

DEPARTAMENTO DE INGENIERIA MECANICA

ESCUELA TECNICA SUPERIOR DE INGENIERIA



INGENIARITZA MEKANIKOA SAILA

BILBOKO INGENIARITZA GOI ESKOLA TEKNIKOA

Master en Ingeniería Mecánica: Diseño y Fabricación

Proyecto Fin de Master

Influencia de las propiedades temporales del tensor de tensiones en la precisión de los métodos de análisis de fatiga multiaxial

Alumno: Ondiz Zarraga Río

Director del Trabajo: Rafael Avilés

Curso 2011 - 2012

Índice

Resumen

1. Introducción	1
2. Objetivos del trabajo	2
3. Estado del arte	2
4. Implementación de métodos	8
5. Comparación de métodos	22
6. Conclusiones	27
7. Referencias	28

Anexo: Programa

Anexo: CD

RESUMEN

Se ha desarrollado un programa en Matlab que, por un lado, aplica los criterios de fallo de fatiga multiaxial clásicos, de enfoque global y de plano crítico y, por otro, representa gráficamente la evolución de los vectores de tensión en un plano dado y aplica y representa los criterios del círculo y de la elipse.

Se ha hecho una comparación de los resultados obtenidos con los diferentes métodos para los casos de tensión uniaxial de componente media nula y no nula, torsión, carga estática, tensión biaxial síncrona y tensión aleatoria.

1. Introducción:

En los últimos se ha invertido mucho tiempo y esfuerzo en desarrollar un criterio de fallo para fatiga multiaxial general. Tristemente, este objetivo no ha podido lograrse y actualmente existen múltiples métodos, sin ser ninguno superior a los demás y con su campo de aplicación específico.

Podría pensarse que esta carencia puede suplirse gracias a la experimentación, pero realizar ensayos para caracterizar perfectamente el fenómeno implicaría ensayar probetas y piezas bajo tensiones alternas, con tensión media no nula, síncronas y asíncronas, con direcciones principales fijas o variables, (Fig 1) ... No resulta viable.

Es por ello, que es importante conocer las ventajas y desventajas de los métodos existentes, para poder aplicar en cada caso el método que mejor se ajuste al problema y evitar obtener resultados erróneos y gastos innecesarios.

