

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Simulación numérica de endurecimiento en ensayos biaxiales cíclicos no-proporcionales mediante una teoría de plasticidad basada en la idea de distancia en el espacio de tensiones

Autor: Angel Hidalgo Armayones

Tutor: Alfredo Navarro Robles

Dpto. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

**Simulación numérica de endurecimiento en ensayos
biaxiales cíclicos no-proporcionales mediante una
teoría de plasticidad basada en la idea de distancia
en el espacio de tensiones**

Autor:

Angel Hidalgo Armayones

Tutor:

Alfredo de Jesús Navarro Robles

Catedrático de Universidad

Dpto. de Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Grado: Simulación numérica de endurecimiento en ensayos biaxiales cíclicos no-proporcionales mediante una teoría de plasticidad basada en la idea de distancia en el espacio de tensiones

Autor: Ángel Hidalgo Armayones

Tutor: Alfredo de Jesús Navarro Robles

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Quiero agradecer a todas las personas cercanas a mí lo mucho que me han inspirado para realizar este trabajo, y el apoyo recibido por la parte docente.

Angel Hidalgo Armayones

Sevilla, 2020

En este proyecto se desarrolla un modelo de cálculo numérico de tensiones y deformaciones elástico-plásticas y cíclicas que puede utilizarse para simulaciones de fatiga a bajo número de ciclos con carga multiaxial general. Este modelo se aplica en el presente trabajo al estudio detallado de qué ocurre durante los ensayos biaxiales fuera de fase, es decir, en los cuales la torsión y la tensión forman un ángulo de 90° . Esto produce un efecto llamado endurecimiento fuera de fase. Este tipo de ensayos tienen unas particularidades que se estudiarán a continuación, mediante una teoría de plasticidad que calcula las tensiones como distancias en un espacio métrico definido por un criterio de plastificación. En este caso, dicho espacio vectorial tendrá la métrica del criterio de plastificación de von Mises.

Se establecerán tres modelos de cálculo diferentes para poder establecer unos resultados de ejecución rápida, sin efectos numéricos indeseados, y ante todo válidos. Para desarrollar estos tres modelos de cálculo se han establecido unas hipótesis que veremos a continuación.

Una vez seleccionado el modelo que mejor se comporta numéricamente y acorde a las previsiones e hipótesis descritas en el proyecto, se analizará la relación que existe entre las diferentes variables que afectan al problema estudiado, y qué alternativas podemos establecer cuando no sea viable aplicar el resultado de las simulaciones a los resultados experimentales reales.

Abstract

In this project, we will establish numerical calculation models that can be used for low cycle-fatigue simulations, which will allow us to apply it to different models. In this case we will focus on simulating out-of-phase biaxial tests, in which the torsion and tension vectors have an angle of 90° . This produces an effect called out-of-phase hardening. This kind of test has some specific features that will be studied below, through a theory of plasticity that calculates stresses as distances in a metric space defined by a plasticization criterion. In this case, the vector space will have the von Mises metric.

Three different models will be established to establish fast execution results, without undesirable numerical effects, and above all valid. To develop these three calculation models, some hypotheses have been established that we will see below.

Once we have a model that behaves numerically good and according to the hypotheses described in the project, some relationships of the behaviour of the results will be analysed.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Figuras	xvi
Notación	xix
1 Teoría de plasticidad basada en la distancia	1
2 Consideraciones teóricas del modelo de cálculo	9
2.1 <i>Sistema de ecuaciones diferenciales deformación-tensión</i>	9
2.2 <i>Análisis de estado límite e hipótesis principales</i>	12
3 Parametrización del modelo matemático en MATLAB	15
3.1 <i>Consideraciones generales</i>	15
3.1.1 Aproximación lineal	15
3.1.2 Aproximación en espiral	16
3.1.3 Tramo con modulo de deformaciones constante	17
3.2 <i>Parámetros y coeficientes del material</i>	17
4 Métodos de cálculo utilizados en MATLAB	19
4.1 <i>Primer modelo de cálculo. Hipótesis de estado tensional creciente.</i>	19
4.1.1 Parametrización	19
4.1.2 Resultado de las simulaciones	21
4.1.3 Análisis de los resultados	31
4.2 <i>Segundo modelo de cálculo. Utilización de ecuaciones de descarga</i>	33
4.2.1 Parametrización de modelo de integración paso a paso mediante método Runge-Kutta de cuarto orden	34
4.2.2 Resultados de las simulaciones	36
4.2.3 Análisis de los resultados	46
4.3 <i>Tercer modelo de cálculo. Simplificación de ecuaciones de descarga</i>	47
4.3.1 Parametrización	47
4.3.2 Resultados de las simulaciones	48
4.3.3 Análisis de los resultados	59
5 Análisis de relación entre variables	61
5.1 <i>Evolución del módulo de tensiones estabilizado frente a las deformaciones impuestas</i>	61
5.2 <i>Ajustes gráficos</i>	63
5.3 <i>Predicciones para deformaciones plásticas superiores</i>	64
5.3.1 Consideraciones generales	64
5.3.2 Resultados de las simulaciones	65
5.3.3 Análisis de los resultados	69
5.3.4 Ajuste gráfico	69
6 Conclusiones	71

Referencias	72
Anexo. Código de programación necesario para las simulaciones	73

ÍNDICE DE FIGURAS

Figura 1.1	4
Figura 1.2	5
Figura 1.3	6
Figura 1.4	7
Figura 3.1.1	15
Figura 3.1.2	16
Figura 4.1.2.1.1	21
Figura 4.1.2.1.2	21
Figura 4.1.2.1.3	22
Figura 4.1.2.1.4	22
Figura 4.1.2.1.5	23
Figura 4.1.2.1.7	23
Figura 4.1.2.1.8	24
Figura 4.1.2.2.1	24
Figura 4.1.2.2.2	25
Figura 4.1.2.2.3	25
Figura 4.1.2.2.4	26
Figura 4.1.2.2.5	26
Figura 4.1.2.2.6	27
Figura 4.1.2.2.7	27
Figura 4.1.2.3.1	28
Figura 4.1.2.3.2	28
Figura 4.1.2.3.3	29
Figura 4.1.2.3.4	29
Figura 4.1.2.3.5	30
Figura 4.1.2.3.6	30
Figura 4.1.2.3.7	31
Figura 4.1.3.1	32
Figura 4.1.3.2	32
Figura 4.2.2.1.1	36
Figura 4.2.2.1.2	36
Figura 4.2.2.1.3	37
Figura 4.2.2.1.4	37
Figura 4.2.2.1.5	38
Figura 4.2.2.1.6	38

Figura 4.2.2.1.7	39
Figura 4.2.2.2.1	39
Figura 4.2.2.2.2	40
Figura 4.2.2.2.3	40
Figura 4.2.2.2.4	41
Figura 4.2.2.2.5	41
Figura 4.2.2.2.6	42
Figura 4.2.2.2.7	42
Figura 4.2.2.3.1	43
Figura 4.2.2.3.2	43
Figura 4.2.2.3.3	44
Figura 4.2.2.3.4	44
Figura 4.2.2.3.5	45
Figura 4.2.2.3.6	45
Figura 4.2.2.3.7	46
Figura 4.3.2.1.1	48
Figura 4.3.2.1.2	49
Figura 4.3.2.1.3	49
Figura 4.3.2.1.4	50
Figura 4.3.2.1.5	50
Figura 4.3.2.1.6	51
Figura 4.3.2.1.7	51
Figura 4.3.2.2.1	52
Figura 4.3.2.2.2	52
Figura 4.3.2.2.3	53
Figura 4.3.2.2.4	53
Figura 4.3.2.2.5	54
Figura 4.3.2.2.6	54
Figura 4.3.2.2.7	55
Figura 4.3.2.3.1	55
Figura 4.3.2.3.2	56
Figura 4.3.2.3.3	56
Figura 4.3.2.3.5	57
Figura 4.3.2.3.6	58
Figura 4.3.2.3.7	58
Figura 5.1.1	62
Figura 5.1.2	62
Figura 5.2.1	63
Figura 5.2.2	64

Figura 5.3.2.1	65
Figura 5.3.2.2	66
Figura 5.3.2.3	66
Figura 5.3.2.4	67
Figura 5.3.2.5	67
Figura 5.3.2.6	68
Figura 5.3.2.7	68
Figura 5.3.4	69

σ	Vector tensión según métrica establecida
ϵ	Vector deformación según métrica establecida
σ	Componente normal del vector tensión
τ	Componente tangencial del vector tensión
ϵ	Componente longitudinal del vector deformación
γ	Componente transversal del vector deformación
$ \sigma $	Módulo efectivo de tensiones
$ \epsilon $	Módulo efectivo de deformaciones
$d()$	Incremento diferencial
θ	Ángulo entre dos vectores
W_p	Trabajo plástico
ϵ_p	Vector deformación plástica
ϵ_e	Vector deformación elástica
$H()$	Función de endurecimiento
$\phi(\sigma)$	Módulo de endurecimiento
\mathbf{n}	Normal a la superficie de plastificación
n_1	Componente x de \mathbf{n}
n_2	Componente y de \mathbf{n}
E	Módulo de Young del material
G	Módulo de Elasticidad transversal del material
K	Parámetro Ramberg-Osgood del material
n	Parámetro Ramberg-Osgood del material
σ_k	Punto de inversión de carga activo
C_k	Circunferencia de plastificación activa asociada a σ_k
$\sigma_{c,k}$	Centro de la circunferencia de plastificación C_k
q	Distancia entre punto actual y punto de descarga
t	Parámetro de muestreo para modelización en MATLAB
$tcarga$	Parámetro t para tramo de aumento de deformaciones impuestas
$ \epsilon_{max} $	Parámetro de deformaciones efectivas impuestas en la simulación
$tfin$	Valor final del parámetro t
N_c	Número de ciclos simulados
S_f	Tensión de fallo del material
ϵ_f	Deformación de fallo del material
b	Parámetro curva $\epsilon - N$
a	Parámetro curva $\epsilon - N$

1 TEORÍA DE PLASTICIDAD BASADA EN LA DISTANCIA

Las simulaciones que se realizarán en este proyecto utilizan una teoría de plasticidad basada en la distancia entre puntos representados en el espacio tensional. Esta teoría está basada en los métodos de cálculos de tensiones y deformaciones utilizados mediante la ecuación de Ramberg-Osgood para deformaciones a bajo número de ciclos en ensayos monoaxiales, permite el cálculo de estos valores para el caso multiaxial. Para ello utilizaremos el criterio de plastificación de von Mises, definiendo así la métrica del espacio tensional que se describirá a continuación.

El método descrito a continuación consiste en una teoría de plasticidad cíclica, de forma que se puedan aplicar al modelo multiaxial procedimientos similares a los que se aplican en los problemas monoaxiales mediante el método de las Deformaciones locales. Para ello, se utilizará una representación espacial de tensiones y deformaciones, cuyos ejes se determinarán según el criterio de plastificación de von Mises. Esto permite utilizar el concepto de distancia entre tensiones para poder visualizar de una forma gráfica y sencilla todo el procedimiento.

Esta teoría se desarrolla completamente en las referencias [2], [3] y [4], pero se ha considerado útil expresar sus ideas claves para una mayor comprensión de este proyecto por parte del lector.

Para entender mejor esta teoría, empezaremos desarrollándola conforme al modelo biaxial, para más adelante aplicarla a casos multiaxiales.

Para comenzar debemos definir la métrica en la cual representaremos las tensiones y deformaciones del modelo. Para ello trataremos los vectores tensión y deformación como vectores, los cuales tendrán componentes normales y tangenciales. En el caso de las tensiones, las definiremos mediante el vector σ , que a su vez se compondrá de las componentes σ (tensión normal) y τ (tensión tangencial). El vector de deformaciones a su vez lo definiremos como el vector ϵ , compuesto por las componentes ϵ (deformación longitudinal) y γ (deformación transversal):

$$\sigma = (\sigma, \tau) \tag{1.1}$$

$$\epsilon = (\epsilon, \gamma) \tag{1.2}$$

El criterio de plastificación de von Mises nos permite definir superficies de plastificación, lo cual significa que la plastificación comienza cuando se supera el valor crítico:

$$|\sigma|^2 = \frac{2}{3}\sigma^2 + 2\tau^2 \tag{1.3}$$

Como esta expresión no es la norma euclidiana, podemos decir que la base vectorial no estará formada por vectores ortonormales (perpendiculares y de módulo unidad), si no por vectores que su producto escalar sea proporcional a la expresión anterior. En el espacio vectorial de tensiones, el ángulo entre dos vectores tensión $\sigma_1 = (\sigma_1, \tau_1)$ y $\sigma_2 = (\sigma_2, \tau_2)$ se calcula mediante:

$$\cos \theta = \frac{\sigma_1 \cdot \sigma_2}{|\sigma_1| \cdot |\sigma_2|} \tag{1.4}$$

Siendo el producto escalar $\boldsymbol{\sigma}_1 \cdot \boldsymbol{\sigma}_2$:

$$\boldsymbol{\sigma}_1 \cdot \boldsymbol{\sigma}_2 = \frac{2}{3} \sigma_1 \sigma_2 + 2\tau_1 \tau_2 \quad (1.5)$$

Sin embargo, para las componentes del vector deformación es diferente, ya que debido a que la expresión para calcular el trabajo plástico es:

$$dW_p = \sigma d\varepsilon^p + \tau d\gamma^p \quad (1.6)$$

Para que esta expresión sea así debemos definir para el espacio vectorial de deformaciones la base dual tomando como referencia el espacio tensional. Como los vectores de las bases duales son ortonormales, podemos calcular el módulo efectivo de deformaciones con la siguiente expresión:

$$|\varepsilon_p| = \sqrt{(\varepsilon_p)^2 + \left(\frac{\varepsilon_p}{2}\right)^2 + \left(\frac{\varepsilon_p}{2}\right)^2 + \left(\frac{\gamma_p}{2}\right)^2 + \left(\frac{\gamma_p}{2}\right)^2} = \sqrt{\frac{3}{2}(\varepsilon_p)^2 + \frac{1}{2}(\gamma_p)^2} \quad (1.7)$$

Avanzando en la materia, tendremos en cuenta para este método una hipótesis de endurecimiento isótropo, asumiendo que el radio de la circunferencia de plastificación según von Mises sólo es función del incremento de deformación efectiva:

$$|\sigma| = H\left(\int |d\varepsilon_p|\right) \quad (1.8)$$

Para la mayoría de los metales que se endurecen plásticamente es una función creciente. La expresión inversa a la ecuación anterior viene dada por:

$$\int |d\varepsilon_p| = H^{-1}(|\sigma|) \quad (1.9)$$

Siendo la función H^{-1} la función inversa a H . Derivando obtenemos:

$$|d\varepsilon_p| = \frac{dH^{-1}(|\sigma|)}{d|\sigma|} = \boldsymbol{\phi}(|\sigma|)d|\sigma| \quad (1.10)$$

Siendo $\boldsymbol{\phi}(|\sigma|)$ la función que relaciona el incremento de la deformación plástica efectiva con el incremento de la tensión efectiva. La denominaremos módulo de endurecimiento. Esta ecuación implica que la deformación plástica sólo ocurre cuando aumenta el módulo efectivo de tensiones, por lo que el endurecimiento sólo depende de la distancia del punto de tensión al origen de coordenadas. El incremento de deformación plástica tendrá una dirección normal a la circunferencia $|\boldsymbol{\sigma}| = cte$, según la expresión:

$$d\varepsilon_p = \boldsymbol{\phi}(|\sigma|)d|\sigma|\mathbf{n} \quad (1.11)$$

Siendo \mathbf{n} el vector normal a la circunferencia $|\boldsymbol{\sigma}| = cte$. Esta normal se define derivando según la métrica descrita en las ecuaciones (1.3) y (1.7):

$$n_1 = \frac{\partial|\sigma|}{\partial\sigma} = \frac{2}{3} \frac{\sigma}{|\sigma|} \quad (1.12)$$

$$n_2 = \frac{\partial|\sigma|}{\partial\tau} = 2 \frac{\tau}{|\sigma|} \quad (1.13)$$

Sustituyendo estas expresiones en la ecuación (1.11) y añadiendo las componentes elásticas según la Ley de Hooke, tendremos a nuestra disposición un sistema de ecuaciones diferenciales el cual describe la evaluación de las deformaciones plásticas y elásticas del material introduciendo como parámetro una evolución en las tensiones. Estas ecuaciones sólo se aplican para el caso $d|\sigma| > 0$, es decir, cuando el módulo efectivo de tensiones aumenta. Si $d|\sigma| = 0$, el punto de tensión representado en el diagrama de tensión normal σ y tensión tangencial τ describirá una trayectoria circular, manteniendo la misma distancia al origen de coordenadas. Cuando esto ocurre, podemos afirmar que solo ocurren cambios en las tensiones y en las deformaciones de forma elástica.

Para obtener la función de endurecimiento $\phi(|\sigma|)$ partiremos de la ecuación de Ramberg-Osgood, según la referencia [1]:

$$\varepsilon = \frac{\sigma}{E} + \left(\frac{\sigma}{K}\right)^{1/n} \quad (1.14)$$

En el cual la deformación plástica se define:

$$\varepsilon_P = \left(\frac{\sigma}{K}\right)^{1/n} \quad (1.15)$$

Teniendo en cuenta que para el caso monoaxial:

$$|\sigma| = \sqrt{\frac{2}{3}} \sigma \quad (1.16)$$

$$|\varepsilon_P| = \sqrt{\frac{3}{2}} \varepsilon_P \quad (1.17)$$

Y sustituyendo en la ecuación (1.14), finalmente tenemos:

$$|\varepsilon_P| = \left(\sqrt{\frac{3}{2}}\right)^{\left(\frac{1}{n}+1\right)} \left(\frac{|\sigma|}{K}\right)^{\frac{1}{n}} \quad (1.18)$$

La cual podemos derivar para obtener la expresión de $\phi(|\sigma|)$ según la expresión (1.10)

$$\phi(|\sigma|) = \left(\sqrt{\frac{3}{2}} \right)^{\left(\frac{1}{n}+1\right)} \frac{1}{n} \left(\frac{|\sigma|}{K} \right)^{\frac{1}{n}} \frac{1}{|\sigma|} \quad (1.19)$$

Expresión que sustituida en (1.11) se puede aplicar para el caso multiaxial, siempre que calculemos los módulos $|\sigma|$ y $|\varepsilon_p|$ según las métricas adoptadas previamente.

Recordamos que esto es válido siempre y cuando el módulo efectivo de tensiones aumente, pero ¿qué ocurre si disminuyen? Tenemos que empezar a hablar de puntos de descarga. Los puntos de descarga se pueden entender de forma análoga a los picos de tensiones que se determinan para aplicar el método de conteo de ciclos Rainflow.

Primero representemos un estado tensional cualquiera:

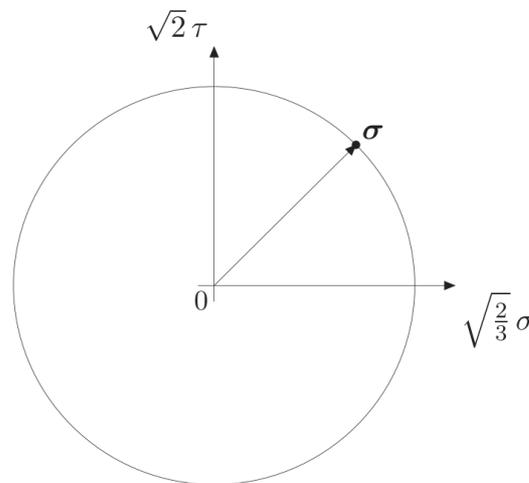


Figura 1.1

Es fácil de entender que si el siguiente paso de σ tiene una distancia al origen mayor al radio de la circunferencia representada en la figura 1.1, decimos que en nuestro ensayo el material está en carga, por lo que pueden aplicarse las ecuaciones descritas en el apartado anterior. Pero si ocurre lo contrario, es decir, la distancia es menor, se crea un punto de descarga. El primer punto de descarga lo definiremos como σ_0 . Este punto se define como el valor cuya distancia al origen es máxima, antes de que se produzca el primer descenso de $|\sigma|$.

Se creará entonces una nueva circunferencia de plastificación según la siguiente imagen:

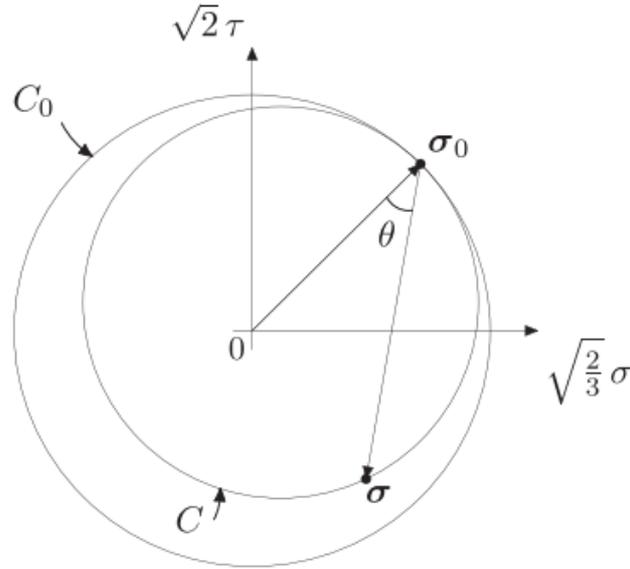


Figura 1.2

Entonces, debemos calcular las distancias respecto a esta nueva circunferencia C de plastificación, cuyo origen, difiere del origen de coordenadas. Para calcular las distancias y poderlas utilizar en el modelo de ecuaciones diferenciales indicado anteriormente, debemos utilizar la siguiente expresión:

$$q = \frac{|\sigma - \sigma_0|}{\cos \theta} \quad (1.20)$$

Donde θ es el ángulo entre los segmentos que conectan σ_0 con el origen de coordenadas y con el punto actual de tensión σ . Si el módulo efectivo de tensiones vuelve a aumentar, $d|\sigma| > 0$, el radio de C crecerá, siendo siempre tangente a C_0 en el punto de descarga σ_0 .

Mediante el producto escalar podemos obtener el ángulo θ :

$$-\sigma_0 * (\sigma - \sigma_0) = |\sigma_0| |\sigma - \sigma_0| \cos \theta \quad (1.21)$$

Con lo cual la ecuación (1.20) nos queda:

$$q = \frac{|\sigma - \sigma_0|}{\cos \theta} = \frac{|\sigma - \sigma_0|^2 |\sigma_0|}{-\sigma_0 * (\sigma - \sigma_0)} \quad (1.22)$$

Simplificando:

$$q = \frac{\frac{2}{3}(\sigma - \sigma_0)^2 + 2(\tau - \tau_0)^2}{\frac{2}{3}(-\sigma_0)(\sigma - \sigma_0) + 2(-\tau_0)(\tau - \tau_0)} \frac{q_0}{2} \quad (1.23)$$

Donde:

$$\frac{q_0}{2} = \sqrt{\frac{2}{3}\sigma_0^2 + 2\tau_0^2} \quad (1.24)$$

Representando con q_0 el módulo efectivo de tensiones del punto de descarga, es decir, el tamaño de la circunferencia C_0 .

Si el vector deformación sigue descendiendo, se crea un nuevo punto de descarga σ_1 y una nueva circunferencia de plastificación C_1 asociada a él:

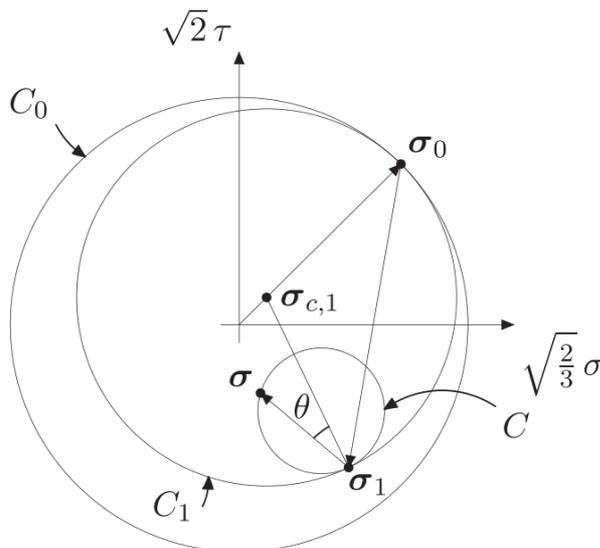


Figura 1.3

Es entonces cuando tenemos que medir las distancias respecto al centro de C_1 , el punto $\sigma_{c,1}$, el cual está conectando el origen y el anterior punto de descarga, σ_0 . Para calcular nuevamente q utilizaremos la siguiente expresión:

$$q = \frac{|\sigma - \sigma_1|}{\cos \theta} = \frac{\frac{2}{3}(\sigma - \sigma_1)^2 + 2(\tau - \tau_1)^2}{\frac{2}{3}(\sigma_{c,1} - \sigma_1)(\sigma - \sigma_1) + 2(\tau_{c,1} - \tau_1)(\tau - \tau_1)} \frac{q_1}{2} \quad (1.25)$$

Donde q_1 es el diámetro de la circunferencia C_1 .

Si esto sucede sucesivamente con más puntos de descarga $\sigma_2, \sigma_3, \dots, \sigma_k$ que su vez crean las circunferencias de plastificación asociadas C_2, C_3, \dots, C_k , podemos establecer una ecuación de q para el caso general:

$$q = \frac{|\sigma - \sigma_k|}{\cos \theta} = \frac{\frac{2}{3}(\sigma - \sigma_k)^2 + 2(\tau - \tau_k)^2}{\frac{2}{3}(\sigma_{c,k} - \sigma_k)(\sigma - \sigma_k) + 2(\tau_{c,k} - \tau_k)(\tau - \tau_k)} \frac{q_k}{2} \quad (1.26)$$

Donde K se refiere al punto de descarga activo y $\sigma_{c,k}$ representa el centro de la circunferencia asociada C_k , cuyo diámetro es q_k .

Para calcular el endurecimiento en estos casos, hay que expresar la función de endurecimiento como:

$$d\varepsilon_p = \phi(q)dq\mathbf{n} \quad (1.27)$$

Siendo las componentes \mathbf{n} definidas desde el centro de la circunferencia de plastificación activa hasta el punto de tensión actual σ , según las expresiones:

$$n_1 = \frac{\partial q}{\partial \sigma} = \frac{2}{3} \frac{2}{q} (\sigma - \sigma_k) - \frac{2}{3} \frac{2}{q_k} (\sigma_{c,k} - \sigma_k) \quad (1.28)$$

$$n_2 = \frac{\partial q}{\partial \tau} = \frac{2}{3} \frac{2}{\tau} (\tau - \tau_k) - \frac{2}{3} \frac{2}{\tau_k} (\tau_{c,k} - \tau_k) \quad (1.29)$$

Si en algún momento el valor de la distancia efectiva entre el centro de la circunferencia de plastificación activa y el último punto de inversión es mayor al radio de dicha circunferencia, este último punto de inversión σ_k y su circunferencia asociada C_k se eliminan, tomándose ahora las referencias desde el anterior punto de inversión. Podemos ver un ejemplo en la siguiente imagen:

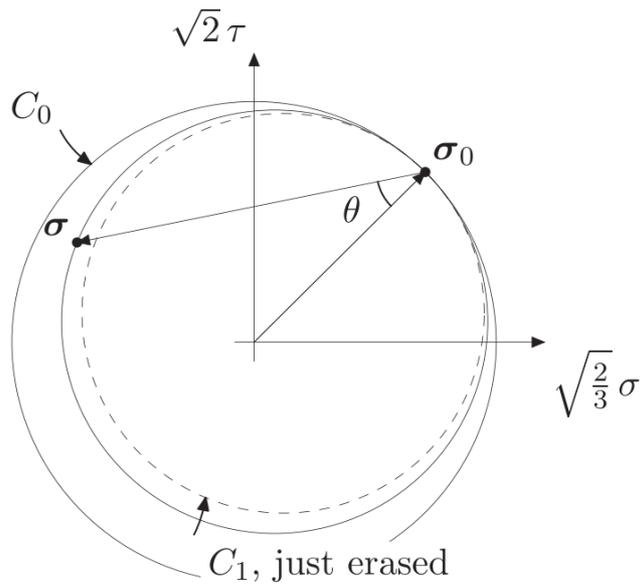


Figura 1.4

Al salirse σ de la circunferencia de plastificación C_1 , esta y su punto de descarga asociado se borran. Si se consigue en algún momento salir de la circunferencia C_0 la cual es la asociada al primer punto de descarga σ_0 , podremos decir que el material está cargando de nuevo y se utilizarán las ecuaciones sin puntos de descarga.

2 CONSIDERACIONES TEÓRICAS DEL MODELO DE CÁLCULO

Llegados a este punto, hemos definido la teoría sobre la que se basarán la totalidad de las simulaciones numéricas. Siguiendo las ecuaciones anteriores, hemos calculado la expresión de la función $\phi(|\sigma|)$ para el caso de un material que sigue el criterio de plastificación de von Mises, y que su deformación plástica podemos ajustarla mediante una curva de Ramberg-Osgood.

2.1 Sistema de ecuaciones diferenciales deformación-tensión

En consecuencia, explicaremos bajo qué circunstancias se aplicarán dichas ecuaciones. En este caso, nuestro estudio se realizará sobre un ensayo de fatiga multiaxial a bajo número de ciclos con las componentes de tracción-torsión, con un desfase de 90° entre ambos esfuerzos. Esto se define comúnmente como endurecimiento fuera de fase. Para el ensayo tracción-torsión, el vectores deformación $\boldsymbol{\varepsilon}$ y el vector tensión $\boldsymbol{\sigma}$ tendrán las siguientes componentes:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon \\ \gamma \end{bmatrix} \quad (2.1.1)$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma \\ \tau \end{bmatrix} \quad (2.1.2)$$

Para definir este ensayo, se impondrán las deformaciones y obtendremos las tensiones siguiendo las ecuaciones expresadas en la teoría de plasticidad descrita anteriormente, según el sistema de ecuaciones diferenciales que usaremos a continuación.

Es de vital importancia indicar que las simulaciones se realizarán imponiendo las deformaciones como parámetros de entrada, obteniendo como soluciones los diferentes puntos del vector tensión.

La primera parte del ensayo será una carga del material en la que incrementaremos el módulo del vector deformación hasta estabilizarlo en un valor. Una vez estabilizado en ese valor, el vector deformación describirá una trayectoria circular $|\varepsilon_{max}| = cte$

siendo este un valor definido previamente. Para que la trayectoria sea circular, debemos representarlas en un sistema de coordenadas de acuerdo a la métrica que hemos utilizado en las ecuaciones, en este caso, la métrica obtenida siguiendo el criterio de plastificación de von Mises, según las ecuaciones (1.3) y (1.7).

Para terminar de definir nuestro problema, desde la ecuación (1.8) hasta la (1.19) aparece el término $\boldsymbol{\varepsilon}_p$, el cual obedece sólo a las deformaciones plásticas. Como bien sabemos, el ensayo se realizará imponiendo deformaciones totales, pues las primeras deformaciones serán elásticas. Para ajustar la expresión, es necesario agregar los términos de deformación elástica $\boldsymbol{\varepsilon}_e$ que vienen dados según la ley de Hooke, la cual para nuestro problema biaxial se manifiesta:

Siendo E y G el módulo de Young y el módulo de cizalladura del material, respectivamente.

$$\boldsymbol{\varepsilon}_e = \begin{bmatrix} \varepsilon_e \\ \gamma_e \end{bmatrix} = \begin{bmatrix} \frac{\sigma}{E} \\ \frac{\tau}{G} \end{bmatrix} \quad (2.1.3)$$

Entonces, definimos el vector $\boldsymbol{\varepsilon}$ como:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_e + \boldsymbol{\varepsilon}_p = \begin{bmatrix} \varepsilon_p + \frac{\sigma}{E} \\ \gamma_p + \frac{\tau}{G} \end{bmatrix} \quad (2.1.4)$$

Despejando $\boldsymbol{\varepsilon}_p$ nos queda:

$$\boldsymbol{\varepsilon}_p = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_e = \begin{bmatrix} \varepsilon - \frac{\sigma}{E} \\ \gamma - \frac{\tau}{G} \end{bmatrix} \quad (2.1.5)$$

Una vez tenemos el vector de deformación plástica definido, calculamos $d\boldsymbol{\varepsilon}_p$ derivando la ecuación (2.1.5):

$$d\boldsymbol{\varepsilon}_p = \begin{bmatrix} d\varepsilon - \frac{1}{E}d\sigma \\ d\gamma - \frac{1}{G}d\tau \end{bmatrix} \quad (2.1.6)$$

Una vez definido el vector de incremento de deformación plástica, podremos utilizar las expresiones de la teoría de plasticidad propuesta anteriormente para expresar las tensiones en función de las deformaciones.

Sustituyendo las ecuaciones (1.8-1.19), en la ecuación (2.1.6), nuestro sistema de ecuaciones toma la siguiente forma:

$$d\boldsymbol{\varepsilon}_p = \begin{bmatrix} d\varepsilon - \frac{1}{E}d\sigma \\ d\gamma - \frac{1}{G}d\tau \end{bmatrix} = \boldsymbol{\phi}(|\sigma|)d|\sigma| \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad (2.1.7)$$

Siendo $d|\sigma|$ dado por como vimos anteriormente:

$$d|\sigma| = \frac{\partial|\sigma|}{\partial\sigma}d\sigma + \frac{\partial|\sigma|}{\partial\tau}d\tau = n_1d\sigma + n_2d\tau \quad (2.1.8)$$

Simplificando la ecuación (2.1.7) e introduciendo el término $d|\sigma|$:

$$\begin{bmatrix} d\varepsilon \\ d\gamma \end{bmatrix} = \boldsymbol{\phi}(|\sigma|)\{n_1 d\sigma + n_2 d\tau\} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{E} d\sigma \\ \frac{1}{G} d\tau \end{bmatrix} \quad (2.1.9)$$

$$\begin{bmatrix} d\varepsilon \\ d\gamma \end{bmatrix} = \begin{bmatrix} \frac{1}{E} + \boldsymbol{\phi}(|\sigma|)n_1^2 & \boldsymbol{\phi}(|\sigma|)n_1n_2 \\ \boldsymbol{\phi}(|\sigma|)n_1n_2 & \frac{1}{G} + \boldsymbol{\phi}(|\sigma|)n_2^2 \end{bmatrix} \begin{bmatrix} d\sigma \\ d\tau \end{bmatrix} \quad (2.1.10)$$

$$\begin{bmatrix} d\varepsilon \\ d\gamma \end{bmatrix} = \begin{bmatrix} \frac{1}{E} + \boldsymbol{\phi}(|\sigma|)n_1^2 & \boldsymbol{\phi}(|\sigma|)n_1n_2 \\ \boldsymbol{\phi}(|\sigma|)n_1n_2 & \frac{1}{G} + \boldsymbol{\phi}(|\sigma|)n_2^2 \end{bmatrix} \begin{bmatrix} d\sigma \\ d\tau \end{bmatrix} \quad (2.1.11)$$

Sustituyendo n_1 y n_2 por sus valores, siguiendo las (1.12) y (1.13):

$$\begin{bmatrix} d\varepsilon \\ d\gamma \end{bmatrix} = \begin{bmatrix} \frac{1}{E} + \boldsymbol{\phi}(|\sigma|)\frac{4}{9}\frac{\sigma^2}{|\sigma|^2} & \boldsymbol{\phi}(|\sigma|)\frac{4}{3}\frac{\sigma\tau}{|\sigma|^2} \\ \boldsymbol{\phi}(|\sigma|)\frac{4}{3}\frac{\sigma\tau}{|\sigma|^2} & \frac{1}{G} + \boldsymbol{\phi}(|\sigma|)\frac{4\tau^2}{|\tau|^2} \end{bmatrix} \begin{bmatrix} d\sigma \\ d\tau \end{bmatrix} \quad (2.1.12)$$

Como en nuestro caso el ensayo se simulará con deformaciones impuestas, tenemos que manipular la expresión para cambiar la disposición actual a aquella que representa las tensiones en función de las deformaciones. Para esto tenemos que invertir la matriz dada en la (2.1.12). Dicha expresión quedaría:

$$\begin{bmatrix} d\sigma \\ d\tau \end{bmatrix} = \frac{1}{\frac{1}{E}\frac{1}{G} + \frac{\boldsymbol{\phi}(|\sigma|)}{|\sigma|^2}\left(\frac{4\tau^2}{E} + \frac{4\sigma^2}{9G}\right)} \begin{bmatrix} \frac{1}{G} + \boldsymbol{\phi}(|\sigma|)\frac{4\tau^2}{|\sigma|^2} & -\boldsymbol{\phi}(|\sigma|)\frac{4}{3}\frac{\sigma\tau}{|\sigma|^2} \\ -\boldsymbol{\phi}(|\sigma|)\frac{4}{3}\frac{\sigma\tau}{|\sigma|^2} & \frac{1}{E} + \boldsymbol{\phi}(|\sigma|)\frac{4}{9}\frac{\sigma^2}{|\sigma|^2} \end{bmatrix} \begin{bmatrix} d\varepsilon \\ d\gamma \end{bmatrix} \quad (2.1.13)$$

La cual es la expresión que debemos integrar para representar las tensiones respecto a las deformaciones. Introduciendo el parámetro t del que dependen las deformaciones, el sistema de ecuaciones diferenciales a integrar es el siguiente:

$$\begin{bmatrix} \frac{d\sigma}{dt} \\ \frac{d\tau}{dt} \end{bmatrix} = \frac{1}{\frac{1}{E}\frac{1}{G} + \frac{\boldsymbol{\phi}(|\sigma|)}{|\sigma|^2}\left(\frac{4\tau^2}{E} + \frac{4\sigma^2}{9G}\right)} \begin{bmatrix} \frac{1}{G} + \boldsymbol{\phi}(|\sigma|)\frac{4\tau^2}{|\sigma|^2} & -\boldsymbol{\phi}(|\sigma|)\frac{4}{3}\frac{\sigma\tau}{|\sigma|^2} \\ -\boldsymbol{\phi}(|\sigma|)\frac{4}{3}\frac{\sigma\tau}{|\sigma|^2} & \frac{1}{E} + \boldsymbol{\phi}(|\sigma|)\frac{4}{9}\frac{\sigma^2}{|\sigma|^2} \end{bmatrix} \begin{bmatrix} \frac{d\varepsilon}{dt} \\ \frac{d\gamma}{dt} \end{bmatrix} \quad (2.1.14)$$

Este parámetro se definirá con mayor detenimiento en el posterior apartado, donde hablaremos del modelo numérico que se creará para realizar las simulaciones.

La expresión de $\phi(|\sigma|)$ viene dada por la ecuación (1.19) la cual expresa el flujo plástico dado por los coeficientes de Ramberg-Osgood.

2.2 Análisis de estado límite e hipótesis principales

Una vez tenemos definido claramente el sistema de ecuaciones que debemos integrar para realizar nuestras simulaciones, es interesante realizar un pequeño análisis de dichas ecuaciones.

Si analizamos las expresiones (1.19) y (2.1.14), podemos tener en cuenta una serie de hipótesis que marcarán la naturaleza de las simulaciones propuestas y el transcurso del presente proyecto. Según la teoría propuesta, al mantener las deformaciones en módulo constante, el material se endurece y plastifica cada vez más, por lo que $|\sigma|$ es cada vez mayor, y con él $\phi(|\sigma|)$ se va incrementando. Al ser n del orden de 0,2 o menor, podemos afirmar que $\phi(|\sigma|)$ crecerá a un ritmo mayor que $|\sigma|^2$, siendo el cociente $\frac{\phi(|\sigma|)}{|\sigma|^2}$ cada vez mayor a medida que aumentan los ciclos a los que está sometido el material.

Es entonces cuando podemos despreciar el término elástico frente al plástico, pues:

$$\frac{1}{E} \frac{1}{G} \ll \frac{\phi(|\sigma|)}{|\sigma|^2}, \quad \frac{1}{G} \ll \phi(|\sigma|); \quad \frac{1}{E} \ll \phi(|\sigma|) \quad (2.2.1)$$

Esta simplificación se puede aplicar cuando $\phi(|\sigma|)$ es suficientemente grande, ya que, al inicio de las simulaciones, en el cual los valores serán menores, no podemos despreciar el término elástico frente al término plástico. Sin embargo, conforme vayan avanzando las simulaciones, es bastante probable que podamos aplicar dicha simplificación.

Simplificando la expresión (2.1.14) según las hipótesis de las ecuaciones (2.2.1):

$$\begin{bmatrix} \frac{d\sigma}{dt} \\ \frac{d\tau}{dt} \end{bmatrix} = \frac{1}{\phi(|\sigma|) \left(\frac{4\tau^2}{E} + \frac{4\sigma^2}{9G} \right)} \begin{bmatrix} \phi(|\sigma|) \frac{4\tau^2}{|\sigma|^2} & -\phi(|\sigma|) \frac{4}{3} \frac{\sigma\tau}{|\sigma|^2} \\ -\phi(|\sigma|) \frac{4}{3} \frac{\sigma\tau}{|\sigma|^2} & \phi(|\sigma|) \frac{4}{9} \frac{\sigma^2}{|\sigma|^2} \end{bmatrix} \begin{bmatrix} \frac{d\varepsilon}{dt} \\ \frac{d\gamma}{dt} \end{bmatrix} \quad (2.2.2)$$

Simplificando:

$$\begin{bmatrix} \frac{d\sigma}{dt} \\ \frac{d\tau}{dt} \end{bmatrix} = \frac{1}{\left(\frac{\tau^2}{E} + \frac{\sigma^2}{9G} \right)} \begin{bmatrix} \tau^2 & -\frac{1}{3}\sigma\tau \\ -\frac{1}{3}\sigma\tau & \frac{1}{9}\sigma^2 \end{bmatrix} \begin{bmatrix} \frac{d\varepsilon}{dt} \\ \frac{d\gamma}{dt} \end{bmatrix} \quad (2.2.3)$$

Una vez hemos conseguido simplificar bastante nuestro sistema de ecuaciones diferenciales, podemos ver que la matriz es singular. Esto nos lleva a un análisis muy interesante de dicha ecuación por componentes. Para ello, compararemos ambas componentes para relacionarlas con el módulo efectivo de tensiones:

$$\frac{d\sigma}{dt} = \frac{1}{\left(\frac{\tau^2}{E} + \frac{\sigma^2}{9G}\right)} \left[\tau^2 \frac{d\varepsilon}{dt} - \frac{1}{3} \sigma \tau \frac{d\gamma}{dt} \right] = \frac{\tau}{\left(\frac{\tau^2}{E} + \frac{\sigma^2}{9G}\right)} \left[\tau \frac{d\varepsilon}{dt} - \frac{1}{3} \sigma \frac{d\gamma}{dt} \right] \quad (2.2.4)$$

$$\frac{d\tau}{dt} = \frac{1}{\left(\frac{\tau^2}{E} + \frac{\sigma^2}{9G}\right)} \left[-\frac{1}{3} \sigma \tau \frac{d\varepsilon}{dt} + \frac{1}{9} \sigma^2 \frac{d\gamma}{dt} \right] = \frac{-\frac{1}{3} \sigma}{\left(\frac{\tau^2}{E} + \frac{\sigma^2}{9G}\right)} \left[\tau \frac{d\varepsilon}{dt} - \frac{1}{3} \sigma \frac{d\gamma}{dt} \right] \quad (2.2.5)$$

Si comparamos estas dos expresiones, podemos llegar a una expresión bastante curiosa, la cual relaciona las hipótesis de simplificación que se han llevado a cabo con el valor del módulo efectivo de tensiones a lo largo del ensayo:

$$\frac{\frac{d\sigma}{dt}}{\frac{d\tau}{dt}} = -\frac{3\tau}{\sigma} \quad (2.2.6)$$

$$\sigma \frac{d\sigma}{dt} + 3\tau \frac{d\tau}{dt} = 0 \quad (2.2.7)$$

$$\sigma \frac{d\sigma}{dt} = -3\tau \frac{d\tau}{dt} \quad (2.2.8)$$

Si calculamos la derivada del módulo de las tensiones según la métrica del espacio tensional, expuesto en la ecuación (1.3):

$$\frac{d|\sigma|}{dt} = \frac{d\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}}{dt} \quad (2.2.9)$$

$$\frac{d|\sigma|}{dt} = \frac{d\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}}{d\sigma} \frac{d\sigma}{dt} + \frac{d\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}}{d\tau} \frac{d\tau}{dt} \quad (2.2.10)$$

$$\frac{d|\sigma|}{dt} = -\frac{1}{2\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}} \frac{4}{3} \sigma \frac{d\sigma}{dt} - \frac{1}{2\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}} 4\tau \frac{d\tau}{dt} \quad (2.2.11)$$

Podemos sustituir en la expresión (2.2.11) la ecuación (2.2.8)

$$\frac{d|\sigma|}{dt} = \frac{1}{2\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}} \frac{4}{3} 3\tau \frac{d\tau}{dt} - \frac{1}{2\sqrt{\frac{2}{3}\sigma^2 + 2\tau^2}} 4\tau \frac{d\tau}{dt} \quad (2.2.12)$$

Por lo tanto:

$$\frac{d|\sigma|}{dt} = 0 \quad (2.2.13)$$

Esto implica que llega un momento en el cual a pesar de que seguimos teniendo ciclos de carga, el módulo de las tensiones se tiene que estabilizar en un valor. En este momento, el vector tensión describiría una trayectoria circular cuyo diámetro es el valor del módulo de sigma $|\sigma|$.

3 PARAMETRIZACIÓN DEL MODELO MATEMÁTICO EN MATLAB

Para determinar si las previsiones que se han determinado en el apartado anterior y las hipótesis utilizadas se cumplen, y si es así, cuándo y en qué valor se estabiliza $|\sigma|$, realizaremos una simulación en MATLAB integrando las ecuaciones diferenciales y representando los resultados gráficamente para analizarlos en detalle.

