Numerical implementation of a multiaxial cyclic plasticity model for the Local Strain Method in low cycle fatigue

C. Madrigal, V. Chaves, A. Navarro
University of Seville, Dpto. Ing. Mecánica y Fabricación, Escuela Técnica Superior de Ingeniería.
Avda. Camino de los Descubrimientos, s/n. 41092. Seville.
cmadrigal@us.es

ABSTRACT. Very often computations on structural elements or machine components subjected to variable loading require using an advanced finite element model. This paper reports the numerical implementation of a model for multiaxial cyclic elasticplastic behaviour developed to extend the tools of the local deformation method under fatigue to multiaxial conditions. A basic computer code for axialtorsional loads was developed with the commercial software Matlab and a more sophisticated code based on the finite element model for general multiaxial loads was developed as a UMAT subroutine in Abaqus. Stress integration was introduced in the two usual forms: implicitly and explicitly. A comparison of the results obtained with the implicit and explicit formulations revealed that, under certain loading conditions, the outcome of the process depends on the particular integration scheme used.

KEYWORDS. Cyclic plasticity; Numerical implementation; Multiaxial fatigue; UMAT user subroutine.

INTRODUCTION

The cyclic plastic behaviour of some materials can be defined via so-called "cyclic stress-strain curves", which are widely used in fatigue studies to introduce steady-state cyclic behaviour in computations of fatigue life at small numbers of cycles in the Local Strain Method [1–3]. Using cyclic stress-strain curves in combination with hysteresis cycles, Neuber's rule, memory rule and $\varepsilon - N$ curves allows one to calculate local stresses and strains at the edge of a notch and lives under uniaxial loading. This is in fact the procedure of choice for a number of sectors in the automobile, aeronautical and aerospatial industries.

In previous work [4–8], we developed a plasticity model to simulate the behaviour of materials under multiaxial loads from cyclic stress-strain curves obtained in uniaxial loading tests with a view to extending the applicability of the Local Strain Method to multiaxial loading. This computational procedure for fatigue life is included in some commercial software packages. In fact, any custom model to be used for this purpose should run in at least one. Also, validating a mathematical model for fatigue life entails performing numerical simulations for comparison with experimental results in order to define the scope of the model. For these reasons, in this work we implemented the proposed model in two different commercial software packages, namely: Matlab, which is useful for computations at specific points under combined tensile–torsional loads, and Abaqus/Standard, which was used to produce a UMAT subroutine for whole elements under general multiaxial loads.

MODEL

The stress tensor is represented by a point in the stress space. Applying a load causes the point to move to another location in the space. The proposed model assumes that the plastic strain caused by the applied load depends on the distance between the starting and ending point in the stress space. If a load-free initial state (i.e., the space origin) is assumed, then application of a load at a given point will result in the point being displaced and in the distance from the origin dictating the beginning of plastic behaviour. Thus, the yield criterion establishes that plastic behaviour starts when the distance Q reaches a value k, which is a property of the material dependent on its extent of cumulative hardening. Q is calculated from the following equation:

$$Q = |\boldsymbol{\sigma}| = \sqrt{g_{ij}\sigma^i\sigma^j} = k \qquad (1)$$

where g_{ij} are the components of the fundamental or metric tensor $\boldsymbol{G}$ in the stress space. This formula contains Einstein's summation convention and can be rewritten in the form of a pseudo-vector with the following components:

$$\begin{array}{lllll} 1-\sigma_x & 2-\sigma_y & 3-\sigma_z & 4-\tau_{xy} & 5-\tau_{yx} \\ 6-\tau_{yz} & 7-\tau_{zy} & 8-\tau_{zx} & 9-\tau_{xz} & \end{array} \qquad (2)$$

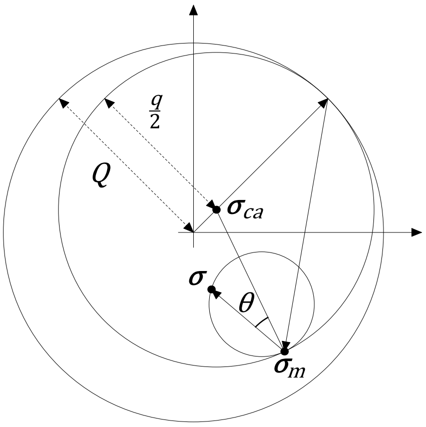

Figure 1: Successive unloading process.