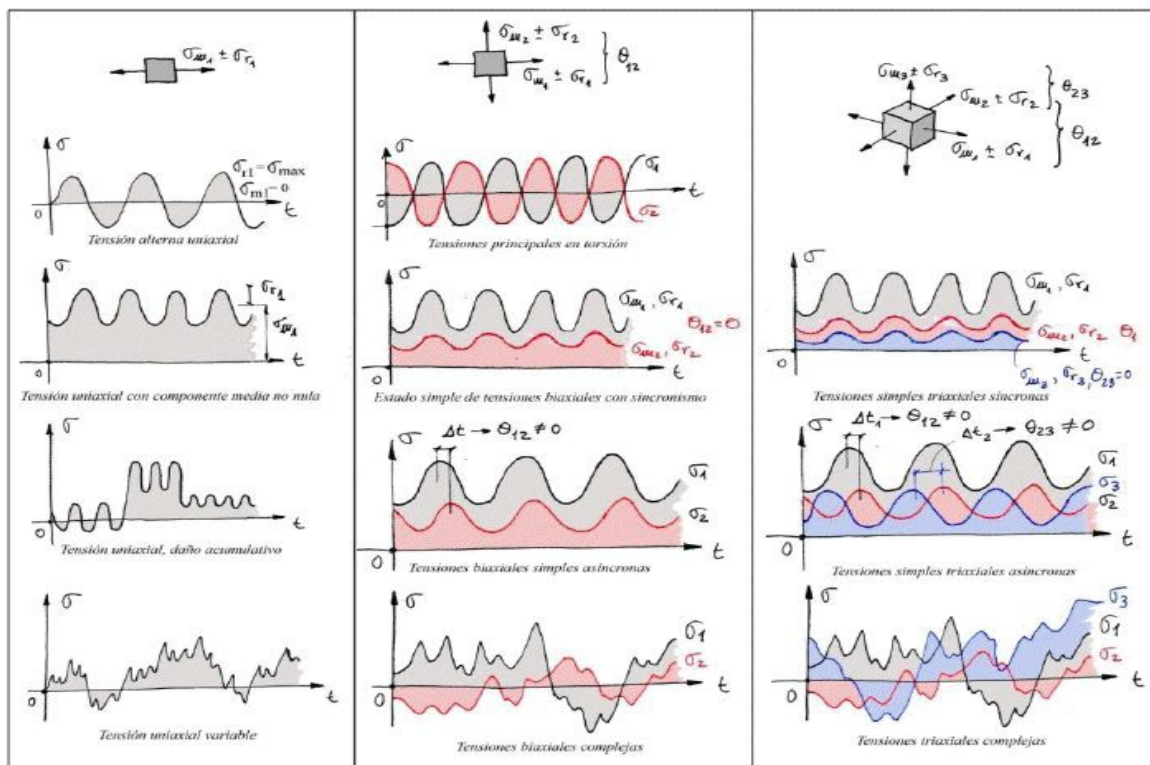


Fig 1. Estados de tensión uniaxial, biaxial y triaxial

2. Objetivos del trabajo:

Se persiguen los siguientes dos objetivos:

- Visualizar la evolución de los componentes del tensor de tensiones a lo largo del tiempo.
- Comparar los resultados obtenidos con los diferentes métodos de fatiga multiaxial.

3. Estado del arte:

Los métodos para fatiga multiaxial se clasifican en cuatro grupos: empíricos, de enfoque global, de plano crítico y mecánica de la fractura. [FAT 2011][YIN 2004]

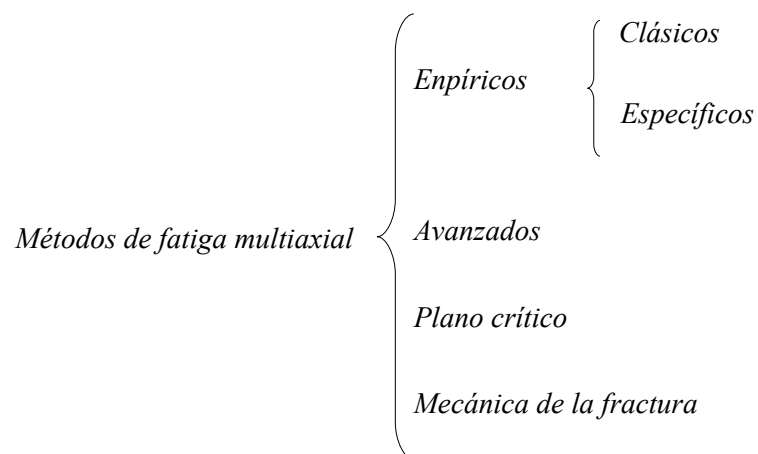


Fig 2 Criterios de fallo de fatiga multiaxial

Los **métodos empíricos** se basan en una combinación de Von Mises – Goodman. Hay dos tipos, los que son una generalización de los métodos utilizados en estática y en el caso uniaxial, conocidos como **clásicos** y los **específicos**, desarrollados para ajustar mejor los resultados obtenidos en ensayos sobre probetas. Estos

métodos no detectan estados de tensión cuasihidrostáticos ni la variación de las direcciones principales. [AVI 2012] Si bien el primer caso no se da demasiado a menudo, el segundo es bastante típico.

Los métodos de enfoque global y de plano crítico se conocen como **métodos avanzados** e intentan evitar los problemas de los métodos clásicos. Tanto unos como otros consideran que la tensión cortante media no afecta a la vida de la pieza, por lo que no la tienen en cuenta. Esto se cumple en el caso de vida ilimitada, pero es discutible en vida finita así que, en principio, sólo pueden utilizarse para ver si va a producirse el fallo y no para estimar la vida de la pieza.