3.1 Consideraciones generales

Para definir el tramo de aumento del módulo del vector deformación $|\varepsilon|$, podemos imponer diferentes trayectorias. Para este caso de estudio, utilizaremos las dos simplificaciones que hemos considerado más ilustradoras para el análisis de nuestro problema.

3.1.1 Aproximación lineal

Por un lado, podemos afirmar que el módulo de las deformaciones $|\varepsilon|$ se incrementará linealmente, y sólo debido a la componente longitudinal ε . En este caso, en el material el módulo del vector deformación alcanzaría el valor de $|\varepsilon_{max}|$ siguiendo el mismo procedimiento que en el caso monoaxial, produciendo a su vez un incremento lineal de la componente axial σ del vector tensión. Una vez alcanzado el punto $|\varepsilon_{max}|$, el vector deformación empezaría a dibujar una trayectoria circular, mediante la cual empezarían a aparecer deformaciones transversales γ y tensiones tangenciales τ . En la siguiente ilustración podemos verlo con más detalle:

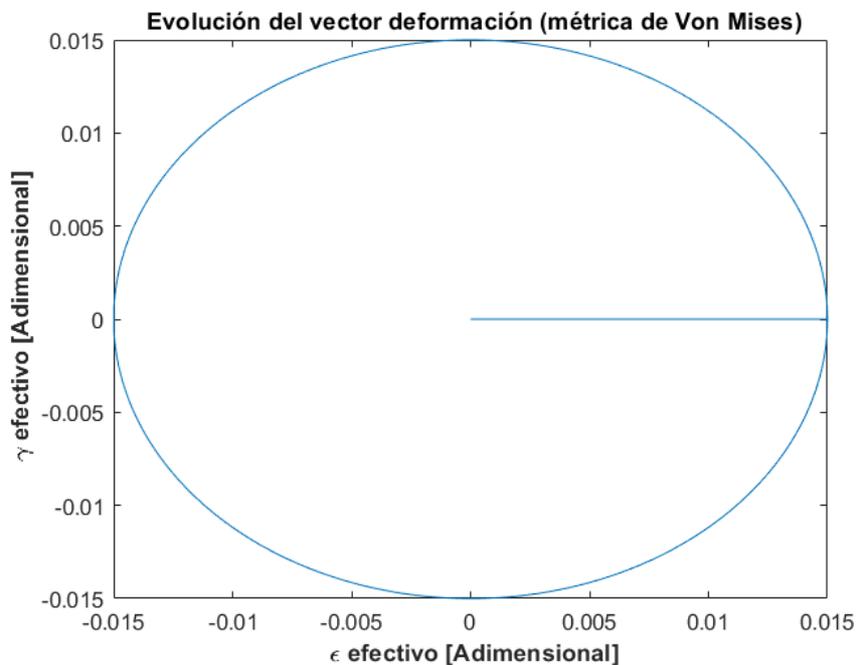


Figura 3.1.1

Las expresiones del vector deformación $\varepsilon(t)$ para este procedimiento de carga vienen dadas por:

$$\varepsilon(t) = \sqrt{\frac{2}{3}} |\varepsilon_{max}| \frac{t}{tcarga} \quad (3.1.1)$$

$$\tau(t) = 0 \quad (3.1.2)$$

Siendo t el parámetro de tiempo que va desde $t = 0$ hasta $tcarga$, instante en el cual el vector deformación ε alcanza el valor $|\varepsilon_{max}|$. Para entender mejor los cálculos que haremos posteriormente, $tcarga$ será múltiplo de 2π .

3.1.2 Aproximación en espiral

Por el otro lado, la carga del material se puede modelar hacia la estabilización en la superficie $|\varepsilon_{max}| = cte$ mediante una aproximación en espiral. La forma más sencilla de modelar este tipo de aproximación es mediante una circunferencia que crece linealmente con el parámetro t . Mediante este método, desde el primer instante el material está experimentando un proceso de carga multiaxial, tanto en tensiones como en deformaciones, a diferencia del caso anterior. Después de alcanzar el valor del módulo del vector de deformaciones $|\varepsilon_{max}|$, proseguiríamos como en el caso anterior, describiendo una circunferencia de radio constante igual a $|\varepsilon_{max}|$. Podemos representarlo gráficamente como:

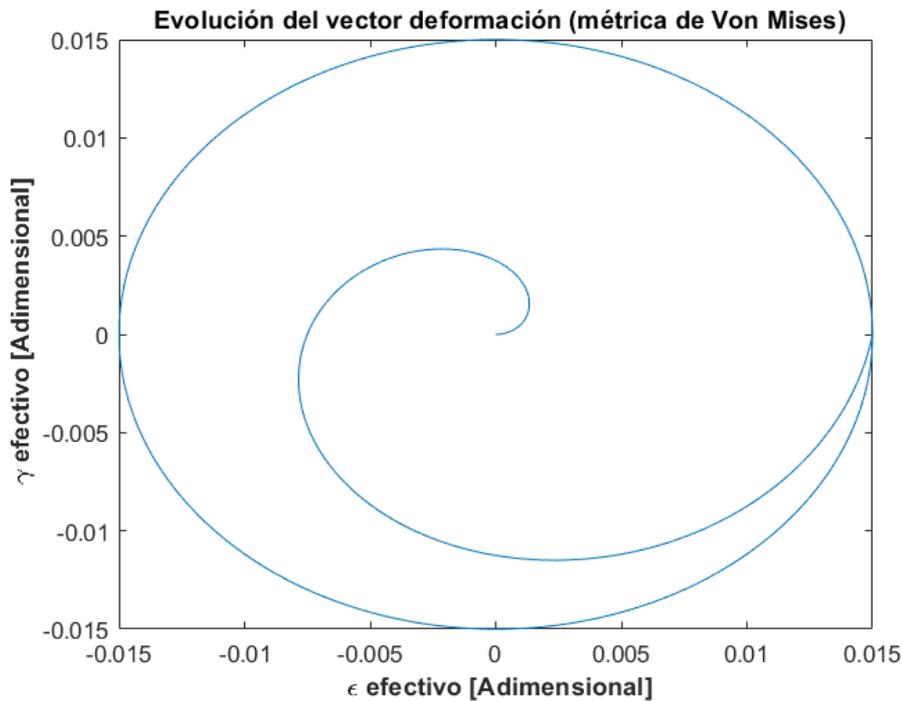


Figura 3.1.2

Las expresiones del vector deformación ε impuesto para este caso particular, serán de la forma:

$$\varepsilon(t) = \sqrt{\frac{2}{3}} |\varepsilon_{max}| \cos(t) \frac{t}{tcarga} \quad (3.1.2.1)$$

$$\tau(t) = \sqrt{2} |\varepsilon_{max}| \sin(t) \frac{t}{tcarga} \quad (3.1.2.2)$$

Los valores constantes que multiplican las variables de las expresiones anteriores vienen dados para que representen una circunferencia en la métrica de nuestro espacio de deformaciones.

Siendo el parámetro t desde $t = 0$ hasta $t = tcarga$ como en el caso anterior. Mirando las expresiones (3.1.2.1) y (3.1.2.2), podemos apreciar en este momento la utilidad de definir $tcarga$ como múltiplo de 2π . Así podemos definir cuantos ciclos tarda el material en alcanzar un vector deformación de módulo $|\varepsilon_{max}|$.

3.1.3 Tramo con modulo de deformaciones constante

Una vez definido el vector deformación $\boldsymbol{\varepsilon}$ para el tramo de aproximación, este pasa a describir una circunferencia de radio constante y valor $|\varepsilon_{max}|$, como vimos en las figuras anteriores.

Para este tramo, que consideramos ya como el endurecimiento cíclico del material, las expresiones del vector deformación $\boldsymbol{\varepsilon}$ son las siguientes:

$$\varepsilon(t) = \sqrt{\frac{2}{3}} |\varepsilon_{max}| \cos(t) \quad (3.1.3.1)$$

$$\tau(t) = \sqrt{2} |\varepsilon_{max}| \sin(t) \quad (3.1.3.2)$$

En este caso el parámetro t empieza desde el valor $t = tcarga$ hasta el valor $t = tfin$. Para determinar con mayor facilidad el número de ciclos que va a durar nuestra simulación, definimos el instante final $tfin$ como $tfin = 2\pi N_c$, siendo N_c el número de ciclos totales que se producen en el ensayo.

Para comparar los mismos resultados, utilizaremos en el transcurso de las simulaciones la aproximación mediante espiral, ya que al aproximarse al segundo tramo de una forma más progresiva, podrá soportar mejor los posibles errores numéricos que un cambio de pendiente tan brusco como ocurre si realizamos el primer tramo de forma lineal.

El problema puede estar mal condicionado, puesto que el valor de entrada son deformaciones (cuyo orden de magnitud es de 0.01) y el valor de salida son tensiones (orden de magnitud 500). Es por esto por lo que debemos utilizar tolerancias de integración pequeñas, aunque esto nos aumente el tiempo de ejecución del código. Se utilizarán tolerancias de integración absolutas y relativas de 10^{-12} .

Para el muestreo de los datos y su representación gráfica, tomaremos una tolerancia $h = 10^{-4}$ para el parámetro t . Este valor permite una resolución gráfica óptima, y a su vez no sobrecarga en exceso la memoria del ordenador que realizará los cálculos.

3.2 Parámetros y coeficientes del material

En esta simulación definiremos los valores de los parámetros siguientes para introducirlos en las ecuaciones diferenciales y expresiones descritas en los apartados anteriores. Estos son los parámetros Ramberg-Osgood, Módulo de Young y coeficiente de Poisson. Utilizaremos los atributos del material Aluminio 7075 T651 según la referencia [5]

Para comprobar la utilidad de los datos utilizados también disponemos de los parámetros de la curva $\varepsilon - N$ asociada, según el método de las deformaciones locales, cuya ecuación es:

$$\varepsilon = \frac{S_f}{E} (2N_c)^b + \varepsilon_f (2N_c)^c \quad (3.2.1)$$

Esta ecuación relaciona el número de ciclos de rotura según una deformación efectiva ε dada, siendo S_f y ε_f , respectivamente, la tensión y deformación de fallo del material en el ensayo de tracción.

Dichos valores, para el Aluminio 7075 T651 vienen dados por [5]:

Parámetros Aluminio 7075 T651	
Parámetros Ramberg-Osgood	
K	852 MPa
n	0,074
Parámetros del ensayo de tracción	
E	71 GPa
G	26,7 GPa
v	0,33
Parámetros de la curva ϵ -N	
S_f	1231 MPa
ϵ_f	0.26
b	-0,122
c	-0,806

4 MÉTODOS DE CÁLCULO UTILIZADOS EN MATLAB

Teniendo en cuenta que estamos ante un problema numérico donde la matriz que define el sistema de ecuaciones diferenciales es posible que sea singular en algunos puntos, se establecerán un total de tres modelos de cálculo. Estos modelos surgirán adaptando al primer modelo las necesidades que encontremos a lo largo de las simulaciones.

4.1 Primer modelo de cálculo. Hipótesis de estado tensional creciente.

Dado que, según las previsiones teóricas, el módulo efectivo de tensiones $|\sigma|$ va creciendo y tiende a estabilizarse en un valor, podemos decir que este valor sólo crece durante la simulación, siendo posible usar sólo las ecuaciones de carga del material, las cuales son bastante más sencillas, lo que simplifica notablemente el problema.

4.1.1 Parametrización

Para evaluar los resultados, haremos varias simulaciones, definiendo diferentes valores de $|\varepsilon_{max}|$ como valor de entrada. Para definir este rango, haremos uso de la ecuación de la curva $\varepsilon - N$, definiendo un rango de ciclos N_C entre 50-250 ciclos, ya que el método de las deformaciones locales y la teoría establecida en la introducción de este proyecto permite predecir el comportamiento del material a bajo número de ciclos (menos de 1000 ciclos). Este rango a su vez nos permitiría estar en el rango medio que abarca desde $N_C = 1$ hasta $N_C = 1000$

Utilizando la ecuación $\varepsilon - N$ mediante el script *curvaen.m* podemos establecer un rango de valores de $|\varepsilon_{max}|$ entre $|\varepsilon_{max}| = 0.011$ y $|\varepsilon_{max}| = 0.017$ según la tabla:

$ \varepsilon_{max} $
0.010
0.011
0.012
0.013
0.014
0.015
0.016
0.017

Para realizar la simulación de este primer modelo (sólo ecuaciones de carga), utilizaremos el script *espiralcl*. Este script se puede utilizar para calcular mediante otros modelos de cálculo más avanzados que veremos en los siguientes apartados, por lo que para este primer modelo ecuación deberemos definir los siguientes parámetros en el script:

- “*ecuacion=1*” para utilizar el primer modelo de cálculo.
- “*esp=1*” para aproximar el primer tramo en espiral, el cual se utilizará durante toda la simulación.
- “*material=1*” para tomar los parámetros del Aluminio 7075 T651
- “ $N_C = 100$ ” para calcular 100 ciclos en cada iteración. Esto se cambiará según los resultados para un mejor muestreo. Es decir, para $|\varepsilon_{max}| = 0.014$ evaluaremos 200 ciclos y para $|\varepsilon_{max}| = 0.017$ 1000 ciclos.
- “ $|\varepsilon_{max}| = VALOR DE TABLA A EVALUAR$ ”

Dentro de dicho script, para integrar el sistema de ecuaciones diferenciales (2.1.14), debemos hacerlo por dos tramos, como explicamos anteriormente. Uno de esos tramos para el aumento de $|\varepsilon|$ en trayectoria espiral y otro para cuando el vector deformación describe una trayectoria circular según la circunferencia $|\varepsilon| = |\varepsilon_{max}|$, parámetro previamente definido. Para ello, disponemos de dos funciones, *sys1.m* y *sys2.m*, (para los tramos 1 y 2, respectivamente), las cuales toman como entrada las componentes del vector deformación según las ecuaciones (3.1.2.1) y (3.1.2.2) para el primer tramo y las ecuaciones (3.1.3.1) y (3.1.3.2.) para el segundo tramo. A su vez, devuelven las componentes σ y τ del vector tensión, a lo largo de cada instante del vector parametrizado t . Estas funciones se integrarán según la función *ode45* que incluye MATLAB.

Al ejecutar el script con dichos parámetros, realizará una serie de representaciones, las cuales son:

- Componente normal de tensión σ frente a componente normal de deformación ε . (Ciclos de histéresis en componente normal).
- Componente transversal de tensión τ frente a componente transversal de deformación γ . (Ciclos de histéresis en componente transversal).
- Componente normal de tensión σ frente a componente transversal de tensión τ . (Evolución del vector tensión σ)
- Componente normal de deformación ε frente a componente transversal de deformación γ . (Evolución del vector tensión ε)
- Norma de tensiones efectivas $|\sigma|$ frente al número de ciclos N_C
- Vector de diferencias de $|\sigma|$ frente al número de ciclos N_C . Esto nos sirve para conocer si ha habido descarga, es decir, $|\sigma|$ ha disminuido. Si esto ocurriera, tendríamos que utilizar las ecuaciones completas (carga y descarga).
- Ángulo de desfase entre los dos vectores.

En primera instancia, nos interesa sobre todo si los resultados van a ser estables numéricamente o no, y entonces analizar los distintos escenarios. En este primer modelo nos centraremos en la curva $N_C - |\sigma|$.

Siguiendo los valores de $|\varepsilon_{max}|$ dados por la tabla 4.1, los resultados gráficos más representativos se indican a continuación. Escogeremos como valores representativos tres valores de la tabla 4.1. Estos valores serán un valor de deformaciones bajas, como $|\varepsilon_{max}|=0.011$; un valor de deformaciones intermedio, el cual será $|\varepsilon_{max}|=0.014$ y un valor de deformaciones elevado, respectivamente al intervalo de deformaciones que hemos escogido, tomado $|\varepsilon_{max}|=0.017$

Los resultados para este primer método de cálculo se muestran a continuación.

4.1.2 Resultado de las simulaciones

4.1.2.1 Resultados para módulo de deformaciones impuesto $|\varepsilon_{max}| = 0.011$

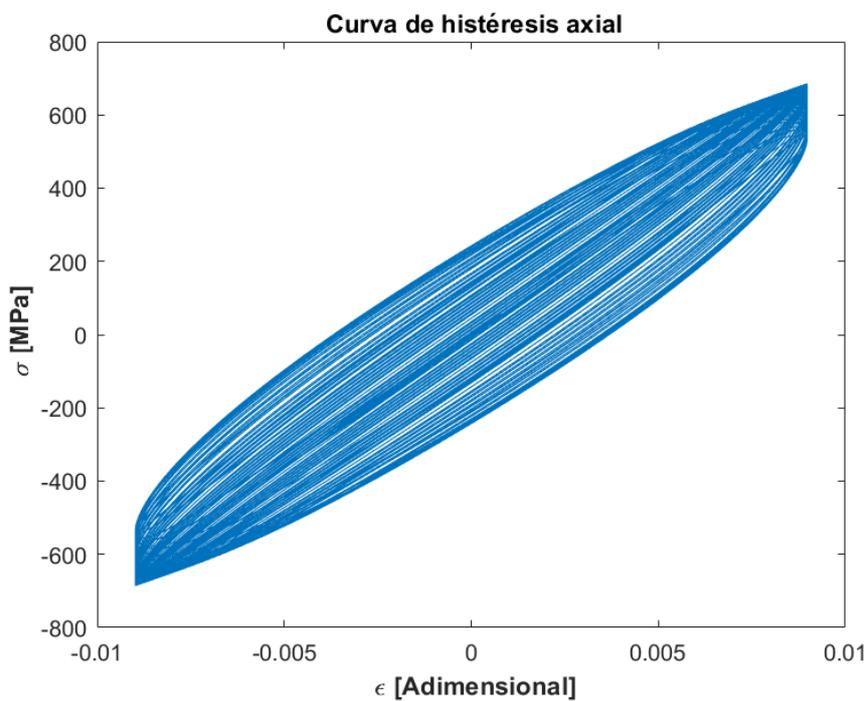


Figura 4.1.2.1.1

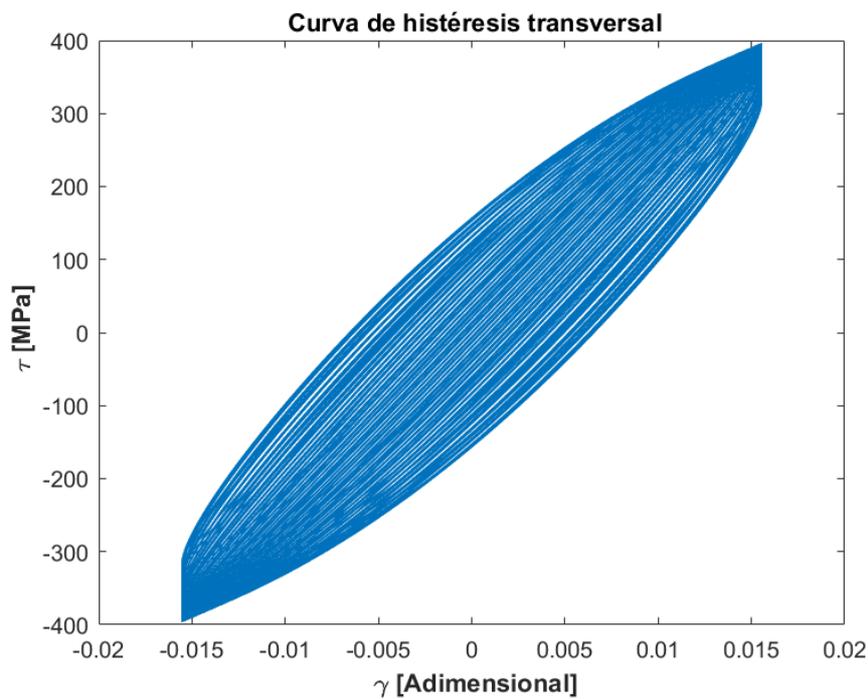
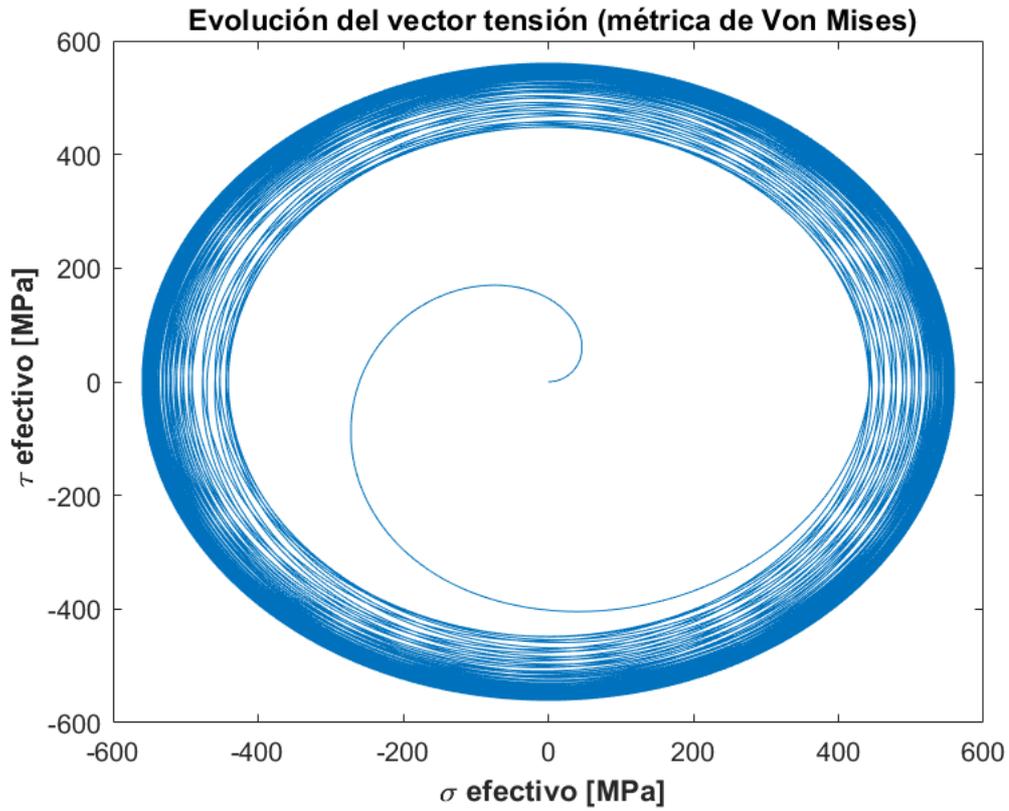
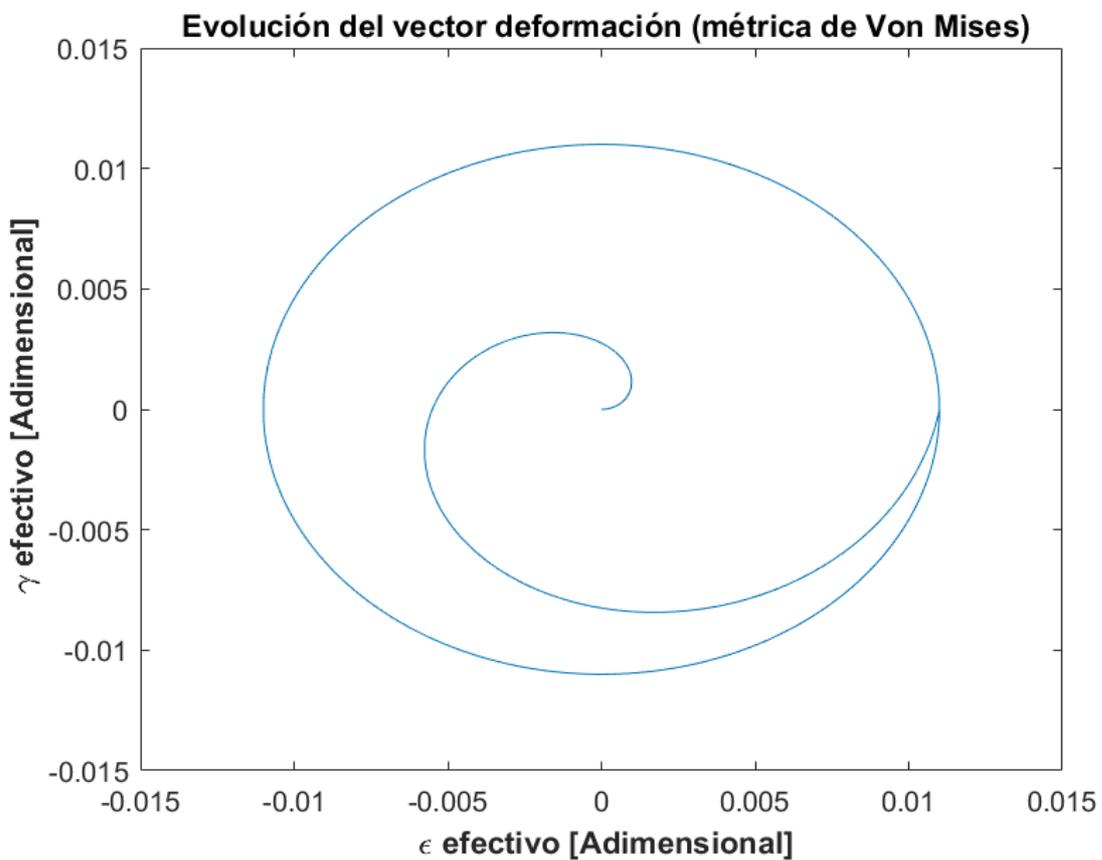