Applying a load to a load-free specimen causes the point representing the stress state at each time to move from the origin and result in an elastic deformation. The distance from the origin increases with increasing loading until the point reaches the yield surface, which is a hypersphere centred in the origin. Under sustained loading, the hypersphere increases in size but continues to contain the load point. If the load is removed, however, the point approaches the origin and the previous equations are inapplicable as the distance ceases to be measured with respect to the origin; also, a new hypersphere forms tangentially to the previous one at the return point $\boldsymbol{\sigma}_m$ (see Fig. 1), which illustrates the situation with two successive unloading events. The unloading distance is defined as the diameter q of the new inner hypersphere, which is calculated from

$$q = \frac{|\boldsymbol{\sigma}-\boldsymbol{\sigma}_m|}{\cos\theta} = \frac{(\sigma_i-\sigma_{mi})(\sigma^i-\sigma^{mi})}{(\sigma_{cai}-\sigma_{mi})(\sigma^{cai}-\sigma^{mi})}\frac{q_0}{2} \qquad (3)$$

where $\boldsymbol{\sigma}_m$ is the return point, $\boldsymbol{\sigma}_{ca}$ the centre of the previous hypersphere - of diameter q_0 - and θ the angle between the $\boldsymbol{\sigma}-\boldsymbol{\sigma}_m$ segment and the line joining $\boldsymbol{\sigma}_m$ with $\boldsymbol{\sigma}_{ca}$.

The amount of plastic strain resulting from loading or unloading is given by the following flow rule:

$$d\varepsilon_i^p = \xi n_i n_j d\sigma^j \qquad (4)$$

where ξ is equal to $\Phi(Q)$ under loading conditions and $\phi(q)/\cos^2\theta$ under unloading conditions. $\Phi(Q)$ and $\phi(q)$ are the *hardening modulus* of the material with and without loading, respectively, and can be calculated numerically or empirically from the cyclic stress-strain curve of the material under uniaxial loading; and $\boldsymbol{n}$ is the vector normal to the yield surface, which is defined by its gradient, i.e., by the gradient of the distance function in Eq. (1) under loads

$$n_i = \frac{\partial Q}{\partial \sigma^i} = \frac{1}{Q} g_{ij}\sigma^j = \frac{\sigma_i}{Q} \tag{5}$$

and Eq. (3) in their absence

$$n_i = \frac{2}{q}\left(\sigma_i - \sigma_{mi}\right) - \frac{2}{q_0}\left(\sigma_{cai} - \sigma_{mi}\right) \tag{6}$$

The stress corresponding to a given deformation increment can be calculated by expanding Eq. (4) with the elastic strain as defined in Hooke's law and solving the resulting expression as follows:

$$d\sigma^j = \left(H_{ij} + \xi n_i n_j\right)^{-1} d\varepsilon_i \tag{7}$$

where H_{ij} denotes the components of the (9×9) Hooke matrix.

A more detailed description of the model can be found elsewhere [4-8].

Matrix notation

As shown in the previous section, the proposed model was defined in 9 dimensions because the stress tensor had 9 components. However, the symmetry of the tensor made using a reduced number of dimensions (6) more practical. This required rewriting the previous equations in matrix form and using an ancillary matrix $\boldsymbol{S}$ to provide for duplicate tangential terms:

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

With this definition, Eq. (1) is written in matrix form as

$$Q = |\boldsymbol{\sigma}| = \sqrt{\boldsymbol{\sigma}^{\boldsymbol{T}}\boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{S}\boldsymbol{\sigma}} \tag{8}$$

and its differential

$$dQ = \nabla \mathrm{Q}\cdot \mathrm{d}\boldsymbol{\sigma} = \boldsymbol{Sn}\cdot\boldsymbol{\sigma} \tag{9}$$