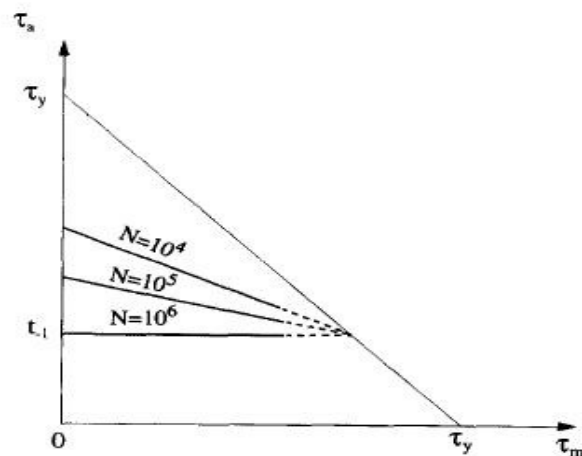


Fig 3 A vida ilimitada el efecto de τ_r puede despreciarse [ref]

Los métodos de **enfoque global** tienen en cuenta el efecto de la tensión normal y de este modo detectan los estados de tensión cuasihidrostáticos, pero no solventan el segundo problema, por lo que no pueden utilizarse en caso de que las direcciones principales varíen en el tiempo.

En estos métodos la tensión equivalente es una combinación de la tensión tangencial y la tensión normal en la forma:

$$\sigma_{eq} = \tau_r + \alpha \sigma_h < \beta \quad (1)$$

donde α y β son constantes que dependen del material y pueden obtenerse a partir de datos de ensayos. Entre ellos están el criterio de Sines y el Crossland:

Criterio de Sines: [SIN 1981]

Este método fue presentado en 1955. Utiliza una combinación de tensión cortante octaédrica y tensión normal media. Predice que no se producirá fallo si:

$$\sigma_{eq\ S} = \tau_{oct} + \alpha_S \sigma_{hm} < \beta_S \quad (2)$$

donde $\alpha_S = \sqrt{2} \frac{\sigma_n'}{\sigma_{ut}}$ y $\beta_S = \tau_n$. Estas constantes pueden obtenerse de cualquier ensayo de fatiga.

También puede utilizarse la tensión de Von Mises en lugar de la tensión cortante octaédrica. En ese caso las constantes que se utilizarán serán diferentes.

Criterio de Crossland: [CROS 1959]

El criterio de Crossland se propuso en 1956 y es similar al de Sines salvo por que este utiliza la tensión hidrostática máxima en lugar de la media. La tensión equivalente será por lo tanto:

$$\sigma_{eq\ C} = \tau_{oct} + \alpha_C \sigma_{hm} < \beta_C \quad (3)$$

En este caso y a diferencia del método de Sines, se consiguen diferentes valores para las constantes dependiendo del ensayo utilizado; el método no es coherente.

En lo que respecta a los métodos de **plano crítico**, estos métodos predicen que el fallo por fatiga ocurrirá en un plano concreto que maximice cierta combinación de tensión cortante y tensión normal. Evitan los problemas presentados por los métodos clásicos, pero necesitan mayor esfuerzo computacional. Los más utilizados son los criterio de Findley, Mataka, McDiarmid y Dang Van, a pesar de que hay más. Estos se presentan a continuación.

Criterio de Findley: [FIN 1957]

Se presentó en 1957 y utiliza como función de fatiga una combinación de tensión cortante variable, tensión normal media y tensión normal variable. El plano crítico será aquel en el que esta función sea máxima.

$$\tau_{eq} = \tau_{nr} + \alpha(\sigma_{nm} + \sigma_{nr}) < \beta \quad (4)$$

donde $\alpha = 2 \frac{\tau_n}{\sigma_n} - 1$ y $\beta = \tau_n$, constantes, que al igual que para los métodos de plano crítico, se obtienen de ensayos. Este método no tiene en cuenta el desfase entre la tensión cortante y la normal. Sólo es válido si $\frac{\tau_n}{\sigma_n} > 0.5$

Criterio de Mataka: [MAT 1977]

Data de 1977 y define el plano crítico como el plano en el que la tensión cortante variable es máxima. Una vez obtenido el plano crítico, se calcula la tensión equivalente con la misma fórmula que en el criterio de Findley. Al igual que este, sólo es válido si $\frac{\tau_n}{\sigma_n} > 0.5$.