Figura 4.1.2.1.2

*Figura 4.1.2.1.3**Figura 4.1.2.1.4*

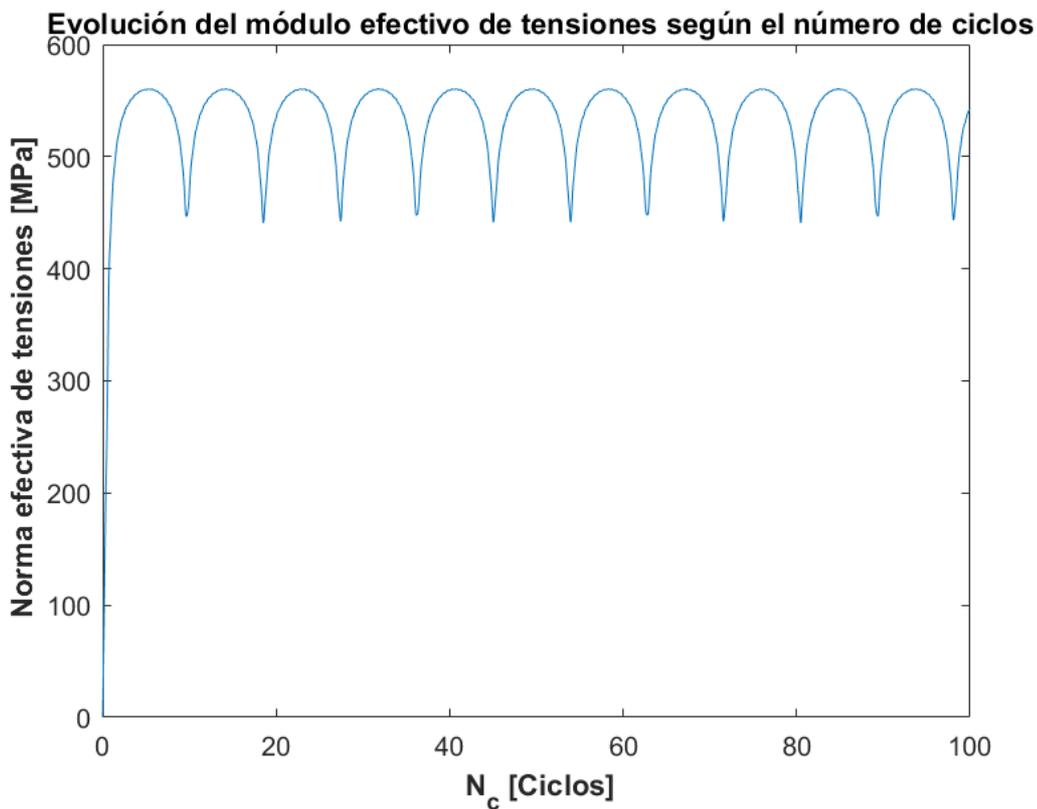


Figura 4.1.2.1.5

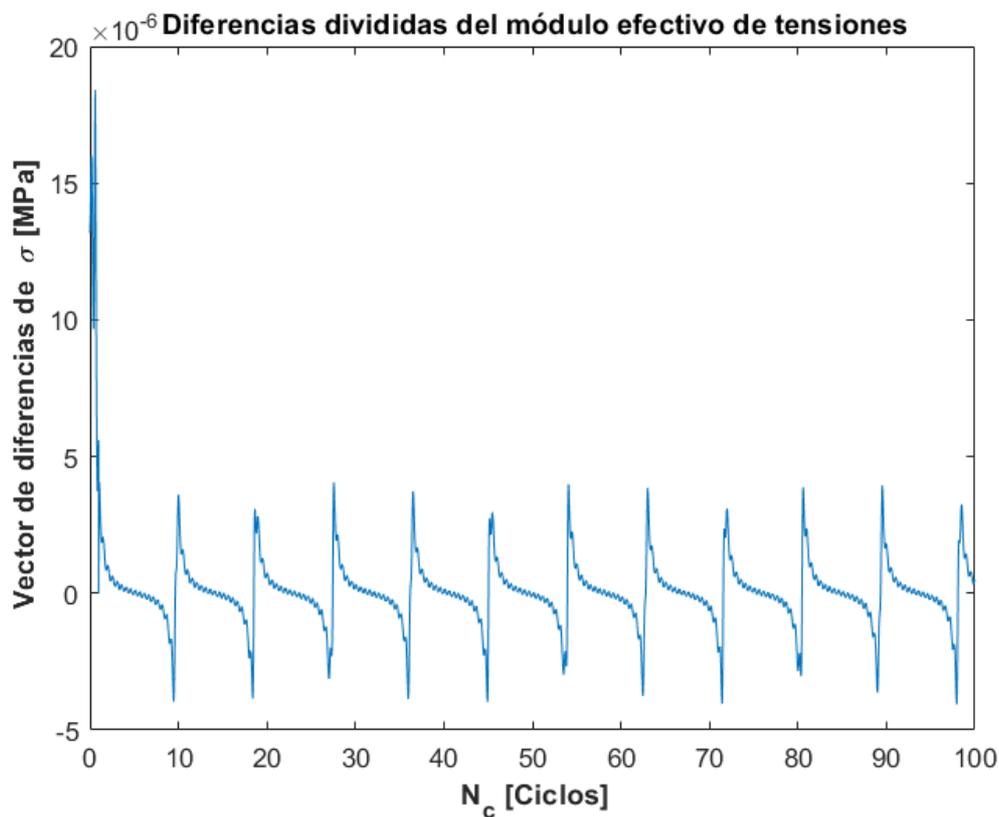


Figura 4.1.2.1.7

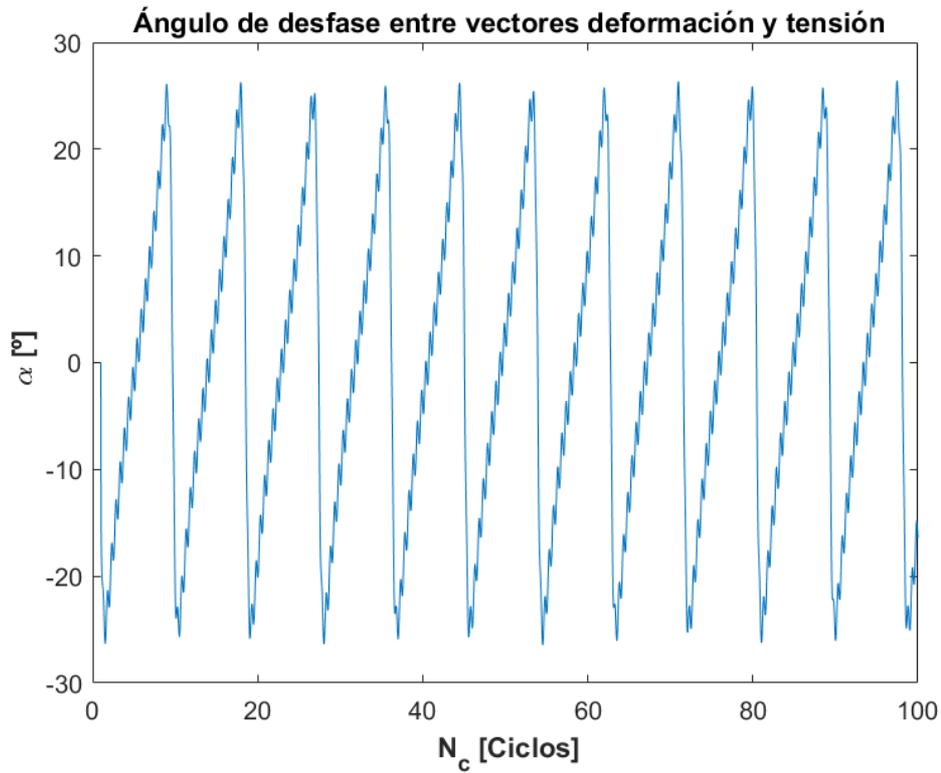


Figura 4.1.2.1.8

4.1.2.2 Resultados para módulo de deformaciones impuesto $|\epsilon_{max}| = 0.014$

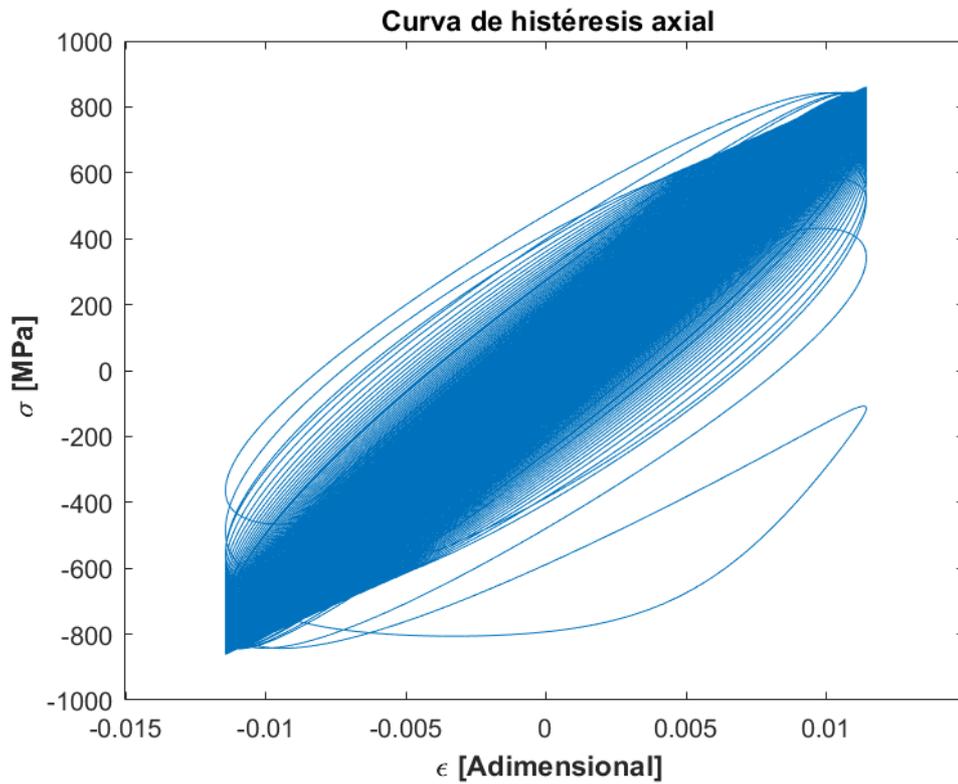


Figura 4.1.2.2.1

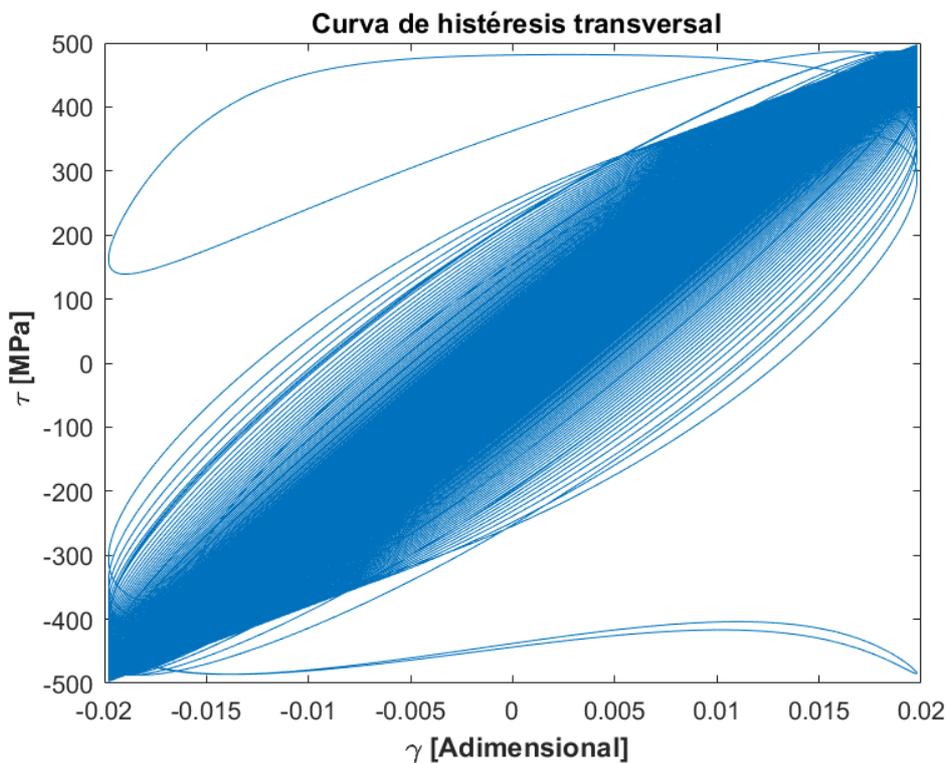


Figura 4.1.2.2.2

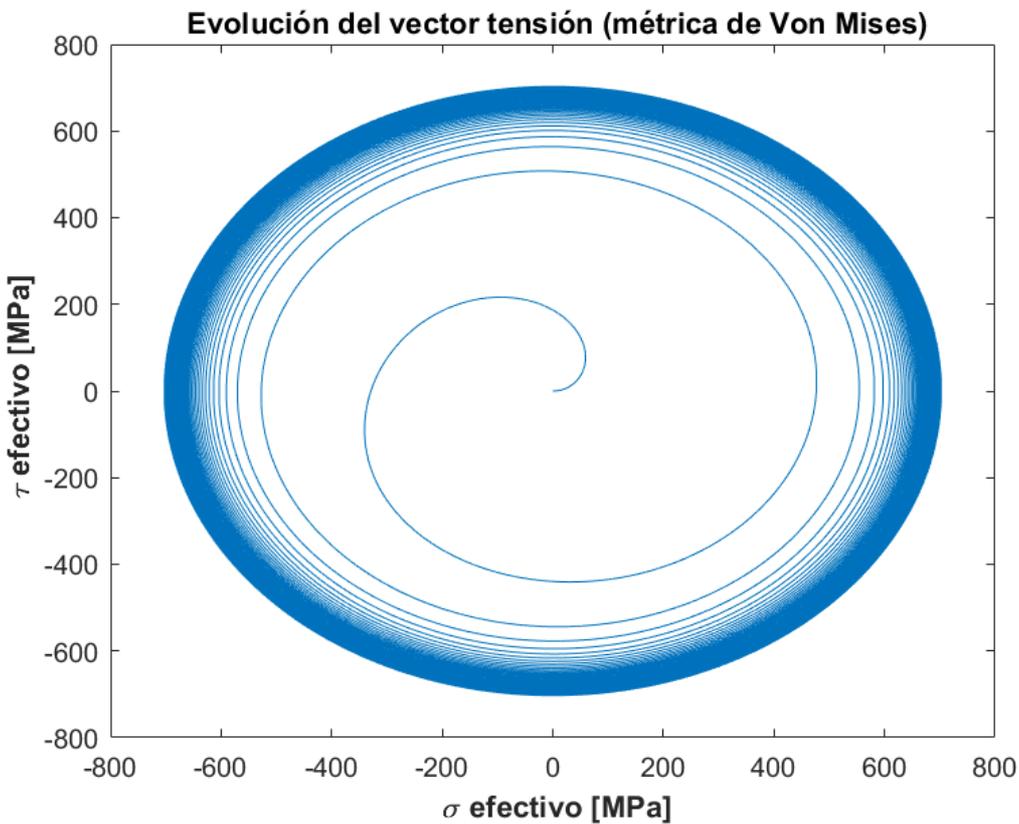


Figura 4.1.2.2.3

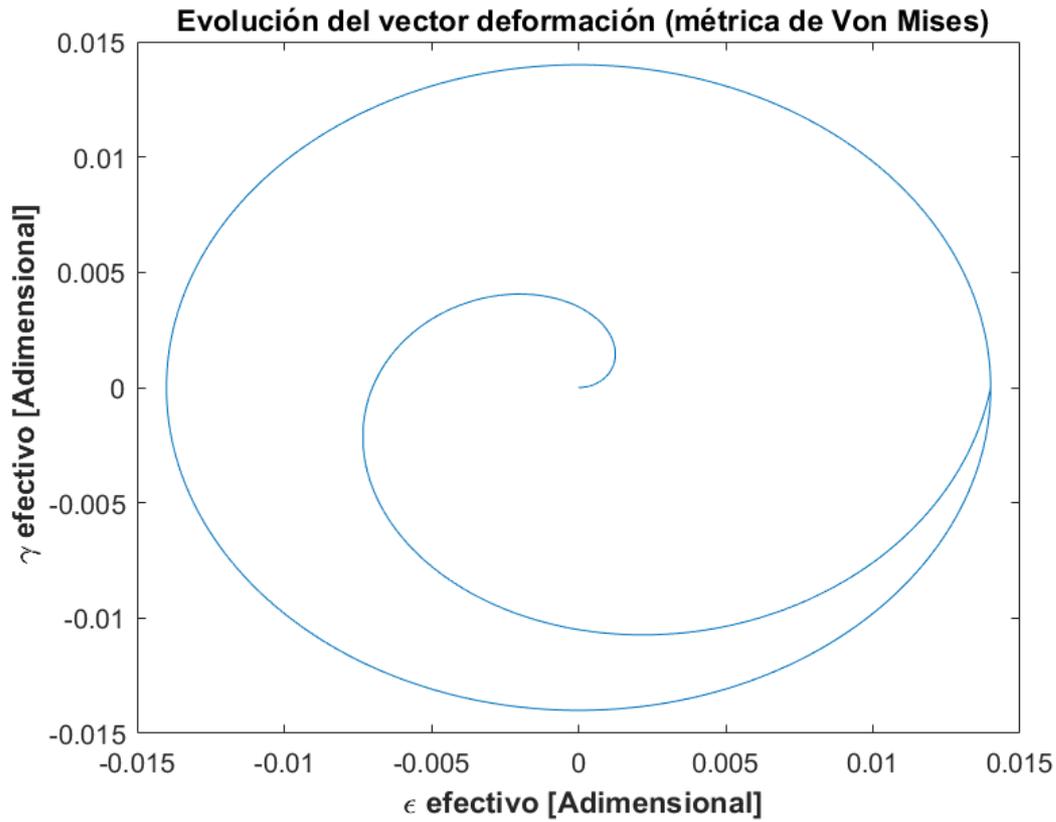


Figura 4.1.2.2.4

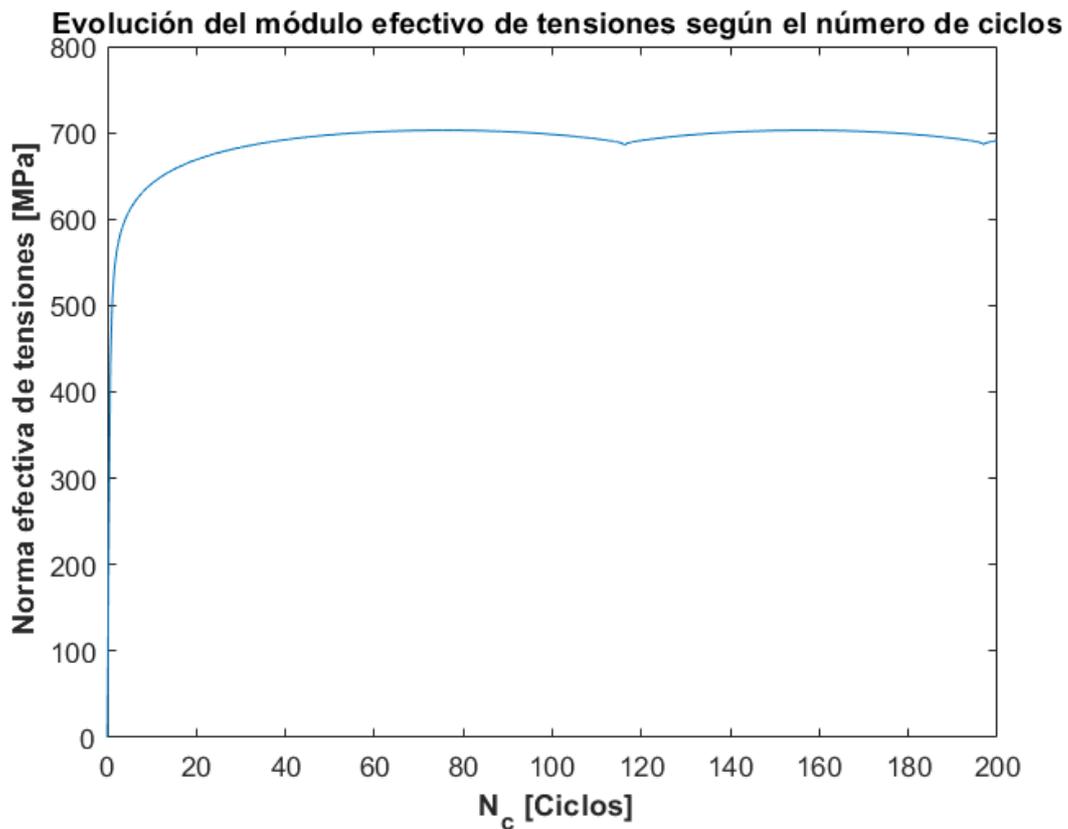


Figura 4.1.2.2.5

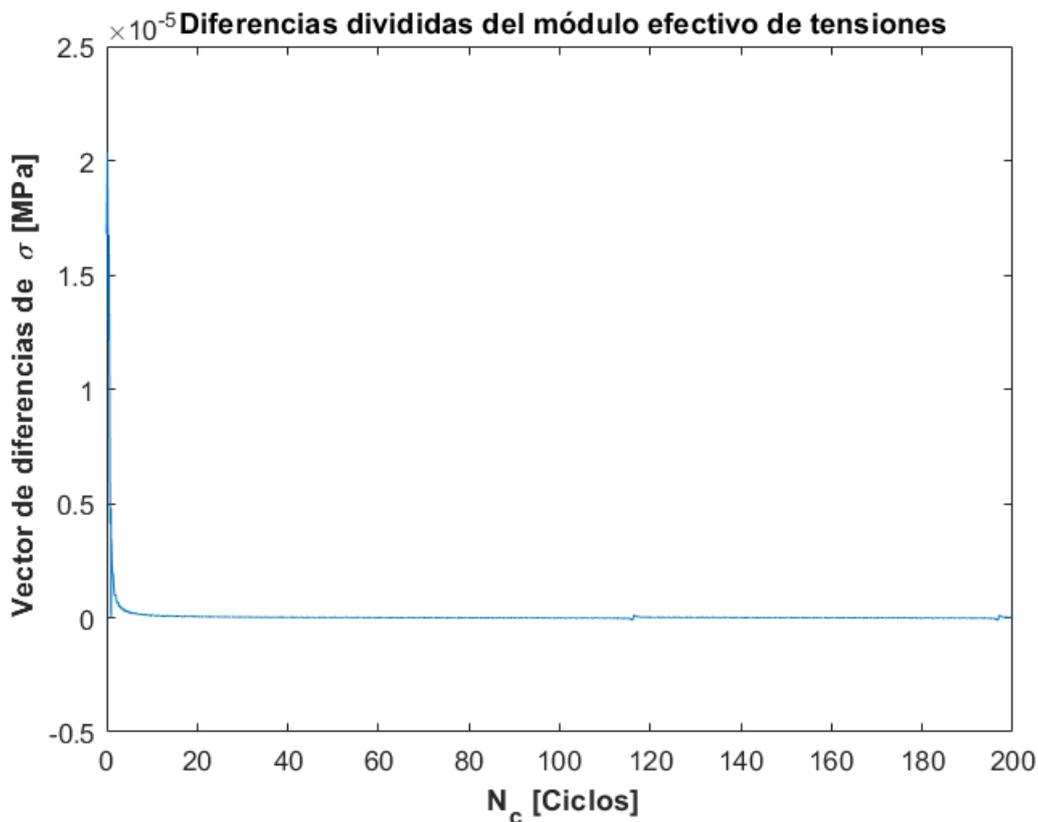


Figura 4.1.2.2.6

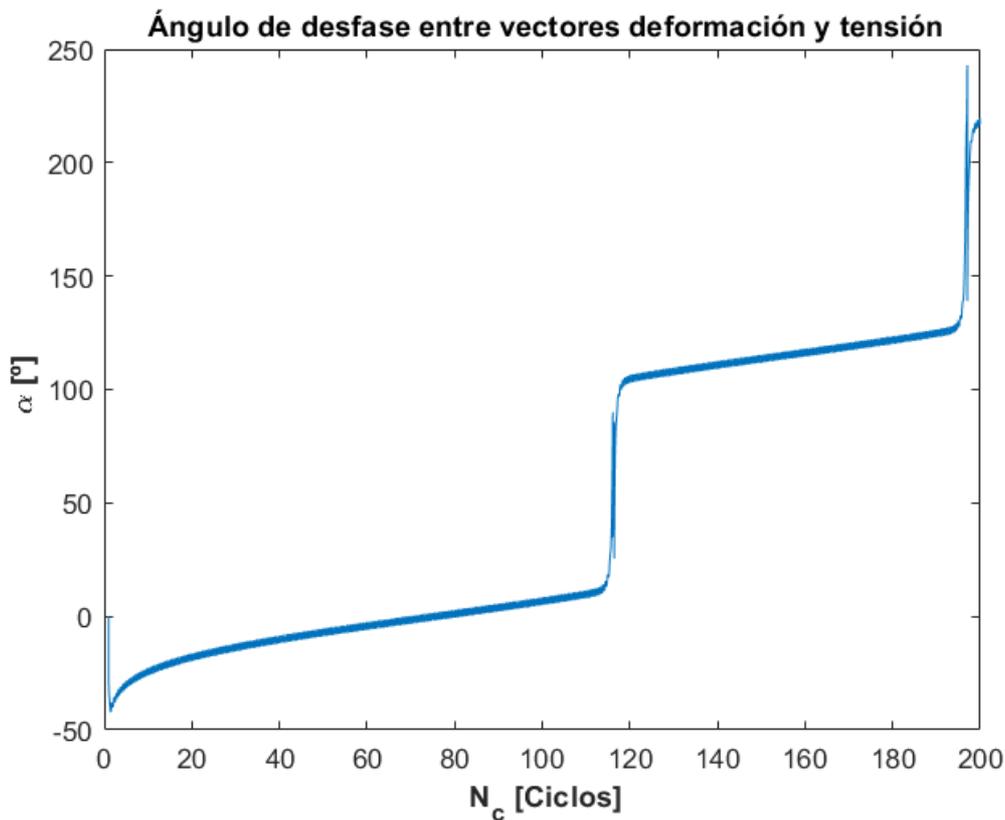


Figura 4.1.2.2.7

4.1.2.3 Resultado para módulo de deformaciones impuesto $|\varepsilon_{max}| = 0.017$

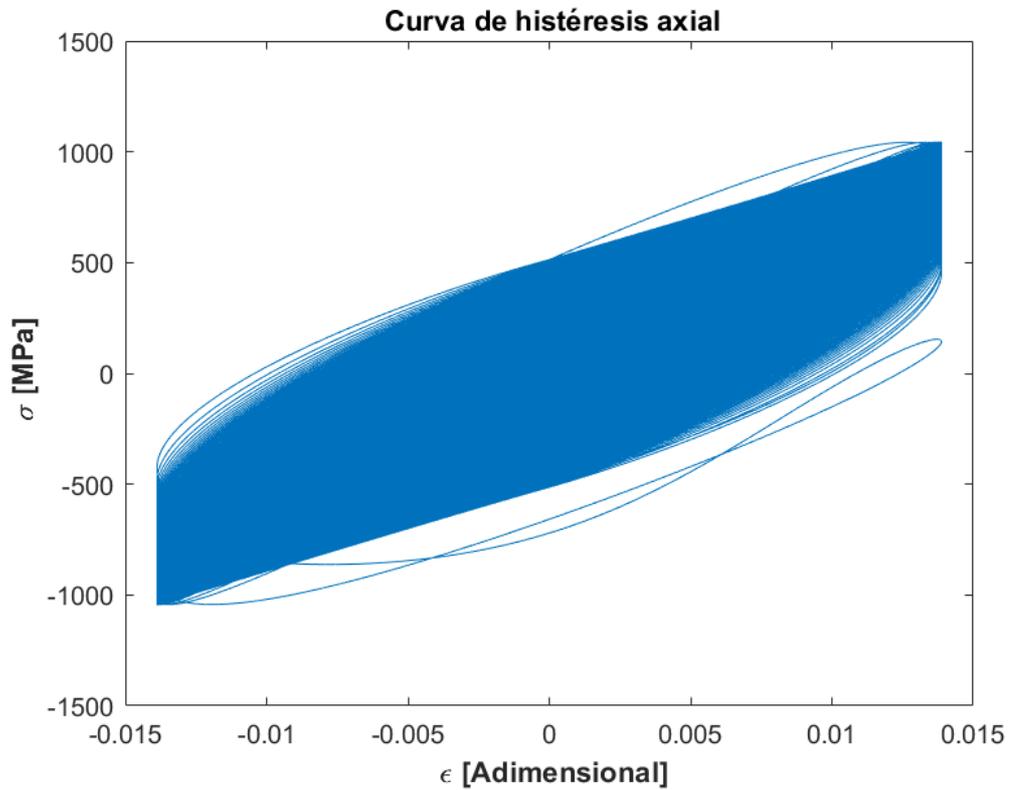


Figura 4.1.2.3.1

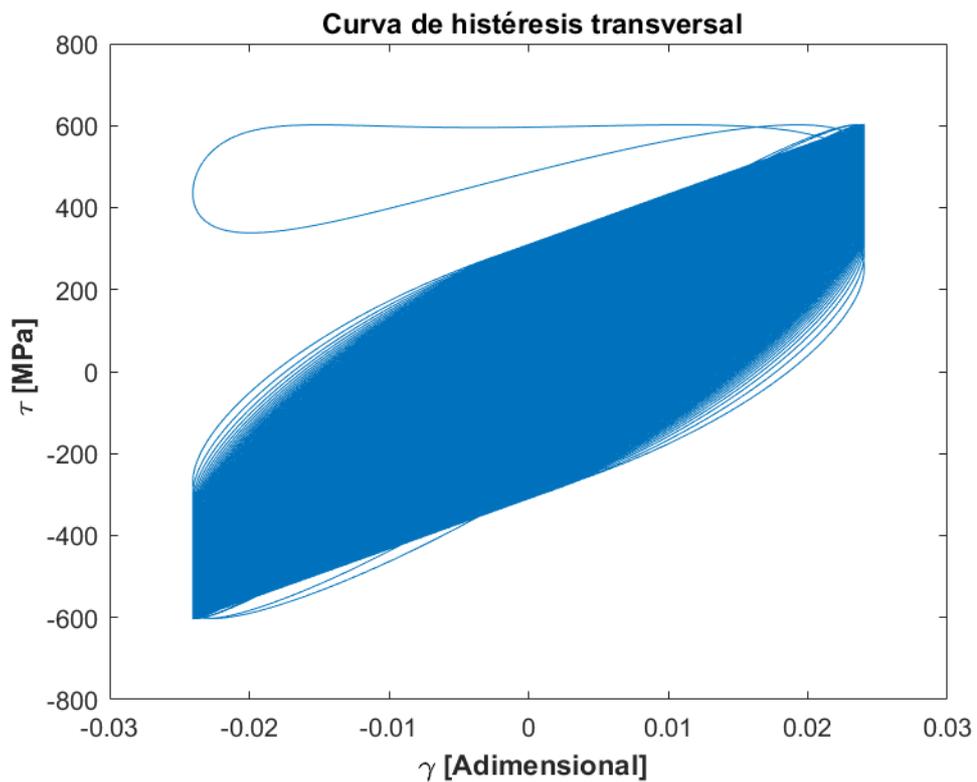


Figura 4.1.2.3.2

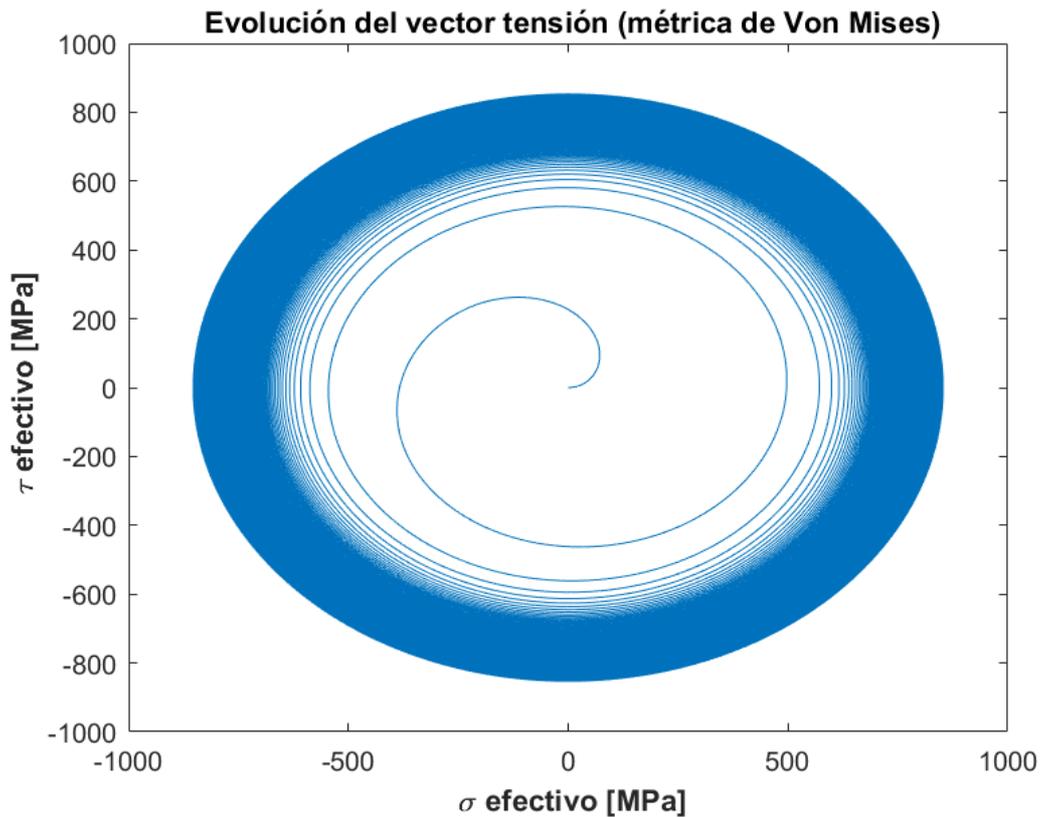


Figura 4.1.2.3.3

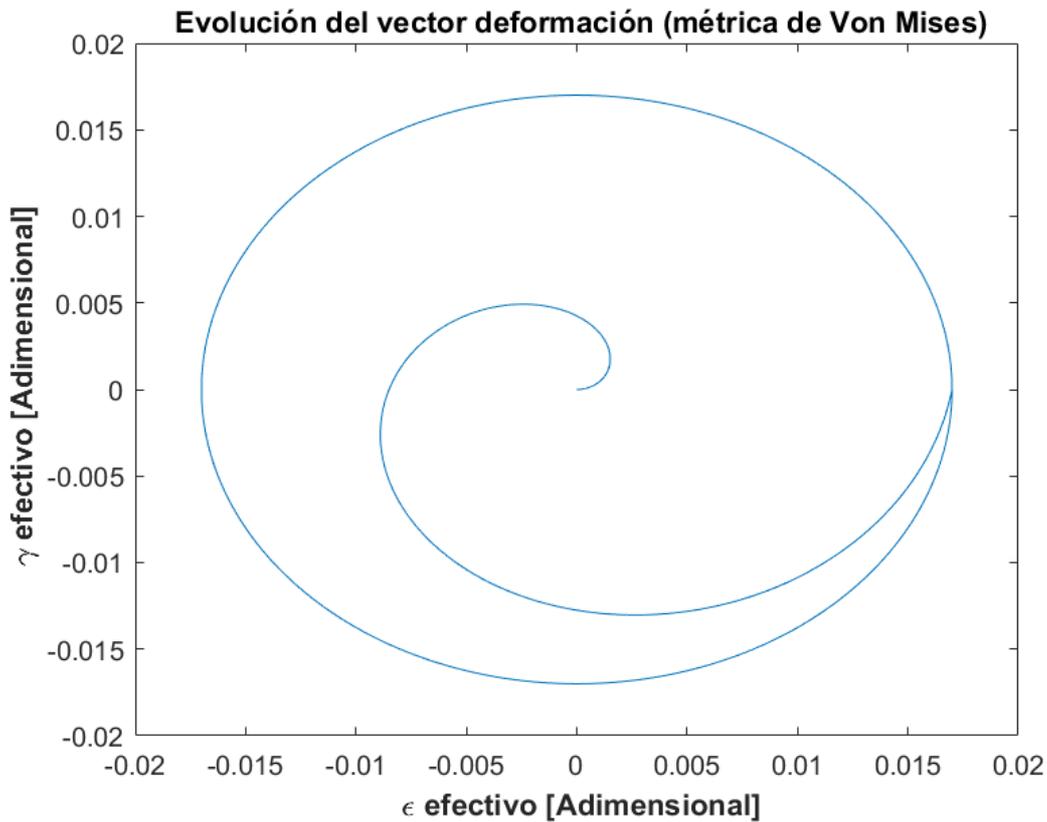


Figura 4.1.2.3.4

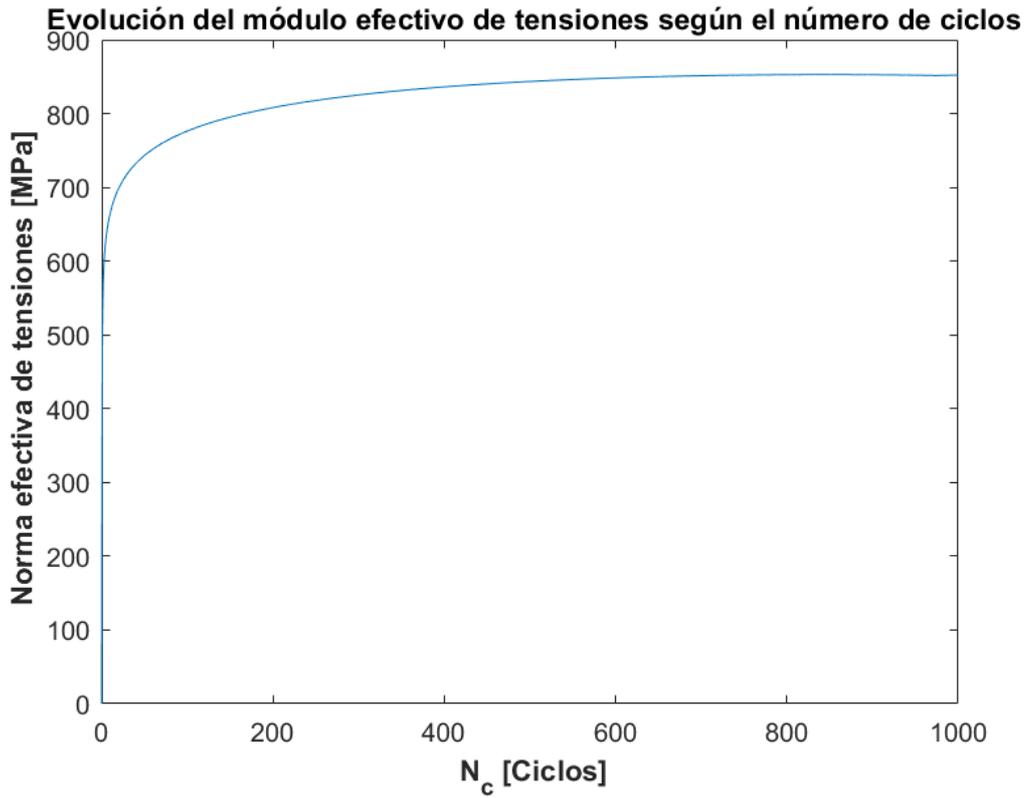


Figura 4.1.2.3.5

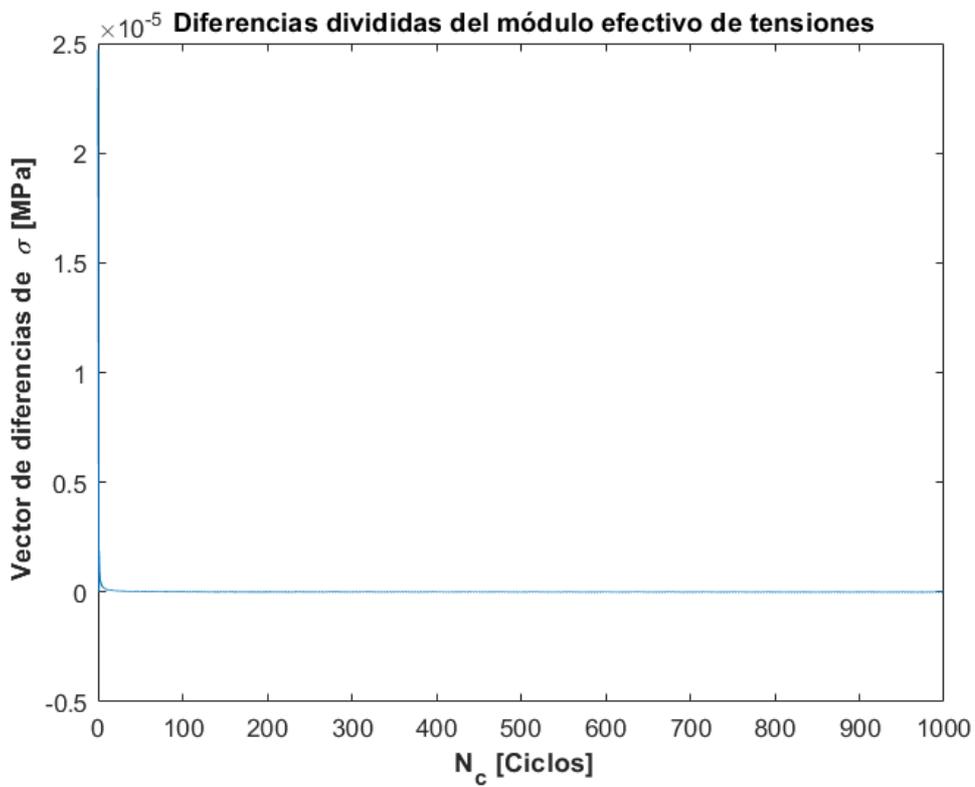


Figura 4.1.2.3.6

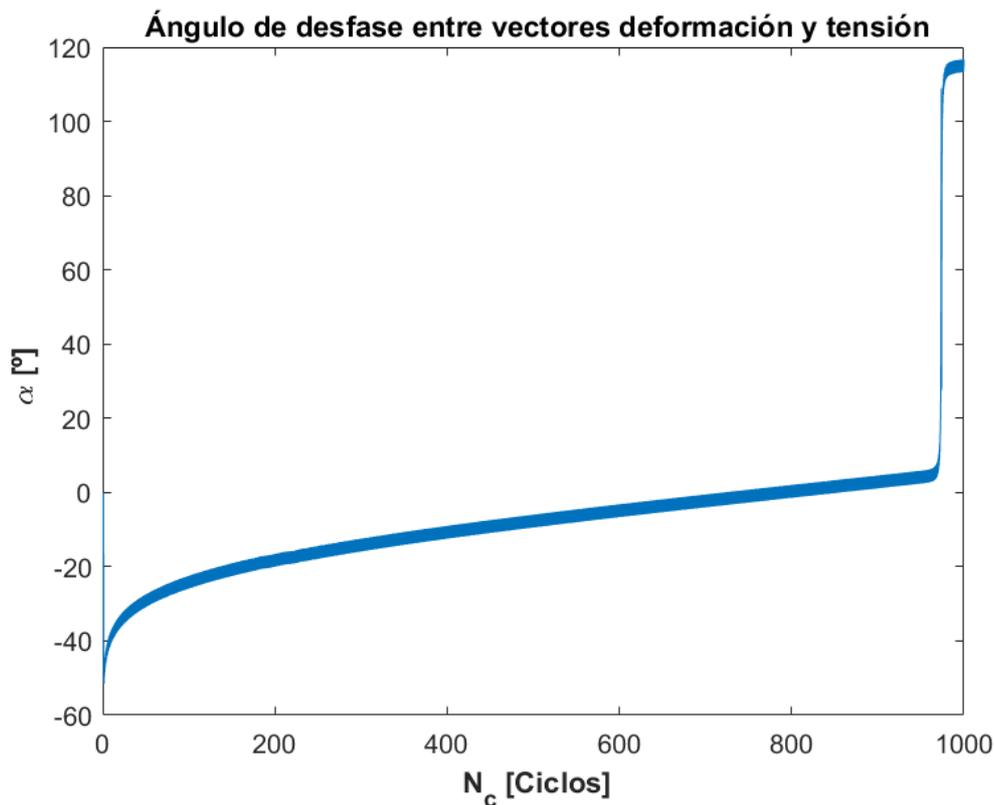


Figura 4.1.2.3.7

4.1.3 Análisis de los resultados

Procedemos a analizar las representaciones gráficas que nos da el primer modelo de cálculo, en el cual sólo se implementan las ecuaciones de carga del material, las cuales son válidas cuando el módulo del vector tensión $|\sigma|$ aumenta, es decir, $d|\sigma| > 0$. Es por esto por lo que las figuras que representan las diferencias divididas de $|\sigma|$ será crucial, ya que si vemos que en algún momento $|\sigma|$ decrece, (vector de diferencias divididas negativo) la validez de los resultados sería muy cuestionable.

En principio, es interesante analizar las figuras 4.1.2.1.1, 4.1.2.1.2, 4.1.2.2.1, 4.1.2.2.2, 4.1.2.3.1 y 4.1.2.3.2, las cuales representan los lazos histéresis tensión-deformación para las componentes axiales y tangenciales, respectivamente. Parece ser que, viendo las figuras de los ciclos de histéresis, en la medida que se incrementa el número de ciclos, el material se va endureciendo cada vez más, por lo que el área que encierra el lazo de histéresis, cada vez se va estrechando más. Esto podemos apreciarlo mejor si utilizamos el comando *comet* en MATLAB para la representación de los resultados, en vez de utilizar el comando *plot*, lo que nos muestra una evolución en tiempo real de cómo se construyen los lazos de histéresis. Llega un momento en el que los lazos se estrechan tanto que el material parece que colapsa, pues se queda sin margen de plastificación. Este momento, podría ser cuando el valor de $|\sigma|$ alcanza su máximo, según las previsiones teóricas vistas en las hipótesis del apartado 3 de esta memoria.

Sin embargo, si observamos las figuras 4.1.2.1.6, 4.1.2.2.6 y 4.1.2.3.6, las cuales representan las diferencias divididas para los tres valores del módulo de deformación impuesto, respectivamente, podemos observar que el módulo efectivo de tensiones $|\sigma|$ decrece, lo que nos hace dudar de la validez de los cálculos realizados. Esto se puede observar a su vez en las figuras 4.1.2.1.5, 4.1.2.2.5 y 4.1.2.3.5, las cuales muestran la evolución de $|\sigma|$ frente al número de ciclos N_c . Se ve claramente un descenso a partir de cierto punto del valor de $|\sigma|$, desde el cual las ecuaciones se comportan de forma extraña. Cabe destacar que el comportamiento de las ecuaciones difiere según cambia el módulo de deformación impuesta $|\epsilon_{max}|$. Se observa un comportamiento más inestable o “turbulento” en el caso de deformaciones más bajas, comportándose nuestro modelo de cálculo de forma más estable o “laminar” en el caso de deformaciones mayores. Podemos establecer que el caso $|\epsilon_{max}| = 0.014$ podría ser el punto de transición entre un tipo de comportamiento y el otro.

Para comprobar qué ocurre en el momento en que las curvas de histéresis se estrechan al límite de colapsar, hemos de calcular sólo hasta el número de ciclos donde $|\sigma|$ alcanza su valor máximo, verificando así si este es el punto en el que los lazos de histéresis colapsan. Representando para los ciclos de histéresis en coordenadas axiales ($\epsilon - \sigma$), nos darían los siguientes resultados:

Para $|\epsilon_{max}|=0.011$:

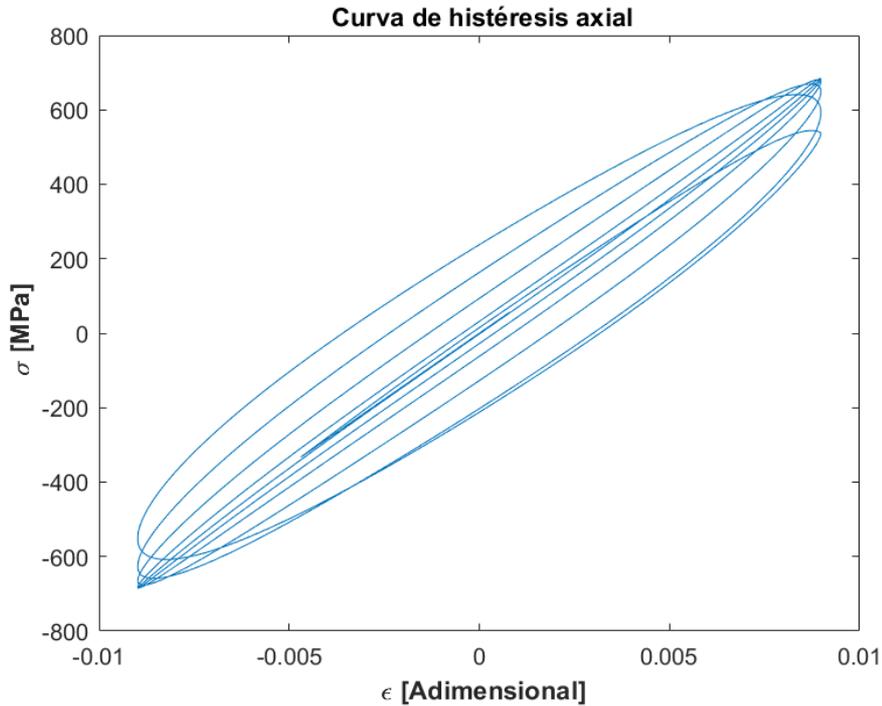


Figura 4.1.3.1

Para $|\epsilon_{max}| = 0.014$:

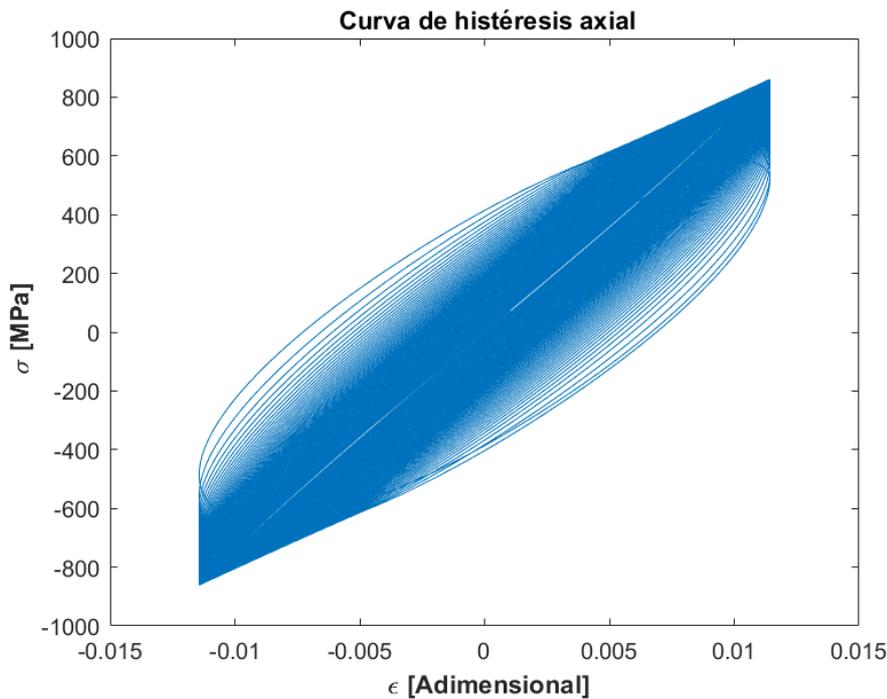


Figura 4.1.3.2

A deformaciones impuestas superiores, como en el caso de $|\epsilon_{max}|=0.017$, la progresión hasta cerrar el ciclo de histéresis es tan lenta que las representaciones no pueden distinguirse de las otras sin hacer zoom, o sin utilizar

el comando *comet*.

Hay que apreciar también que según la deformación impuesta $|\varepsilon_{max}|$ es mayor, mayor número de ciclos son necesarios para que $|\sigma|$ alcance el valor máximo. Se ve claramente que hay inestabilidades numéricas, que van cambiando según cambia el valor de $|\varepsilon_{max}|$ impuesto. Para valores más pequeños, las fluctuaciones son rápidas y de amplitud pequeña. Sin embargo, para valores más altos, las fluctuaciones tienen una amplitud considerable, aunque tengan una frecuencia menor. Esto podría poner en duda la validez de los resultados obtenidos. Sin embargo, estas fluctuaciones ocurren después de que $|\sigma|$ sea máximo. Al estar utilizando las ecuaciones sólo teniendo en cuenta la carga, es posible que se deba a esto, pues después del pico de $|\sigma|$, su valor desciende.

Teniendo sólo en cuenta el tramo en el que $|\sigma|$ va aumentando, podemos afirmar que el punto en el que $|\sigma|$ es máximo, es cuando el ciclo de histéresis no se puede cerrar más. Según la teoría propuesta, el valor de $|\sigma|$ se debería de estabilizar entonces. ¿Qué ocurriría si el valor de $|\sigma|$ se estabiliza? ($d|\sigma| = 0$)

Al ser la deformación plástica ε_p directamente proporcional a $d|\sigma|$ según la expresión 1.11, podemos decir que, en este caso, en el que $|\sigma|$ se estabiliza, no hay deformación plástica. Si no hay deformación plástica, la evolución del material debe ser elástica, quedando el sistema de ecuaciones diferenciales dependiendo sólo del tensor de Hooke, sin el término plástico, cuya expresión sería:

$$\begin{bmatrix} \frac{d\sigma}{dt} \\ \frac{d\tau}{dt} \end{bmatrix} = \begin{bmatrix} E & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} \frac{d\varepsilon}{dt} \\ \frac{d\gamma}{dt} \end{bmatrix} \quad (4.1.3)$$

Pero al usar las ecuaciones sin tener en cuenta la descarga, tenemos un problema asociado a ello, pues después de que $|\sigma|$ alcance el máximo, este valor comienza a decrecer en todos los ensayos realizados, en unos más rápido (a menores deformaciones impuestas), y en otros de forma más progresiva (a mayores deformaciones impuestas). Esto es totalmente incompatible con este modelo, pues a partir de este momento deberíamos utilizar las ecuaciones teniendo en cuenta la descarga (efecto memoria), descritas en la teoría de plasticidad propuesta al inicio del documento, que no se han implementado en este modelo.

Debemos entonces, implementar las ecuaciones teniendo en cuenta la descarga, para verificar si efectivamente después de que $|\sigma|$ alcance el máximo, este valor se estabiliza, como hemos previsto con las hipótesis anteriormente señaladas.

4.2 Segundo modelo de cálculo. Utilización de ecuaciones de descarga

En el apartado anterior hemos podido ver que nuestros resultados no podrían tener la validez necesaria, puesto que hemos utilizado un modelo de cálculo suponiendo que $|\sigma|$ es siempre creciente, es decir, $d|\sigma| \geq 0$ y vemos que esto se incumple en las simulaciones que llevamos hasta ahora. Debemos entonces calcular las tensiones como distancias respecto a la circunferencia de plastificación C_k que se genere, como hemos explicado detalladamente en el apartado introductorio de este proyecto.

Pero esto tiene una problemática asociada. Para que este método funcione, en nuestro sistema de ecuaciones diferenciales en derivadas parciales, debemos detectar cuando existe un paso de integración en el que $d|\sigma| \leq 0$ y a partir de este, almacenar este paso como primer punto de descarga σ_0 y calcular su circunferencia de plastificación asociada C_0 . Hasta aquí no hay problema, pero ¿qué ocurriría si se almacenaran sucesivos puntos de descarga, y una vez que respecto a estas circunferencias de plastificación $|\sigma|$ vuelva a crecer y estas se borren debido al efecto memoria? Pues que los sistemas de integración optimizados que se incluyen en el paquete MATLAB no permiten volver a pasos de integración anteriores, por lo que no podríamos almacenar los puntos de descarga y seguir integrando, para volver a ellos cuando se diera el efecto memoria.

Para poder implementar este complejo sistema de ecuaciones, hay una alternativa, establecer un método de integración manual paso a paso siguiendo el método Runge-Kutta de orden 4.

Para ello se crearán las funciones *syst2rk4d.m* y *syst2d.m*, que sustituirán respectivamente a las funciones *ode45.m* de MATLAB y *syst2.m* que hemos utilizado para calcular el segundo tramo de nuestras ecuaciones en el modelo de cálculo anterior. Cabe destacar, que, en el primer tramo de cálculo, es decir, el tramo de carga en el que $|\varepsilon|$ se aproxima a $|\varepsilon_{max}|$ mediante una espiral, no es necesario establecer estas ecuaciones ya que en este tramo $|\sigma|$ siempre crece y por lo tanto $d|\sigma|$ es siempre positivo, siendo el modelo de cálculo anterior efectivo en este caso.

4.2.1 Parametrización de modelo de integración paso a paso mediante método Runge-Kutta de cuarto orden

El método Runge-Kutta de cuarto orden nos permite establecer un método de integración paso a paso de una forma más exacta y optimizada que el conocido método de Euler. Explicado brevemente, consiste en establecer las siguientes expresiones para calcular cada paso de integración:

$$Y(i + 1) = Y(i) + h(K_1 + 2K_2 + 2K_3 + K_4) \quad (4.2.1)$$

$$K_1 = dY(X(i), Y(i)) \quad (4.2.2)$$

$$K_2 = dY\left(X(i) + \frac{h}{2}, Y(i) + K_1 \frac{h}{2}\right) \quad (4.2.3)$$

$$K_3 = dY\left(X(i) + \frac{h}{2}, Y(i) + K_2 \frac{h}{2}\right) \quad (4.2.4)$$

$$K_4 = dY(X(i) + h, Y(i) + K_3 h) \quad (4.2.5)$$

Expresiones en las que Y es la función incognita a obtener, i es el paso del ultimo valor de Y obtenido, h es el paso de integración establecido, y X constituye el conjunto de variables independientes de las que depende Y .

Mediante este método podremos acceder a todos los puntos de integración y poder así almacenar los sucesivos puntos de descarga σ_k que se vayan calculando y sus circunferencias de plastificación asociadas C_k , calculando a su vez el centro de estas circunferencias $\sigma_{c,k}$ desde el cual se medirán relativamente las distancias en el espacio tensional para calcular los siguientes puntos. ¿Cómo podemos implementar entonces las ecuaciones teniendo en cuenta la descarga? Explicaremos brevemente el algoritmo utilizado en el fichero *syst2rk4d.m*:

- Se establecen primero los valores iniciales de centros de circunferencias de plastificación (empieza en el $[0,0]$), índice de paso i en 1, y el parámetro c en 1 también, parámetro que nos ayudará a contabilizar más adelante cuál es la circunferencia de plastificación activa.
- Empezamos a utilizar las ecuaciones sin tener en cuenta la descarga, integrando nuestro sistema basado en RK4 introduciendo la función *syst2.m* como dY . Se establecerá un condicional al final de cada paso, que verificará si $|\sigma|$ en el último paso calculado ha aumentado respecto al paso anterior. Si esto ocurre, seguiremos integrando con estas ecuaciones. Sin embargo, si ocurre lo contrario, establecemos el parámetro d con valor 1, parámetro respecto a qué punto de descarga medimos las distancias. Al tomar valor 1, se crea el primer punto de descarga.
- Si al calcular un nuevo paso de integración el parámetro d es distinto de 0, volveremos al punto de integración anterior, calcularemos la distancia q_0 , respecto a la cual se calculará el siguiente punto, mediante las ecuaciones (1.23), (1.24) y (2.1.14), sustituyendo en esta última la función de endurecimiento $\phi(|\sigma|)$ por $\phi(q)$. Esto se implementa en MATLAB introduciendo como función a integral la función *syst2d.m*, que toma como valores de entrada $\sigma_{c,k}$, el último valor de σ , y el punto de descarga activo σ_k .

- Si este nuevo valor de $|\sigma|$ es mayor que el anterior, se produce efecto memoria y borramos el punto de descarga, estableciendo de nuevo el parámetro d en 0. Volvemos a utilizar las ecuaciones sin tener en cuenta la descarga, según el primer modelo de cálculo, mediante la función *sys2.m*.
- Esto cambia radicalmente si se produce más de un punto de descarga sucesivo, pues entonces se crearía la primera circunferencia de plastificación asociada C_0 y su centro $\sigma_{c,0}$, respecto al que se calculara entonces la distancias q y q_0 , según la expresión 1.26. Cabe destacar q_k es igual al diámetro de la circunferencia de plastificación C_k activa.
- A partir de entonces, se establece un bucle repetitivo por el cual si las distancias respecto al centro de la circunferencia de plastificación activa $\sigma_{c,k}$ se reduce, se crearán nuevos puntos de descarga y se incrementarán los valores de d y de c . Cada vez que aumentan estos parámetros, se almacenan los puntos de descarga y centros correspondientes. Al contrario, si las distancias aumentan y superan el radio de la circunferencia de plastificación activa, esta y su punto de descarga asociado se borrarán, tomándose como referencia el punto de descarga y su circunferencia de plastificación anteriores. Para entender mejor esto podemos examinar detenidamente las figuras 1.1, 1.2, 1.3, y 1.4

Para mayor detalle, los cálculos están perfectamente descritos y comentados en el fiche.

Tenemos entonces implementado nuestro sistema para calcular de forma más precisa nuestras simulaciones, ya que en este caso sí hemos tenido en cuenta que el material puede descargar, por lo que si vemos en las figuras que $|\sigma|$ desciende en algún instante, los resultados no tendrían por qué perder validez.

Pero no todo son ventajas, pues la exactitud de nuestros resultados dependerá de la tolerancia de integración h que escojamos. Aun así, podremos establecer como valor límite una tolerancia de integración para este método paso a paso de $h = 10^{-4}$ ya que el costo computacional de almacenar variables con pasos de integración (y por ende de muestreo) pequeños es altísimo, tanto en consumo de memoria como en tiempo de ejecución. Como simple comparativa, el fichero *espiral.cl* tarda en ejecutar las simulaciones del primer modelo de cálculo unos 20 segundos. En este modelo de cálculo esta cifra asciende hasta horas incluso si mantenemos el paso de integración en $h = 10^{-4}$. Veremos más adelante si hay alguna forma de corregir esto.

Estableceremos las simulaciones para los mismos valores que en el apartado 4.1, representando las figuras para $|\varepsilon_{max}|=0.011$, $|\varepsilon_{max}| = 0.014$ y $|\varepsilon_{max}| = 0.017$.

