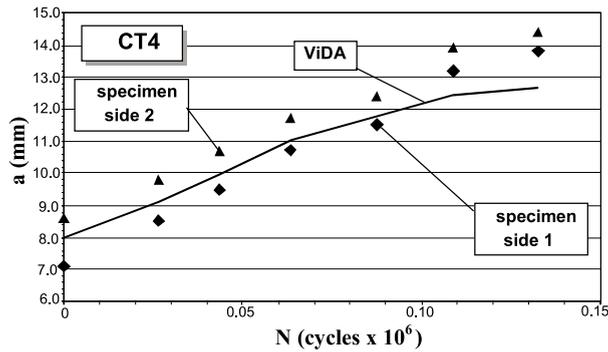


Fig. 20. Predicted and measured fatigue life for CT4.

7. Conclusions

A two-phase methodology was presented to predict fatigue crack propagation in generic 2D structures. First, self-adaptive finite elements were used to calculate, by means of three different methods, the fatigue crack path and the stress-intensity factors along the crack length $K_I(a)$ and $K_{II}(a)$, at each propagation step.

The computed $K_I(a)$ was then used to calculate the propagation fatigue life by the local approach, using the cycle-by-cycle integration method considering overload-induced crack retardation effects.

Two complementary software packages have been developed to implement this methodology. The first is an interactive graphical program for simulating 2D fracture processes based on a FE adaptive mesh-generation strategy. The second is a general-purpose fatigue design software developed to predict both initiation and propagation fatigue lives under variable loading by all classical design methods. Particularly, its crack propagation module accepts any stress-intensity factor expression, including the ones generated by the FE software.

Experimental results showed that the presented methodology and its software implementation could effectively and economically predict crack propagation paths and fatigue lives of arbitrary 2D structural components.

References

- [1] Shih CF, de Lorenzi HG, German MD. Crack extension modeling with singular quadratic isoparametric elements. *Int J Fract* 1976;12:647–51.
- [2] Raju IS. Calculation of strain-energy release rates with higher order and singular finite elements. *Engng Fract Mech* 1987;28:251–74.
- [3] Rybicki EF, Kanninen MF. A finite element calculation of stress-intensity factors by a modified crack closure integral. *Engng Fract Mech* 1977;9:931–8.
- [4] Dodds Jr RH, Vargas PM. Numerical evaluation of domain and contour integrals for nonlinear fracture mechanics. Report, Department of Civil Engineering, University of Illinois, Urbana-Champaign, 1988.
- [5] Nikishkov GP, Atluri SN. Calculation of fracture mechanics parameters for an arbitrary three-dimensional crack by the equivalent domain integral method. *Int J Numer Meth Engng* 1987;24:1801–21.
- [6] Bittencourt TN, Wawrzynek PA, Ingraffea AR, Sousa JLA. Quasi-automatic simulation of crack propagation for 2D LEFM problems. *Engng Fract Mech* 1996;55:321–34.
- [7] Miranda ACO, Cavalcante Neto JB, Martha LF. An algorithm for two-dimensional mesh generation for arbitrary regions with cracks. In: Stolfi J, Tozzi C, editors. *Proceedings of XII Brazilian Symposium on Computer Graphics, Image Processing and Vision IEEE Computer Society Order Number PRO0481, ISBN 0-7695-0481-7*. 1999. p. 29–38.
- [8] Erdogan F, Sih GC. On the crack extension in plates under plane loading and transverse shear. *J Basic Engng* 1963;85: 519–27.
- [9] Hussain MA, Pu SU, Underwood J. Strain energy release rate for a crack under combined mode I and II. *ASTM STP* 1974;560: 2–28.
- [10] Sih GC. Strain-energy-density factor applied to mixed mode crack problems. *Int J Fract Mech* 1974;10:305–21.
- [11] Araújo TDP, Cavalcante Neto JB, Carvalho MTM, Bittencourt TN, Martha LF. Adaptive simulation of fracture processes based on spatial enumeration techniques. *Int J Rock Mech Mining Sci* 1997;34:551.
- [12] Carvalho CVA, Araújo TDP, Cavalcante Neto JB, Martha LF, Bittencourt TN. Automatic fatigue crack propagation using a self-adaptive strategy. *PACAM VI—Sixth Pan-American Congress of Applied Mechanics, Rio de Janeiro 1999;6:377–80*.
- [13] Wawrzynek PA, Ingraffea AR. Interactive finite element analysis of fracture processes: an integrated approach. *Theor Appl Fract Mech* 1987;8:137–50.
- [14] Carpinteri A, Valente S, Ferrara G, Imperato L. Experimental and numerical fracture modelling of a gravity dam. In: *Proceedings of the First International Conference on Fracture Mechanics of Concrete Structures (FRAMCOS I), Colorado, USA. 1992. p. 1–5*.
- [15] Cavalcante Neto JB, Wawrzynek PA, Carvalho MTM, Martha LF, Ingraffea AR. An algorithm for three-dimensional mesh generation for arbitrary regions with cracks. *Engng Comput* 2001;17(1):75–91.
- [16] Chan CT, Anastasiou K. An automatic tetrahedral mesh generation scheme by the advancing front method. *Commun Numer Meth Engng* 1997;13:33–46.
- [17] Jin H, Tanner RI. Generation of unstructured tetrahedral meshes by advancing front technique. *Int J Numer Meth Engng* 1993;36:1805–23.
- [18] Lo SH. A new mesh generation scheme for arbitrary planar domains. *Int J Numer Meth Engng* 1985;21:1403–26.
- [19] Lohner R, Parikh P. Generation of three-dimensional unstructured grids by the advancing-front method. *Int J Numer Meth Fluid* 1988;8:1135–49.
- [20] Moller P, Hansbo P. On advancing front mesh generation in three dimensions. *Int J Numer Meth Engng* 1995;38:3551–69.