Criterio de McDiarmid: [MCD 1991]

Existen dos criterios de McDiarmid diferentes, el primero toma como plano crítico el de mayor tensión cortante variable, como Mataka y su función de fatiga es:

$$\frac{\tau_{nr} + B(\sigma_{nr})^{1.5}}{A} < 1 \quad (5)$$

donde $A = \tau_n$ y $B = \frac{\tau_n - 0.5 \sigma_n}{(0.5 \sigma_n)^{1.5}}$

Este criterio sólo es válido si $\frac{\tau_n}{\sigma_n} > 0.5$

La segunda variante tiene en cuenta el efecto de la tensión media:

$$\frac{\left(1 - 2 \frac{\sigma_m}{\sigma_{ut}}\right)^{-0.5} \tau_r + B \sigma_r^{1.5}}{A} < 1 \quad (6)$$

Para que sea válido debe cumplirse, a parte de que $\frac{\tau_n}{\sigma_n} > 0.5$, que

$$\frac{\sigma_m}{\sigma_{ut}} < 0.5$$

Criterio de Dang Van: [DAN 1993]

Este método tiene la particularidad de que usa tensiones microscópicas en lugar de macroscópicas. Define como plano crítico aquel que maximiza una combinación de tensión cortante variable y tensión hidrostática:

$$\frac{1}{\beta} (\tau_r(t) + \alpha \sigma_h(t)) < 1 \quad (7)$$

Como se utiliza el estado de tensiones en cada instante, este método sí tiene en cuenta el desfase entre la tensión cortante e hidrostática, a diferencia de los métodos anteriores.

Las constante toman los valores: $\alpha = 3\left(\frac{\tau_n}{\sigma_n} - 0.5\right)$ y $\beta = \tau_n$

En este caso también debe cumplirse $\frac{\tau_n}{\sigma_n} > 0.5$ para que el método tenga validez.

4. Implementación de métodos:

Para lograr los objetivos marcados se ha desarrollado un programa en Matlab que realiza las siguientes funciones:

- Aplicación del criterio de Von Mises – Goodman
- Aplicación de los criterios de enfoque global de Sines y de Crossland
- Representación gráfica de los vectores de tensión en un punto a lo largo del tiempo y aplicación de los criterios del círculo y de la elipse
- Aplicación de los criterios de plano crítico

A continuación se explicará cómo se procede en cada uno de los apartados, así como un ejemplo de utilización. El programa completo puede encontrarse en el Anexo.

4.1 Datos de entrada:

Se importa un archivo XLS en el que están guardados los valores de los componentes del tensor de tensiones en el tiempo y otro que contiene los datos del material. Los datos requeridos son el límite de fatiga del ensayo de probeta rotatoria σ_n' , la tensión cortante límite del ensayo de probeta rotatoria, la tensión cortante límite de fatiga en el ensayo de torsión alterna τ_n , la tensión de rotura ingenieril σ_{ut} y el límite de fatiga en el ensayo de tracción alterna σ_n . Estos datos serán necesarios para aplicar los criterios de fallo de fatiga multiaxial.

t	σ_{xx}	σ_{yy}	σ_{zz}	τ_{xy}	τ_{xz}	τ_{yz}
0	$\sigma_{xx}(0)$	$\sigma_{yy}(0)$	$\sigma_{zz}(0)$	$\tau_{xy}(0)$	$\tau_{xz}(0)$	$\tau_{yz}(0)$
Δt	$\sigma_{xx}(\Delta t)$	$\sigma_{yy}(\Delta t)$	$\sigma_{zz}(\Delta t)$	$\tau_{xy}(\Delta t)$	$\tau_{xz}(\Delta t)$	$\tau_{yz}(\Delta t)$
$2\Delta t$	$\sigma_{xx}(2\Delta t)$	$\sigma_{yy}(2\Delta t)$	$\sigma_{zz}(2\Delta t)$	$\tau_{xy}(2\Delta t)$	$\tau_{xz}(2\Delta t)$	$\tau_{yz}(2\Delta t)$
...