Para utilizar este modelo de cálculo, configuraremos el fichero *espiral.cl* de la siguiente forma:

- “*ecuacion=1*” para utilizar el primer modelo de cálculo.
- “*esp=1*” para aproximar el primer tramo en espiral, el cual se utilizará durante toda la simulación.
- “*material=1*” para tomar los parámetros del Aluminio 7075 T651
- “ $N_C = 100$ ” para calcular 100 ciclos en cada iteración. Esto se cambiará según los resultados para un mejor muestreo. Es decir, para $|\varepsilon_{max}| = 0.014$ evaluaremos 200 ciclos y para $|\varepsilon_{max}| = 0.017$ 1000 ciclos.
- “ $|\varepsilon_{max}| = VALOR DE TABLA A EVALUAR$ ”

El resto de la ejecución que no tenga que ver con la integración de las variables se hará análoga al procedimiento del primer modelo de cálculo descrito en el apartado 4.1

4.2.2 Resultados de las simulaciones

4.2.2.1 Resultado para módulo de deformaciones impuesto $|\varepsilon_{max}| = 0.011$

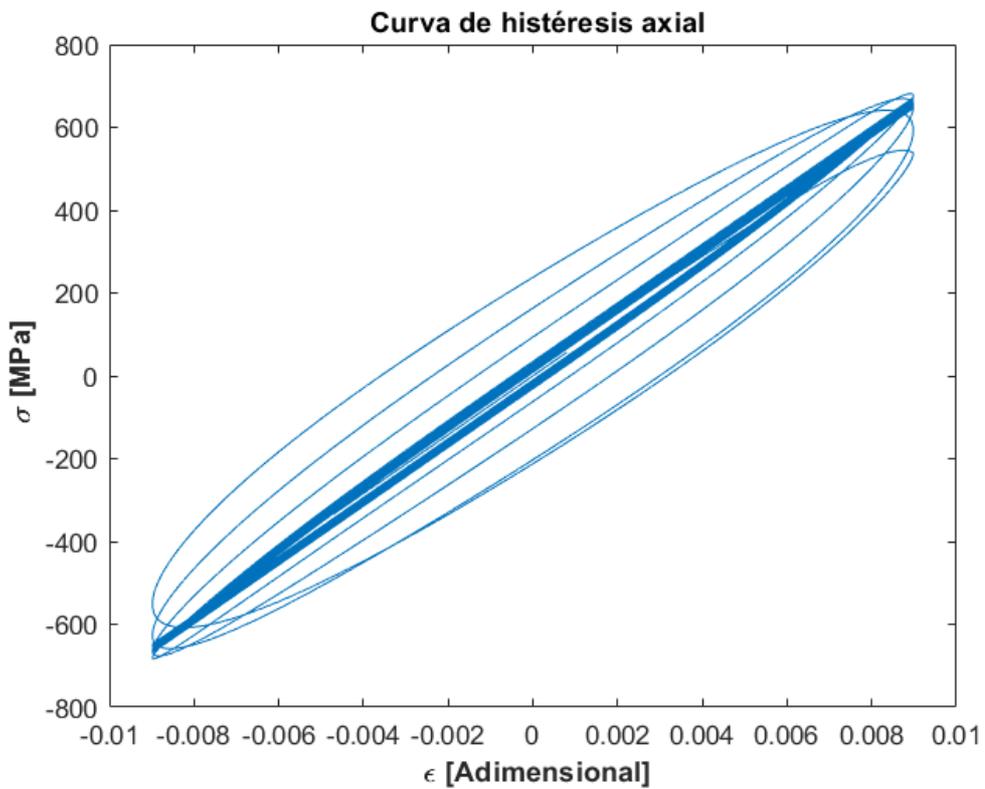


Figura 4.2.2.1.1

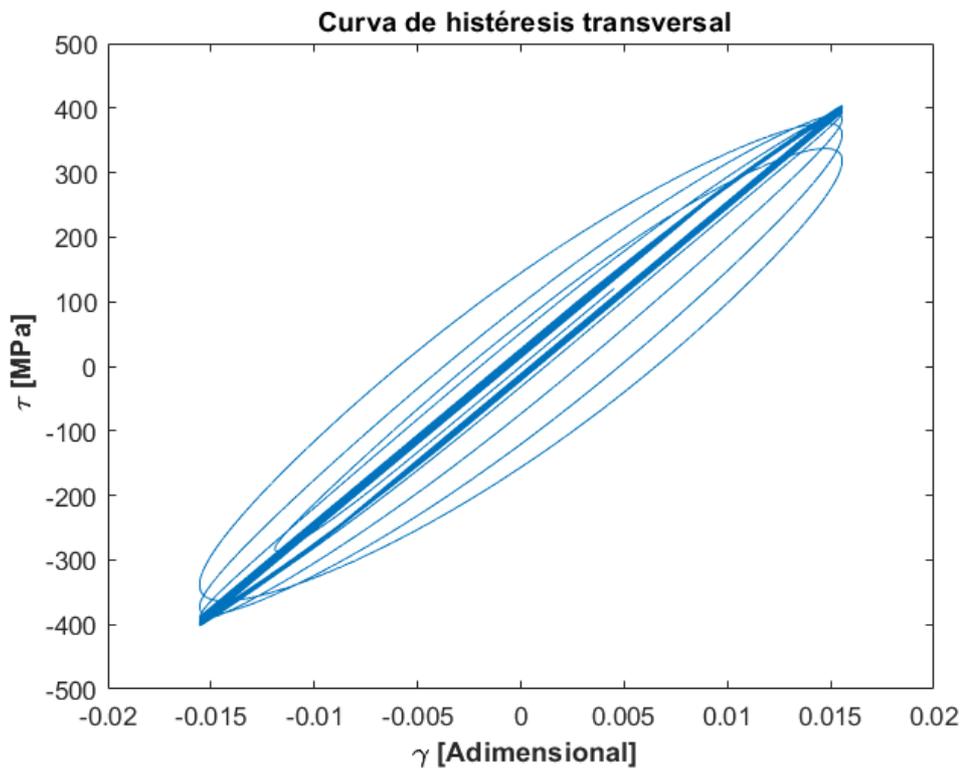


Figura 4.2.2.1.2

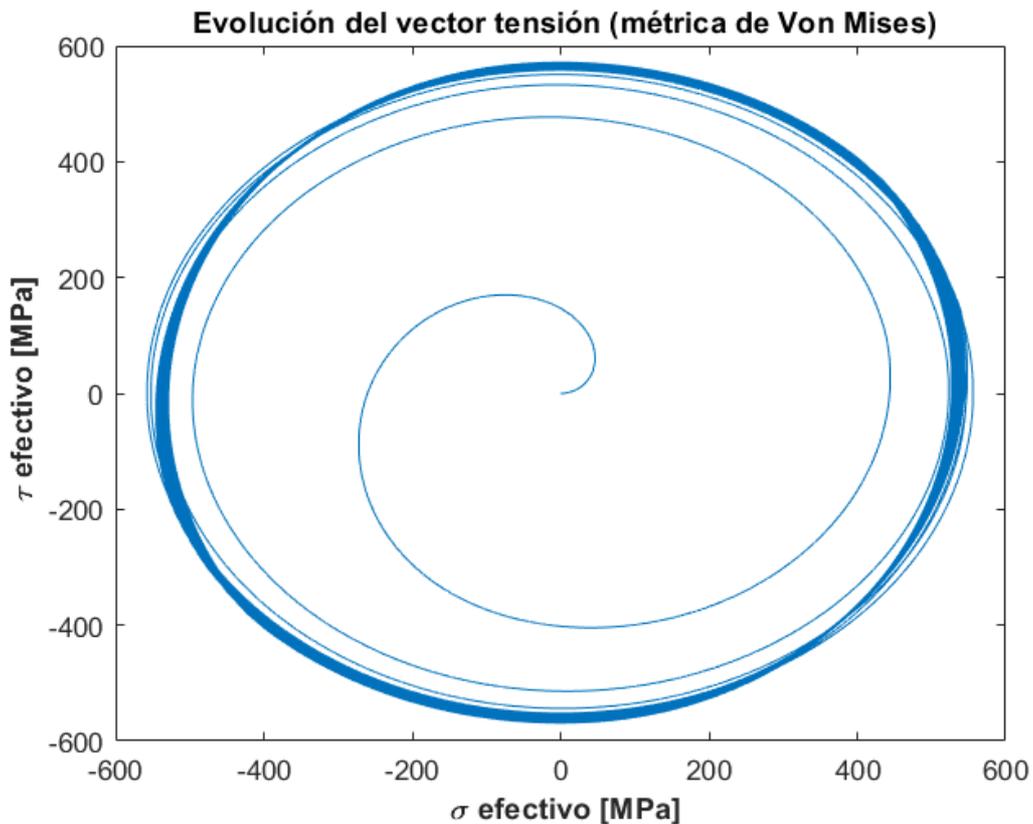


Figura 4.2.2.1.3

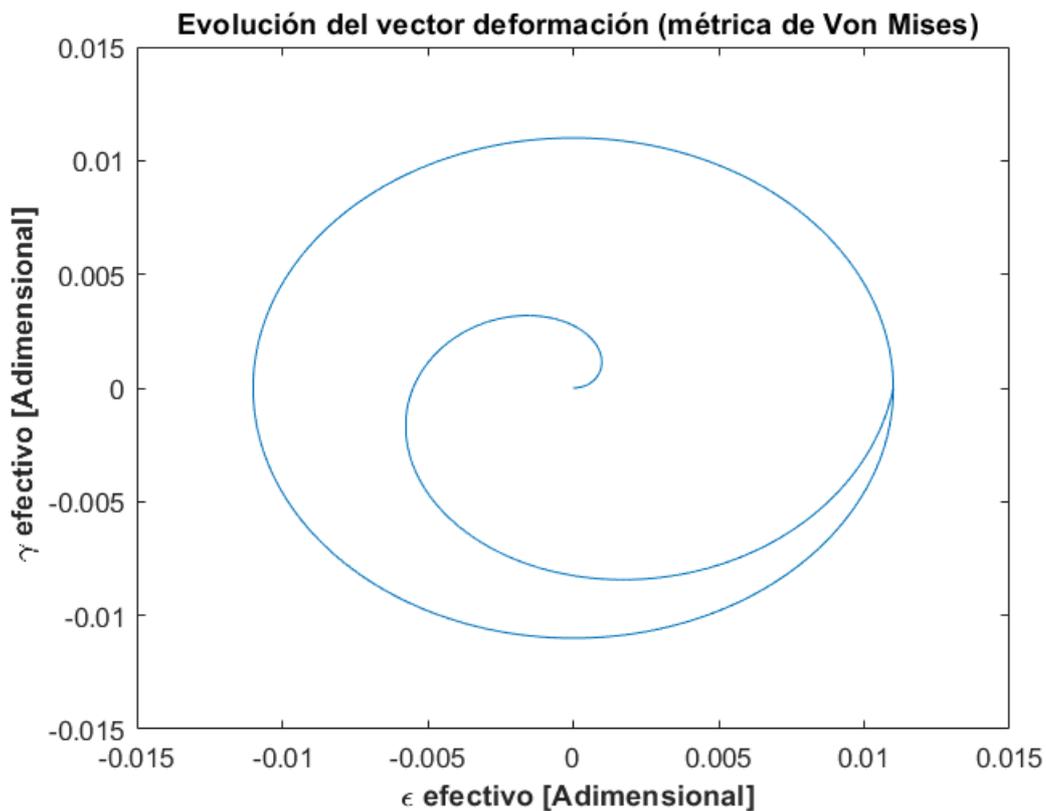


Figura 4.2.2.1.4

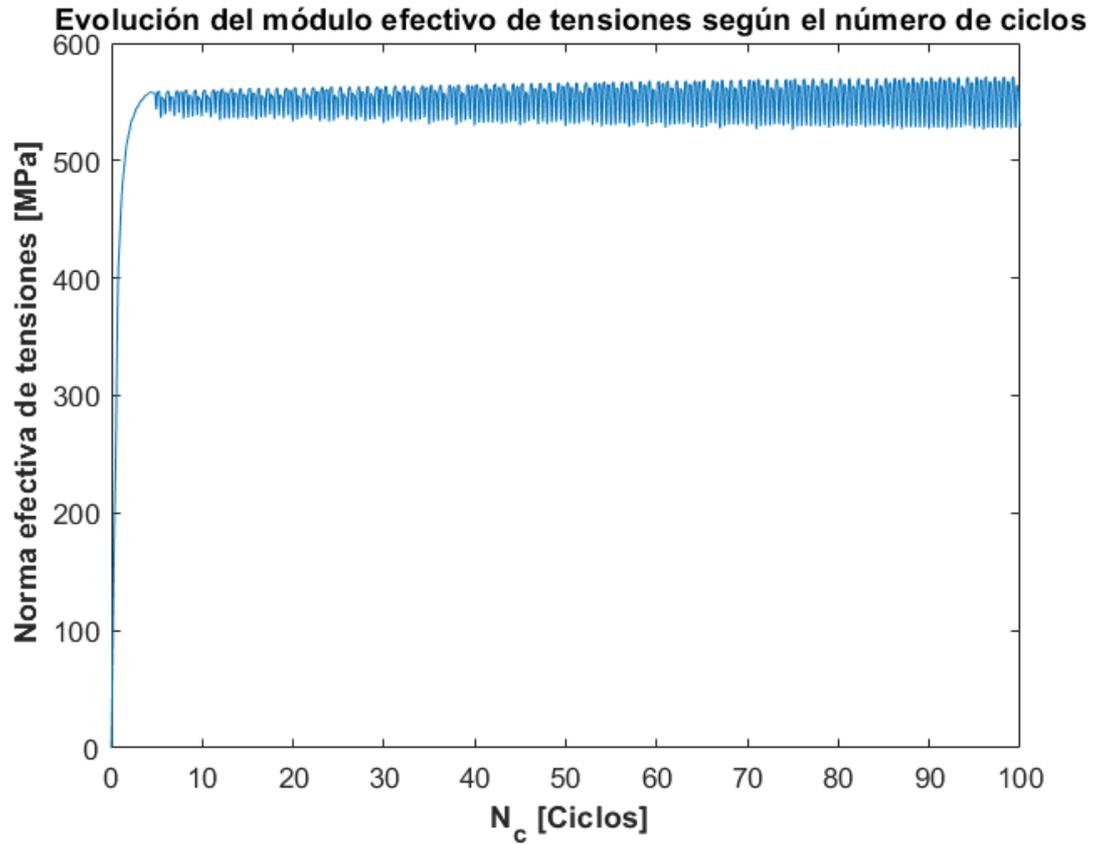


Figura 4.2.2.1.5

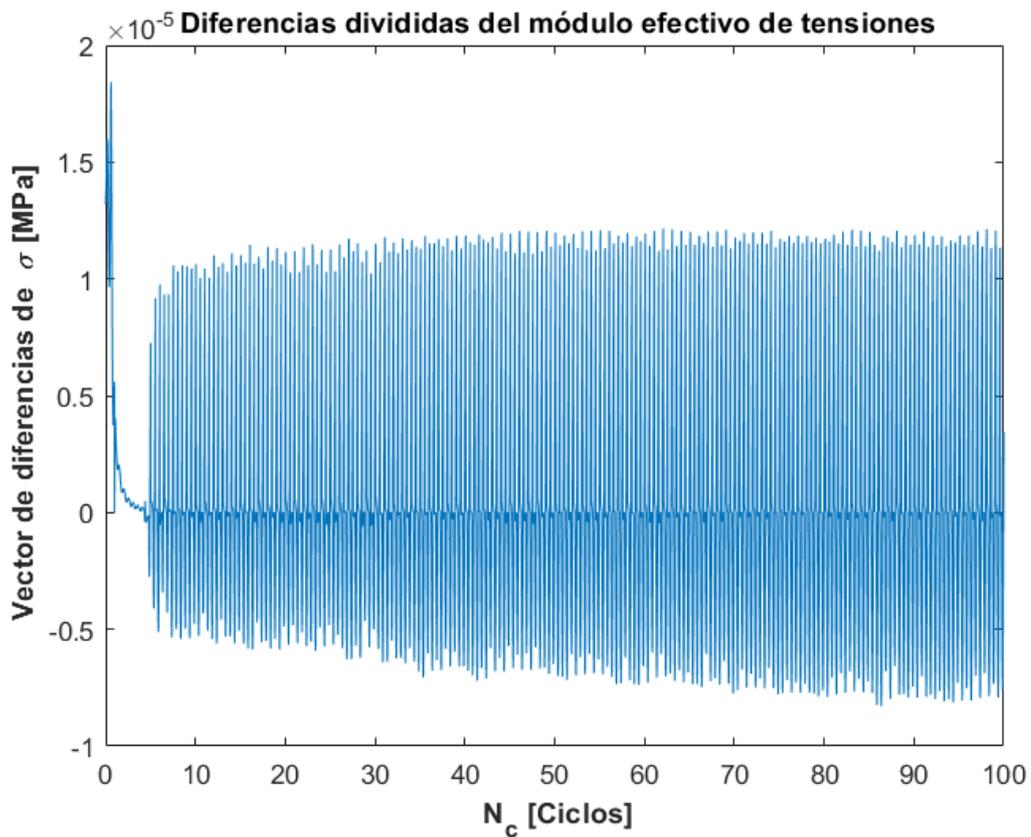


Figura 4.2.2.1.6

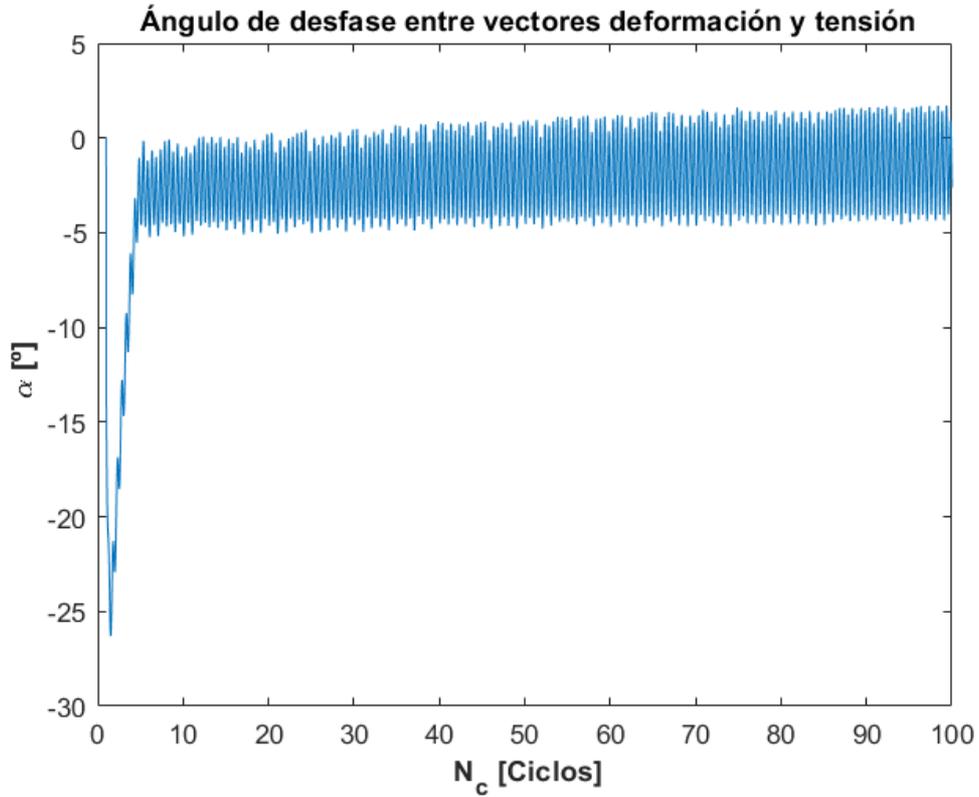


Figura 4.2.2.1.7

4.2.2.2 Resultado para modulo de deformaciones impuesto $|\epsilon_{max}| = 0.014$

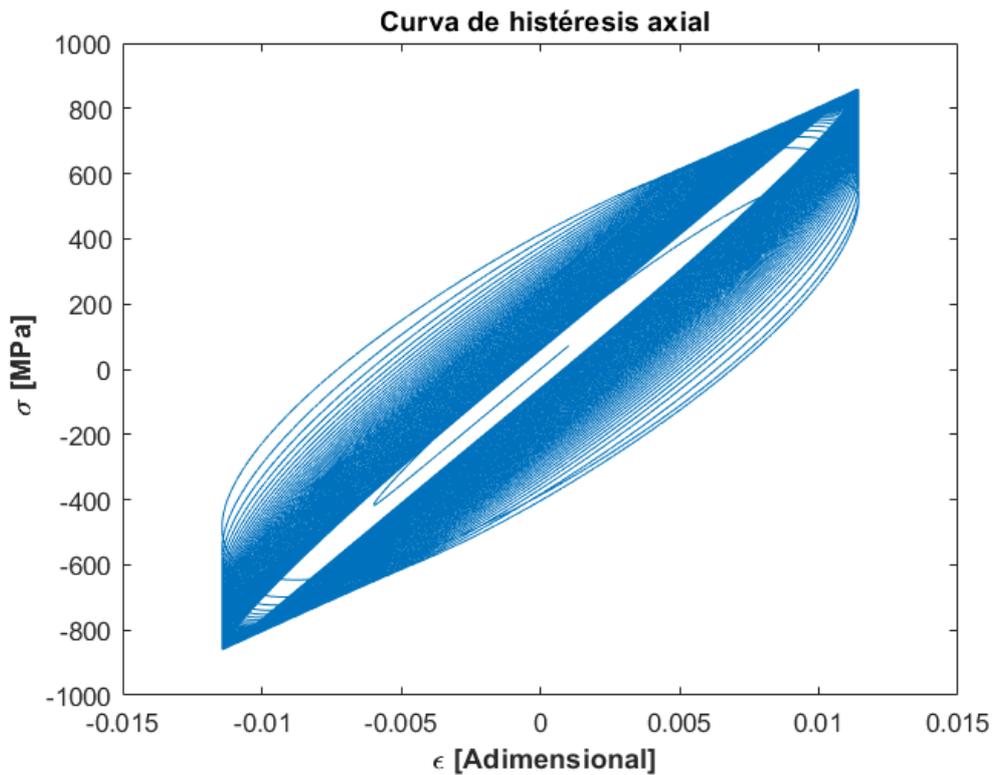


Figura 4.2.2.2.1

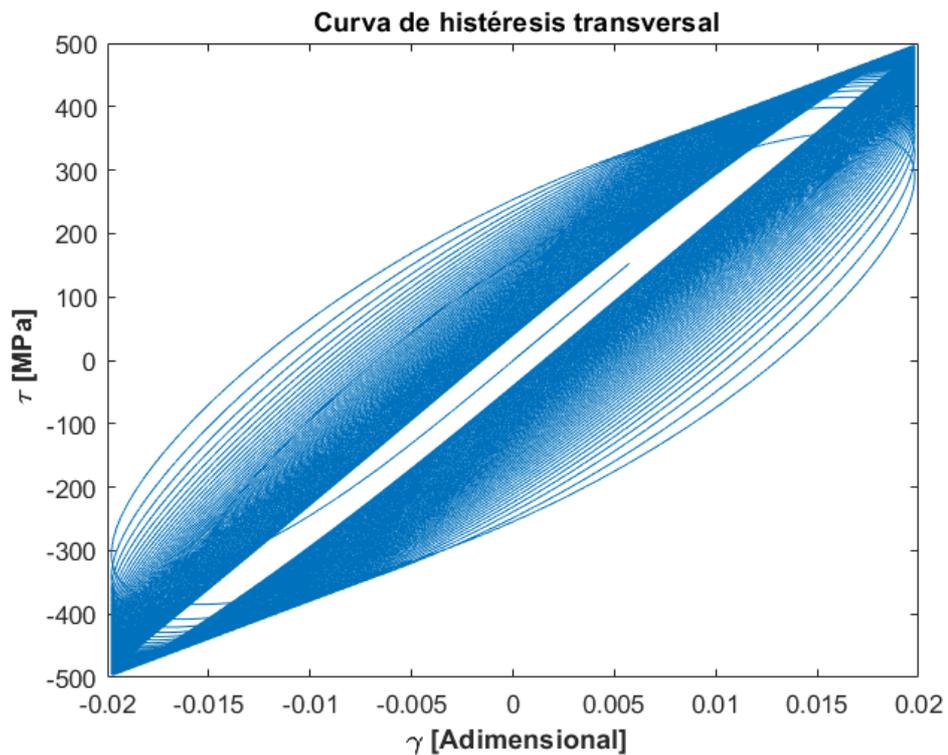


Figura 4.2.2.2.2

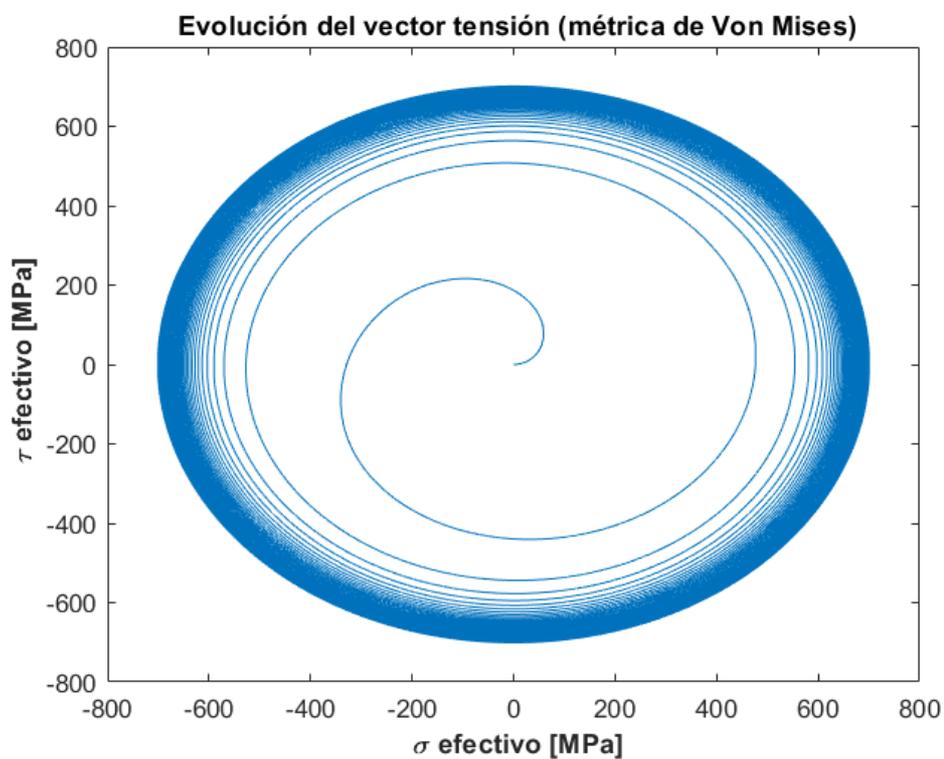


Figura 4.2.2.2.3

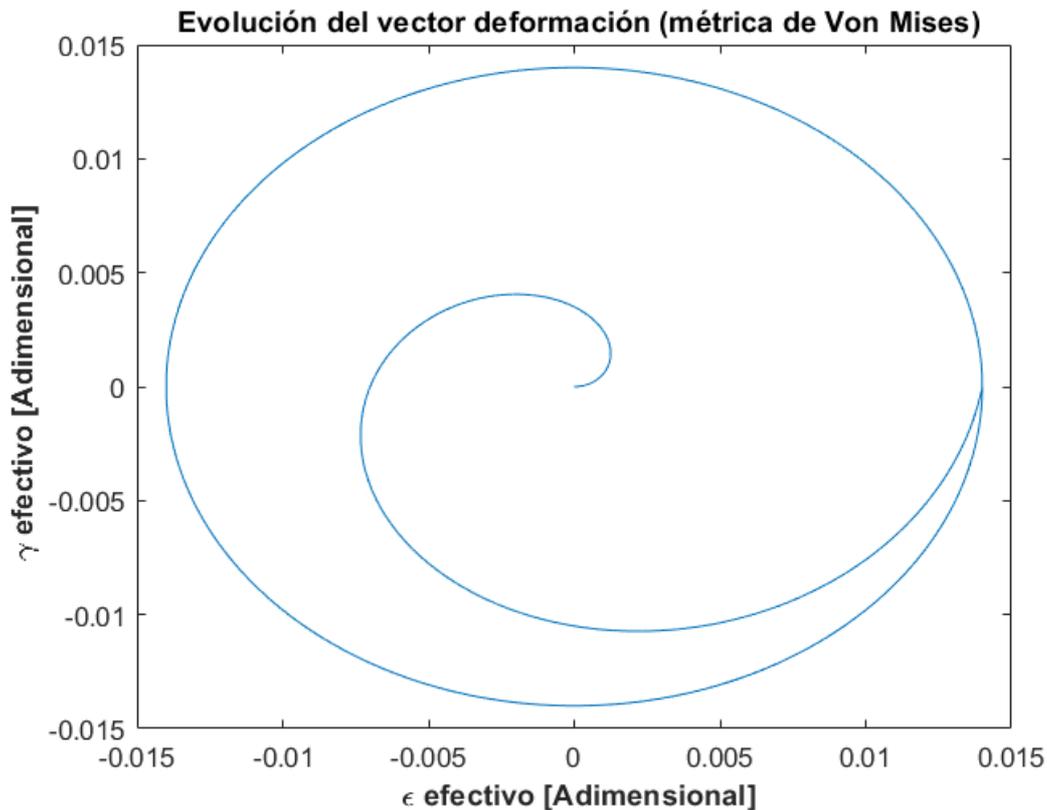


Figura 4.2.2.2.4

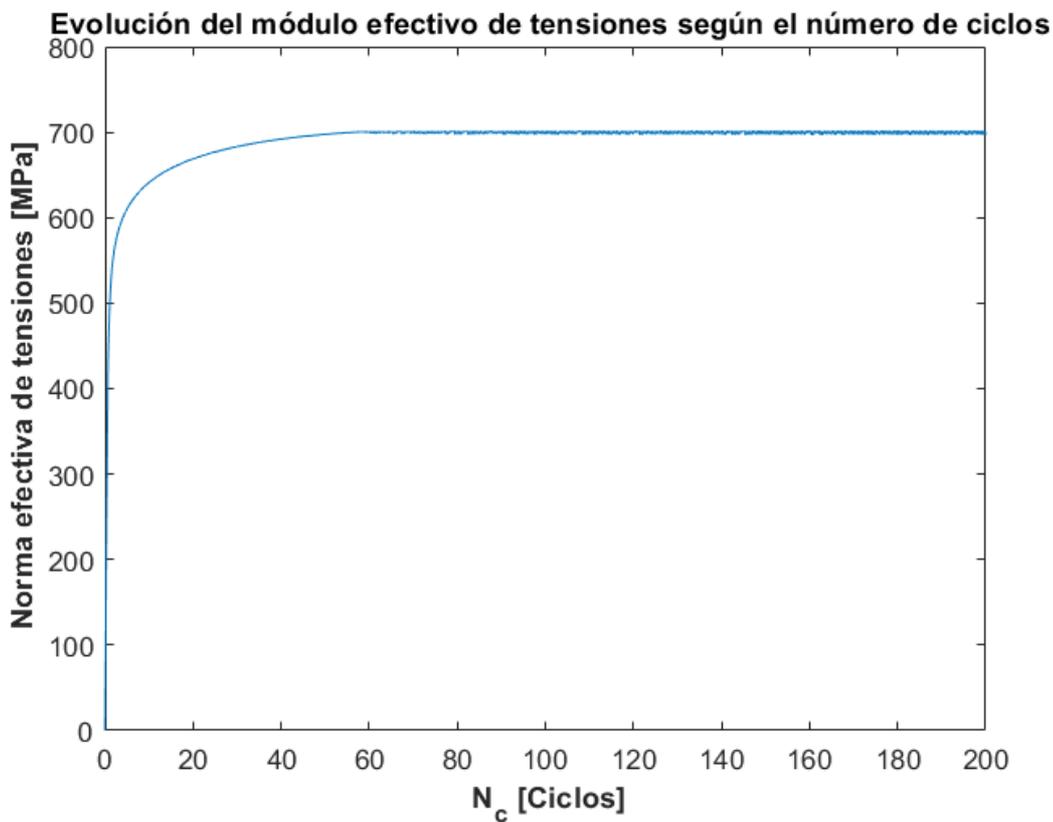


Figura 4.2.2.2.5

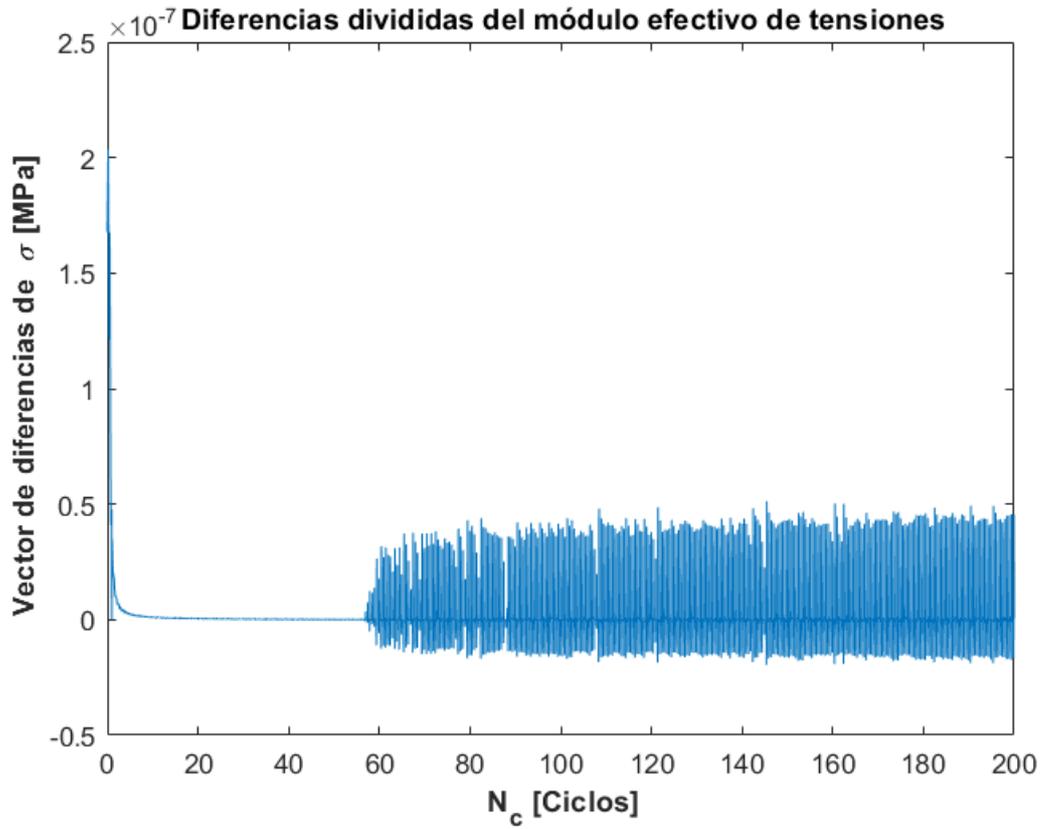


Figura 4.2.2.2.6

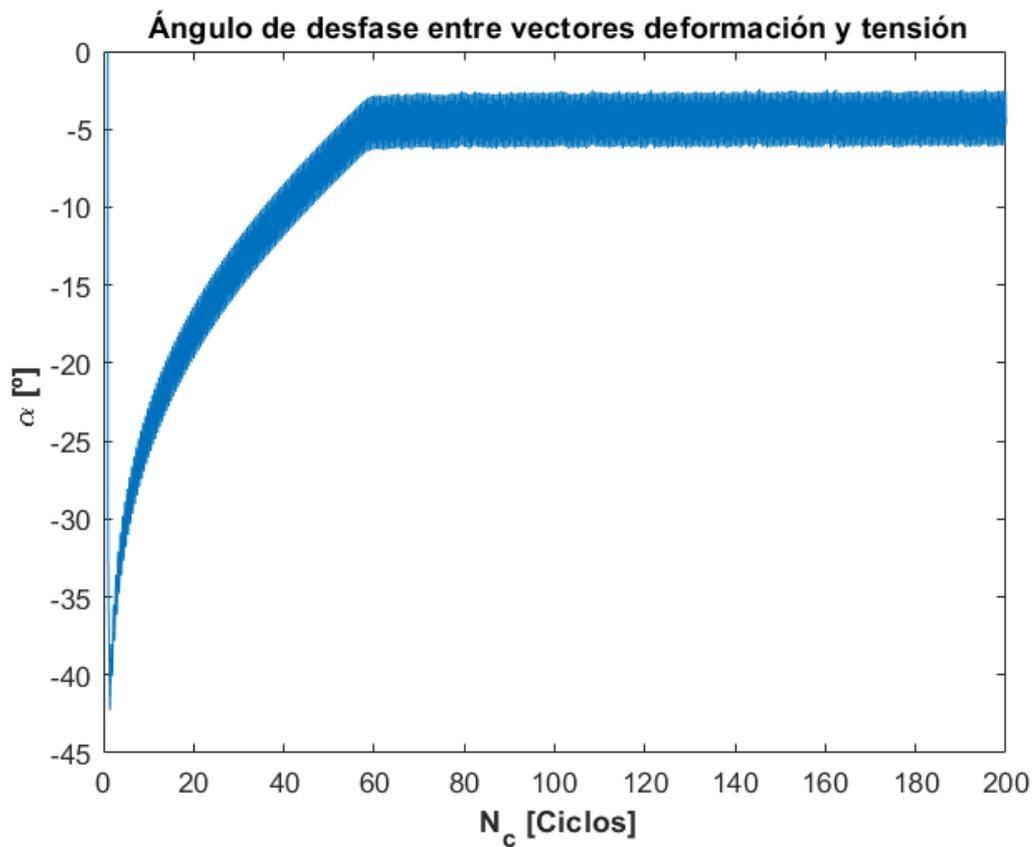


Figura 4.2.2.2.7

4.2.2.3 Resultado para módulo de deformaciones impuesto $|\epsilon_{max}| = 0.017$

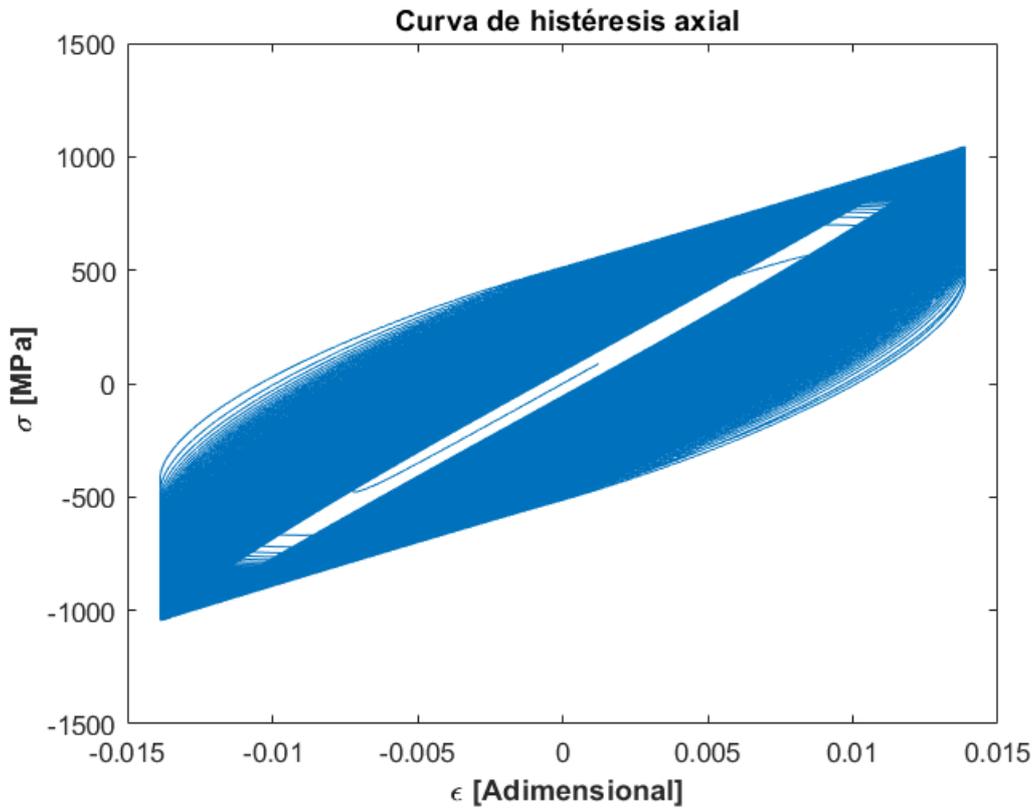


Figura 4.2.2.3.1

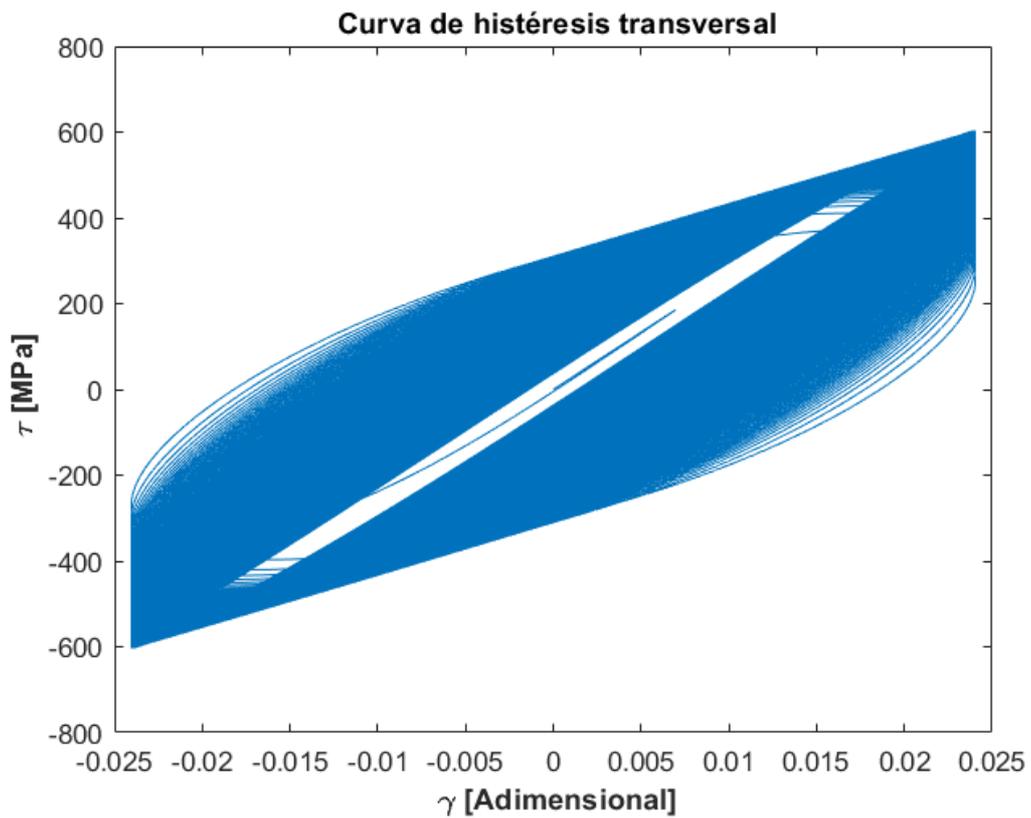


Figura 4.2.2.3.2

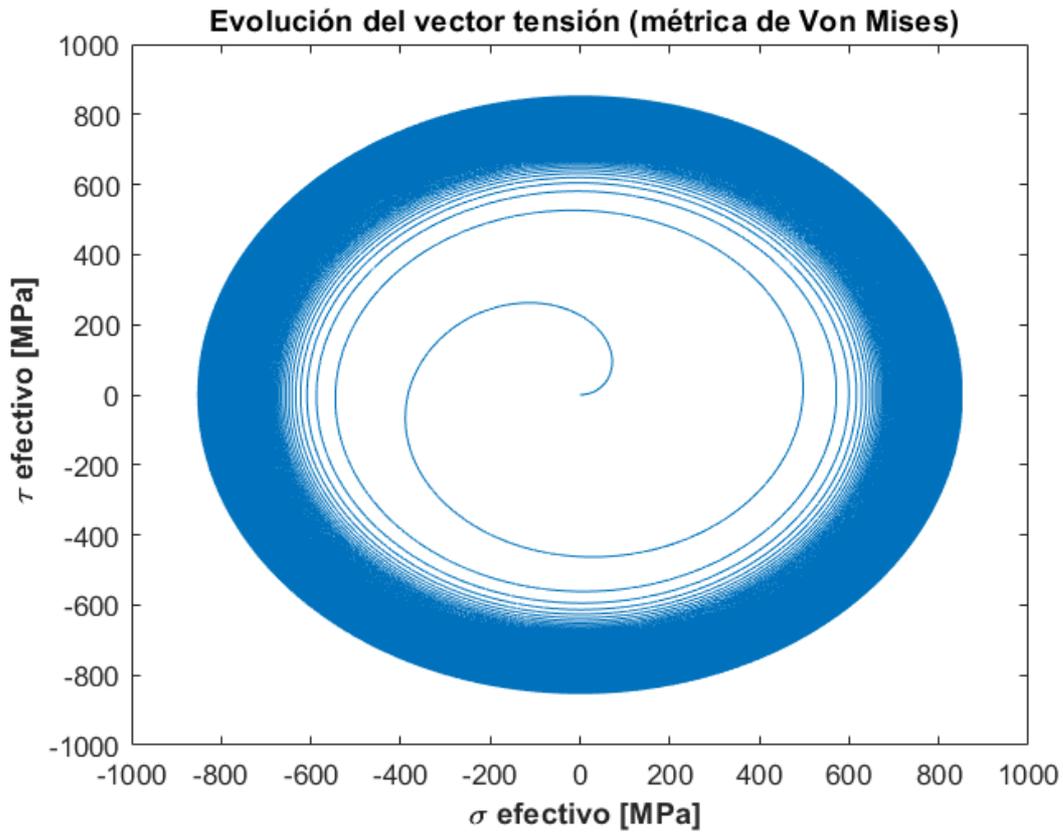


Figura 4.2.2.3.3

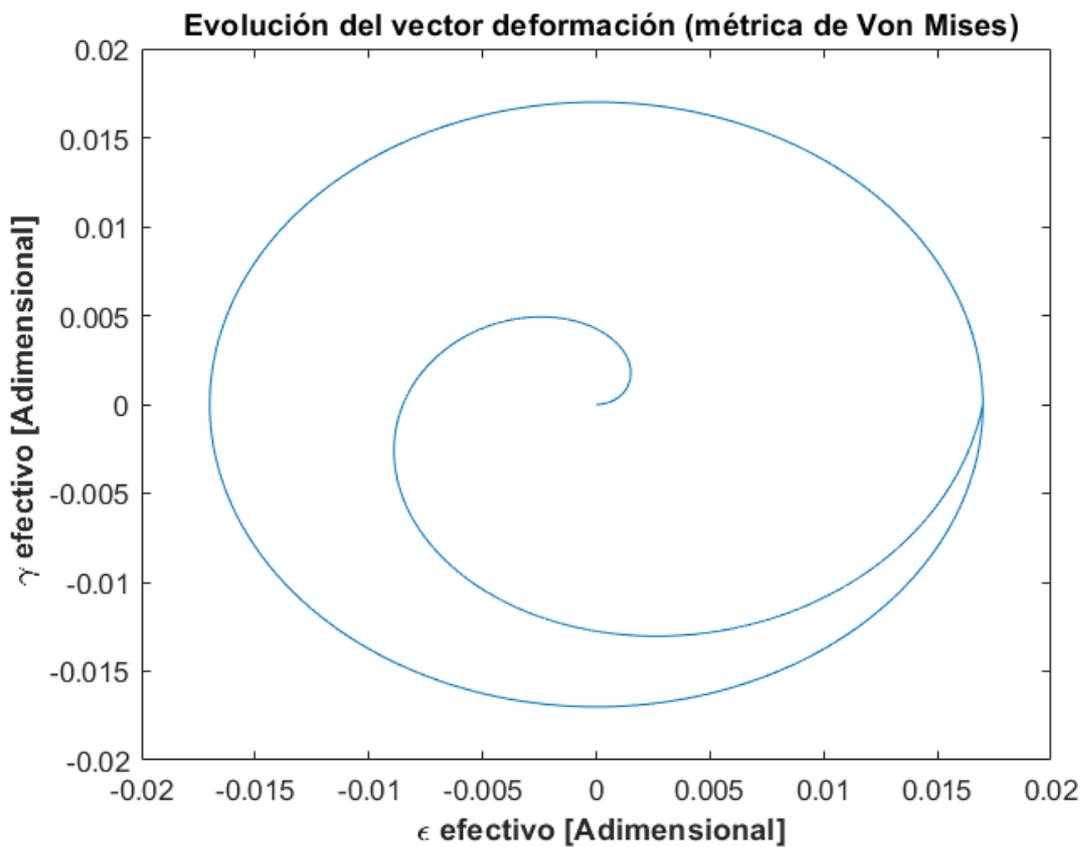


Figura 4.2.2.3.4

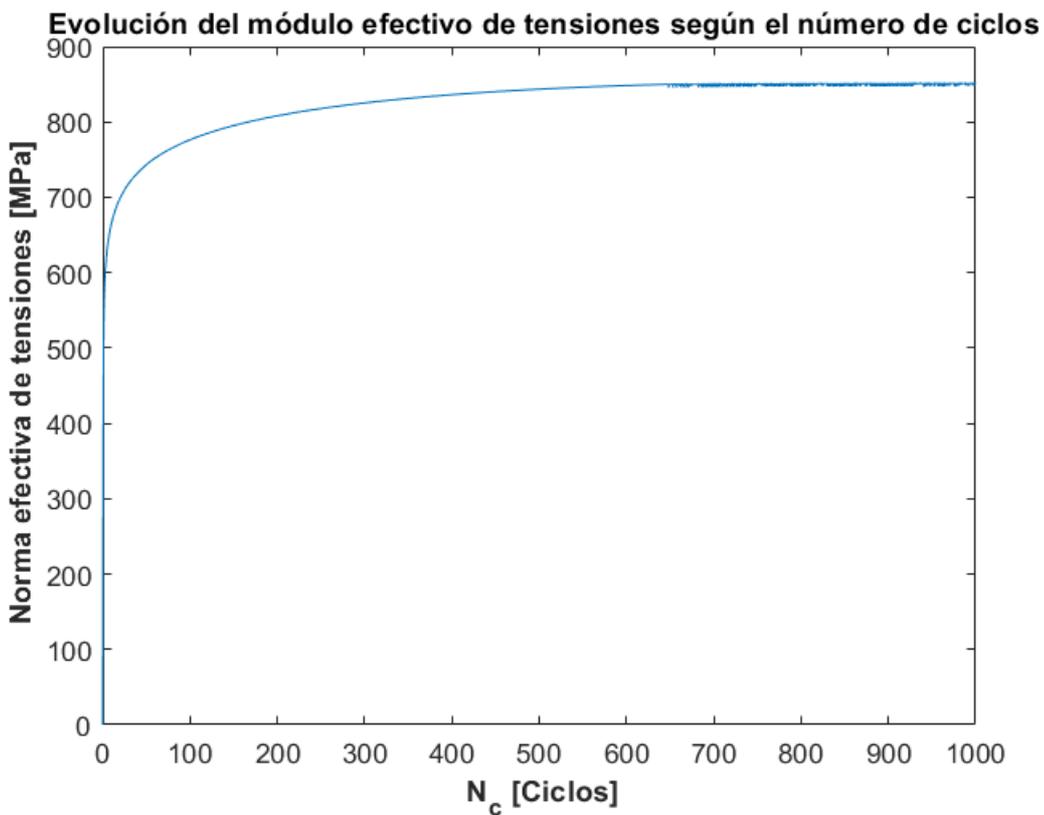


Figura 4.2.2.3.5

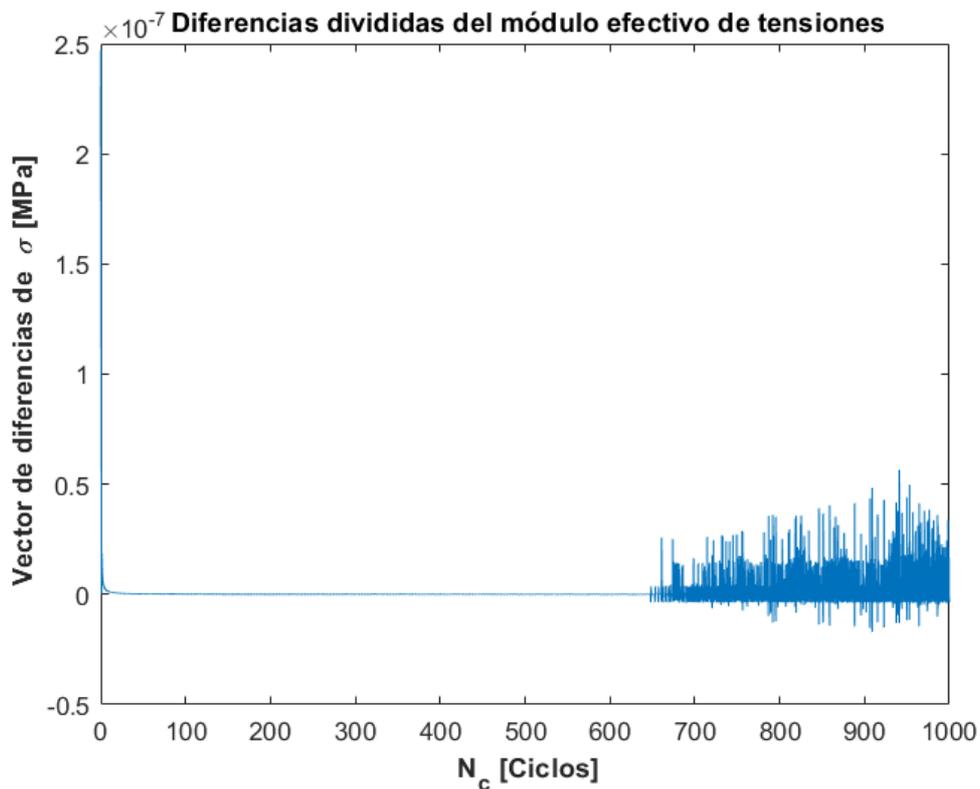


Figura 4.2.2.3.6

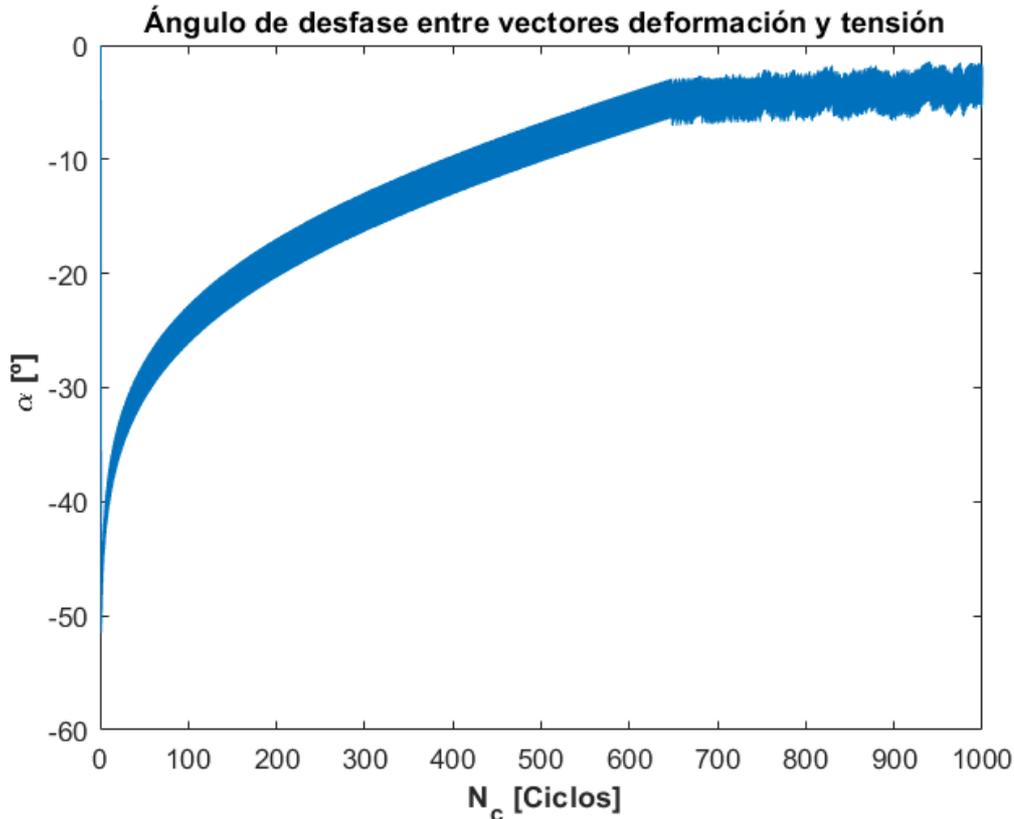


Figura 4.2.2.3.7

4.2.3 Análisis de los resultados

Una vez implementadas las ecuaciones de plasticidad teniendo en cuenta la existencia de puntos de inversión y del efecto memoria, hemos obtenido los resultados según las figuras (4.2.2.1.1) hasta (4.2.2.3.7), manteniendo las mismas condiciones y parámetros de las simulaciones realizadas con el primer modelo de cálculo, que no tenía estos factores. Esto nos permite comparar ambas figuras para determinar el comportamiento de ambos modelos.

El factor más importante de los resultados obtenidos con este modelo es que ahora los resultados no tienen por qué invalidarse por ver un descenso en el módulo efectivo de tensiones $|\sigma|$, o apreciar que en las figuras de diferencias divididas de $|\sigma|$ los valores son negativos. Aunque las previsiones teóricas vistas en los apartados 1 y 2 de este documento expresan que $|\sigma|$ no puede descender debido a que la deformación plástica se mantiene constante; esta vez sí es posible permitimos pequeñas oscilaciones numéricas.

Debemos tener en cuenta que el sistema de ecuaciones diferenciales indicado en la expresión (2.1.14) es complejo y por ello puede estar no exento de comportamientos numéricos indeseados. En los resultados del primer modelo de cálculo podemos apreciar una inestabilidad y un comportamiento errático de las ecuaciones una vez el módulo efectivo de tensiones $|\sigma|$ llega a su máximo y se detecta el primer descenso de este valor. Este descenso hacía invalidar los resultados puesto que dichas ecuaciones no podían calcular verazmente a partir de ahí. Esto cambia con este modelo de cálculo.

Ahora puede verse en las figuras (4.2.2.1.5), (4.2.2.2.5) y (4.2.2.3.5) que indican la evolución del valor $|\sigma|$ respecto al número de ciclos totales transcurridos N_c , que el valor de $|\sigma|$ llega a estabilizarse, con oscilaciones debido a un comportamiento numérico indeseado. Estas oscilaciones se producen tanto para deformaciones bajas como para deformaciones altas. Sin embargo, esto no invalida los resultados al haber tenido en cuenta las ecuaciones de descarga.

Hay que tener en cuenta que hemos definido un paso de integración no muy bajo $h = 10^{-4}$ debido al elevado consumo de recursos que requiere implementar un modelo de integración paso a paso para poder almacenar los

sucesivos puntos de descarga σ_k y sus circunferencias de plastificación asociadas C_k . Este paso de integración puede ser el responsable de las oscilaciones numéricas, siendo necesaria una simplificación de estas ecuaciones, para poder realizar un análisis con mayor profundidad.

4.3 Tercer modelo de cálculo. Simplificación de ecuaciones de descarga

Volviendo al apartado 4.1.3, si verificamos que $|\sigma|$ se estabiliza, lo que ocurre, pero con oscilaciones de naturaleza numérica; no existe deformación plástica, puesto que según la expresión 1.11 vemos que el incremento de deformación plástica $d\varepsilon_p$ es directamente proporcional al incremento del módulo efectivo de tensiones $d|\sigma|$, el cual es 0 a partir de la estabilización de $|\sigma|$. Entonces podríamos decir que a partir de la estabilización de $|\sigma|$ el comportamiento es elástico al no existir plastificación. Las figuras (4.2.2.1.1), (4.2.2.1.2), (4.2.2.2.1), (4.2.2.2.2), (4.2.2.3.1), (4.2.2.3.2), nos indican que la órbita de estabilización es casi una línea recta, lo cual sería un comportamiento plástico perfecto. Esto se aprecia mejor utilizando el comando *comet* para las figuras en vez del comando *plot*, lo que nos permite apreciar el trazado de las figuras en tiempo real. Recordemos que, en el primer modelo de cálculo, el máximo de $|\sigma|$ es el punto donde los ciclos de histéresis se estrechan al máximo y a partir de ahí $|\sigma|$ comienza a descender.

Esto sugiere la siguiente simplificación, con la cual no necesitaríamos utilizar las ecuaciones de descarga, en las cuales no podemos introducir un paso de integración más pequeño y en las cuales el tiempo de ejecución y un consumo de recursos es altísimo.

Teniendo en cuenta esto último, podemos utilizar el primer modelo de cálculo, pero con un condicionante: puesto que $|\sigma|$ comenzará a descender por comportamientos numéricos indeseados una vez alcanzado el máximo, que coincide con el valor de estabilización; una vez se detecte que $d|\sigma| < 0$ en un paso de integración, sustituiremos el sistema de ecuaciones (2.1.14) por el (4.1.3), el cual indica un comportamiento elástico, sólo para este paso de integración.

La diferencia de esto con el segundo método de cálculo es que este pequeño algoritmo no necesita un sistema de integración paso a paso a mano, sino que se puede realizar utilizando las subrutinas de integración que nos ofrece el paquete MATLAB, las cuales están perfectamente optimizadas para dar unos resultados rápidos y precisos, como es el caso de la función *ode45* utilizada en el primer modelo de cálculo. Esto, aparentemente, nos permitirá obtener unos resultados con mejor comportamiento numérico y con un tiempo de ejecución indiscutiblemente más rápido.

4.3.1 Parametrización

La parametrización de este modelo es muy sencilla una vez definida la del primer modelo de cálculo, ya que se basa estrictamente en la de este.

Sólo debemos introducir un pequeño cambio, en el script que definimos el sistema de ecuaciones diferenciales (2.1.14), *syst.m*, y *syst2.m* para el primer modelo, introduciremos la condición de que si $d|\sigma|$ es negativo en algún paso de integración, en dicho paso de integración sustituiremos las ecuaciones (2.1.14) por las de la expresión (4.1.3), siendo este paso de integración puramente elástico. Se crean a su vez los ficheros *systcl.m* y *syst2cl.m* en el que se incluye este condicionante. Utilizaremos la subrutina de integración *ode45* la cual trabajará con estas dos funciones, para el primer y segundo tramos, respectivamente.

Estableceremos las simulaciones para los mismos valores que en el apartado 4.1, para $|\varepsilon_{max}| = 0.011$, $|\varepsilon_{max}| = 0.014$ y $|\varepsilon_{max}| = 0.017$.

Para utilizar este modelo de cálculo, configuraremos el fichero *espiral.cl* de la siguiente forma:

- “*ecuacion=3*” para utilizar el primer modelo de cálculo.
- “*esp=1*” para aproximar el primer tramo en espiral, el cual se utilizará durante toda la simulación.
- “*material=1*” para tomar los parámetros del Aluminio 7075 T651
- “*N_C = 100*” para calcular 100 ciclos en cada iteración. Esto se cambiará según los resultados para un mejor muestreo. Es decir, para $|\varepsilon_{max}| = 0.014$ evaluaremos 200 ciclos y para $|\varepsilon_{max}| = 0.017$ 1000 ciclos.
- “ $|\varepsilon_{max}| = \text{VALOR DE TABLA A EVALUAR}$ ”

Los resultados que se representarán serán análogos a los vistos anteriormente en los apartados 4.1 y 4.2, para poder hacer una comparación efectiva sobre estos.