Under unloading conditions, diameter q was expressed as

$$q = \frac{q_0}{2}\frac{\left(\boldsymbol{\sigma}-\boldsymbol{\sigma}_m\right)^{\boldsymbol{T}}\boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{GS}\left(\boldsymbol{\sigma}-\boldsymbol{\sigma}_m\right)}{\left(\boldsymbol{\sigma}_{ca}-\boldsymbol{\sigma}_m\right)^{\boldsymbol{T}}\boldsymbol{S}^{\boldsymbol{T}}\boldsymbol{GS}\left(\boldsymbol{\sigma}-\boldsymbol{\sigma}_m\right)} \tag{10}$$

or, in differential form, as

$$dq = \nabla \mathrm{q}\cdot\mathrm{d}\boldsymbol{\sigma} = \frac{\boldsymbol{Sn}\cdot\boldsymbol{\sigma}}{\cos^2\theta} \tag{11}$$

The flow rule took the form

$$d\boldsymbol{\varepsilon}^{p} = \xi\left(\boldsymbol{Sn}\otimes\boldsymbol{Sn}\right)d\boldsymbol{\sigma} \qquad (12)$$

where ξ is equal to $\Phi\left(Q\right)$ with loading and $\phi\left(q\right)/\cos^{2}\theta$ in the absence of loads. The normal vector was formulated as

$$\boldsymbol{n} = \frac{\boldsymbol{GS\sigma}}{Q} \qquad (13)$$

with loads and as

$$\boldsymbol{n} = \frac{2}{q}\boldsymbol{GS}\left(\boldsymbol{\sigma}-\boldsymbol{\sigma}_{m}\right) - \frac{2}{q_{0}}\boldsymbol{GS}\left(\boldsymbol{\sigma}_{ca}-\boldsymbol{\sigma}_{m}\right) \qquad (14)$$

without them. Combining the elastic and plastic terms, and inversion, yielded the stress increment:

$$d\boldsymbol{\sigma} = \left(\boldsymbol{H}+\xi\boldsymbol{Sn}\otimes\boldsymbol{Sn}\right)^{-1}d\boldsymbol{\varepsilon} \qquad (15)$$

where $\boldsymbol{H}$ is a (6×6) Hooke matrix.

NUMERICAL IMPLEMENTATION

The previous equations were implemented numerically in two different commercial software packages. Thus, Matlab was used to implement the particular equations for combined axial–torsional loads, which accurately reflect the behaviour of a material under multiaxial loads and considerably simplify the model equations. This enables easy numerical integration by effect of the vector space being reduced to two dimensions. Also, most experimental results at low cycle fatigue under multiaxial loads are obtained by subjecting hollow tubes to combined tensile–torsional loads; as a consequence, these simulations are highly suitable for expeditious validation of models.

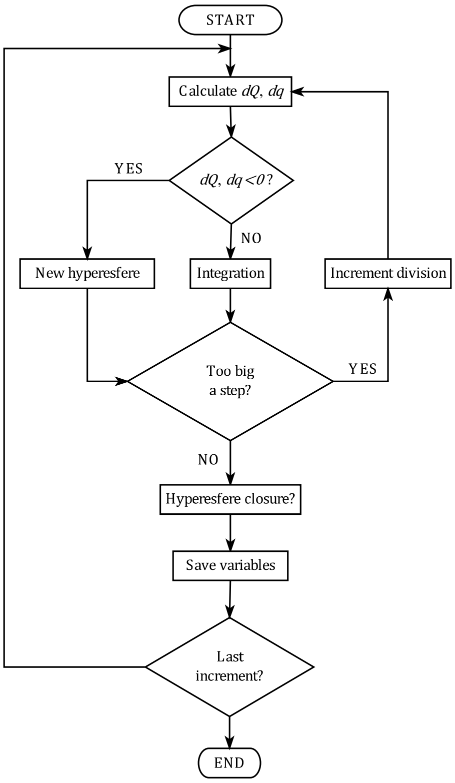

Figure 2: Flow chart for the code.

Calculating the amount of deformation in a whole element entails implementing the proposed equations in finite element software. We used a UMAT subroutine for Abaqus/Standard for this purpose. The scheme of Fig. 2 depicts a very simplified version of the flow chart for the code. On calling the subroutine, the software feeds it with a given total strain increment that is used to calculate the corresponding stress increment. The procedure is identical with and without loads. The total strain increment is used in combination with Eq. (9) and (11) to calculate the distance increment (dQ or dq). If the increment is positive, then the hypersphere continues to grow and Eq. (15) is integrated to calculate the associated stress increment. Under unloading conditions, the hypersphere must be checked not to grow as much as needed to exceed the size of the following outer hypersphere - otherwise, unloading is finished, the inner hypersphere is closed and the process continues with the outer hypersphere in order to record the memory effect. If dQ (or dq) is negative, then unloading starts and an inner hypersphere forms tangentially to the previous one at the load return point.