- [21] Peraire J, Peiro J, Formaggia L, Morgan K, Zienkiewicz OC. Finite Euler computation in three-dimensions. *Int J Numer Meth Engng* 1988;26:2135–59.
- [22] Rassinoux A. Generation and optimization of tetrahedral meshes by advancing front technique. *Int J Numer Meth Engng* 1998;41:651–74.
- [23] O'Rourke J. *Art gallery theorems and algorithms*. Oxford University Press; 1987.
- [24] Watson DF. Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput J* 1981;24(2):167–72.
- [25] Paulino GH, Menezes IFM, Cavalcante Neto JB, Martha LF. A methodology for adaptive finite element analysis: towards an integrated computational environment. *Comput Mech* 1999;23:361–88.
- [26] MSC.Fatigue 9.0 User's Guide. MSC Software Corporation, 2000.
- [27] Forman RG, Shivakumar V, Mettu SR, Newman Jr JC. *Fatigue crack growth computer program NASGRO version 3.0. Reference Manual*, NASA JSC-22267B, 2000.
- [28] Castro JTP, Meggiolaro MA. Some comments on the sN method automation for fatigue dimensioning under complex loading. *Brazilian J Mech Sci* 1999;21:294–312 [in Portuguese].
- [29] Meggiolaro MA, Castro JTP. ViDa 98—a visual damagemeter to automate the fatigue design under complex loading. *Brazilian J Mech Sci* 1998;20:666–85 [in Portuguese].
- [30] Suresh S. *Fatigue of materials*. Cambridge: Cambridge University Press; 1998.
- [31] Broek D. *The practical use of fracture mechanics*. Kluwer; 1988.
- [32] Meggiolaro MA, Castro JTP. Comparison of load interaction models in fatigue crack propagation. In: *Proceedings of the XVI Brazilian Congress in Mechanical Engineering (COBEM), ABCM*, vol. 12. 2001. p. 247–56.
- [33] Elber W. The Significance of Fatigue Crack Closure. *ASTM STP* 1971;486:230–42.
- [34] Miranda ACO, Meggiolaro MA, Castro JTP, Martha LF, Bittencourt TN. Fatigue crack propagation under complex loading in arbitrary 2D geometries. In: *Braun AA, McKeighan PC, Lohr RD, editors. Applications of Automation Technology in Fatigue and Fracture Testing and Analysis*, vol. 4. *ASTM STP* 1411; 2002. p. 120–46.