4.3.2 Resultados de las simulaciones

4.3.2.1 Resultado para módulo de deformaciones impuesto $|\epsilon_{max}| = 0.011$

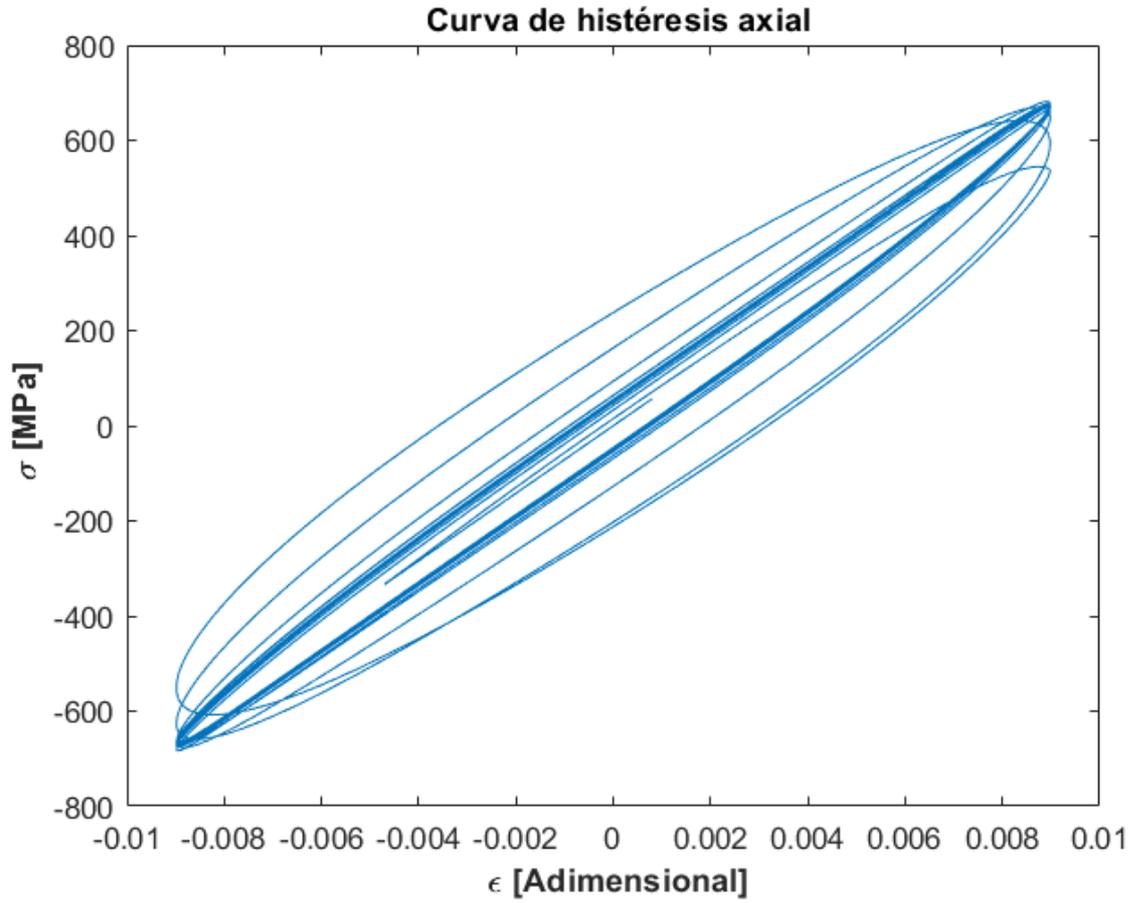


Figura 4.3.2.1.1

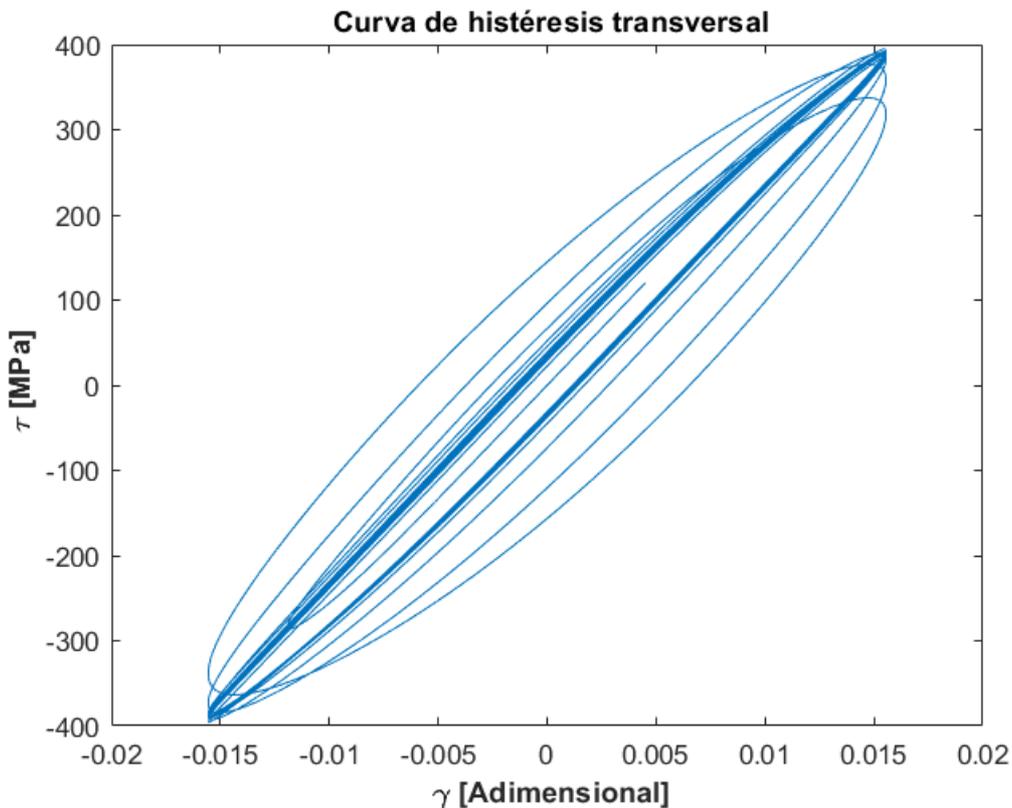


Figura 4.3.2.1.2

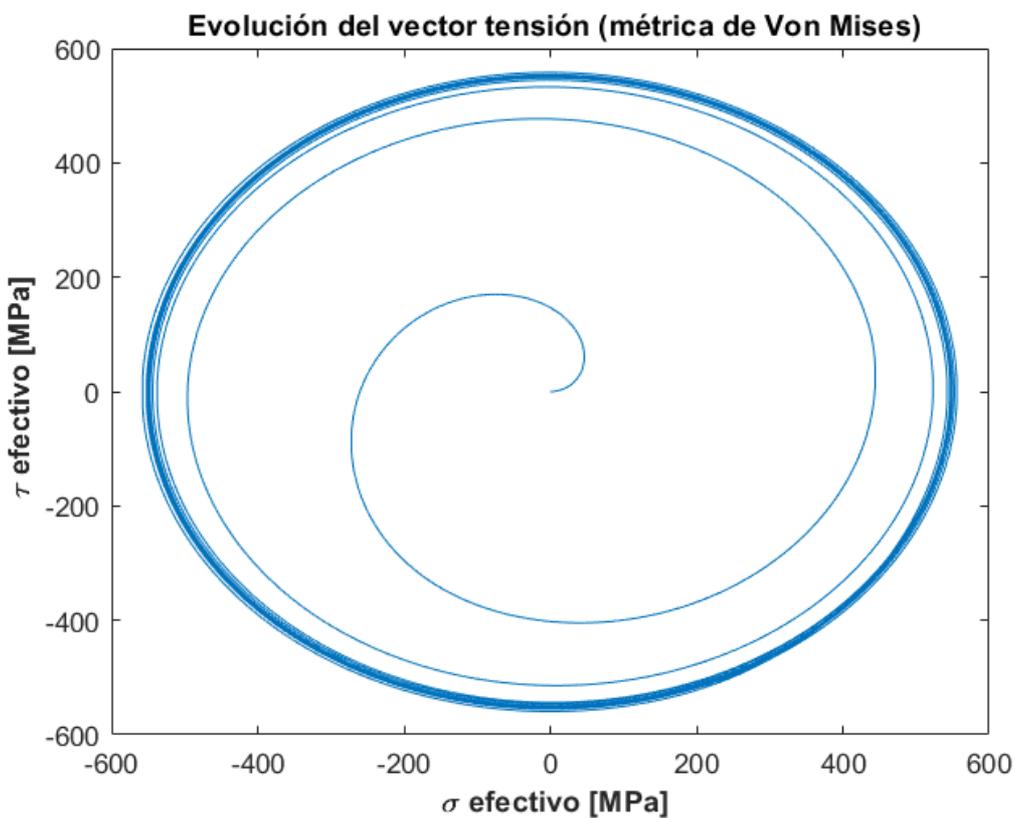


Figura 4.3.2.1.3

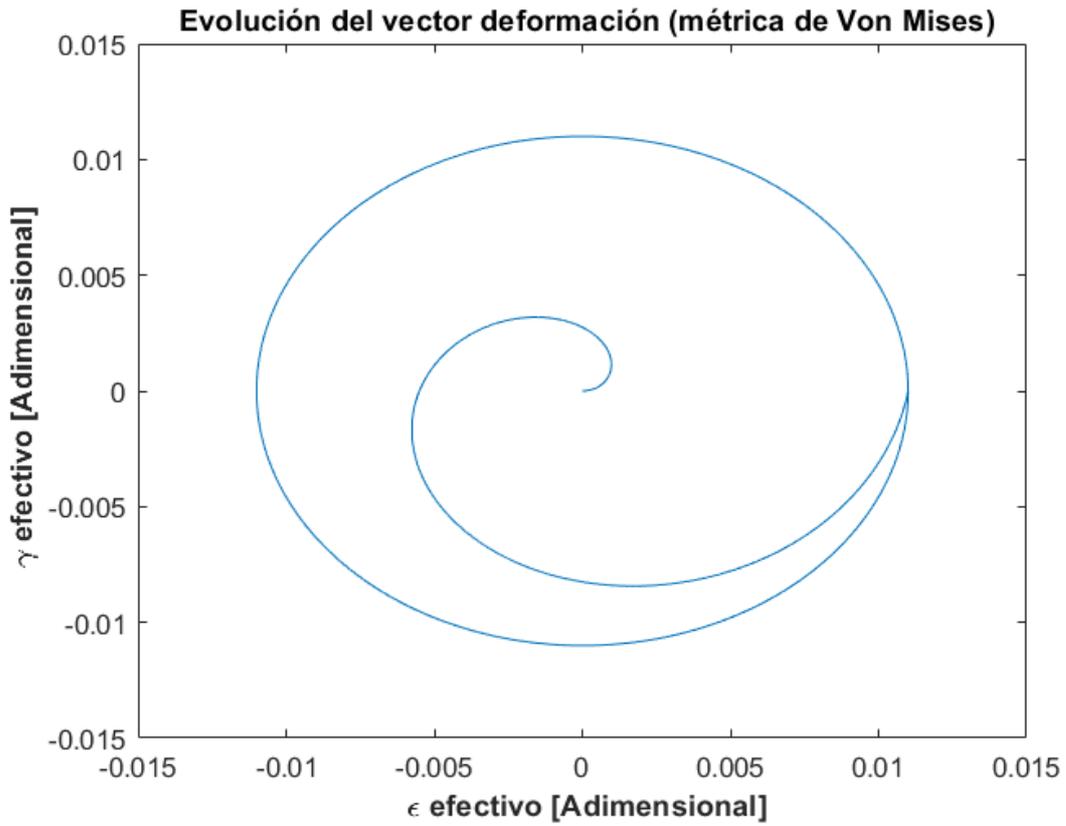


Figura 4.3.2.1.4

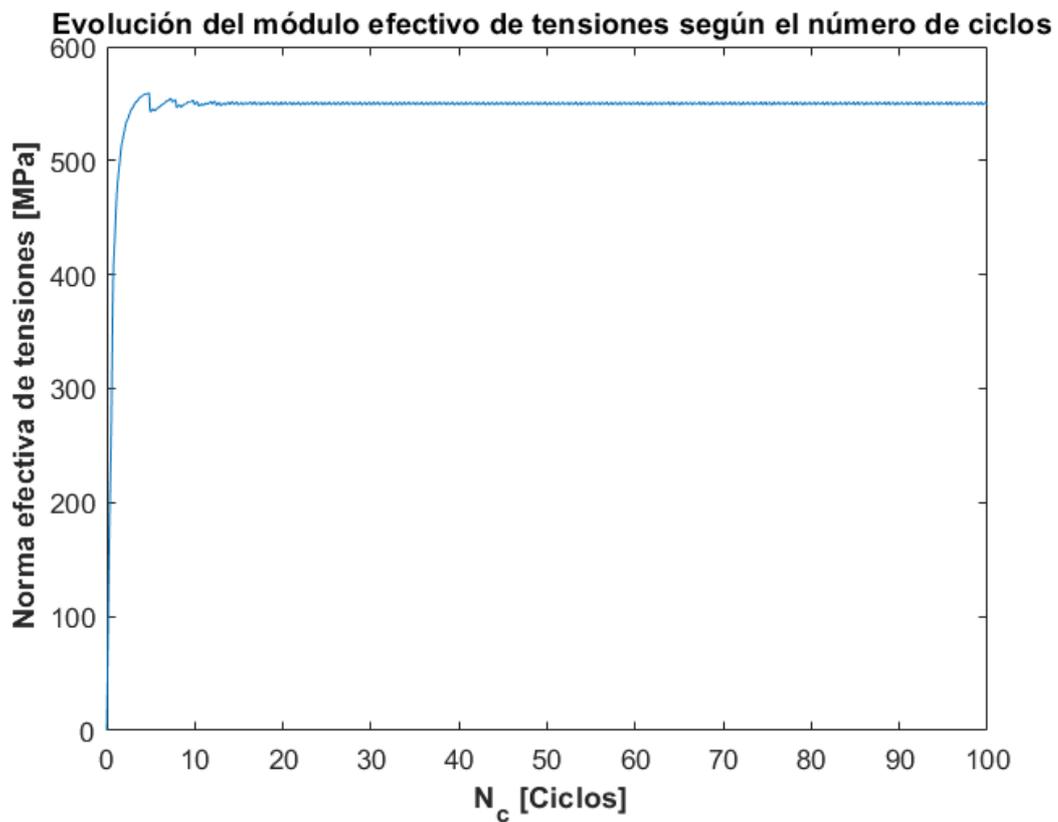


Figura 4.3.2.1.5

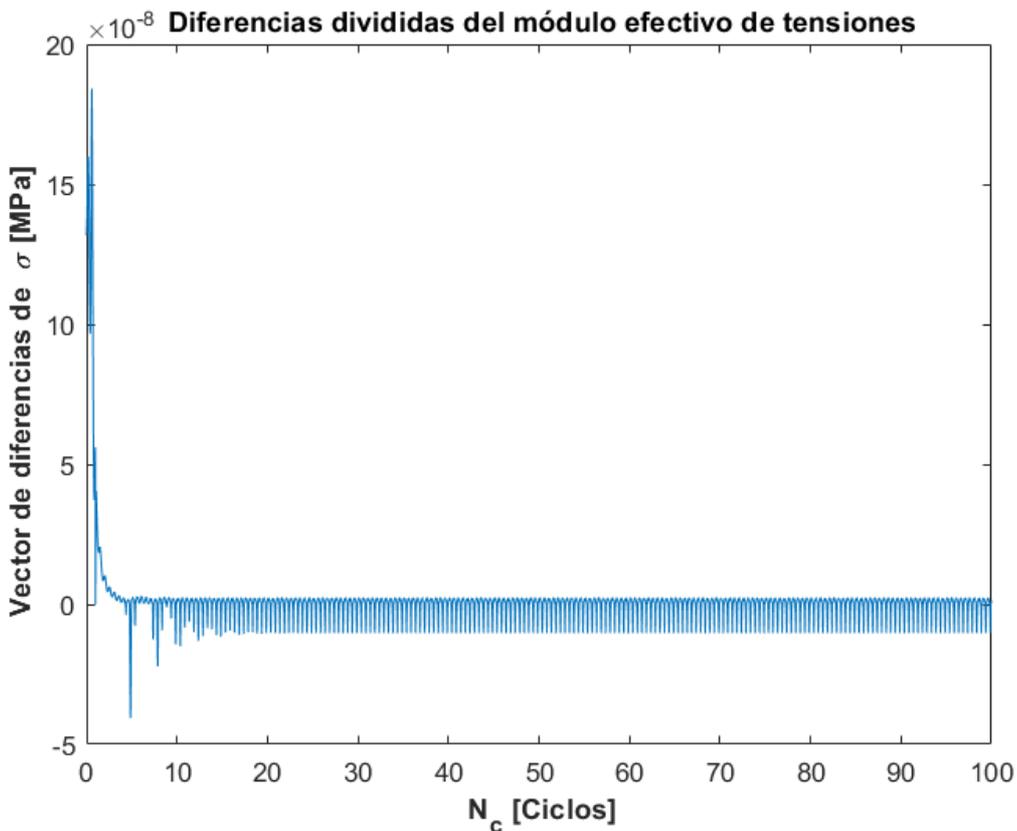


Figura 4.3.2.1.6

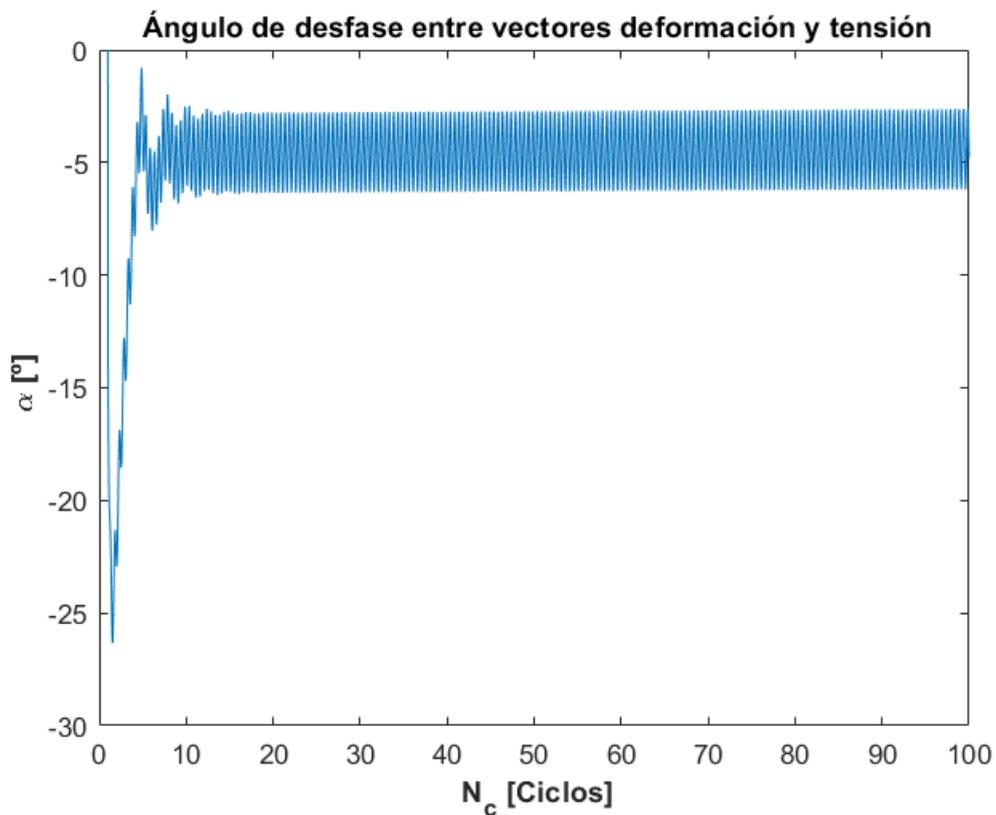


Figura 4.3.2.1.7

4.3.2.2 Resultados para módulo de deformaciones impuesto $|\varepsilon_{max}| = 0.014$

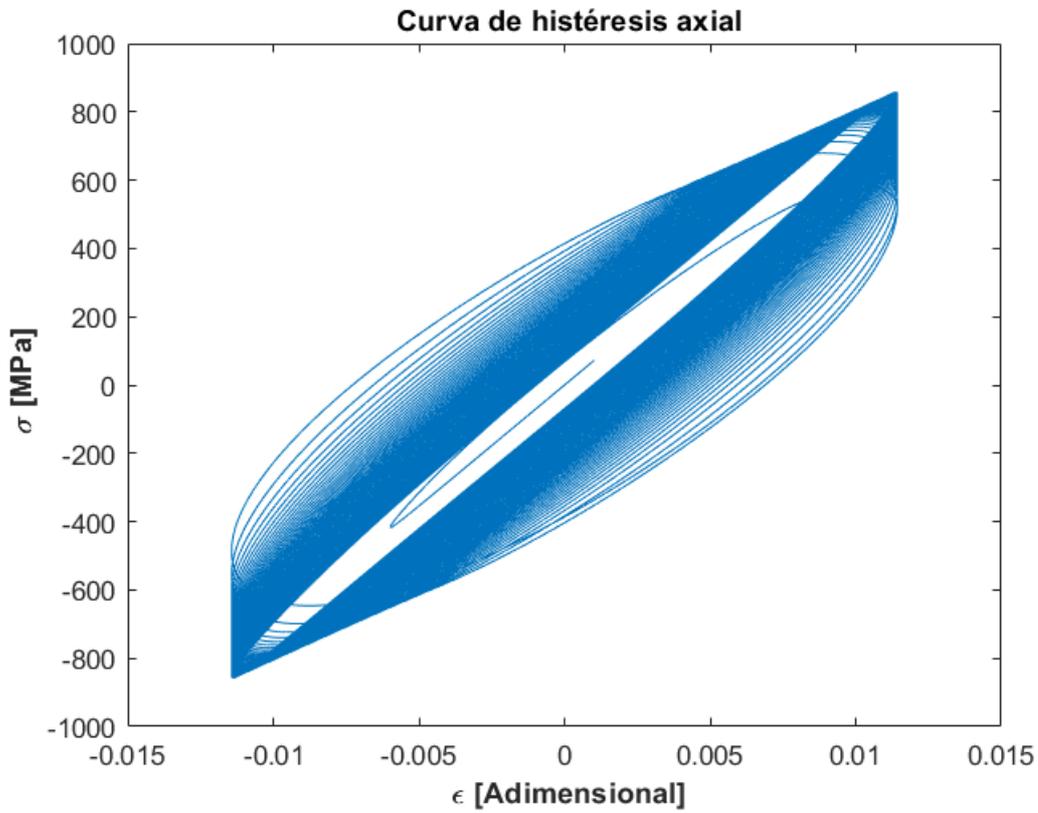


Figura 4.3.2.2.1

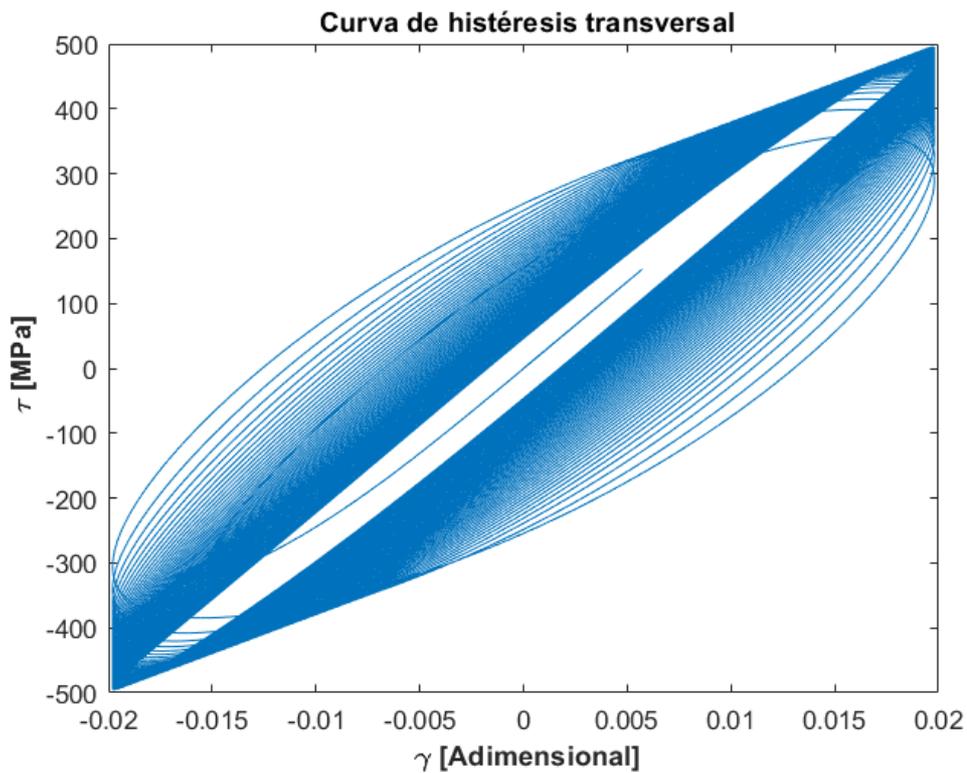


Figura 4.3.2.2.2

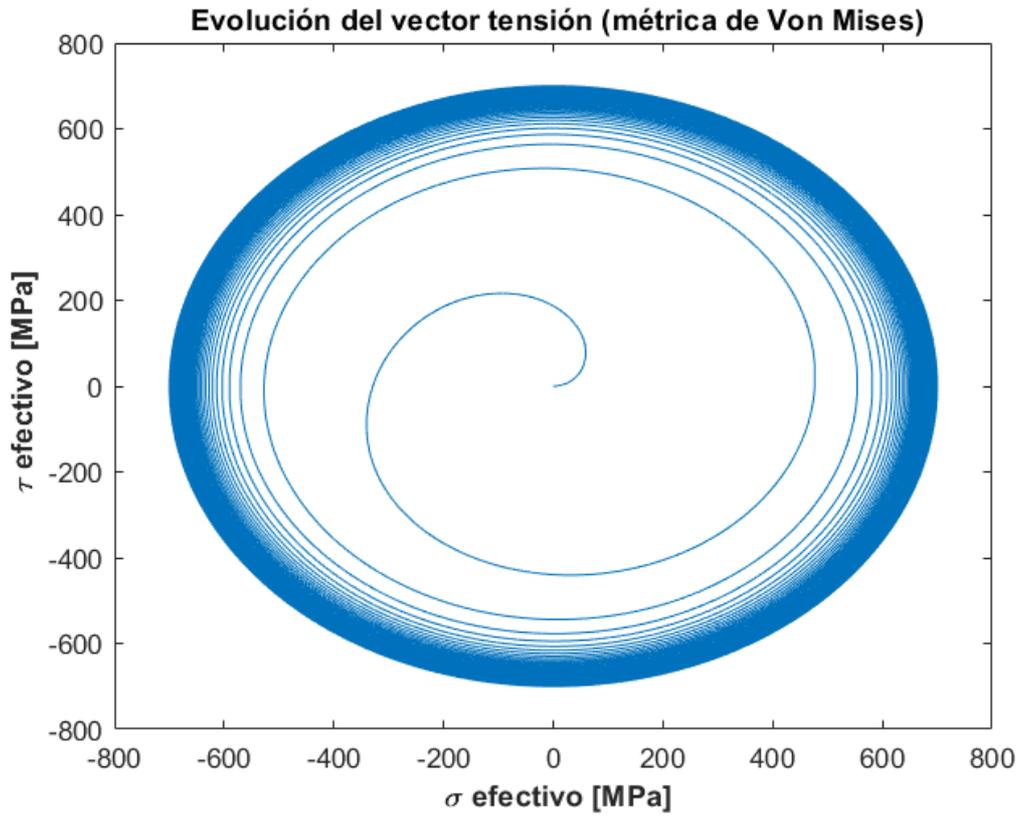


Figura 4.3.2.2.3

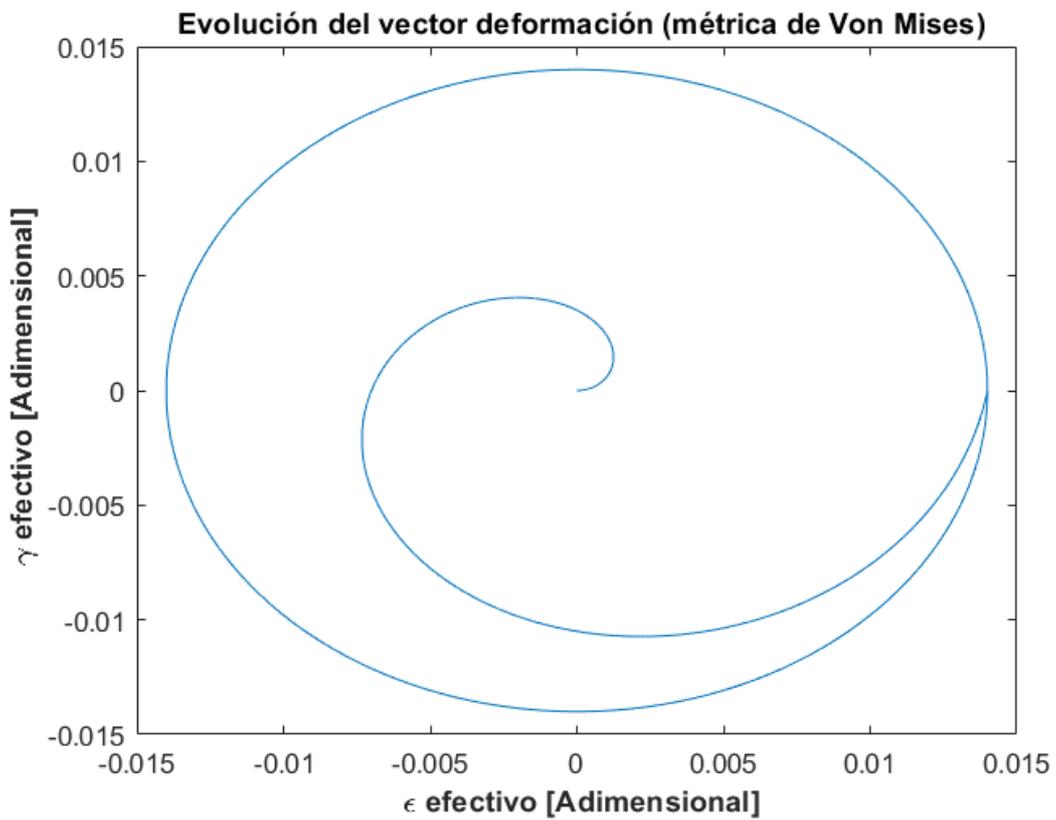


Figura 4.3.2.2.4

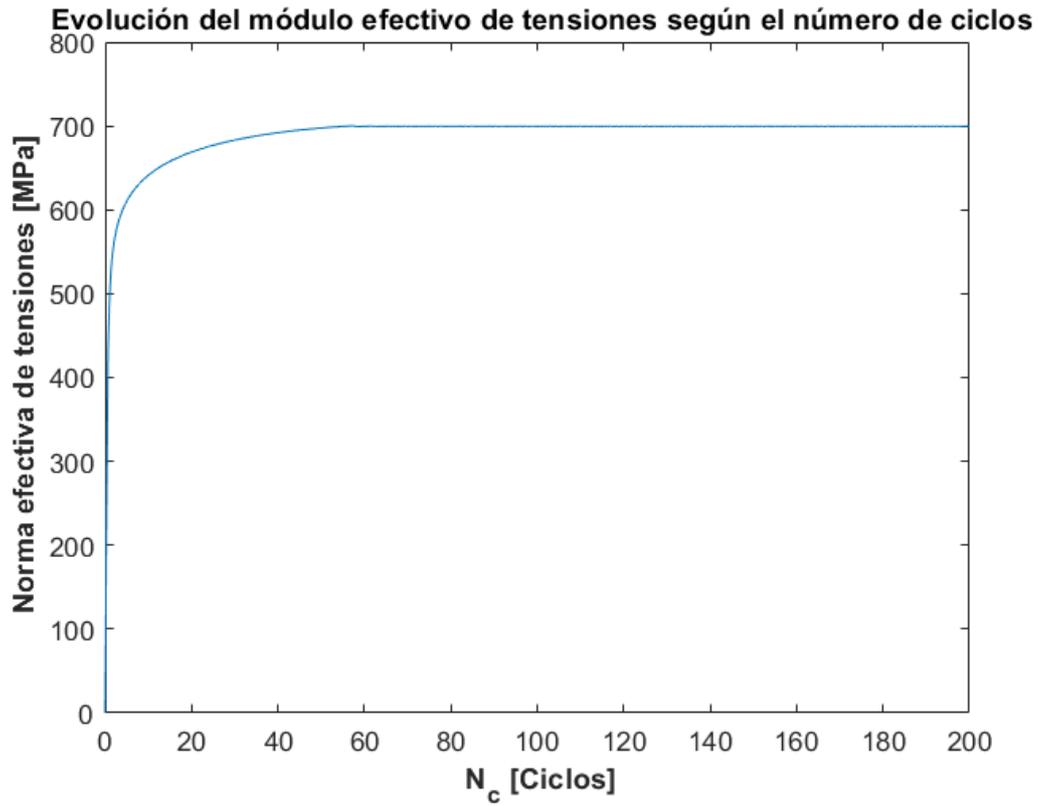


Figura 4.3.2.2.5

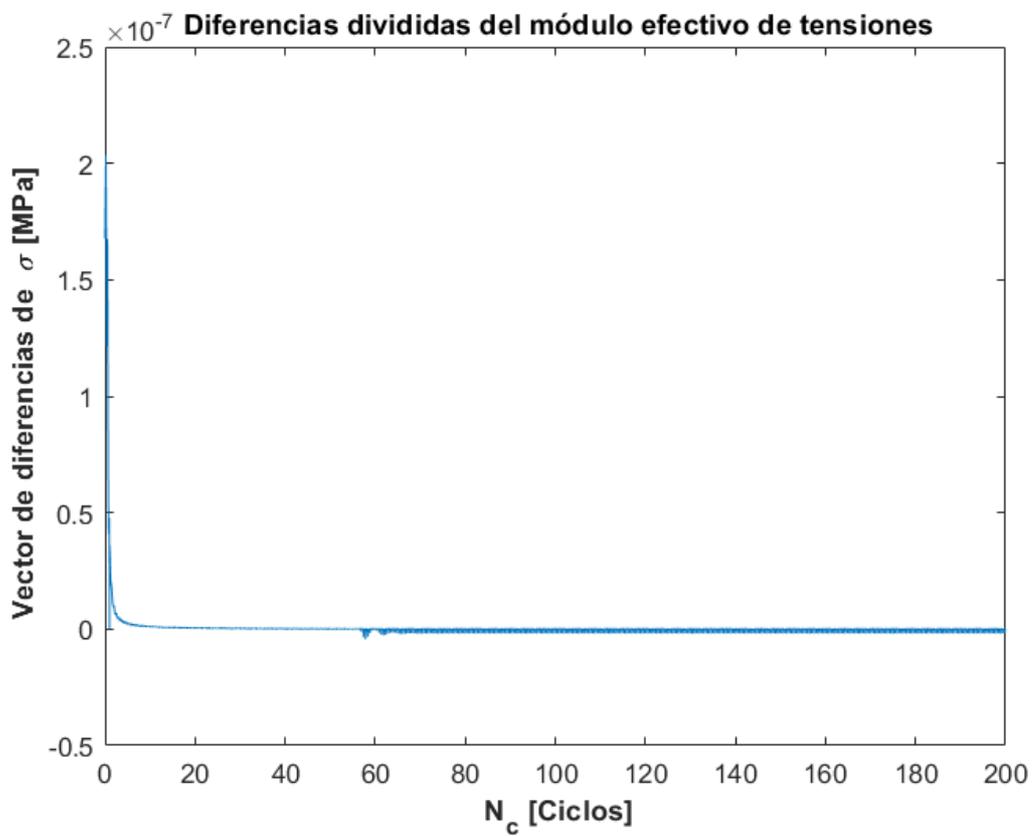


Figura 4.3.2.2.6

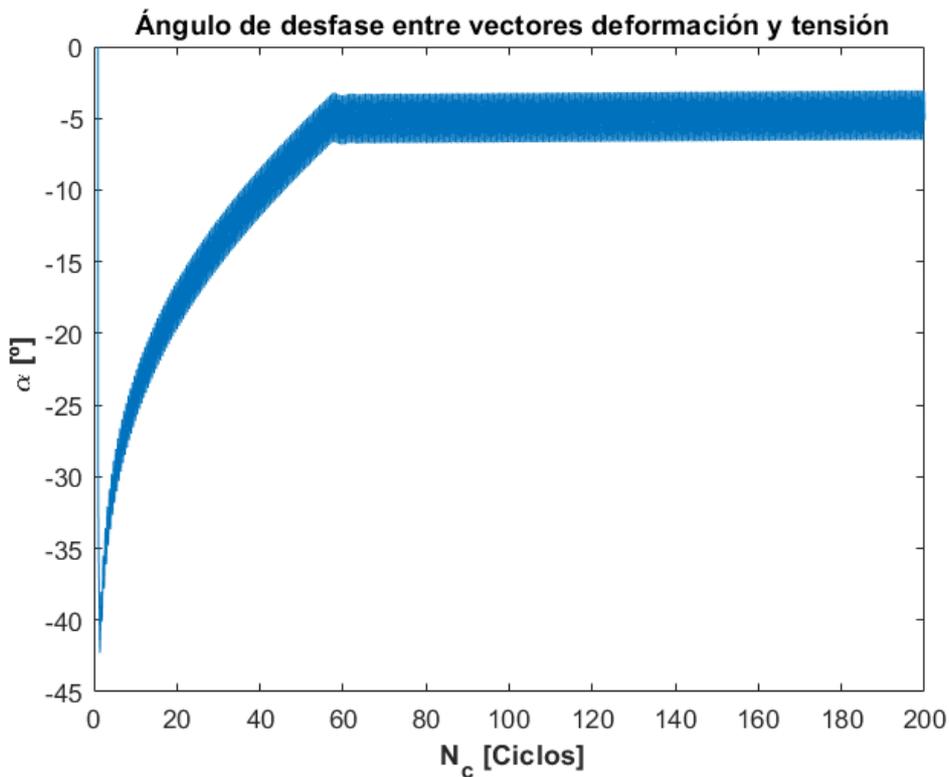


Figura 4.3.2.2.7

4.3.2.3 Resultados para módulo de deformaciones impuesto $|\epsilon_{max}| = 0.017$

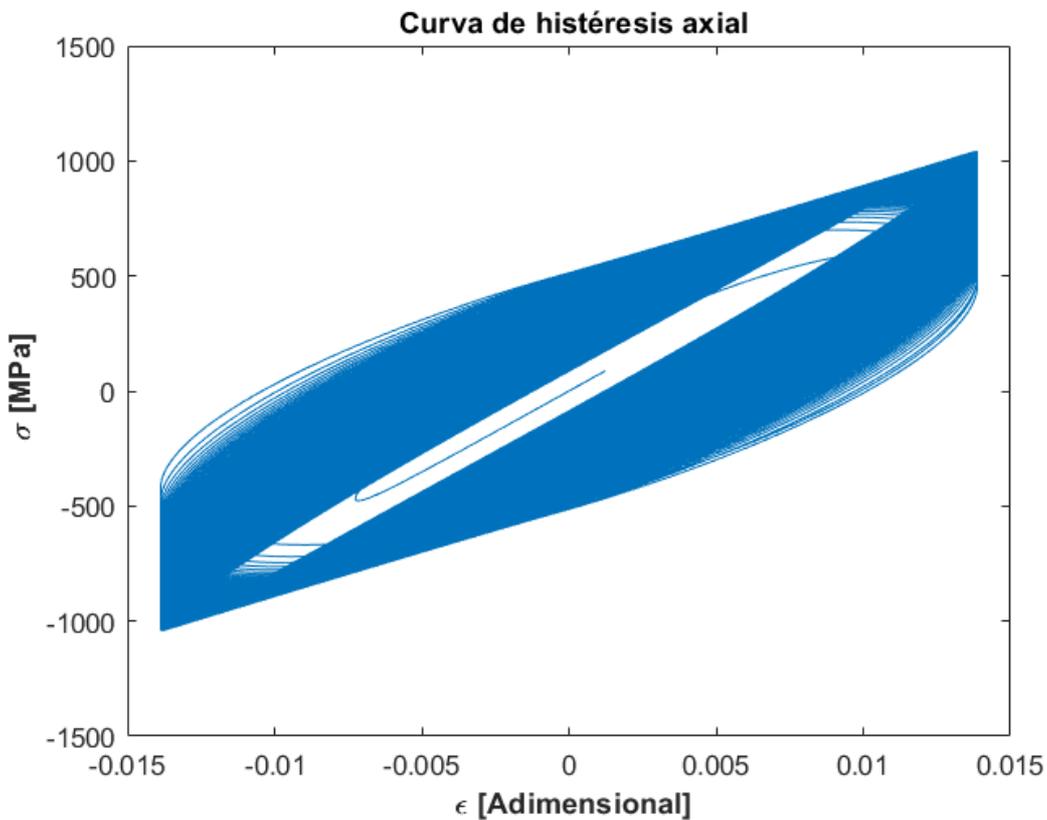


Figura 4.3.2.3.1

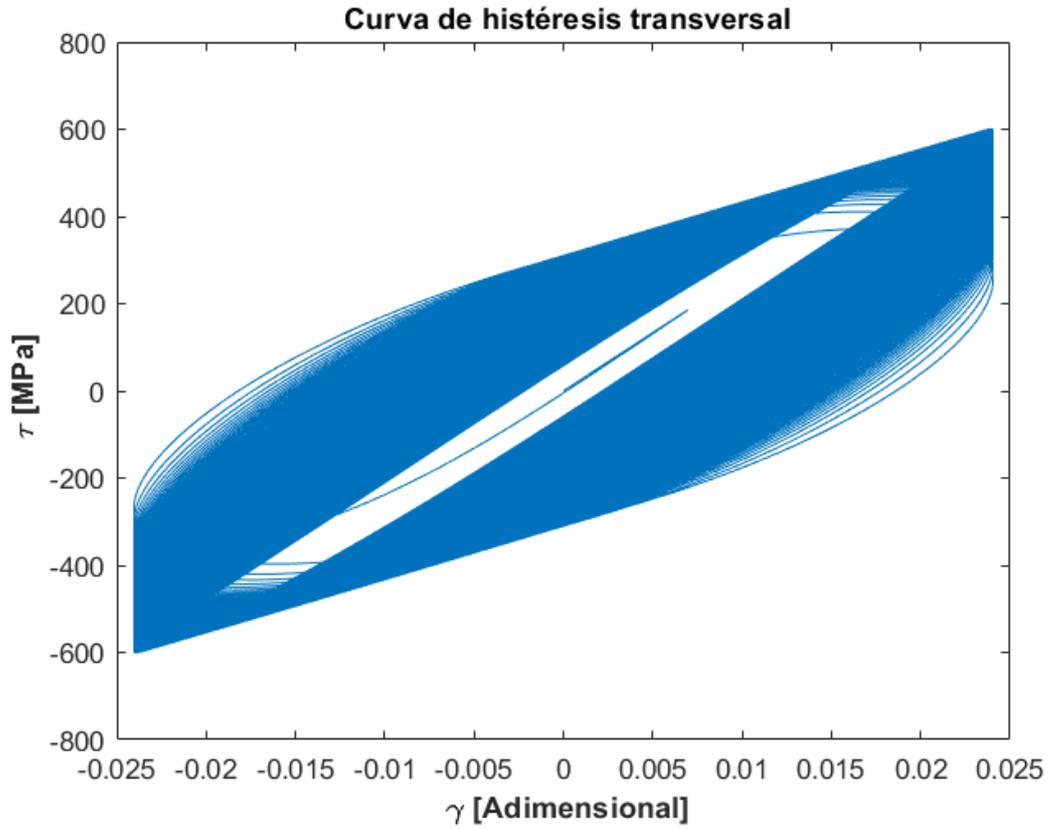


Figura 4.3.2.3.2

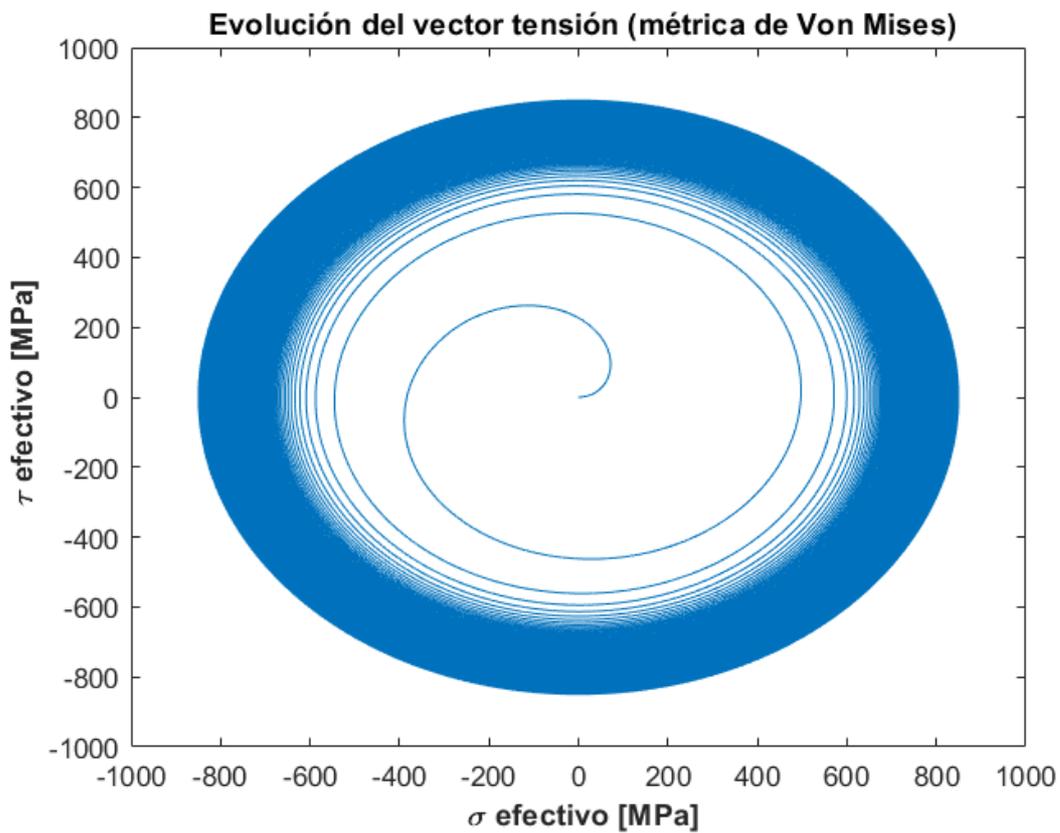


Figura 4.3.2.3.3

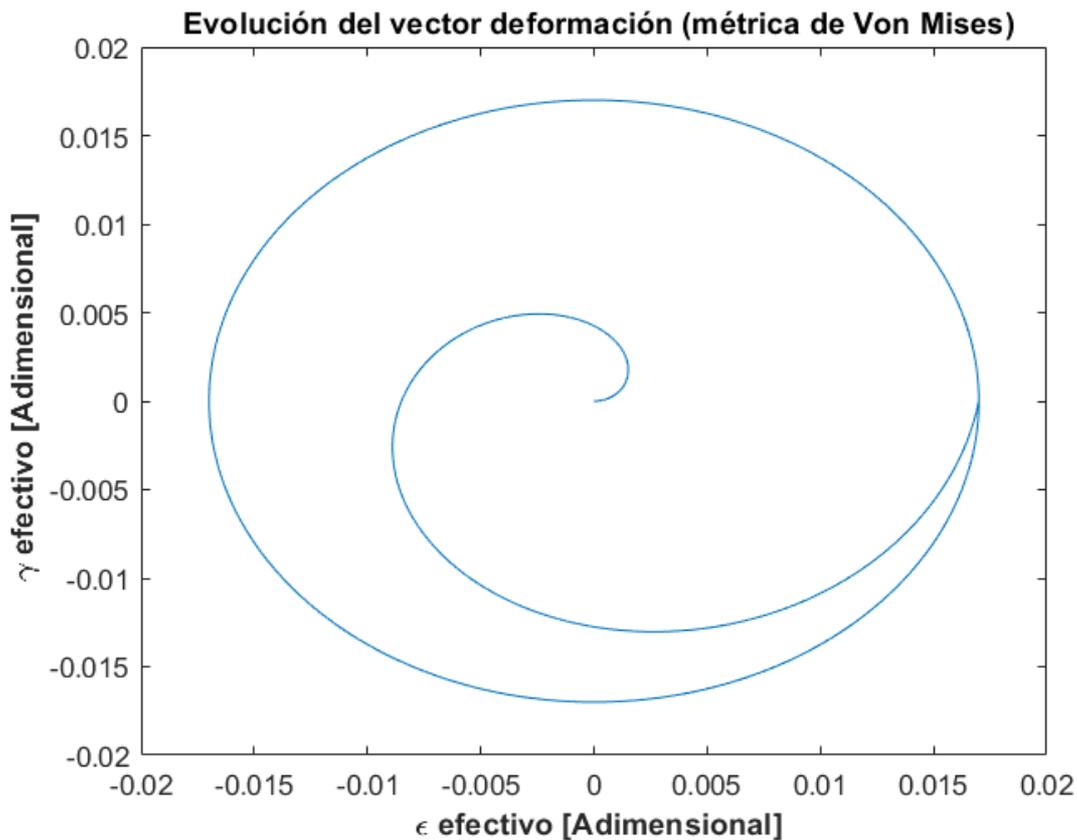


Figura 4.3.2.3.4

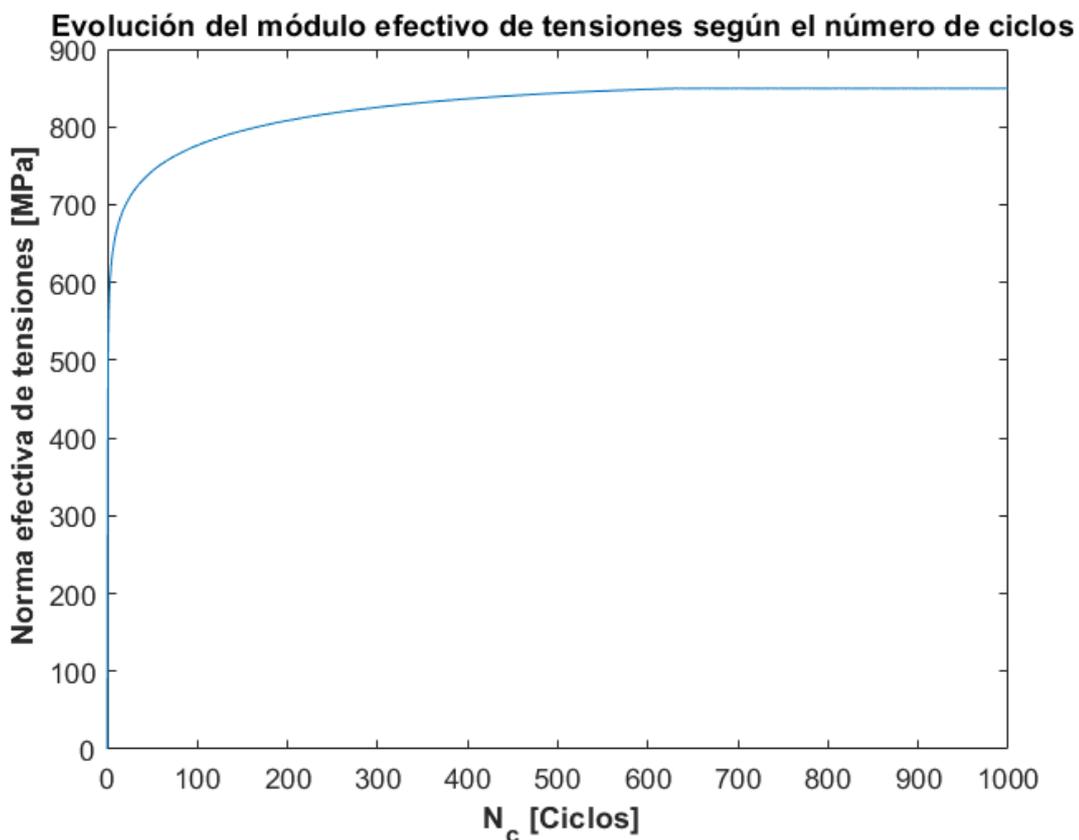


Figura 4.3.2.3.5

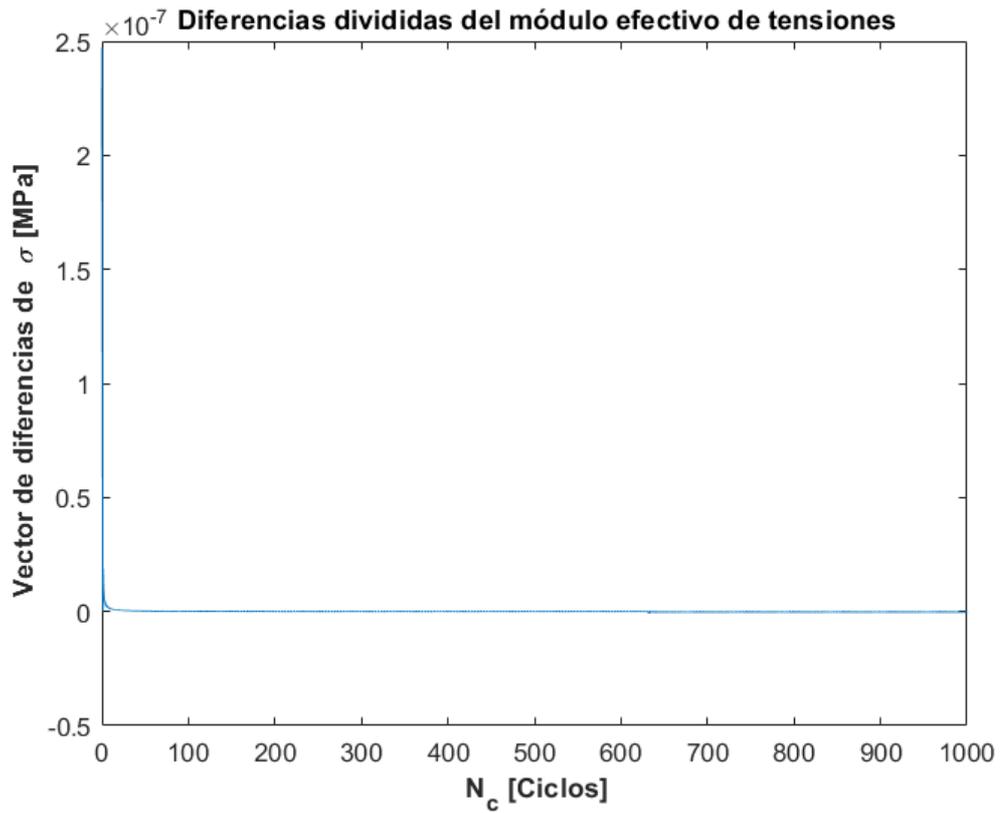


Figura 4.3.2.3.6

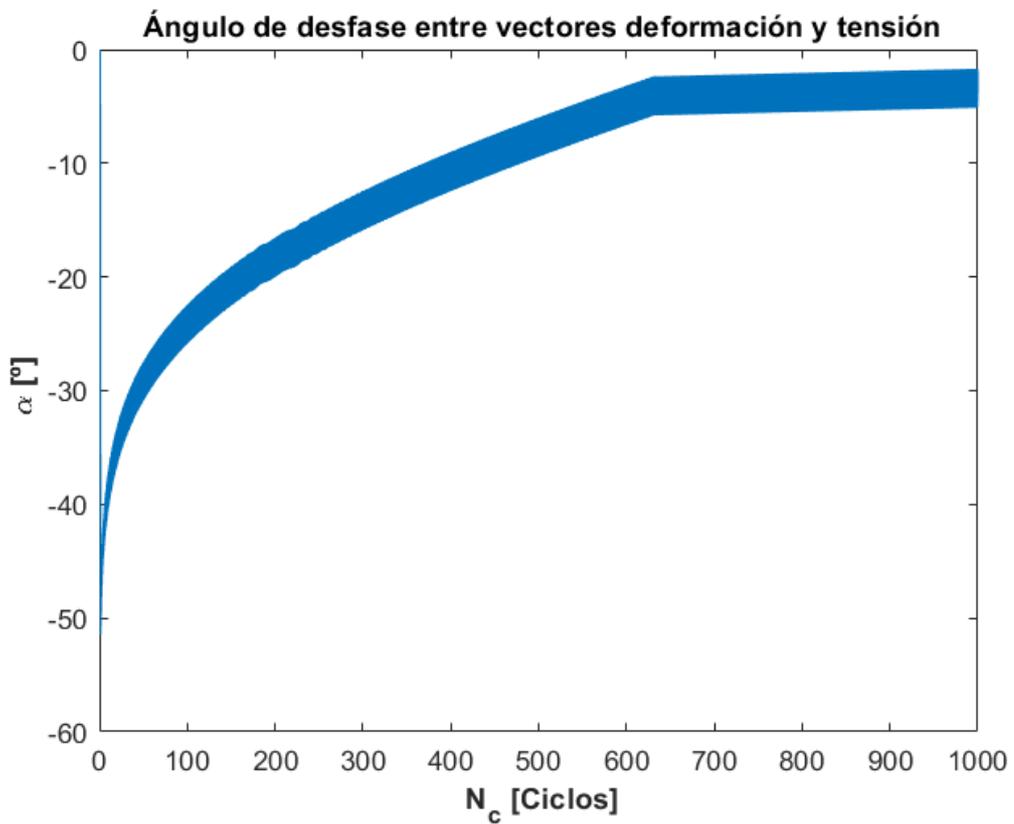


Figura 4.3.2.3.7

4.3.3 Análisis de los resultados

Una vez obtenidos los resultados de los tres modelos de cálculo utilizados bajo los mismos parámetros de simulación, podemos hacer un análisis preliminar.

Para empezar, tenemos que destacar que los resultados obtenidos con el tercer método de cálculo son muy similares a los obtenidos con el segundo método, cuyos resultados están en las figuras del apartado 4.2.2. Los resultados son en esencia los mismos, pero se han eliminado todas las oscilaciones numéricas que provocaban utilizar un método de integración paso a paso a mano con una tolerancia no muy precisa. Por lo tanto, podemos considerar que las simplificaciones que hemos realizado en este método no afectan a la veracidad de los resultados y que aparentemente han sido un éxito. Además, el tiempo de ejecución es el mismo que para el primer modelo de cálculo. Esto convierte a este modelo de cálculo en el modelo de referencia a utilizar para las simulaciones de este proyecto.

Podemos observar que las previsiones teóricas se han cumplido, puesto que el módulo efectivo de tensiones se estabiliza, según las figuras (4.3.2.1.5), (4.3.2.2.5) y (4.3.2.3.5). Se estabiliza en el mismo valor en el cual en el primer modelo de cálculo se alcanzaba el máximo de $|\sigma|$, y en el mismo valor en el cual se estabilizaba en el segundo modelo de cálculo, pero con las oscilaciones numéricas asociadas. Podemos asegurar la veracidad de los resultados y que se cumplen las previsiones teóricas desarrolladas en el apartado 2.

Observando las figuras de los ciclos de histéresis, (4.2.3.1.1), (4.2.3.1.2), (4.2.3.2.1), (4.2.3.2.2), (4.2.3.3.1), (4.2.3.3.2), podemos ver que en cuanto $|\sigma|$ se estabiliza, también la órbita que describe el diagrama se estabiliza, apreciándose con mayor detalle con el comando *comet*. La órbita en la cual se estabiliza se acerca mucho a un comportamiento elástico, lo que cumple las previsiones e hipótesis establecidas.

El ángulo desfase entre los vectores tensión σ y deformación ϵ , representado en las figuras (4.3.2.1.7), (4.3.2.2.7) y (4.3.2.3.7), el cual no había sido analizado hasta ahora, nos indica también algo curioso. Según la teoría de plasticidad utilizada en este proyecto, el valor del ángulo se asemejaría a 90° cuando la plasticidad predomina. Pero vemos que a medida que los lazos de histéresis se van estrechando cada vez más, viendo que el ángulo de desfase se reduce, los vectores tensión σ y deformación ϵ se van alineando. Llega el punto de la estabilización, en un valor cercano a -5° para todos los ensayos. Esto significa que el comportamiento del material es casi completamente elástico, lo que valida una vez más nuestros resultados.

En cuanto al vector de diferencias divididas $d\sigma$ frente al número de ciclos N_c , podemos observar en las figuras (4.3.2.1.6), (4.3.2.2.6) y (4.3.2.3.6) que este valor se estabiliza en 0, excepto pequeñas variaciones que pueden despreciarse, comportándose según lo previsto.

Para finalizar, estos resultados indican que nuestro tercer modelo se comporta según las hipótesis descritas previamente. Hemos conseguido obtener un método numérico de referencia para posteriores ensayos de este tipo de problemas, en el cual las dos componentes del vector deformación ϵ tengan 90° , es decir, tracen una circunferencia manteniendo $|\epsilon| = cte$.