INTEGRATION

We used two different integration modes. The explicit formulation was more simple and expeditious but conditionally stable; also, the precision was dependent on the size increment chosen —which should therefore be not too large. The implicit formulation was more computationally expensive but afforded a greater integration step and led to faster solutions; also, it was unconditionally stable and ensured that the increment end-point was on the yield surface by effect of all terms and variables being assessed at the end of the increment.

Explicit formulation

The explicit formulation was based on Euler's direct first-order scheme, which involves updating stresses at the end of step $\boldsymbol{\sigma}_{n+1}$ from those at the start (i.e., those calculated at the previous step, $\boldsymbol{\sigma}_n$):

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \mathrm{d}\boldsymbol{\sigma} \tag{16}$$

where the stress increment resulting from the total deformation increment, $d\boldsymbol{\varepsilon}$, was calculated from Eq. (15).

It should be borne in mind that all properties included in the equations are assessed at the start of the increment, which can lead to gross errors if the deformation step is very large.

Implicit formulation

One of the most widely used methods for implicit integration is based on radial return [9–11], which forces the stress point to remain on the yield surface.

Stresses at the end of the increment are calculated with provision for the fact that the deformation increment can be split into an elastic component and a plastic component. Therefore,

$$\boldsymbol{\varepsilon}_{n+1}^{e} = \boldsymbol{\varepsilon}_{n}^{e} + d\boldsymbol{\varepsilon}^{e} = \boldsymbol{\varepsilon}_{n}^{e} + d\boldsymbol{\varepsilon} - d\boldsymbol{\varepsilon}^{p} \tag{17}$$

so thus

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{D}\left(\boldsymbol{\varepsilon}_{n}^{e} + d\boldsymbol{\varepsilon}\right) - \boldsymbol{D}d\boldsymbol{\varepsilon}^{p} = \boldsymbol{\sigma}^{tr} - \boldsymbol{D}d\boldsymbol{\varepsilon}^{p} \tag{18}$$

where $\boldsymbol{D}$ is the elastic constant matrix and $\boldsymbol{\sigma}^{tr}$ the *trial stress* or *elastic estimator*, which coincides with the stress level that would be reached if the strain increment were purely elastic. Since part of the deformation is plastic, the test stress falls off the yield surface and must be combined with the second term (the *plastic corrector*) to bring it back on the surface.

These non-linear equations can be solved by using the Newton–Raphson method, which uses an iterative procedure to calculate stresses at the end of the increment, $\boldsymbol{\sigma}_{n+1}$. In what follows, $\boldsymbol{\sigma}_{n+1}$ is denoted in simplified form by $\boldsymbol{\sigma}$. The final stress at the end of the increment will be the $\boldsymbol{\sigma}$ value reached at the end of the iterative process. Such a value is calculated from a residual stress $\boldsymbol{\Psi}$ given by

$$\boldsymbol{\Psi} = \boldsymbol{\sigma} - \boldsymbol{\sigma}^{tr} + \boldsymbol{D}d\boldsymbol{\varepsilon}^{p} = 0 \tag{19}$$

substitution of the plastic strain increment (Eq. (12)) into which yields

$$\boldsymbol{\Psi} = \boldsymbol{\sigma} - \boldsymbol{\sigma}^{tr} + \boldsymbol{D}\xi\left(\boldsymbol{Sn}\otimes\boldsymbol{Sn}\right)\left(\boldsymbol{\sigma} - \boldsymbol{\sigma}_0\right) = 0 \tag{20}$$

where the stress differential term is simply the difference between the values at the start ($\boldsymbol{\sigma}_0$) and end of the increment. The residual stress can be approximated via the Taylor series

$$\boldsymbol{\Psi} + \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{\sigma}} \partial \boldsymbol{\sigma} = 0 \tag{21}$$

which can be solved for the stress increment upon each iteration:

$$\partial \boldsymbol{\sigma} = -\left(\frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{\sigma}} \right)^{-1} \boldsymbol{\Psi} \tag{22}$$