Tabla 1. Formato de los datos importados

4.2 Von Mises – Goodman:

Esta función calcula la tensiones equivalentes de Goodman – Von Mises y Von Mises – Goodman. Para ello, a partir de los datos importados, calcula las tensiones principales en cada instante de tiempo y sus respectivos componentes medios y variables según:

$$\sigma_{im} = \frac{\sigma_{MAX} + \sigma_{min}}{2} , \text{ tensión media} \quad (8)$$

$$\sigma_{ir} = \frac{\sigma_{MAX} - \sigma_{min}}{2} , \text{ tensión variable,} \quad i=1,2,3 \quad (9)$$

Para seguir el camino Goodman – Von Mises, se calcula la tensión equivalente de Goodman para cada par tensión media – variable siendo:

$$\sigma_{ieq} = \frac{\sigma_{ir}}{1 - \frac{\sigma_{im}}{\sigma_{ut}}} , \quad i=1,2,3 \quad (10)$$

Una vez obtenidas estas tres tensiones, se aplica la fórmula de Von Mises [AVI 2005]:

$$\sigma_{GM} = \sqrt{\frac{1}{2} [(\sigma_{1eq} - \sigma_{2eq})^2 + (\sigma_{1eq} - \sigma_{3eq})^2 + (\sigma_{2eq} - \sigma_{3eq})^2]} \quad (11)$$

Hay que tener en cuenta que en los casos en los que $\sigma_1 + \sigma_2 + \sigma_3 < 0$, σ_{GM} deberá ser negativa.

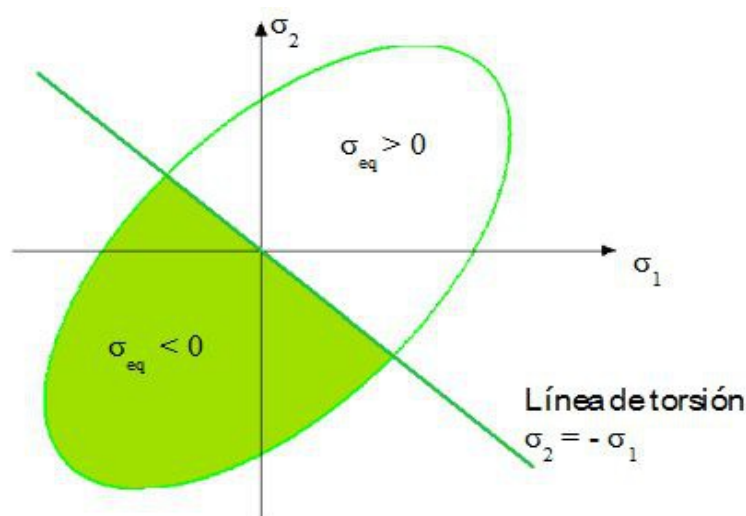


Fig 4 Criterio de signos en el caso biaxial.

Para el caso Von Mises – Goodman, se procede del modo inverso: en primer lugar se obtienen la tensión equivalente media a partir de las tres tensiones medias aplicando una fórmula análoga a la (10) donde en lugar de tensiones equivalente se utilizan tensiones medias. La tensión equivalente variable se calcula del mismo modo, sin más que cambiar las componentes medias por las variables. Con estas dos tensiones, σ_m y σ_r , se aplica la fórmula de Goodman (11) y se obtiene σ_{MG} .

Las tensiones equivalentes calculadas, ya sea según Von Mises – Goodman o Goodman – Von Mises, pueden llevarse al diagrama $\log \sigma - \log N$ del material para obtener su duración.