5 ANÁLISIS DE RELACIÓN ENTRE VARIABLES

Concluido este análisis preliminar, se asegura que se cumplen las previsiones teóricas. Sin embargo, hay un pequeño detalle a tener en cuenta. Cuanto mayor es la deformación impuesta $|\varepsilon_{max}|$, mayor es el número de ciclos N_c necesario para la estabilización de $|\sigma|$.

Sería interesante establecer una relación de cómo varía este valor N_c de estabilización según la deformación máxima impuesta $|\varepsilon_{max}|$ en nuestro modelo. A partir de este punto utilizaremos el tercer modelo de simulaciones, visto su buen comportamiento numérico y acorde a las previsiones teóricas del apartado 2

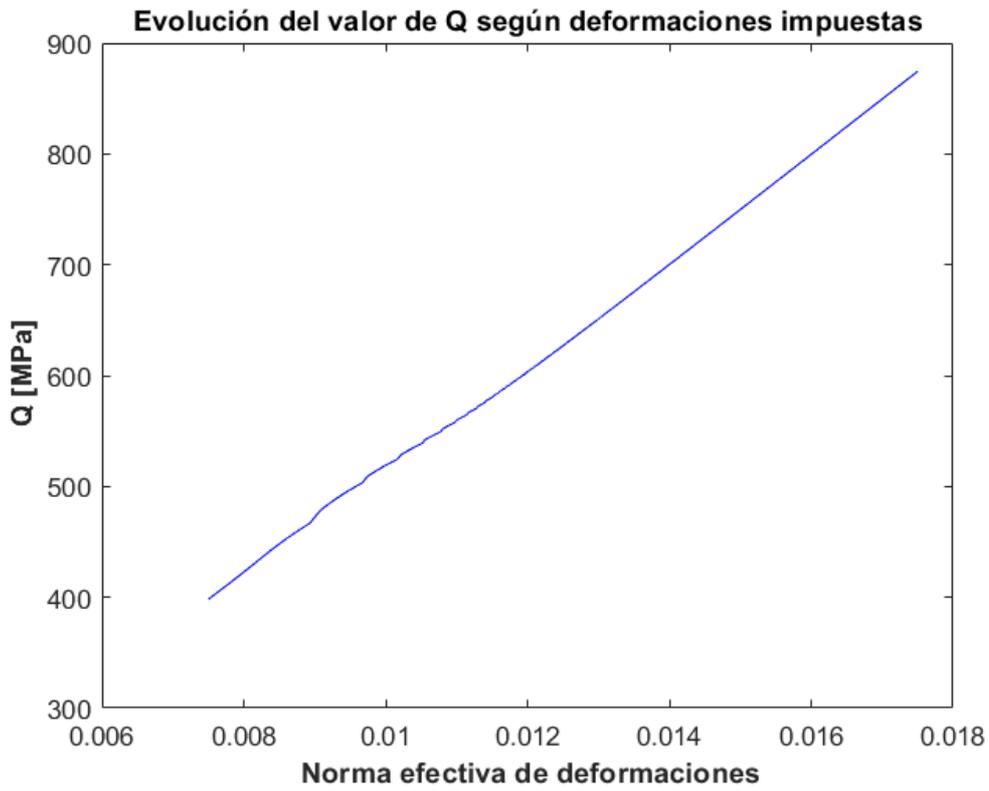
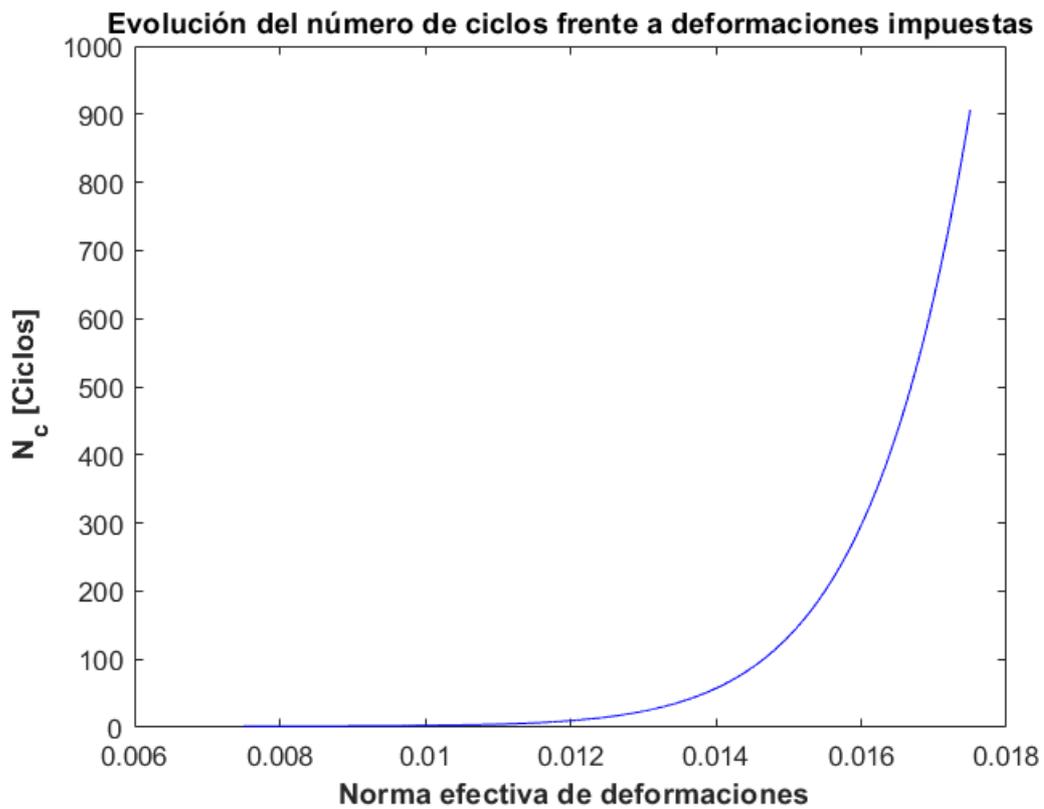
5.1 Evolución del módulo de tensiones estabilizado frente a las deformaciones impuestas

En este apartado veremos cómo varía el módulo de tensiones efectivo $|\sigma|$ en función de las deformaciones impuestas constantes $|\varepsilon_{max}|$ que definamos en la simulación. Para ello crearemos un nuevo fichero en MATLAB apoyándonos en el script principal de simulación *espiral.cl*. En dicho script utilizaremos los métodos de integración del tercer modelo de cálculo, el cual se describe en el apartado 4.3, ya que son los más adecuados y precisos para realizar las simulaciones, además de ser bastante más rápidos de ejecutar que los del segundo modelo de cálculo descrito en el apartado 4.2.

El programa para este cometido es el fichero *ordenadatos.m*. Consiste en el siguiente algoritmo:

- Utilizamos la estructura base del programa *espiral.cl*, configurando los parámetros exactamente igual que en el tercer modelo de cálculo, fijando el parámetro “*ecuación*” en 3.
- Se definirá un parámetro h_e el cual nos servirá para ir iterando y almacenando distintas simulaciones con diferentes valores de $|\varepsilon_{max}|$. h_e será el incremento de $|\varepsilon_{max}|$ entre una simulación y otra.
- Se definen dos parámetros, $|\varepsilon_{max}|_i$ y $|\varepsilon_{max}|_f$, que definirán el valor inicial y final de $|\varepsilon_{max}|$ de las simulaciones a almacenar; cuyos valores serán 0.075 y 0.0175, respectivamente.
- En cada iteración, se almacenarán los valores Q y N_{est} , que representarán el valor en el cual se estabiliza el módulo de tensiones efectivo $|\sigma|$ y el número de ciclos necesario para que se estabilice, respectivamente.
- Se almacena esta información en un archivo de texto, *tabl.txt*, para su posterior tratamiento, ya que al realizar bastantes simulaciones, el tiempo de ejecución es alto.

Una vez ejecutado podemos representar los datos obtenidos con las siguientes figuras:

*Figura 5.1.1**Figura 5.1.2*

5.2 Ajustes gráficos

Podemos observar claramente que la relación entre el valor estabilizado de $|\sigma|$ y las deformaciones impuestas $|\varepsilon_{max}|$ es lineal. Además, el número de ciclos necesario para la estabilización N_{est} parece ser que aumenta exponencialmente a medida que aumenta $|\varepsilon_{max}|$.

Para comprobar que las evoluciones de estas dos relaciones son lineal y exponencial, respectivamente, se utilizará la función *ajuste.m*. Esta función comparará los resultados gráficos reales con unas gráficas generadas mediante ajustes de regresión. Esto puede implementarse fácilmente gracias a la función incluida en MATLAB *lsqcurvefit*, la cual recibe una función deseada, y los vectores de las variables independientes y dependientes, devolviendo los coeficientes de la función introducida. Dicha función con estos coeficientes representa el ajuste más cercano a los puntos de muestreo.

Esto se realizará con una función lineal para la gráfica $Q - |\varepsilon_{max}|$ del tipo:

$$f(x) = c_1 + c_2 x \tag{5.2.1}$$

y para la gráfica $N_{est} - |\varepsilon_{max}|$ una función exponencial del tipo:

$$f(x) = c_1 c_2^x \tag{5.2.2}$$

Siendo en ambos casos c_1 y c_2 los coeficientes que calculará *lsqcurvefit* para los datos recogidos en el fichero *tabl.txt*, que se deben calcular anteriormente con el ejecutable *ordenadatos.m*

También podremos comparar ambos datos y calcular el error relativo máximo que se comete con el ajuste. Los resultados se ven a continuación:

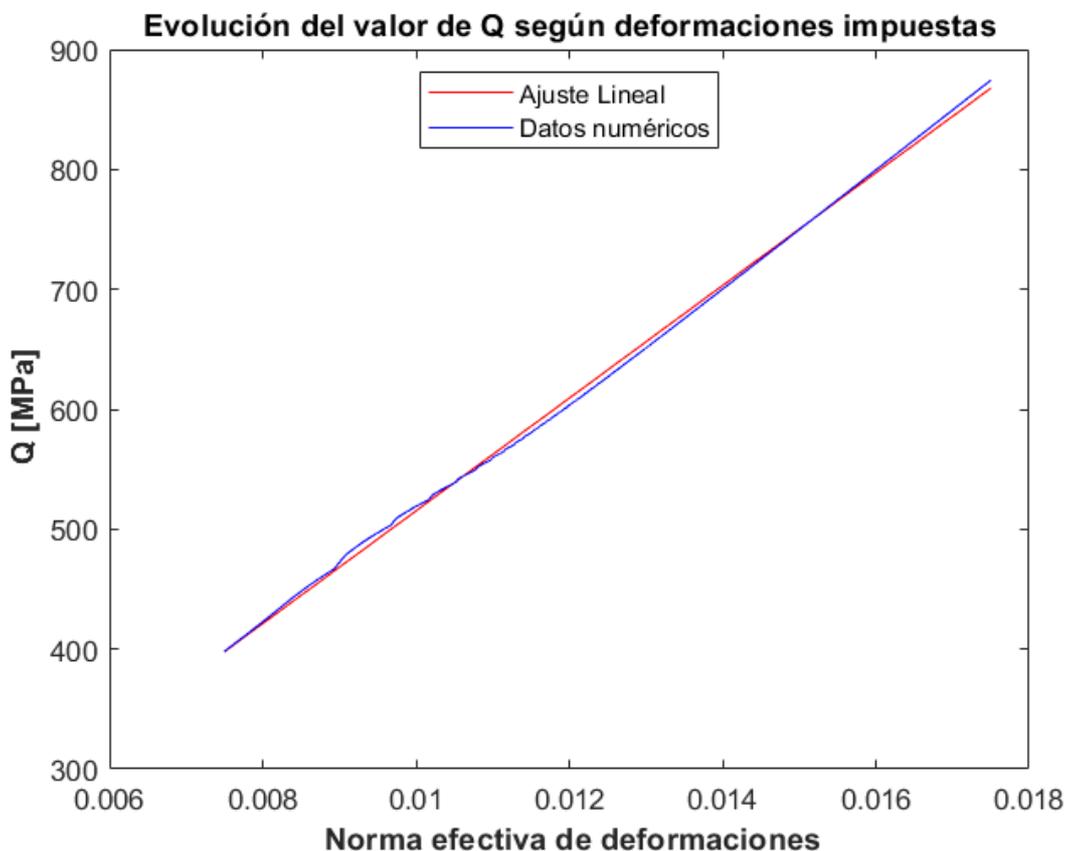


Figura 5.2.1

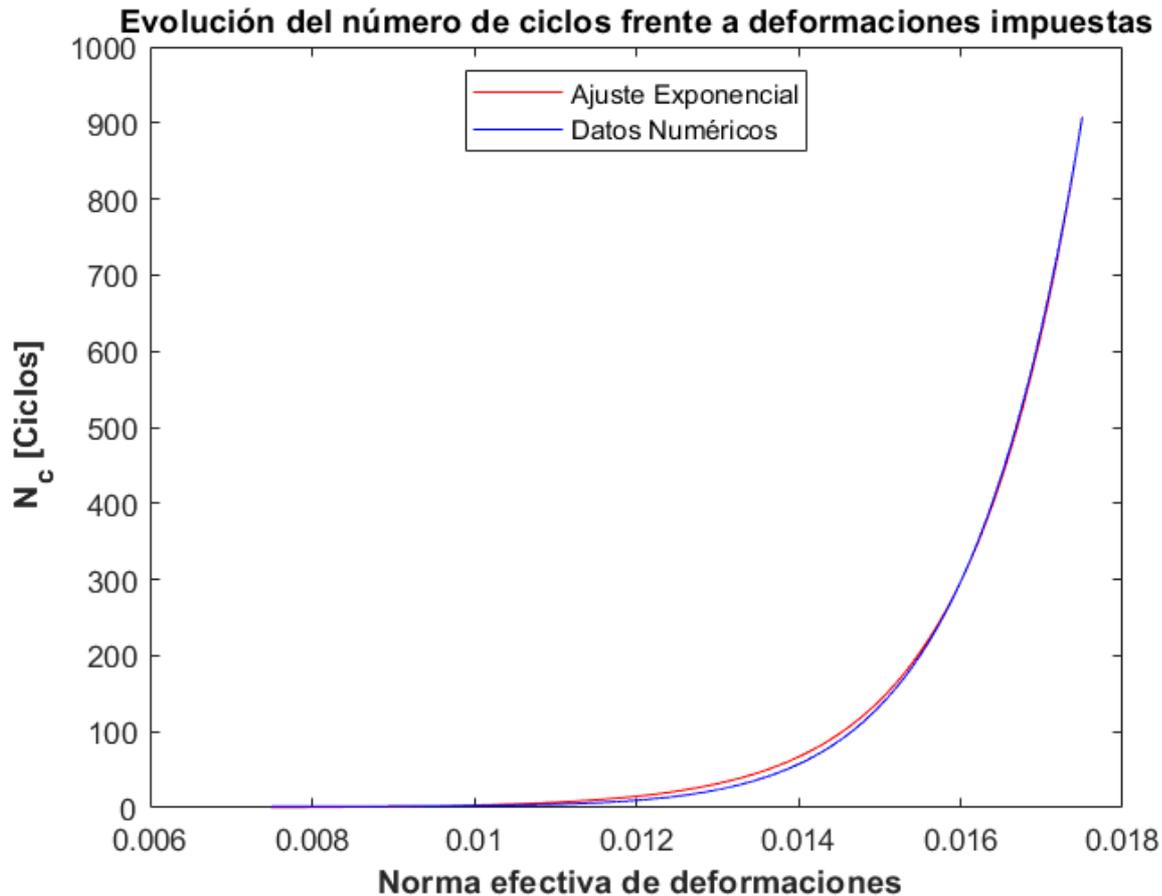


Figura 5.2.2

Los errores relativos máximos son en el caso del ajuste lineal representado en la figura 5.2.1 $e_{max} = 0,53 \%$, y en el caso del ajuste exponencial $e_{max} = 0,91 \%$. Estos valores son muy pequeños, por lo que vemos que el ajuste es excelente en ambos casos.