The final stress is updated in each iteration by using the increment in (22). Thus,

$$\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^{k} + \delta \boldsymbol{\sigma} \tag{23}$$

and the updated stress is used to recalculate the residual and update the stress iteratively until the preset tolerance level is reached.

$$\left| \boldsymbol{\Psi} \right| = \left| \boldsymbol{\sigma} - \boldsymbol{\sigma}^{tr} + \boldsymbol{D} d \boldsymbol{\varepsilon}^{p} \right| \leq 10^{-12} \tag{24}$$

Each iteration updates all terms in the equations. As a result, the stress at the end of the increment, $\boldsymbol{\sigma}_{n+1}$, is calculated from the material properties at the corresponding stress level.

VALIDATION OF THE CODE

The proposed code was validated by comparison of its numerical output with experimental values. A number of simulations spanning a wide range of complexity from uniaxial, proportional loads to complex, non-proportional loads were conducted to this end. By way of example, Fig. 3 shows the simulation of a complex loading history reported by Lamba *et al.* [12]. The process, implemented in Abaqus, involved modelling a tube mimicking the straight portion of the gauge cross-section of the specimens used in the experimental tests that was 1.15 cm in outer radius, 0.95 cm in inner radius and 3.6 cm in length. The mesh was constructed from three-dimensional C3D20R solid elements, which allow the presence of a thickness stress gradient. Boundary conditions were applied by linking the upper and lower surface to a reference node each allowing for radial motion. The reference node for the lower surface was fixed whereas that for the upper surface was allowed to move vertically and rotate about the same axis in order to facilitate recording of the whole loading history. Therefore, the boundary conditions involved node displacement rather than deformation.

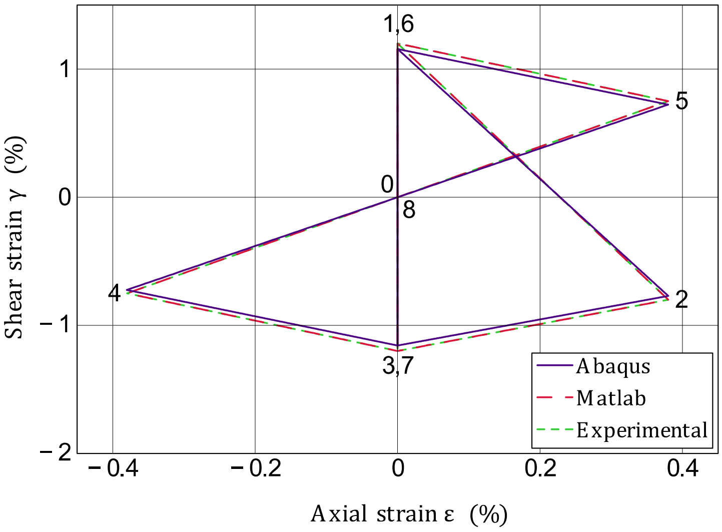

Figure 3: Loading history imposed in the experiments and simulations.

Fig. 3 shows the loading history together with the deformations imposed in the tests and the Matlab simulation. As can be seen, the amount of deformation recorded by the Abaqus simulation was not identical with that applied in the tests. This was a result of the boundary conditions in Abaqus being imposed via displacements rather than directly as deformations. The axial displacement to be applied was calculated as the product of the gauge length by the desired axial strain. The shear strain was calculated from the relation between the rotation angle φ and the cross-section of radius (R) and gauge length (L): $\varphi = \gamma L / R$. The resulting estimates led to axial strains nearly identical with the experimentally determined values and shear strains slightly lower than their experimental counterparts.