Los coeficientes obtenidos para ambas relaciones son los siguientes, teniendo cuenta los parámetros para el Aluminio 7075 T651 según la referencia [5]:

$$\begin{aligned} c_1^{lin} &= 47.000 ; c_2^{lin} = 45 \\ c_1^{exp} &= 0 ; c_2^{exp} = 1.727 \end{aligned} \quad (5.2.3)$$

Siendo c_1^{lin} y c_2^{lin} los parámetros para el ajuste lineal de la relación $Q - |\epsilon_{max}|$ y c_1^{exp} y c_2^{exp} los parámetros para el ajuste exponencial de la relación $Q - N_{est}$,

5.3 Predicciones para deformaciones plásticas superiores

5.3.1 Consideraciones generales

Si nos fijamos más detenidamente en la figura 5.2.2 vemos que al verificarse que el número de ciclos necesarios para que el material se estabilice, N_{est} , evoluciona exponencialmente a deformaciones altas. Por ello, a deformaciones sensiblemente más altas, la estabilización se produciría tan tarde que se saldría del campo de la fatiga a bajo número de ciclos, ya que el valor N_{est} superaría holgadamente los 1000 ciclos.

Es por ello de utilidad examinar detenidamente el comportamiento a deformaciones más altas para estudiar si podemos realizar una predicción mejor que declarar que el valor de $|\sigma|$ se estabilizará.

Para ello simularemos de nuevo con el tercer modelo de cálculo, pero esta vez estableceremos $|\epsilon_{max}|$ en un valor bastante mayor, por ejemplo, $|\epsilon_{max}| = 0.05$, evaluando 500 ciclos de carga. Esto nos permitirá ver si $|\sigma|$ crece muy lentamente o si podemos aproximararlo con otro tipo de ajuste.

Los resultados se muestran en el siguiente apartado.

5.3.2 Resultados de las simulaciones

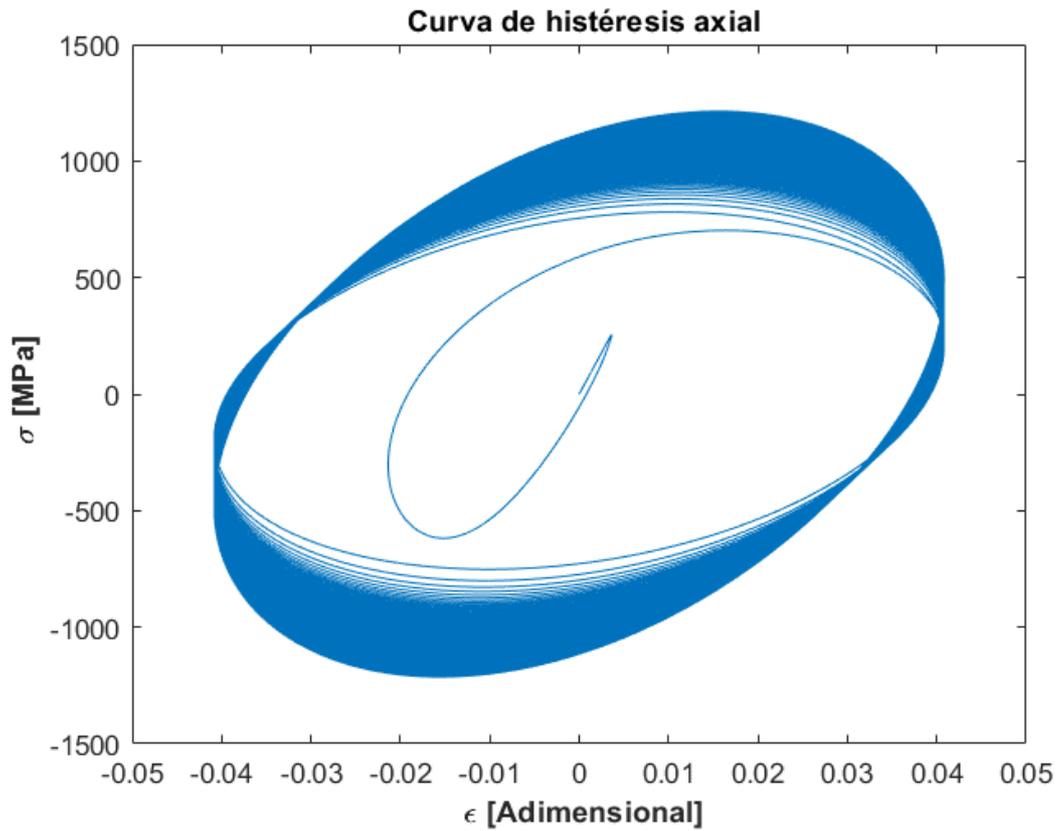


Figura 5.3.2.1

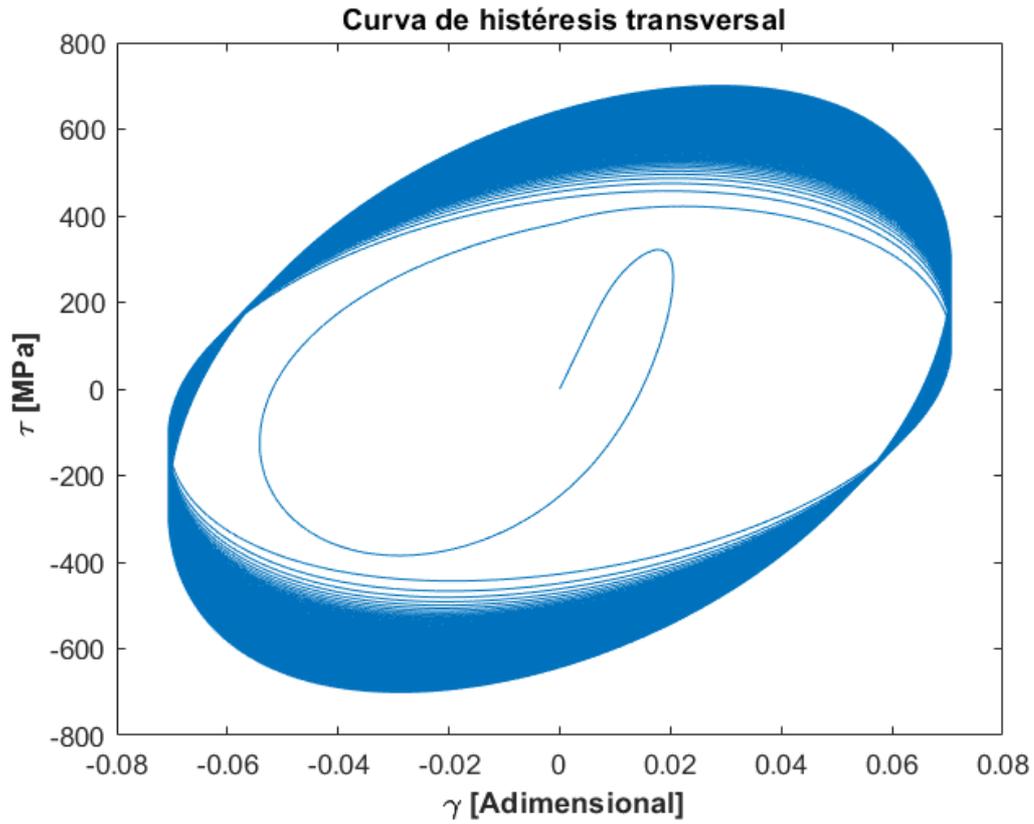


Figura 5.3.2.2

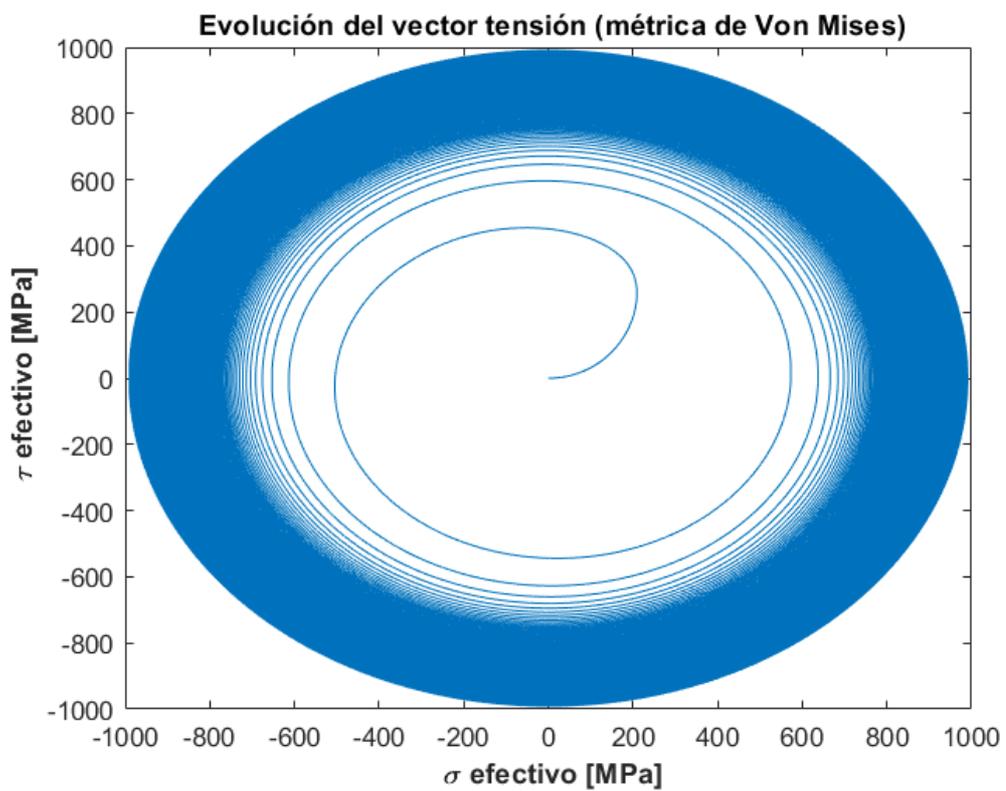


Figura 5.3.2.3

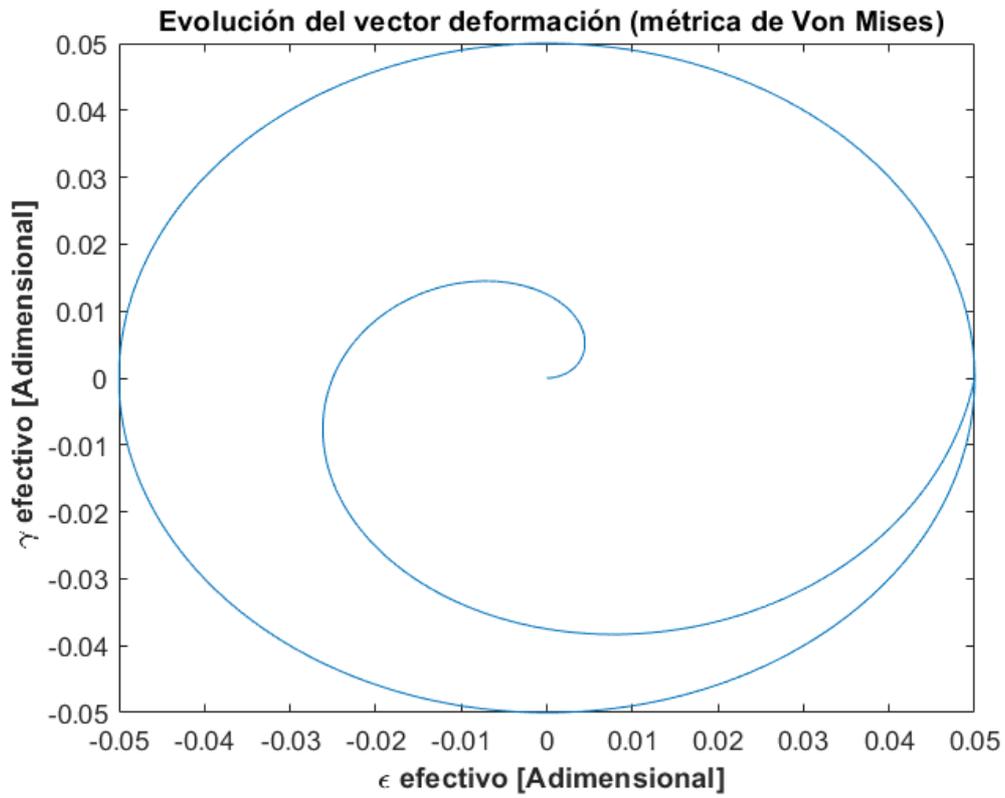


Figura 5.3.2.4

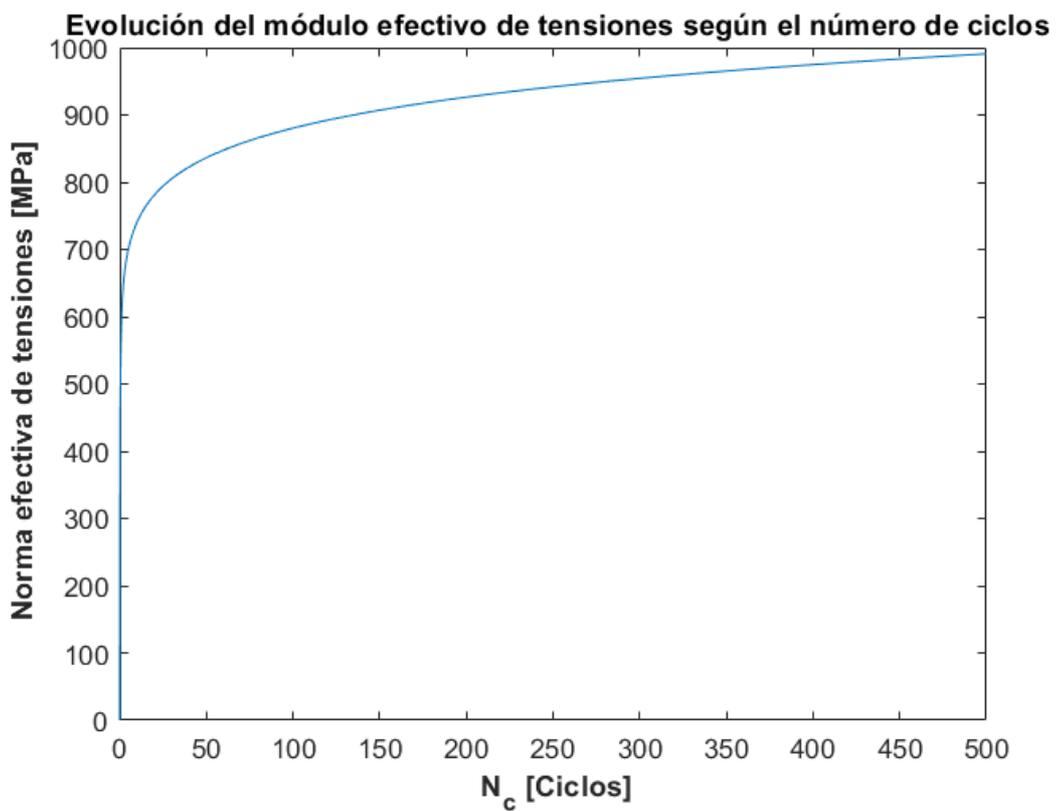


Figura 5.3.2.5

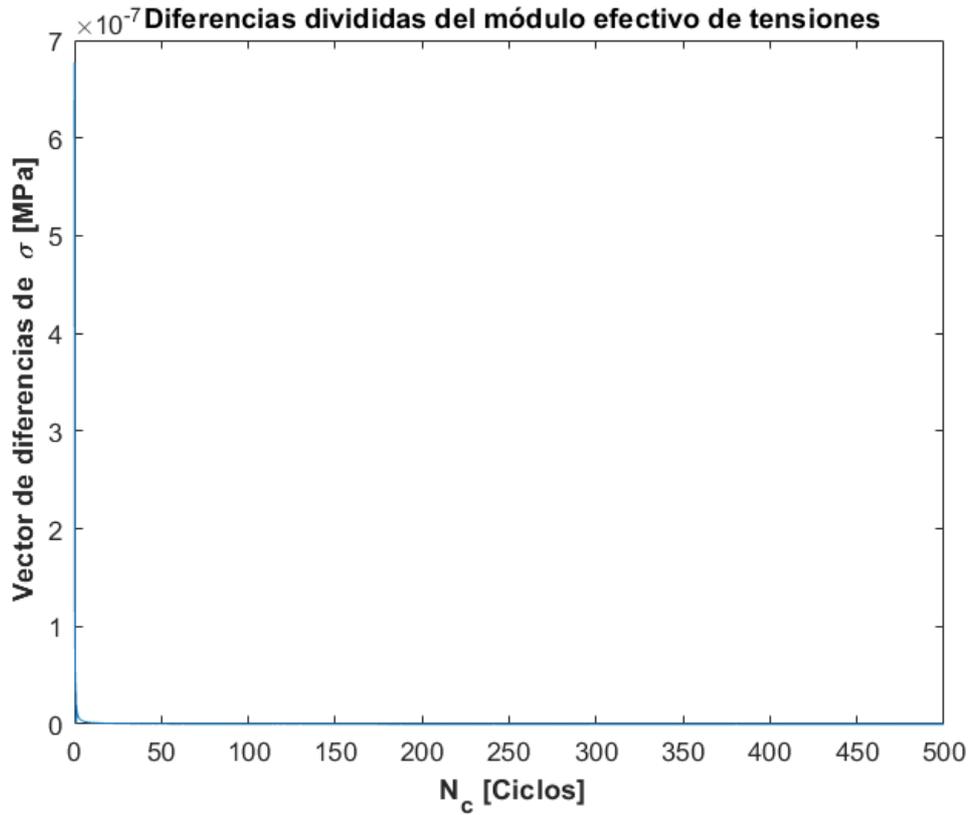


Figura 5.3.2.6

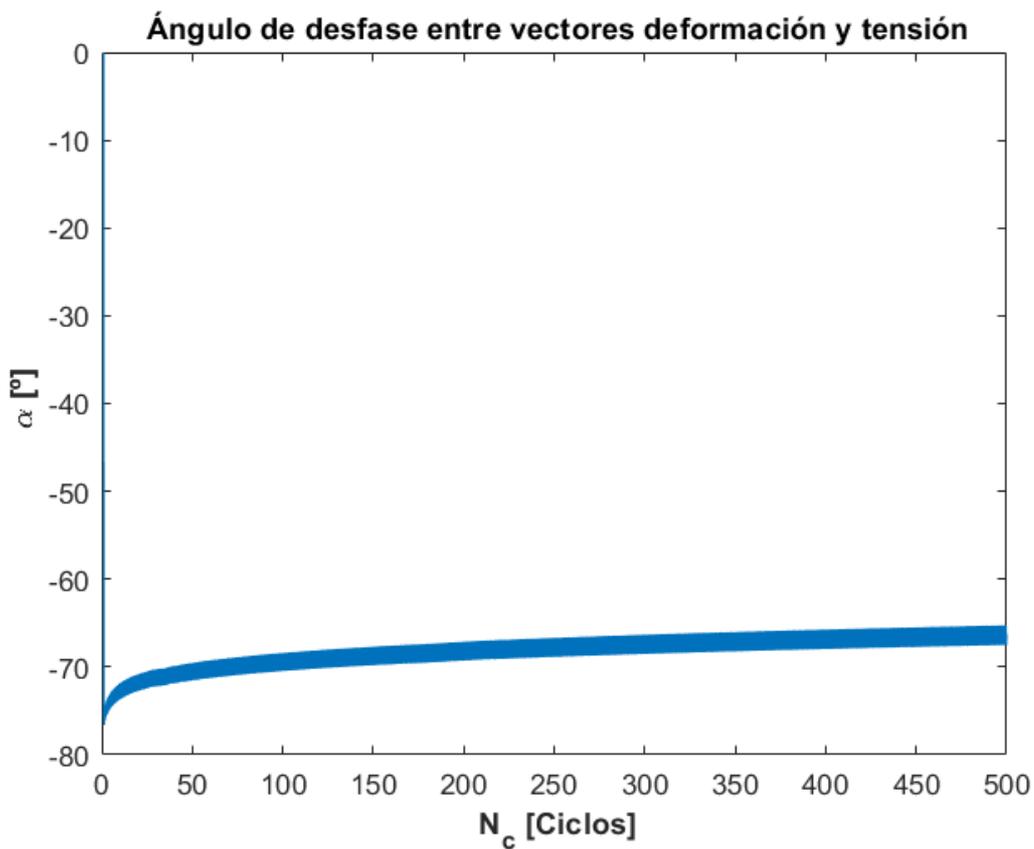


Figura 5.3.2.7

5.3.3 Análisis de los resultados

Empezaremos a analizar los ciclos o lazos de histéresis de las figuras 5.3.2.1 y 5.3.2.2, en comparación con lo que hemos visto anteriormente para las deformaciones impuestas menores. Vemos que los lazos de histéresis son anchos y que se estrechan a un ritmo bastante lento, lo que nos indica que la supuesta estabilización, si llegase, habría llegado mucho después de la rotura del material, por lo que dicho cálculo no nos sería muy útil.

El ángulo entre los vectores tensión σ y deformación ϵ se acerca bastante a 90° , y evoluciona muy lentamente, diciéndonos también que faltan muchos ciclos para la estabilización. Esto nos dice también que el comportamiento plástico predomina todavía mucho en material. Podemos apreciar esta información en la figura 5.3.2.7

Por último, si nos fijamos en la figura 5.3.2.5 vemos que el comportamiento se asemeja más a una gráfica logarítmica que a una asíntota, que es a lo que tendría que asemejarse dicha figura para asegurar que $|\sigma|$ se va a estabilizar en no muchos ciclos relativamente hablando.

5.3.4 Ajuste gráfico

Análogamente al procedimiento del apartado 5.2, volveremos a utilizar el comando *lsqcurvefit* de MATLAB para realizar un ajuste a la figura 5.3.2.5 según una función logarítmica de la forma:

$$f(x) = c_1 \log(x - c_2) + c_3 \quad (5.3.4.1)$$

Siendo c_1 , c_2 y c_3 los coeficientes a buscar para representar la función del ajuste. Análogamente al apartado 5.2, calcularemos también el error relativo máximo para este ajuste. El ajuste se realizará para el segundo tramo, ya que el tramo de carga hasta que el módulo de deformación efectiva alcanza el valor de $|\epsilon_{max}|$ es casi una línea recta, lo que invalidaría el ajuste

Los resultados se muestran a continuación:

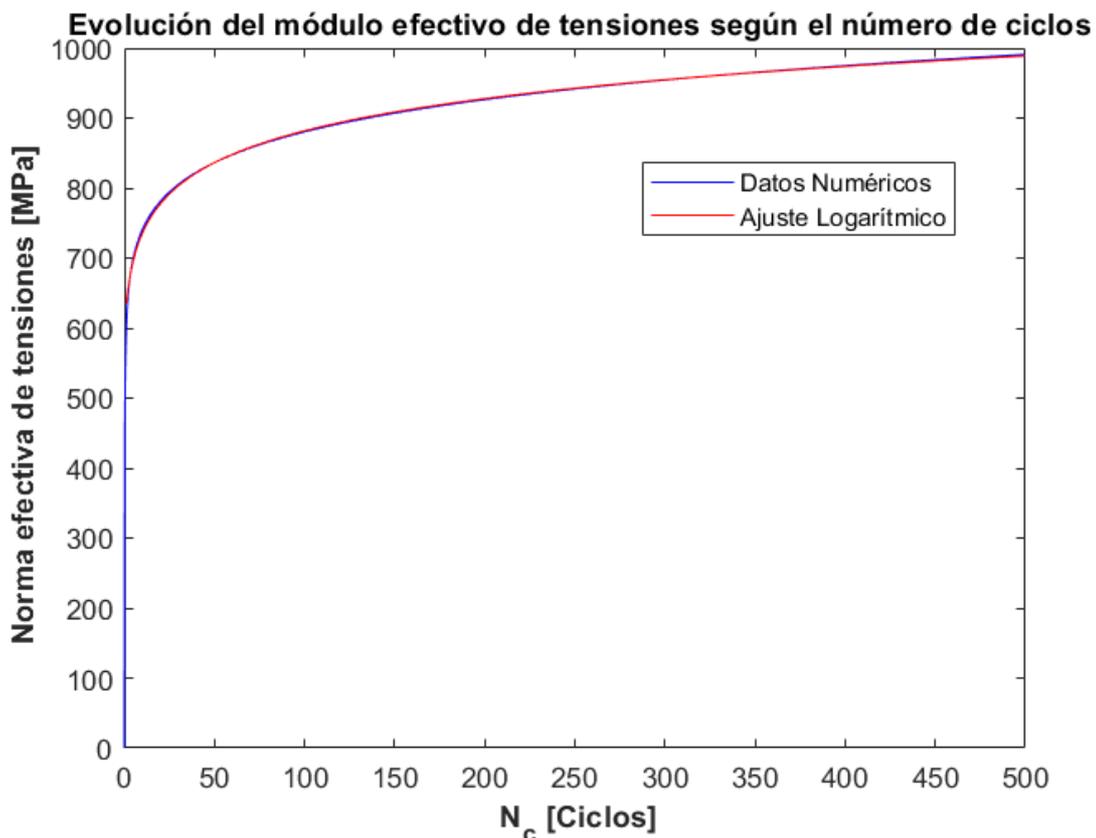


Figura 5.3.4

Como vemos, el ajuste encaja a la perfección con los datos reales de las simulaciones, por lo que vemos que, en caso de deformaciones altas, podemos predecir el comportamiento como una tendencia tipo logarítmica, en el caso de la evolución de $|\sigma|$ respecto al número de ciclos que ensayamos.

El error relativo para este ajuste es de $e_{max} = 0.05 \%$, un valor muy pequeño, con lo que podemos asegurar que el ajuste nos vale para prevenir este tipo de comportamiento a la perfección.

Los coeficientes del ajuste logarítmico realizado en la figura 5.3.4 son los siguientes, utilizando los parámetros para el Aluminio 7075 T651 según la referencia [5]:

$$c_1 = 67 \quad c_2 = -1,4 \quad \text{y} \quad c_3 = 572 \quad (5.3.4.2)$$

6 CONCLUSIONES

Esta teoría de plasticidad nos ha dado la oportunidad de construir un sistema de ecuaciones las cuales nos permite analizar el comportamiento de materiales Ramberg-Osgood utilizando en este caso el criterio de plastificación de von Mises. En este proyecto se ha analizado exhaustivamente el caso particular del endurecimiento fuera de fase, es decir, las componentes del vector deformación forman un ángulo de 90° y trazan una circunferencia de radio constante en el espacio métrico definido.

A lo largo del documento hemos realizado diferentes simulaciones para diferentes condiciones.

Hemos encontrado un modelo de cálculo eficiente cuyos resultados son lo esperados, según las previsiones teóricas e hipótesis que hemos tomado en el proceso. Aplicándolo podemos llegar a una conclusión clara: existe un tramo de valores de deformaciones constantes impuestas $|\varepsilon_{max}|$ para los que el valor del módulo efectivo de tensiones $|\sigma|$ se estabiliza en un margen de ciclos dentro del campo de estudio de la fatiga a bajo número de ciclos. Este número está en torno a los 1000 ciclos.

Cuando esta estabilización ocurre, el material se comporta de manera casi perfectamente elástica, puesto que los ciclos de histéresis han colapsado en este punto y se ha agotado la deformación plástica (área que encierran los lazos de histéresis). Sin embargo, a medida que aumentamos las deformaciones constantes impuestas $|\varepsilon_{max}|$, podemos ver que el material tarda cada vez más en estabilizarse, y la evolución es exponencial.

Eso nos ha llevado a realizar una simulación tipo para valores de deformación más altos, observando que en estos casos una previsión más eficiente que la estabilización de $|\sigma|$ es una evolución logarítmica, la cual se ajusta perfectamente como hemos visto en las representaciones gráficas.

REFERENCIAS

- [1] Norman E. Dowling, *Mechanical Behavior of Materials. Engineering Methods for Deformation, Fracture, and Fatigue*. Always Learning Pearson, 4th Edition. p. 638-683
- [2] A. Navarro, C. Madrigal, V. Chaves. *Plasticity theory for the multiaxial Local Strain-Life Method*. *International Journal of Fatigue* 100 575-582; 2016; p. 1-6
- [3] A. Navarro, C. Madrigal, V. Chaves. *A simplified plasticity model for multiaxial non-proportional cyclic loading*. *Theoretical and Applied Fracture Mechanics* 103 102247; 2019; p. 2-3
- [4] A. Navarro. *Plastic flow and memory rules for the local strain method in the multiaxial case*; 2005; p. 8-11
- [5] C. Navarro. *Práctica de bases para el diseño de sistemas mecánicos. Ensayo de fatiga*; 2016

ANEXO. CÓDIGO DE PROGRAMACIÓN NECESARIO PARA LAS SIMULACIONES

Espiral.cl

```
%SCRIPT PRINCIPAL PARA SIMULACIÓN%

clear,clc
close all          %LIMPIADO DE MEMORIA%
format long
tic              %EMPEZAMOS A CONTAR EL TIEMPO DE CÁLCULO%
global K na E G  %CONSTANTES GLOBALES DEL MATERIAL%

%CONJUNTO DE VARIABLES QUE DEFINEN LOS ENSAYOS QUE SE REALIZARAN, SÓLO
ES NECESARIO
%CAMBIAR ESTAS VARIABLES PARA REALIZAR UNA SIMULACIÓN DISTINTA

%TIPO DE ECUACIONES A UTILIZAR=
%1=ECUACIONES SIN DESCARCA
%2=ECUACIONES CON DESCARGA
%3=ECUACIONES SIMPLIFICADAS
%UNA VEZ DEFINIDAS LAS ECUACIONES TIPO 3, SE UTILIZARAN ESTAS%

ecuacion=2;

material=1; %ELECCION DE MATERIAL PARA LAS SIMULACIONES, ALUMINIO 1,
ACERO 0
esp=1;      %ELECCION DEL TIPO DE CARGA HASTA LLEGAR AL MÁXIMO DE
DEFORMACIÓN, 0 UNIAXIAL, 1 APROXIMACIÓN EN ESPIRAL
h=1e-4;    %TOLERANCIA PARA LA VISUALIZACIÓN DE LAS GRÁFICAS. EN CASO
DE ECUACIONES TIPO 2,
%TAMBIÉN ES LA TOLERANCIA DE INTEGRACIÓN
Nccarga=1; %NUMERO DE CICLOS PARA LLEGAR A DEFORMACIÓN MÁXIMA%
emax=0.017; %MÓDULO EFECTIVO DE DEFORMACIÓN MÁXIMO%
Nc=1000; %NÚMERO DE CICLOS TOTALES%

if material==0 %ELECCIÓN DE MATERIAL PARA LAS SIMULACIONES%

% ACERO
K=1655;          %PARÁMETROS PARA EL ACERO%
na=0.131;
E=207e3;
v=0.3;
G=E/(2*(1+v));

else

% ALUMINIO%
K=852;           %PARÁMETROS PARA EL ALUMINIO%
na=0.074;
```

```

E=71e3;
v=0.33;
G=E/(2*(1+v));

end

tcarga=Nccarga*2*pi;      %CREAMOS LA VARIABLE T, QUE RESULTA DE
MULTIPLICAR 2PI POR EL NUMERO DE CICLOS CORRESPONDIENTE%
tspan=0:h:tcarga;       %INICIALIZACIÓN DEL VECTOR T PARA EL TRAMO DE
CARGA%
l1=length(tspan);       %PARÁMETRO UTILIZADO PARA EMPEZAR A MEDIR EL
DESFASE DE ÁNGULOS UNA VEZ TERMINA LA CARGA%

mepsilon=emax*sqrt(2/3); %DEFORMACIÓN EFECTIVA EN COMPONENTES EPSILON
Y GAMMA PARA DESFASE DE 90°, ES DECIR, EMAX SIEMPRE ES CONSTANTE%
mgamma=emax*sqrt(2);

if esp==1                %SELECCIÓN DEL TIPO DE CARGA%

% CARGA EN ESPIRAL%
eps.e=@(t)mepsilon*t.*cos(t)/tcarga;      %DEFINICIÓN DE FUNCIÓN DE
APROXIMACIÓN EN ESPIRAL. LOS TERMINOS DE ARRIBA SE METEN SI QUEREMOS
DERIVAR NUMERICAMENTE%
eps.g=@(t)mgamma*t.*sin(t)/tcarga;
epsc.e=mepsilon/tcarga;                    %ESTOS TERMINOS SE UTILIZAN
SI PODEMOS CALCULAR LA DERIVADA EXACTA, COMO COEFICIENTES CONSTANTES%
epsc.g=mgamma/tcarga;

else
% CARGA LINEAL%

eps.e=@(t)mepsilon/tcarga*t;              %DEFINICIÓN DE FUNCIÓN DE CARGA LINEAL
UNIAXIAL (TRACCIÓN). %
eps.g=@(t)0*t;                            %SE INTRODUCEN EN LA FUNCION DE
INTEGRACIÓN ANÁLOGAMENTE A LAS DE ARRIBA%

epsc.e=mepsilon/tcarga;
epsc.g=0;

end

tspan2=(tcarga):h:Nc*2*pi; %SE INICIALIZA EL VECTOR T PARA LA SEGUNDA
PARTE DE LA SIMULACIÓN%

eps2.e=@(t)mepsilon*cos(t); %SE DEFINE LA ROTACIÓN DEL VECTOR
DEFORMACIONES MANTENIENDO SU MÓDULO EFECTIVO CONSTANTE%
eps2.g=@(t)mgamma*sin(t); %SE DEFINEN LAS FUNCIONES ANALOGAMENTE A
COMO HICIMOS ANTES, PARA DERIVACIÓN NUMÉRICA O PARA CALCULO DE
COEFICIENTES CONSTANTES SABRIENDO LA DERIVADA EXACTA%

epsc2.e=mepsilon;
epsc2.g=mgamma;

```

```

options=odeset('RelTol',1e-12,'AbsTol',1e-12); %DEFINICIÓN DE
TOLERANCIAS DE INTEGRACIÓN%
%SELECCIÓN DE SISTEMA DE INTEGRACIÓN SEGÚN EL TIPO DE ECUACIONES A
%UTILIZAR
if ecuacion==1
[tspan,sigma]=ode45(@syst,tspan,[0;0],options,epsc,esp);
%INTEGRACIÓN DE LAS ECUACIONES DE FLUJO PLÁSTICO EN DERIVADAS
PARCIALES PARA EL PRIMER TRAMO%

[tspan2,sigma2]=ode45(@syst2,tspan2,[sigma(end,1);sigma(end,2)],option
s,epsc2); %INTEGRACIÓN DE LAS ECUACIONES DE FLUJO PLÁSTICO EN
DERIVADAS PARCIALES PARA EL SEGUNDO TRAMO%
end
if ecuacion==2
[tspan,sigma]=systrk4d(0,tcarga,[0;0],epsc,h,esp);
[tspan2,sigma2]=syst2rk4d(tcarga,tspan2(end),[sigma(1,end);sigma(2,end
)],epsc2,h);
tspan=tspan';
tspan2=tspan2';
sigma=sigma';
sigma2=sigma2';
end
if ecuacion==3
[tspan,sigma]=ode45(@systc1,tspan,[0;0],options,epsc,esp);
%INTEGRACIÓN DE LAS ECUACIONES DE FLUJO PLÁSTICO EN DERIVADAS
PARCIALES PARA EL PRIMER TRAMO%

[tspan2,sigma2]=ode45(@systc12,tspan2,[sigma(end,1);sigma(end,2)],opti
ons,epsc2); %INTEGRACIÓN DE LAS ECUACIONES DE FLUJO PLÁSTICO EN
DERIVADAS PARCIALES PARA EL SEGUNDO TRAMO%
end

epsilon=[eps.e(tspan) ;eps2.e(tspan2)]; %SE CALCULAN LAS DEFORMACIONES
EN COMPONENTES EPSILON Y GAMMA PARA CADA INSTANTE DEL VECTOR T%
gamma=[eps.g(tspan) ;eps2.g(tspan2)]; %SE AGRUPAN A SU VEZ LOS
TRAMOS 1 Y 2%

sigma=[sigma; sigma2]; %SE AGRUPAN LAS TENSIONES DE LOS TRAMOS 1 Y
2 , POR COMPONENTES SIGMA Y TAU%
tspan=[tspan ;tspan2]; %SE AGRUPA EL VECTOR DEL PARÁMETRO T DE LOS
TRAMOS 1 Y 2%

Q=modu(sigma'); %SE CALCULA EL MÓDULO EFECTIVO DE LAS TENSIONES
PARA CADA PUNTO CALCULADO%
dm=diff(Q); %SE CALCULA EL VECTOR DE DIFERENCIAS DEL MÓDULO DE
SIGMA, PARA VER SI HA DECRECIDO EN ALGÚN PUNTO%

N=tspan/(2*pi); %DESHACEMOS EL CAMBIO DE VARIABLE DE T A NÚMERO DE
CICLOS TOTALES, PARA CADA INSTANTE EVALUADO%

toc %TERMINAMOS DE CONTAR EL TIEMPO QUE TARDAMOS EN
HACER LOS CÁLCULOS%

```

```

%REPRESENTACIÓN DE FIGURAS%
figure (1)

plot(epsilon, (sigma(:,1)))
title('\bf Curva de histéresis axial')
xlabel('\bf \epsilon [Adimensional]')
ylabel('\bf \sigma [MPa]')

figure (2)

plot(gamma, sigma(:,2))
title('\bf Curva de histéresis transversal')
xlabel('\bf \gamma [Adimensional]')
ylabel('\bf \tau [MPa]')

figure (3)

plot(sqrt(2/3)*sigma(:,1), sqrt(2)*sigma(:,2))
title('\bf Evolución del vector tensión (métrica de von Mises)')
xlabel('\bf \sigma efectivo [MPa]')
ylabel('\bf \tau efectivo [MPa]')

figure (4)

plot(sqrt(3/2)*epsilon, sqrt(1/2)*gamma)
title('\bf Evolución del vector deformación (métrica de von Mises)')
xlabel('\bf \epsilon efectivo [Adimensional]')
ylabel('\bf \gamma efectivo [Adimensional]')

figure (5)

plot(N,Q)
title('\bf Evolución del módulo efectivo de tensiones según el número
de ciclos')
xlabel('\bf N_{c} [Ciclos]')
ylabel('\bf Norma efectiva de tensiones [MPa]')

figure (6)

plot(N(1:end-1), dm*h/(2*pi))
title('\bf Diferencias divididas del módulo efectivo de tensiones')
xlabel('\bf N_{c} [Ciclos]')
ylabel('\bf Vector de diferencias de \bf \sigma [MPa]')

% figure (7)
%
% plot(Q*1e3, flujo(K, na, sigma'))
% title('EVOLUCIÓN DEL FLUJO PLÁSTICO FRENTE A LA NORMA EFECTIVA DE
TENSIONES')

% % REPRESENTACIÓN EN TIEMPO REAL DE LAS GRAFICAS SIGMA-TAU, EPSILON-
GAMMA

```

```

% figure (1)
% title('Tensión axial frente a tensión tangencial')
%
% hold on
% figure (2)
% title('Deformación normal frente a deformación transversal')
% hold on
%
%
%
% for t=1:length(tspan)
%     figure (1)
%
%     plot(sqrt(2/3)*sigma(t,1),sqrt(2)*sigma(t,2),'r')
%     figure (2)
%
%     plot(sqrt(3/2)*epsilon(t),sqrt(1/2)*gamma(t),'b')
%
% end

%CÁLCULO DEL DESFASE ENTRE LOS VECTORES TENSIÓN Y DEFORMACIÓN%

angeps=acos(sqrt(3/2)*epsilon(11:end)/emax); %CÁLCULO DEL ÁNGULO
GIRADO POR EL VECTOR DEFORMACIONES A PARTIR DEL SEGUNDO TRAMO%

angsig=acos(sqrt(2/3)*sigma(11:end,1)./Q(11:end)'); %CÁLCULO DEL
ÁNGULO GIRADO POR EL VECTOR TENSIONES A PARTIR DEL SEGUNDO TRAMO%

angsig(1)=0; %PONEMOS A CERO LOS PRIMEROS INSTANTES, POR SI
FUERAN INFINITO%
angsig2=dp(angsig); %CON UNA FUNCIÓN, PASAMOS LOS ÁNGULOS EN UNIDADES
DE REFERENCIA, A ÁNGULO RECORRIDO TOTAL POR EL VECTOR%
angeps(1)=0;
angeps2=dp(angeps); %ESTO SE HACE PARA LOS DOS ÁNGULOS (DE LAS
DEFORMACIONES Y DE LAS TENSIONES%

%AHORA QUE TENEMOS LOS ÁNGULOS RECORRIDOS POR LOS DOS VECTORES,
PODEMOS
%CALCULAMOS EL DESFASE ENTRE AMBOS VECTORES

desf=angeps2-angsig2;

%REPRESENTACIÓN DEL DESFASE ENTRE VECTORES DEFORMACIÓN Y TENSIÓN%

figure(15)
plot(N(11:end-1),desf*360/(2*pi))
title('\bf Ángulo de desfase entre vectores deformación y tensión')
ylabel('\bf \alpha [°]')
xlabel('\bf N_{c} [Ciclos]')

```

Syst.m

```

function dsigma=syst(t,sigma,eps,esp)
global K na E G

%FUNCIÓN DE INTEGRACIÓN DE ECUACIONES DIFERENCIALES SIMPLIFICADA PARA
EL
%TRAMO1

Q=sqrt(2/3*sigma(1)^2+2*sigma(2)^2); %SE DEFINE EL MÓDULO EFECTIVO
DE TENSIONES%
aux=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(Q^(1/na-3)); %SE DEFINE
UNA VARIABLE AUXILIAR= PHI(Q)/Q^2 PARA REDUCIR POSIBLES PROBLEMAS
NUMÉRICOS%
%SE DEFINE LA MATRIZ DE LAS DERIVADAS PARCIALES%
A=1/(1/E*1/G+(aux)*((4*sigma(2)^2)/E+(4*sigma(1)^2)/(9*G)))*[1/G+(4*s
igma(2)^2)*aux -4/3*sigma(1)*sigma(2)*aux;-4/3*sigma(1)*sigma(2)*aux
1/E+4/9*(sigma(1)^2)*aux];

%CÁLCULO DE LA DERIVADA DEL VECTOR DEFORMACIÓN%
%MÉTODO CONOCIENDO LA DERIVADA EXACTA%
%CARGA LINEAL%
if esp==0
deps=@(t)[eps.e;0];
%CARGA EN ESPIRAL%
else
deps=@(t)[eps.e*(cos(t)-t*sin(t)) ; eps.g*(sin(t)+t*cos(t))];
end
% DIFERENCIACIÓN NUMÉRICA %
% h=1e-10;
%
% deps=@(t)[(eps.e(t+h)-eps.e(t-h))/(2*h);(eps.g(t+h)-eps.g(t-
h))/(2*h)];
dsigma=A*deps(t); %CALCULAMOS LA DERIVADA DEL VECTOR SIGMA A PARTIR
DE LA MATRIZ A Y LA DERIVADA DEL VECTOR DEFORMACIÓN

```

Syst2.m

```

function dsigma=syst2(t,sigma,eps)
global K na E G

%FUNCIÓN DE INTEGRACIÓN DE ECUACIONES DIFERENCIALES SIMPLIFICADA PARA
EL
%TRAMO2

Q=sqrt(2/3*sigma(1)^2+2*sigma(2)^2); %SE DEFINE EL MÓDULO EFECTIVO
DE TENSIONES%
aux=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(Q^(1/na-3)); %SE DEFINE
UNA VARIABLE AUXILIAR= PHI(Q)/Q^2 PARA REDUCIR POSIBLES PROBLEMAS
NUMÉRICOS%
%SE DEFINE LA MATRIZ DE LAS DERIVADAS PARCIALES%
A=1/(1/E*1/G+(aux)*((4*sigma(2)^2)/E+(4*sigma(1)^2)/(9*G)))*[1/G+(4*s
igma(2)^2)*aux -4/3*sigma(1)*sigma(2)*aux;-4/3*sigma(1)*sigma(2)*aux
1/E+4/9*(sigma(1)^2)*aux];

```

```
%CÁLCULO DE LA DERIVADA DEL VECTOR DEFORMACIÓN%
%MÉTODO CONOCIENDO LA DERIVADA EXACTA%
deps=@(t) [(-eps.e*sin(t)); (eps.g*cos(t))];
% DIFERENCIACIÓN NUMÉRICA %
% h=1e-10;
%
%      deps=@(t) [(eps.e*(t+h)-eps.e*(t-h))/(2*h); (eps.g*(t+h)-eps.g*(t-
h))/(2*h)];

dsigma=A*deps(t); %CALCULAMOS LA DERIVADA DEL VECTOR SIGMA A PARTIR DE
LA MATRIZ A Y LA DERIVADA DEL VECTOR DEFORMACIÓN

%se pueden meter la funcion y derivarla o meter directamente la funcion
%derivada %
```

Systrk4d.m

```
function [t,sigma]=systrk4d(tini,tfin,sigma0,eps,h,esp)
d=0; %PARÁMETRO DE DESCARGA%

sigma(:,1)=sigma0; %PUNTO INICIAL DE TENSIONES%

t=tini:h:tfin; %INICIALIZACIÓN DEL VECTOR T A PARTIR DEL PASO DE
INTEGRACIÓN H%

sigmac=[0;0]; %CENTRO DE CIRCUNFERENCIA DE PLASTIFICACIÓN INICIAL%
c=1; %CIRCUNFERENCIA DE PLASTIFICACIÓN ACTIVA%
i=1; %INDICE DE PASO PARA CADA INSTANTE DE T%
while i<=(length(t)-1) %MIENTRAS QUE NO LLEGUEMOS AL INSTANTE FINAL DE
T

if d>=1 %SI SE HA DETECTADO DESCARGA%
    sigmad(:,1)=sigma(:,i-1); %EL PRIMER PUNTO DE DESCARGA ES EL PASO
ANTERIOR A LA DESCARGA%
    q0=2*modu(sigmad(:,1)); %DEFINIMOS EL MÓDULO EFECTIVO DE LA
PRIMERA CIRCUNFERENCIA DE PLASTIFICACIÓN
    %RESOLVEMOS EL SISTEMA CON LAS ECUACIONES DE DESCARGA, MEDIANTE
MÉTODO
    %RUNGE KUTTA PASO A PASO DE 4TH ORDEN
    K1=systd(t(i),sigma(:,i),sigmad(:,d),sigmac(:,c),q0,eps,esp);

K2=systd(t(i)+h/2,(sigma(:,i)+K1/2*h),sigmad(:,d),sigmac(:,c),q0,eps,esp);

K3=systd(t(i)+h/2,(sigma(:,i)+K2/2*h),sigmad(:,d),sigmac(:,c),q0,eps,esp);

K4=systd(t(i)+h,(sigma(:,i)+K3*h),sigmad(:,d),sigmac(:,c),q0,eps,esp);
    sigma(:,i+1)=sigma(:,i)+h*(K1+2*K2+2*K3+K4)/6;

%
%      q0=abs((2/3*(sigma(1,i)-sigmad(1,d))^2+2*(sigma(2,i)-
sigmad(2,d))^2)/(2/3*(sigmac(1,c)-sigmad(1,d))*(sigma(1,i)-
```

```

sigmad(1,d))+2*(sigmac(2,c)-sigmad(2,d))*(sigma(2,i)-
sigmad(2,d))*q0/2);

    i=i+1; %INCREMENTAMOS INDICE DE PASO%
    while modu(sigma(:,i))<modu(sigmad(:,1)) && i<=(length(t)-1)
%MIENTRAS QUE LAS TENSIONES NO SEAN MAYORES AL PRIMER PUNTO DE DESCARGA%

        while modu(sigma(:,i)-sigmac(:,c))<modu(sigmad(:,d)-sigmac(:,c))
&& i<=(length(t)-1) %MIENTRAS LA DISTANCIA DEL PUNTO DE TENSIÓN AL CENTRO
ACTUAL DE PLASTIFICACIÓN SEA MENOR A LA DISTANCIA DEL PUNTO DE DESCARGA
ACTIVO AL CENTRO ACTUAL DE PLSATIFICACIÓN%

            qaux=abs((2/3*(sigma(1,i-1)-sigmad(1,d))^2+2*(sigma(2,i-1)-
sigmad(2,d))^2)/(2/3*(sigmac(1,c)-sigmad(1,d))*(sigma(1,i-1)-
sigmad(1,d))+2*(sigmac(2,c)-sigmad(2,d))*(sigma(2,i-1)-
sigmad(2,d)))*q0/2); %CALCULAMOS EL DIÁMETRO DE LA NUEVA CIRCUNFERENCIA
PROVISIONAL DE PLASTIFICACIÓN, QUE SE USARÁ PARA CALCULAR EL NUEVO
CENTRO%
            %CALCULAMOS EL CENTRO DE LA NUEVA CIRCUNFERENCIA PROVISIONAL DE
PLASTIFICACIÓN%
            c=c+1;
            sigmac(:,c)=sigmad(:,d)-qaux/2*(sigmad(:,d)-sigmac(:,c-
1))/(modu(sigmad(:,d)-sigmac(:,c-1)));

            if modu(sigma(:,i)-sigmac(:,c))<modu(sigmad(:,d)-sigmac(:,c)) %SI
EL NUEVO PUNTO DE TENSIONES CALCULADO ANTERIORMENTE ESTA DENTRO DE LA
CIRCUNFERENCIA PROVISIONAL DE PLASTIFICACIÓN%

                %SE CREA UN NUEVO PUNTO DE DESCARGA%
                d=d+1;
                sigmad(:,d)=sigma(:,i-1);
                q0=abs((modu(sigmad(:,d)-sigmad(:,d-1)))^2)/(2/3*(sigmac(1,c-1)-
sigmad(1,d-1))*(sigmad(1,d)-sigmad(1,d-1))+2*(sigmac(2,c-1)-
sigmad(2,d-1))*(sigmad(2,d)-sigmad(2,d-1)))*q0/2);
                %SE CALCULA EL DIÁMETRO DE LA NUEVA CIRCUNFERENCIA DE
PLASTIFICACIÓN%
            else %SI EL NUEVO PUNTO DE TENSIONES CALCULADO ANTERIORMENTE NO
ESTÁ DENTRO DE LA CIRCUNFERENCIA PROVISIONAL DE PLASTIFICACIÓN
                c=c-1; %SE BORRA DICHA CIRCUNFERENCIA AUXILIAR%

            end

            %CALCULAMOS UN NUEVO PUNTO DE TENSIONES USANDO EL MÉTODO RUNGE
KUTTA DE
            %CUARTO ORDEN
            K1=systd(t(i),sigma(:,i),sigmad(:,d),sigmac(:,c),q0,eps,esp);

K2=systd(t(i)+h/2,(sigma(:,i)+K1/2*h),sigmad(:,d),sigmac(:,c),q0,eps,e
sp);

K3=systd(t(i)+h/2,(sigma(:,i)+K2/2*h),sigmad(:,d),sigmac(:,c),q0,eps,e
sp);

```

```

K4=systd(t(i)+h, (sigma(:,i)+K3*h), sigmad(:,d), sigmac(:,c), q0, eps, esp);
    sigma(:,i+1)=sigma(:,i)+h*(K1+2*K2+2*K3+K4)/6;

    %          q0=abs(((modu(sigma(:,i)-sigmad(:,d)))^2)/(2/3*(sigmac(1,c)-
sigmad(1,d))*(sigma(1,i)-sigmad(1,d))+2*(sigmac(2,c)-
sigmad(2,d))*(sigma(2,i)-sigmad(2,d)))*q0/2);

        i=i+1; %INCREMENTAMOS INDICE DE PASO%

        while modu(sigma(:,i)-sigmac(:,c))>modu(sigmad(:,d)-sigmac(:,c))
&& d>1 && c>1
            %MIENTRAS QUE EL NUEVO PUNTO CALCULADO SALGA DE LA CIRCUNFERENCIA
DE
            %PLASTIFICACIÓN ACTIVA%
            d=d-1; %BORRAMOS EL PUNTO DE DESCARGA ACTIVO Y LA CIRCUNFERENCIA
DE PLASTIFICACIÓN ACTIVA, VOLVIENDO A LA ANTERIOR
            c=c-1;
            %EL PUNTO DE DESCARGA
            end

            if d>1 %SI EL PUNTO DE DESCARGA ACTIVO NO ES EL PRIMERO%
                %RECALCULAMOS LA CIRCUNFERENCIA DE PLASTIFICACIÓN QUE PASA A
SER
                %LA ACTUAL
                q0=abs(((modu(sigmad(:,d)-sigmad(:,d-
1))))^2)/(2/3*(sigmac(1,c-1)-sigmad(1,d-1))*(sigmad(1,d)-sigmad(1,d-
1))+2*(sigmac(2,c-1)-sigmad(2,d-1))*(sigmad(2,d)-sigmad(2,d-
1)))*q0/2);
                else %SI EL PUNTO DE DESCARGA ACTIVO ES EL PRIMERO%
                %RECALCULAMOS LA CIRCUNFERENCIA DE PLASTIFICACIÓN QUE PASA A
SER
                %LA ACTUAL
                q0=2*modu(sigmad(:,1));
                end
            end
        end
    end
    %SI SALIMOS DEL BUCLE, ES POR QUE LAS TENSIONES HAN SUPERADO EL PRIMER
%PUNTO DE DESCARGA%

    d=0; %PONEMOS EL PARÁMETRO DE DESCARGA A 0%
    if i<=(length(t)-1)
        %RESOLVEMOS EL SISTEMA CON LAS ECUACIONES DE DESCARGA, MEDIANTE MÉTODO
%RUNGE KUTTA PASO A PASO DE 4TH ORDEN
        K1=syst(t(i), sigma(:,i), eps, esp);
        K2=syst(t(i)+h/2, (sigma(:,i)+K1/2*h), eps, esp);
        K3=syst(t(i)+h/2, (sigma(:,i)+K2/2*h), eps, esp);
        K4=syst(t(i)+h, (sigma(:,i)+K3*h), eps, esp);
        sigma(:,i+1)=sigma(:,i)+h*(K1+2*K2+2*K3+K4)/6;
    end

```

```

if modu(sigma(:,i+1))<modu(sigma(:,i)) %SI LAS TENSIONES DEL NUEVO
PUNTO HAN DISMINUIDO RESPECTO AL ANTERIOR%
    d=1; %SE CREA EL PRIMER PUNTO DE DESCARGA%
end
i=i+1; %INCREMENTAMOS ÍNDICE DE PASO%
end
end

```

Syst2rk4d.m

```

function [t,sigma]=syst2rk4d(tini,tfin,sigma0,eps,h)
d=0; %PARÁMETRO DE DESCARGA%

sigma(:,1)=sigma0; %PUNTO INICIAL DE TENSIONES%

t=tini:h:tfin; %INICIALIZACIÓN DEL VECTOR T A PARTIR DEL PASO DE
INTEGRACIÓN H%

sigmac=[0;0]; %CENTRO DE CIRCUNFERENCIA DE PLASTIFICACIÓN INICIAL%
c=1; %CIRCUNFERENCIA DE PLASTIFICACIÓN ACTIVA%
i=1; %INDICE DE PASO PARA CADA INSTANTE DE T%
while i<=(length(t)-1) %MIENTRAS QUE NO LLEGUEMOS AL INSTANTE FINAL DE
T

    if d>=1 %SI SE HA DETECTADO DESCARGA%
        sigmad(:,1)=sigma(:,i-1); %EL PRIMER PUNTO DE DESCARGA ES EL PASO
ANTERIOR A LA DESCARGA%
        q0=2*modu(sigmad(:,1)); %DEFINIMOS EL MÓDULO EFECTIVO DE LA
PRIMERA CIRCUNFERENCIA DE PLASTIFICACIÓN
        %RESOLVEMOS EL SISTEMA CON LAS ECUACIONES DE DESCARGA, MEDIANTE
MÉTODO
        %RUNGE KUTTA PASO A PASO DE 4TH ORDEN
        K1=syst2d(t(i),sigma(:,i),sigmad(:,d),sigmac(:,c),q0,eps);

        K2=syst2d(t(i)+h/2,(sigma(:,i)+K1/2*h),sigmad(:,d),sigmac(:,c),q0,eps)
;

        K3=syst2d(t(i)+h/2,(sigma(:,i)+K2/2*h),sigmad(:,d),sigmac(:,c),q0,eps)
;

        K4=syst2d(t(i)+h,(sigma(:,i)+K3*h),sigmad(:,d),sigmac(:,c),q0,eps);
        sigma(:,i+1)=sigma(:,i)+h*(K1+2*K2+2*K3+K4)/6;

        %
        q0=abs((2/3*(sigma(1,i)-sigmad(1,d))^2+2*(sigma(2,i)-
sigmad(2,d))^2)/(2/3*(sigmac(1,c)-sigmad(1,d))*(sigma(1,i)-
sigmad(1,d))+2*(sigmac(2,c)-sigmad(2,d))*(sigma(2,i)-
sigmad(2,d)))*q0/2);

        i=i+1; %INCREMENTAMOS INDICE DE PASO%
    while modu(sigma(:,i))<modu(sigmad(:,1)) && i<=(length(t)-1)
%MIENTRAS QUE LAS TENSIONES NO SEAN MAYORES AL PRIMER PUNTO DE DESCARGA%

```

```

while modu(sigma(:,i)-sigmac(:,c))<modu(sigmad(:,d)-sigmac(:,c))
&& i<=(length(t)-1) %MIENTRAS LA DISTANCIA DEL PUNTO DE TENSIÓN AL CENTRO
ACTUAL DE PLASTIFICACIÓN SEA MENOR A LA DISTANCIA DEL PUNTO DE DESCARGA
ACTIVO AL CENTRO ACTUAL DE PLSATIFICACIÓN%

    qaux=abs((2/3*(sigma(1,i-1)-sigmad(1,d))^2+2*(sigma(2,i-1)-
sigmad(2,d))^2)/(2/3*(sigmac(1,c)-sigmad(1,d))*(sigma(1,i-1)-
sigmad(1,d))+2*(sigmac(2,c)-sigmad(2,d))*(sigma(2,i-1)-
sigmad(2,d)))*q0/2); %CALCULAMOS EL DIÁMETRO DE LA NUEVA CIRCUNFERENCIA
PROVISIONAL DE PLASTIFICACIÓN, QUE SE USARÁ PARA CALCULAR EL NUEVO
CENTRO%
    %CALCULAMOS EL CENTRO DE LA NUEVA CIRCUNFERENCIA PROVISIONAL DE
PLASTIFICACIÓN%
    c=c+1;
    sigmac(:,c)=sigmad(:,d)-qaux/2*(sigmad(:,d)-sigmac(:,c-
1))/(modu(sigmad(:,d)-sigmac(:,c-1)));

    if modu(sigma(:,i)-sigmac(:,c))<modu(sigmad(:,d)-sigmac(:,c)) %SI
EL NUEVO PUNTO DE TENSIONES CALCULADO ANTERIORMENTE ESTA DENTRO DE LA
CIRCUNFERENCIA PROVISIONAL DE PLASTIFICACIÓN%

        %SE CREA UN NUEVO PUNTO DE DESCARGA%
        d=d+1;
        sigmad(:,d)=sigma(:,i-1);
        q0=abs(((modu(sigmad(:,d)-sigmad(:,d-1)))^2)/(2/3*(sigmac(1,c-1)-
sigmad(1,d-1))*(sigmad(1,d)-sigmad(1,d-1))+2*(sigmac(2,c-1)-
sigmad(2,d-1))*(sigmad(2,d)-sigmad(2,d-1)))*q0/2);
        %SE CALCULA EL DIÁMETRO DE LA NUEVA CIRCUNFERENCIA DE
PLASTIFICACIÓN%
        else %SI EL NUEVO PUNTO DE TENSIONES CALCULADO ANTERIORMENTE NO
ESTÁ DENTRO DE LA CIRCUNFERENCIA PROVISIONAL DE PLASTIFICACIÓN
            c=c-1; %SE BORRA DICHA CIRCUNFERENCIA AUXILIAR%
        end
        %CALCULAMOS UN NUEVO PUNTO DE TENSIONES USANDO EL MÉTODO RUNGE
KUTTA DE
        %CUARTO ORDEN
        K1=syst2d(t(i),sigma(:,i),sigmad(:,d),sigmac(:,c),q0,eps);

K2=syst2d(t(i)+h/2,(sigma(:,i)+K1/2*h),sigmad(:,d),sigmac(:,c),q0,eps)
;

K3=syst2d(t(i)+h/2,(sigma(:,i)+K2/2*h),sigmad(:,d),sigmac(:,c),q0,eps)
;

K4=syst2d(t(i)+h,(sigma(:,i)+K3*h),sigmad(:,d),sigmac(:,c),q0,eps);
        sigma(:,i+1)=sigma(:,i)+h*(K1+2*K2+2*K3+K4)/6;

        %
        q0=abs(((modu(sigma(:,i)-sigmad(:,d)))^2)/(2/3*(sigmac(1,c)-
sigmad(1,d))*(sigma(1,i)-sigmad(1,d))+2*(sigmac(2,c)-
sigmad(2,d))*(sigma(2,i)-sigmad(2,d)))*q0/2);

        i=i+1; %INCREMENTAMOS INDICE DE PASO%

```

```

    while modu(sigma(:,i)-sigmac(:,c))>modu(sigmad(:,d)-sigmac(:,c))
&& d>1 && c>1
    %MIENTRAS QUE EL NUEVO PUNTO CALCULADO SALGA DE LA CIRCUNFERENCIA
DE
    %PLASTIFICACIÓN ACTIVA%
    d=d-1; %BORRAMOS EL PUNTO DE DESCARGA ACTIVO Y LA CIRCUNFERENCIA
DE PLASTIFICACIÓN ACTIVA, VOLVIENDO A LA ANTERIOR
    c=c-1;
    %EL PUNTO DE DESCARGA
end

    if d>1 %SI EL PUNTO DE DESCARGA ACTIVO NO ES EL PRIMERO%
        %RECALCULAMOS LA CIRCUNFERENCIA DE PLASTIFICACIÓN QUE PASA A
SER
        %LA ACTUAL
        q0=abs((modu(sigmad(:,d)-sigmad(:,d-
1)))^2)/(2/3*(sigmac(1,c-1)-sigmad(1,d-1))*(sigmad(1,d)-sigmad(1,d-
1))+2*(sigmac(2,c-1)-sigmad(2,d-1))*(sigmad(2,d)-sigmad(2,d-
1)))*q0/2);
    else %SI EL PUNTO DE DESCARGA ACTIVO ES EL PRIMERO%
        %RECALCULAMOS LA CIRCUNFERENCIA DE PLASTIFICACIÓN QUE PASA A
SER
        %LA ACTUAL
        q0=2*modu(sigmad(:,1));
    end
end
end

end
%SI SALIMOS DEL BUCLE, ES POR QUE LAS TENSIONES HAN SUPERADO EL PRIMER
%PUNTO DE DESCARGA%

d=0; %PONEMOS EL PARÁMETRO DE DESCARGA A 0%
if i<=(length(t)-1)
%RESOLVEMOS EL SISTEMA CON LAS ECUACIONES DE DESCARGA, MEDIANTE MÉTODO
%RUNGE KUTTA PASO A PASO DE 4TH ORDEN
K1=syst2(t(i),sigma(:,i),eps);
K2=syst2(t(i)+h/2,(sigma(:,i)+K1/2*h),eps);
K3=syst2(t(i)+h/2,(sigma(:,i)+K2/2*h),eps);
K4=syst2(t(i)+h,(sigma(:,i)+K3*h),eps);
sigma(:,i+1)=sigma(:,i)+h*(K1+2*K2+2*K3+K4)/6;

    if modu(sigma(:,i+1))<modu(sigma(:,i)) %SI LAS TENSIONES DEL NUEVO
PUNTO HAN DISMINUIDO RESPECTO AL ANTERIOR%
        d=1; %SE CREA EL PRIMERO PUNTO DE DESCARGA%
    end
    i=i+1; %INCREMENTAMOS ÍNDICE DE PASO%
end
end
end

```

Systd.m

```

function dsigma=systd(t,sigma,sigmad,sigmac,q0,eps,esp)
global K na E G

q=abs(modu(sigma-sigmad))^2/(2/3*(sigmac(1)-sigmad(1))*(sigma(1)-sigmad(1))+2*(sigmac(2)-sigmad(2))*(sigma(2)-sigmad(2)))/2;
%CÁLCULO DEL NUEVO DIÁMETRO, DIVIDIA ENTRE EL DIAMETRO ANTERIOR (PARA EVITAR PROBLEMAS NUMERICOS SI Q0
%ES CERCANO A CERO%
n=[4/(3*q)*(sigma(1)-sigmad(1))-4/3*(sigmac(1)-sigmad(1));4/q*(sigma(2)-sigmad(2))-4*(sigmac(2)-sigmad(2))];
%DIRECCIÓN DE ENDURECIMIENTO MULTIPLICADO POR Q0 (PARA EVITAR PROBLEMAS NUMÉRICOS%
aux=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(q/2)^(1/na-1))*(q0)^(1/na-3);
%FLUJO PLÁSTICO DE LA NUEVA CIRCUNFERENCIA DE PLASTIFICACIÓN, DIVIDO ENTRE
%Q0^2

B=[(aux*n(1)^2+1/E) (aux*n(1)*n(2));(aux*n(1)*n(2)) (aux*n(2)^2+1/G)];
%MONTAJE DE LA MATRIZ B, EL 1/Q0^2 QUE MULTIPLICA A AUX SE COMPENSA CON EL Q0 QUE
%MULTIPLICA EN CADA COMPONENTE DEL VECTOR N

%CÁLCULO DE LA DERIVADA DEL VECTOR DEFORMACIÓN%
%DIFERENCIACIÓN SIMBÓLICA%
%LINEAL%
if esp==0
deps=@(t)[eps.e;0];

%ESPIRAL%
else
deps=@(t)[eps.e*(cos(t)-t*sin(t)) ; eps.g*(sin(t)+t*cos(t))];
end
% DIFERENCIACIÓN NUMÉRICA %
% h=1e-10;
%
deps=@(t)[(eps.e(t+h)-eps.e(t-h))/(2*h);(eps.g(t+h)-eps.g(t-h))/(2*h)];
dsigma=B\deps(t);
Syst2d.m
function dsigma=syst2d(t,sigma,sigmad,sigmac,q0,eps)
global K na E G

q=abs((modu(sigma-sigmad))^2/(2/3*(sigmac(1)-sigmad(1))*(sigma(1)-sigmad(1))+2*(sigmac(2)-sigmad(2))*(sigma(2)-sigmad(2))))/2;
%CÁLCULO DEL NUEVO DIÁMETRO, DIVIDIA ENTRE EL DIAMETRO ANTERIOR (PARA EVITAR PROBLEMAS NUMERICOS SI Q0
%ES CERCANO A CERO%
n=[4/(3*q)*(sigma(1)-sigmad(1))-4/3*(sigmac(1)-sigmad(1));4/q*(sigma(2)-sigmad(2))-4*(sigmac(2)-sigmad(2))];

```

```

%DIRECCIÓN DE ENDURECIMIENTO MULTIPLICADO POR Q0 (PARA EVITAR PROBLEMAS
%NUMÉRICOS%
aux=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*((q/2)^(1/na-
1))* (q0)^(1/na-3);
%FLUJO PLÁSTICO DE LA NUEVA CIRCUNFERENCIA DE PLASTIFICACIÓN, DIVIDO
ENTRE
%Q0^2

B=[ (aux*n(1)^2+1/E) (aux*n(1)*n(2)); (aux*n(1)*n(2))
(aux*n(2)^2+1/G)];
%MONTAJE DE LA MATRIZ B, EL 1/Q0^2 QUE MULTIPLICA A AUX SE COMPENSA
CON EL Q0 QUE
%MULTIPLICA EN CADA COMPONENTE DEL VECTOR N

%CÁLCULO DE LA DERIVADA DEL VECTOR DEFORMACIÓN%
deps=@(t)[(-eps.e*sin(t));(eps.g*cos(t))];

dsigma=B\deps(t); %CALCULAMOS LA DERIVADA DEL VECTOR SIGMA A PARTIR DE
LA MATRIZ b Y LA DERIVADA DEL VECTOR DEFORMACIÓN

%se pueden meter la funcion y derivarla o meter directamente la funcion
%derivada %

```

Systcl.m

```

function dsigma=systcl(t,sigma,eps,esp)
global K na E G

%FUNCIÓN DE INTEGRACIÓN DE ECUACIONES DIFERENCIALES SIMPLIFICADA PARA
EL
%TRAMO1

Q=sqrt(2/3*sigma(1)^2+2*sigma(2)^2); %SE DEFINE EL MÓDULO EFECTIVO
DE TENSIONES%
aux=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(Q^(1/na-3)); %SE DEFINE
UNA VARIABLE AUXILIAR= PHI(Q)/Q^2 PARA REDUCIR POSIBLES PROBLEMAS
NUMÉRICOS%
%SE DEFINE LA MATRIZ DE LAS DERIVADAS PARCIALES%
A=1/(1/E*1/G+(aux)*((4*sigma(2)^2)/E+(4*sigma(1)^2)/(9*G)))*[1/G+(4*s
igma(2)^2)*aux -4/3*sigma(1)*sigma(2)*aux;-4/3*sigma(1)*sigma(2)*aux
1/E+4/9*(sigma(1)^2)*aux];

%CÁLCULO DE LA DERIVADA DEL VECTOR DEFORMACIÓN%
%MÉTODO CONOCIENDO LA DERIVADA EXACTA%
%CARGA LINEAL%
if esp==0
deps=@(t)[eps.e;0];
%CARGA EN ESPIRAL%
else

```

```

deps=@(t) [eps.e*(cos(t)-t*sin(t)) ; eps.g*(sin(t)+t*cos(t))];
end
% DIFERENCIACIÓN NUMÉRICA %
% h=1e-10;
%
%      deps=@(t) [(eps.e(t+h)-eps.e(t-h))/(2*h); (eps.g(t+h)-eps.g(t-
h))/(2*h)];
dsigma=A*deps(t);      %CALCULAMOS LA DERIVADA DEL VECTOR SIGMA A PARTIR
DE LA MATRIZ A Y LA DERIVADA DEL VECTOR DEFORMACIÓN

dQ=dsigma'*[2/3*sigma(1)/Q; 2*sigma(2)/Q];      %CÁLCULO DE LA DERIVADA
DEL MÓDULO EFECTIVO DE TENSIONES%
%CONDICIÓN DE DESCARGA SIMPLIFICADA,
% SI SE DETECTA QUE EL MATERIAL VA A DESCARGAR, ESTE PASO DE INTEGRACIÓN
SOLO TIENE INFLUENCIA ELÁSTICA%
if dQ<0
aux=0;
A=1/(1/E*1/G+(aux)*((4*sigma(2)^2)/E+(4*sigma(1)^2)/(9*G)))*[1/G+(4*s
igma(2)^2)*aux      -4/3*sigma(1)*sigma(2)*aux;-4/3*sigma(1)*sigma(2)*aux
1/E+4/9*(sigma(1)^2)*aux];
dsigma=A*deps(t);
else
end

```

Systcl2.m

```

function dsigma=systcl2(t,sigma,eps)
global K na E G

%FUNCIÓN DE INTEGRACIÓN DE ECUACIONES DIFERENCIALES SIMPLIFICADA PARA
EL
%TRAMO2

Q=sqrt(2/3*sigma(1)^2+2*sigma(2)^2);      %SE DEFINE EL MÓDULO EFECTIVO
DE TENSIONES%
aux=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(Q^(1/na-3));      %SE DEFINE
UNA VARIABLE AUXILIAR= PHI(Q)/Q^2 PARA REDUCIR POSIBLES PROBLEMAS
NUMÉRICOS%
%SE DEFINE LA MATRIZ DE LAS DERIVADAS PARCIALES%
A=1/(1/E*1/G+(aux)*((4*sigma(2)^2)/E+(4*sigma(1)^2)/(9*G)))*[1/G+(4*s
igma(2)^2)*aux      -4/3*sigma(1)*sigma(2)*aux;-4/3*sigma(1)*sigma(2)*aux
1/E+4/9*(sigma(1)^2)*aux];

%CÁLCULO DE LA DERIVADA DEL VECTOR DEFORMACIÓN%
%MÉTODO CONOCIENDO LA DERIVADA EXACTA%
deps=@(t) [(-eps.e*sin(t)); (eps.g*cos(t))];
% DIFERENCIACIÓN NUMÉRICA %
% h=1e-10;
%
%      deps=@(t) [(eps.e(t+h)-eps.e(t-h))/(2*h); (eps.g(t+h)-eps.g(t-
h))/(2*h)];

```

```

dsigma=A*deps(t); %CALCULAMOS LA DERIVADA DEL VECTOR SIGMA A PARTIR DE
LA MATRIZ A Y LA DERIVADA DEL VECTOR DEFORMACIÓN

dQ=dsigma'*[2/3*sigma(1)/Q; 2*sigma(2)/Q]; %CÁLCULO DE LA DERIVADA DEL
MÓDULO EFECTIVO DE TENSIONES%
%CONDICIÓN DE DESCARGA SIMPLIFICADA,

if dQ<0 %CONDICIÓN DE DESCARGA SIMPLIFICADA,
% SI SE DETECTA QUE EL MATERIAL VA A DESCARGAR, ESTE PASO DE INTEGRACIÓN
SOLO TIENE INFLUENCIA ELÁSTICA%
aux=0;
A=1/(1/E*1/G+(aux)*((4*sigma(2)^2)/E+(4*sigma(1)^2)/(9*G)))*[1/G+(4*sigma(2)^2)*aux -4/3*sigma(1)*sigma(2)*aux;-4/3*sigma(1)*sigma(2)*aux
1/E+4/9*(sigma(1)^2)*aux];
dsigma=A*deps(t);
else

end

%se pueden meter la funcion y derivarla o meter directamente la funcion
%derivada %

```

Curvaen.m

```

clear,clc
close all

%CURVA E - N %
%PARÁMETROS DE LA CURVA E-N PARA EL ALUMINIO%
sf=1231;
ef=0.26;
E=71e3;
b=-0.122;
c=-0.806;
N=250; %INSTERTAR NÚMERO DE CICLOS%
epsilon=sf/E*(2*N)^b+ef*(2*N)^c %DEFORMACIÓN/2 QUE ROMPE A ESE NÚMERO
DE CICLOS%

```

Dp.m

```

function depur=dp(angsig)
%FUNCIÓN PARA REPRESENTAR UN ÁNGULO RELATIVO (DESDE 0 HASTA PI) A
ÁNGULO
%RECORRIDO TOTAL
d=1;
depur(d)=angsig(d); %ESTE ANGULO RECORRIDO TOTAL LO IREMOS ALMACENANDO
EN UNA NUEVA VARIABLE%
a=2; %SE ALMACENA EL VALOR INICIAL Y SE EMPIEZA A
CONTAR DESDE EL PASO 2%

```

```

while a<length(angsig)           %MIENTRAS EL ÍNDICE SEA MENOR AL INDICE
FINAL DEL VECTOR DE ÁNGULOS
    while angsig(a)>=angsig(a-1) && a<length(angsig) %MIENTRAS QUE EL
VALOR DEL ÁNGULO RELATIVO AUMENTE%
        depur(a)=depur(d)+angsig(a); %EL ÁNGULO TOTAL ES EL ÁNGULO DEL
PRIMER PUNTO DE RETROCESO MAS EL ÁNGULO RELATIVO%
        a=a+1;
    end
    d=a-1; %CUANDO EL VALOR DEL ÁNGULO DISMINUYE, EL VALOR
ANTERIOR SE CONVIERTE EN EL NUEVO PUNTO DE RETROCESO%
    while angsig(a)<angsig(a-1) && a<length(angsig)
        depur(a)=depur(d)+pi-angsig(a); %MIENTRAS EL VALOR DEL ÁNGULO
RELATIVO SIGA BAJANDO, EL ÁNGULO TOTAL SERÁ EL ÁNGULO
%DEL PUNTO DE DESCARGA MAS (PI-ÁNGULO RELATIVO)
        a=a+1;
    end
    d=a-1; %CUANDO EL VALOR DEL ÁNGULO VUELVE A CRECER, EL VALOR
ANTERIOR SE CONVIERTE EN EL NUEVO PUNTO DE DESCARGA%

end

```

Flujo.m

```

function phi=flujo(K,na,sigma)
Q=(2/3*abs(sigma(1,:)).^2+2*abs(sigma(2,:)).^2).^(1/2);
phi=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(Q^(1/na-1));
%REPRESENTACIÓN DEL FLUJO PLÁSTICO EN FUNCIÓN DE Q,K,na

```

Flujod.m

```

function phid=flujod(K,na,q)

phid=((sqrt(3/2))^(1/na+1))/na*((1/K)^(1/na))*(q^(1/na-1))

```

Modu.m

```

function Q=modu(sigma)
Q=(2/3*abs(sigma(1,:)).^2+2*abs(sigma(2,:)).^2).^(1/2);
%FUNCIÓN PARA EL CÁLCULO DEL MÓDULO EFECTIVO DE TENSIONES%

```

Ordenados.m

```

%SCRIPT PARA SIMULACIÓN AUTOMATIZADA CALCULANDO PARA VARIOS VALORES DE
EMAX
%ANÁLISIS DE LA TENDENCIA EMAX-NÚMERO DE CICLOS PARA LA ESTABILIZACIÓN
%ANÁLISIS DE LA TENDENCIA EMAX-QMAX EN LA ESTABILIZACIÓN

clear,clc
close all          %LIMPIADO DE MEMORIA%
format long
tic              %EMPEZAMOS A CONTAR EL TIEMPO DE CÁLCULO%
global K na E G  %CONSTANTES GLOBALES DEL MATERIAL%

%CONJUNTO DE VARIABLES QUE DEFINEN LOS ENSAYOS QUE SE REALIZARAN, SÓLO
ES NECESARIO
%CAMBIAR ESTAS VARIABLES PARA REALIZAR UNA SIMULACIÓN DISTINTA

material=1;      %ELECCION DE MATERIAL PARA LAS SIMULACIONES, ALUMINIO 1,
ACERO 0
esp=1;          %ELECCION DEL TIPO DE CARGA HASTA LLEGAR AL MÁXIMO DE
DEFORMACIÓN, 0 UNIAXIAL, 1 APROXIMACIÓN EN ESPIRAL
h=0.001;        %TOLERANCIA PARA LA VISUALIZACIÓN DE LAS GRÁFICAS Y CÁLCULO
DE LOS MÁXIMOS%
he=0.00001;     %INCREMENTO DE EMAX ENTRE SIMULACIONES%
Nccarga=1;      %NUMERO DE CICLOS PARA LLEGAR A DEFORMACIÓN MÁXIMA%
emaxf=0.0075;   %MÓDULO EFECTIVO DE DEFORMACIÓN SIMULACIÓN FINAL%
emaxi=0.0175;   %MÓDULO EFECTIVO DE DEFORMACIÓN SIMULACIÓN INICIAL%
Nc=300;         %NÚMERO DE CICLOS TOTALES%

if material==0  %ELECCIÓN DE MATERIAL PARA LAS SIMULACIONES%

% ACERO
K=1655;         %PARÁMETROS PARA EL ACERO%
na=0.131;
E=207e3;
v=0.3;
G=E/(2*(1+v));

else

% ALUMINIO%
K=852;          %PARÁMETROS PARA EL ALUMINIO%
na=0.074;
E=71e3;
v=0.33;
G=E/(2*(1+v));

end

tcarga=Nccarga*2*pi; %CREAMOS LA VARIABLE T, QUE RESULTA DE
MULTIPLICAR 2PI POR EL NUMERO DE CICLOS CORRESPONDIENTE%
tspan=0:h:tcarga;   %INICIALIZACIÓN DEL VECTOR T PARA EL TRAMO DE
CARGA%

```

```

pos=1;
for emax=emaxi:he:emaxf

mepsilon=emax*sqrt(2/3); %DEFORMACIÓN EFECTIVA EN COMPONENTES EPSILON
Y GAMMA PARA DESFASE DE 90°, ES DECIR, EMAX SIEMPRE ES CONSTANTE%
mgamma=emax*sqrt(2);

if esp==1 %SELECCIÓN DEL TIPO DE CARGA%

% CARGA EN ESPIRAL%
eps.e=@(t)mepsilon*t.*cos(t)/tcarga; %DEFINICIÓN DE FUNCIÓN DE
APROXIMACIÓN EN ESPIRAL. LOS TERMINOS DE ARRIBA SE METEN SI QUEREMOS
DERIVAR NUMERICAMENTE%
eps.g=@(t)mgamma*t.*sin(t)/tcarga;
epsc.e=mepsilon/tcarga; %ESTOS TERMINOS SE UTILIZAN
SI PODEMOS CALCULAR LA DERIVADA EXACTA, COMO COEFICIENTES CONSTANTES%
epsc.g=mgamma/tcarga;

else
% CARGA LINEAL%

eps.e=@(t)mepsilon/tcarga*t; %DEFINICIÓN DE FUNCIÓN DE CARGA LINEAL
UNIAXIAL (TRACCIÓN). %
eps.g=@(t)0*t; %SE INTRODUCEN EN LA FUNCION DE
INTEGRACIÓN ANÁLOGAMENTE A LAS DE ARRIBA%

epsc.e=mepsilon/tcarga;
epsc.g=0;

end

options=odeset('RelTol',1e-8,'AbsTol',1e-8); %DEFINICIÓN DE
TOLERANCIAS DE INTEGRACIÓN%
[tspan,sigma]=ode45(@systcl,tspan,[0;0],options,epsc,esp);
%INTEGRACIÓN DE LAS ECUACIONES DE FLUJO PLÁSTICO EN DERIVADAS
PARCIALES PARA EL PRIMER TRAMO%

tspan2=(tcarga):h:Nc*2*pi; %SE INICIALIZA EL VECTOR T PARA LA SEGUNDA
PARTE DE LA SIMULACIÓN%

eps2.e=@(t)mepsilon*cos(t); %SE DEFINE LA ROTACIÓN DEL VECTOR
DEFORMACIONES MANTENIENDO SU MÓDULO EFECTIVO CONSTANTE%
eps2.g=@(t)mgamma*sin(t); %SE DEFINEN LAS FUNCIONES ANALOGAMENTE A
COMO HICIMOS ANTES, PARA DERIVACIÓN NUMÉRICA O PARA CALCULO DE
COEFICIENTES CONSTANTES SABRIENDO LA DERIVADA EXACTA%

epsc2.e=mepsilon;
epsc2.g=mgamma;

```

```

[tspan2,sigma2]=ode45(@systc12,tspan2,[sigma(end,1);sigma(end,2)],opti
ons,epsc2); %INTEGRACIÓN DE LAS ECUACIONES DE FLUJO PLÁSTICO EN
DERIVADAS PARCIALES PARA EL SEGUNDO TRAMO%

epsilon=[eps.e(tspan) ;eps2.e(tspan2)]; %SE CALCULAN LAS DEFORMACIONES
EN COMPONENTES EPSILON Y GAMMA PARA CADA INSTANTE DEL VECTOR T%
gamma=[eps.g(tspan) ;eps2.g(tspan2)]; %SE AGRUPAN A SU VEZ LOS
TRAMOS 1 Y 2%

sigma=[sigma; sigma2]; %SE AGRUPAN LAS TENSIONES DE LOS TRAMOS 1 Y
2 , POR COMPONENTES SIGMA Y TAU%
tspan=[tspan ;tspan2]; %SE AGRUPA EL VECTOR DEL PARÁMETRO T DE LOS
TRAMOS 1 Y 2%

Q=modu(sigma'); %SE CALCULA EL MÓDULO EFECTIVO DE LAS TENSIONES
PARA CADA PUNTO CALCULADO%
dm=diff(Q); %SE CALCULA EL VECTOR DE DIFERENCIAS DEL MÓDULO DE
SIGMA, PARA VER SI HA DECRECIDO EN ALGÚN PUNTO%

N=tspan/(2*pi); %DESHACEMOS EL CAMBIO DE VARIABLE DE T A NÚMERO DE
CICLOS TOTALES, PARA CADA INSTANTE EVALUADO%

[m,i]=max(Q); %SE CALCULA EL MÁXIMO DEL MÓDULO EFECTIVO DE
TENSIONES, Y EL ÍNDICE EN EL CUAL APARECE%
%ESTE VALOR SERÁ MÁXIMO JUSTO ANTES DE LA ESTABILIZACIÓN, LO TOMAREMOS
COMO ORDEN DE MAGNITUD%
i=tspan(i)/(2*pi); %CAMBIAMOS LA VARIABLE ÍNDICE A NÚMERO DE CICLOS

tabeps(pos)=emax; %GUARDAMOS LOS VALORES DE EMAX, NEST, Y QMAX PARA
CADA SIMULACIÓN%
Nest(pos)=i;
Qmax(pos)=m;
pos=pos+1;

end
toc
figure (1)
plot(tabeps,Qmax) %REPRESENTAMOS LAS TENDENCIAS E-QMAX Y E-NEST
title('EVOLUCIÓN DEL NUMERO DE CICLOS EN EL CUAL SE ESTABILIZA Q')
xlabel('MÓDULO DE EPSILON')
ylabel('NÚMERO DE CICLOS NECESARIO')

figure (2)
plot(tabeps,Nest)
xlabel('MÓDULO DE EPSILON')
ylabel('MÓDULO DE SIGMA Q')
title('EVOLUCIÓN DEL VALOR DE Q EN LA ESTABILIZACIÓN')

save('tabl.txt','tabeps','Nest','Qmax','-ascii') %ALMACENAMOS LOS
DATOS, PUES EL TIEMPO DE EJECUCION DE ESTE
%SCRIPT ES ALTO

```

Ajuste.m

```
%SCRIPT PARA AJUSTAR LAS TENDENCIAS EMAX-QMAX Y EMAX-NEST DE FORMAS
LINEAL
%Y EXPONENCIAL RESPECTIVAMENTE
%SE DEBEN EJECUTAR DESPUES DE ORDENADATOS.M, O TENER LOS FICHEROS QUE
ESTE
%GENERA

clear,clc
close all

A=load('tabl.txt'); %CARGA DE LOS DATOS ALMACENADOS POR EL SCRIPT
ORDENADATOS.M

tabeps=A(1,:); %ORGANIZACIÓN DE LOS DATOS PARA CADA VARIABLE%
Nest=A(2,:);
Qmax=A(3,:);

fact=100; %FACTOR DE MULTIPLICACIÓN PARA MEJOR LA CONVERGENCIA%

tabepsch=tabeps*fact; %MULTIPLICAMOS EMAX POR UN FACTOR DE 100 PARA
FACILITAR
%LA CONVERGENCIA DEL AJUSTE EXPONENCIAL

fun=@(coef,x)coef(1)*(coef(2)).^(x); %DEFINICION DE UNA FUNCIÓN DE
AJUSTE TIPO EXPONENCIAL%

coef0=[10;10]; %VALORES INICIALES DE LOS COEFICIENTES%

coef = lsqcurvefit(fun,coef0,tabepsch,Nest); %CÁLCULO DE LOS
COEFICIENTES PARA NUESTROS DATOS (EMAX, NEST)%

Nest2=fun(coef,tabepsch); %CÁLCULO DEL NUMERO DE CICLOS NEST SEGÚN EL
AJUSTE%

fun2=@(coef2,x)coef2(1).*x+coef2(2); %DEFINICIÓN DE UNA FUNCIÓN DE
AJUSTE TIPO LINEAL%

coef20=[1,1]; %VALORES INICIALES DE LOS COEFICIENTES%

coef2 = lsqcurvefit(fun2,coef20,tabeps,Qmax);
Qmax2=fun2(coef2,tabeps);
%COMPARACIÓN GRÁFICA ENTRE EL AJUSTE REALIZADO, Y LOS VALORES
NÚMERICOS
%OBTENIDOS POR INTEGRACIÓN%
%GRÁFICA EMAX-NEST%
figure (1)
plot(tabepsch/fact,Nest2,'b',tabeps,Nest,'r')
title('Evolución del número de ciclos frente a deformaciones
impuestas')
ylabel('\bf N_{c} [Ciclos]')
```

```

xlabel('\bf Norma efectiva de deformaciones')
legend('Ajuste Exponencial','Datos Numéricos','location','north')

error1=max(abs(Nest2-Nest)/Nest)*100; %ERROR MÁXIMO COMETIDO CON LA
APROXIMACIÓN EN TANTO POR CIENTO%
er=num2str(error1);
ab=[er '%'];
disp(ab)
%GRÁFICA EMAX-QMAX%
figure (2)
plot(tabeps,Qmax2,'b',tabeps,Qmax,'r')
ylabel('\bf Q [MPa]')
xlabel('\bf Norma efectiva de deformaciones')
title('Evolución del valor de Q según deformaciones impuestas')
legend('Ajuste Lineal','Datos numéricos','location','north')

error2=max(abs(Qmax2-Qmax)/Qmax)*100; %ERROR MÁXIMO COMETIDO CON LA
APROXIMACIÓN EN TANTO POR CIENTO%

```

Ajuste2.m

```

%SCRIPT QUE SE EJECUTA DESPUES DE ESPIRAL.M%
%PERMITE CALCULAR UN AJUSTE LOGARITMICO EN LA EVOLUCIÓN N-Q PARA
%DEFORMACIONES ALTAS EN LAS QUE EL VECTOR Q NO SE ESTABILIZA DE FORMA
%TEMPRANA

tc=round(tcarga/h)+1/h;
N1=N(tc:end,:); %INICIALIZAMOS LOS VECTORES N Y Q A PARTIR DE CUANDO
TERMINA LA CARGA, CON UN CIERTO MARGEN
Q1=Q(:,tc:end);

fun=@(coef,N1)coef(1)*log(N1-coef(2))+coef(3); %DEFINICION DE UNA
FUNCIÓN DE AJUSTE TIPO LOGARÍTMICA%

coef0=[1,1,10]; %VALORES INICIALES DE LOS COEFICIENTES%

coef = lsqcurvefit(fun,coef0,N1,Q1); %CÁLCULO DE LOS COEFICIENTES
PARA NUESTROS DATOS (N1,Q1)%

Q1=fun(coef,N1);
Q1=real(Q1); %CÁLCULO DE LA GRAFICA N-Q SEGÚN EL AJUSTE%

Q2=Q(tc:end);
error=Q1-Q2;
error=error./Q2;
error=abs(error);
[error,i]=max(error); %CÁLCULO DEL ERROR MAXIMO QUE SE COMETE CON EL
AJUSTE%

%COMPARACIÓN GRÁFICA ENTRE EL AJUSTE REALIZADO, Y LOS VALORES
NÚMERICOS
%OBTENIDOS POR INTEGRACIÓN%

figure (9)

```

```
plot(N,Q,'r',N1,Q1,'b')
title('\bf Evolución del módulo efectivo de tensiones según el número
de ciclos')
xlabel('\bf N_{c} [Ciclos]')
ylabel('\bf Norma efectiva de tensiones [MPa]')
legend('Datos Numéricos','Ajuste Logarítmico','location','best')
```