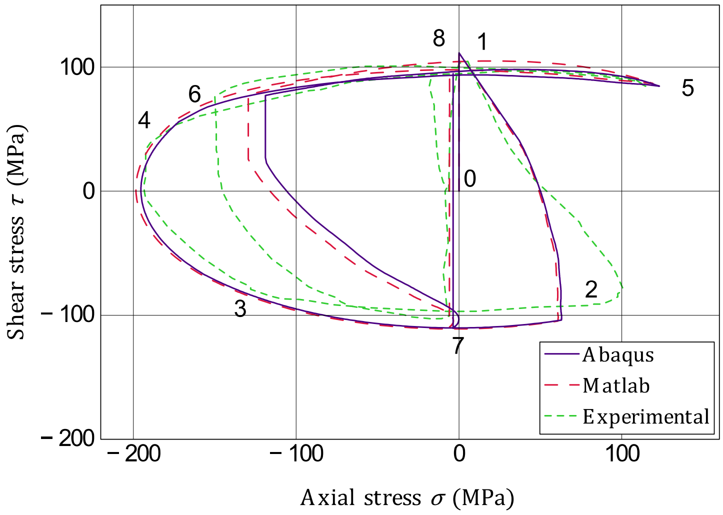

Figure 4: Comparison of the experimental and numerical results for the loading history shown in Fig. 3.

The Abaqus and Matlab predictions are compared with the experimental results of Lamba et al. [1] in Fig. 4. As can be seen, both numerical simulations led to very similar results that were also similar to the experimental values. The two curves differed in the torsional branch by effect of slight differences in their imposed shear deformation. The difference could be lessened by using a finer mesh containing several elements across its thickness in order to more accurately simulate its deformation gradient.

INFLUENCE OF THE INTEGRATION MODE

Once the required equations were implemented and the resulting code was verified, the influence of the particular integration mode used on the numerical results was examined in various types of tests. With uniaxial and proportional loads, the integration mode had no effect on the results. Fig. 5 illustrates a simulated test involving proportional loads and both integration modes. The graph only shows the torsional loops. As can be seen, the curves were virtually indistinguishable.

On the other hand, the outcome of the tests with non-proportional loads was strongly affected by the integration mode used. Fig. 6 shows its influence on the results of a 90° out-of-phase loading test. Thus, explicit integration failed to predict the empirically observed levelling of stresses. By contrast, implicit integration of the same load history led to a stable stress orbit. This was a result of the out-of-phase test involving a continuous neutral load once stresses peaked. Explicit integration calculated the increment from the values of the variables at the start. With a neutral load, this caused the loading point in the stress space to move tangentially to the yield surface at the start of the step. With a finite increment, the end point obviously lay off the yield surface, the displacement from each increasing on each successive integration step. Explicit integration should therefore be avoided when the loading history includes a substantial neutral load. Implicit integration corrects the initial estimate by forcing the point to return to the yield surface iteratively before a new step is taken. Therefore, implicit integration is mandatory in simulating out-of-phase loading tests.

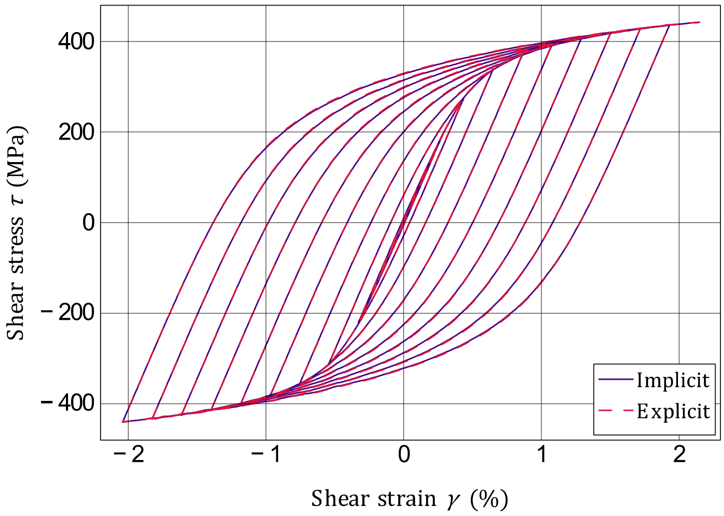

Figure 5: Implicit and explicit simulation of a proportional test. Torsional branch.

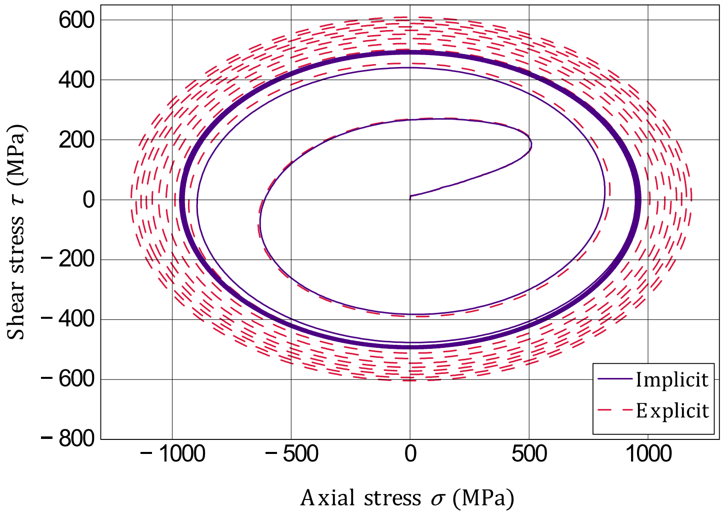

Figure 6: Simulations of a $90°$ out-of-phase test.

Conclusions

Implementing a non-standard plasticity model in commercial software enabled application of constitutive equations to elements under various types of loads. A simplified Matlab scheme allowed the stress status at a preset point under axial and torsional loading to be determined; also, a UMAT subroutine in Abaqus/Standard allowed simulation of components subjected to general multiaxial loads. Stresses in both codes were integrated explicitly and implicitly. The codes were validated by simulating various loading histories and comparing the outcome with experimental values. Based on the results, the UMAT subroutine is accurate and applicable to any type of multiaxial load, whether proportional or otherwise.

The integration mode used was found to have a strong impact on the simulated results for non-proportional loading histories involving substantial segments of neutral loads. Under these conditions, the explicit scheme cannot retain stresses on the yield surface and considerable errors in the simulated values arise as a result. With uniaxial, proportional loads, however, the two integration modes provide essentially identical results.

ACKNOWLEDGEMENTS

The authors would like to thank the Spanish Ministry of Education for its financial support through grant DPI2011-27019.

REFERENCES

[1] Wetzel, R. M., Fatigue Under Complex Loading: Analyses and Experiments, Advances in Engineering 6, Society of Automotive Engineers, (1977).

[2] Dowling, N. E., Mechanical behavior of materials. Engineering Methods for Deformation, Fracture, and Fatigue, Prentice-Hall, (1993).

[3] Bannantine, J. A., Comer, J. J., Handrock, J., Fundamentals of Metals Fatigue Analysis, Prentice-Hall, Inc., (1990).

[4] Navarro, A., Brown, M. W., A constitutive model for elastic-plastic deformation under cycling multiaxial straining, Fatigue Fract. Eng. Mater. Struct., 20 (1997) 747–758.

[5] Navarro, A., Giráldez, J. M., Vallellano, C., A constitutive model for elastoplastic deformation under variable amplitude multiaxial cyclic loading, Int. J. Fatigue, 27 (2005) 838–846.

[6] Navarro, A., Madrigal, C., Ecuaciones de flujo plástico para el Método de las Deformaciones Locales con carga multiaxial, Anal. Mec. Fract, 26-II (2009) 417–423.

[7] Madrigal, C., Navarro, A., Distancia en el espacio de tensiones y criterios de plastificación, Anal. Mec. Fract, 28-II (2011) 571–576.

[8] Madrigal, C., Chaves, V., Navarro, A., Aplicación de ecuaciones de flujo plástico basadas en la distancia en el espacio de tensiones a ensayos cíclicos en Acero F125, Anal. Mec. Fract, 30-I (2013) 187–192.

[9] Doghri, I., Fully implicit integration and consistent tantent modulus in elasto-plasticity, Int. J. Numer. Meth. Eng., 36 (1993) 3915–3932.

[10] Hopperstad, O. S., Remseth, S., A return mapping algorithm for a class of cyclic plastticity models, Int. J. Numer. Meth. Eng., 38 (1995) 549–564.

[11] Kobayashi, M., Ohno, N., Implementation of cyclic plasticity models based on a general form of kinematic hardening, Int. J. Numer. Meth. Eng., 53, (2002) 2217–2238.

[12] Lamba, H. S., Sidebottom, O. M., Cyclic plasticity for nonproportional paths: Part 1 - Cyclic hardening, erasure of memory, and subsequent strain hardening experiments, J. Eng. Mater. Technol., 100 (1978) 96–103.