4.3 Criterios de enfoque global de Sines y de Crossland:

En primer lugar, se forma el tensor de tensiones y se calculan las tensiones principales para cada instante de tiempo. Con estos valores se obtienen los valores de la tensión octaédrica y la tensión hidrostática.

$$\tau_{oct}(t) = \frac{1}{3} \sqrt{[(\sigma_1(t) - \sigma_2(t))^2 + (\sigma_1(t) - \sigma_3(t))^2 + (\sigma_2(t) - \sigma_3(t))^2]} \quad (12)$$

$$\sigma_h(t) = \frac{\sigma_1(t) + \sigma_2(t) + \sigma_3(t)}{3} \quad (13)$$

La tensión octaédrica variable será: $\tau_{oct\ r} = \frac{\tau_{oct\ MAX} - \tau_{oct\ min}}{2} \quad (14)$

y la tensión hidrostática media: $\sigma_{hm} = \frac{\sigma_{h\ MAX} - \sigma_{h\ min}}{2} \quad (15)$

Con estos valores se calculan las tensiones equivalentes de Sines y Crossland según las siguientes fórmulas, dónde ya se han sustituido los valores de las constantes:

$$\text{Sines: } \sigma_{eq} = \tau_{oct} + \sqrt{2} \frac{\sigma_n'}{\sigma_{ut}} \sigma_{hm} \quad (16)$$

$$\text{Crossland: } \sigma_{eq} = \sqrt{J_2} + \left(3 \frac{\tau_n}{\sigma_n'} - \sqrt{3} \right) \frac{\sigma_n'}{\sigma_{ut}} \sigma_{h\ MAX} \quad (17)$$

4.4 Representación gráfica:

En este apartado se representa la evolución de los vectores de tensión normal σ_{nn} y tensión tangencial τ_{nt} en un plano determinado y se dibujan los criterios del círculo y la elipse.

Para ello, se pide al usuario el vector normal del plano a estudiar. A partir de este dato se calculan los vectores σ_{nn} y τ_{nt} en el plano definido por la normal para cada instante de tiempo. Así, se logra una animación en la que se ve la evaluación de dichos vectores.

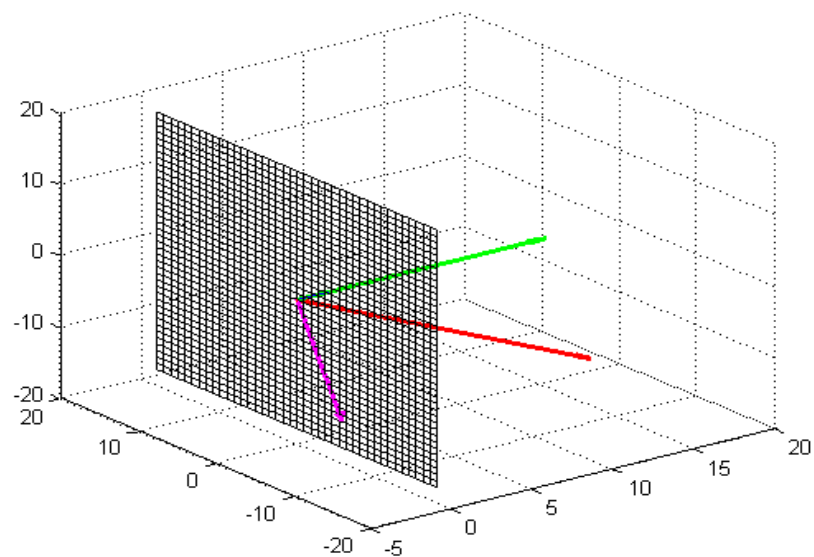


Fig 5 Representación de los vectores de tensión σ_n (rojo) σ_{nn} (verde) τ_{nt} (rosa)

Además, se dibuja la trayectoria del extremo del vector τ_{nt} en el plano. Las coordenadas de los puntos de esta trayectoria se utilizarán para aplicar los criterios del círculo y elipse.

El criterio del círculo dice que si se representa el círculo de menor radio que contenga los puntos de la trayectoria del extremo de τ_{nt} , la tensión tangencial media será el vector que une el origen de coordenadas con el centro del círculo representado y que la tensión tangencial variable vendrá dada por el vector que une el centro del círculo con cada punto de la trayectoria en cada instante de tiempo.

El círculo de menor tamaño que contiene un grupo de puntos se define como:

- Un círculo que pase por dos de los puntos del conjunto. Su radio será la mitad de la distancia entre esos puntos y el centro la posición intermedia.
- Un círculo que pase por tres de los puntos del conjunto. Su centro estará en el circuncentro del triángulo que forman y el radio será la distancia desde el centro a cualquiera de los puntos.

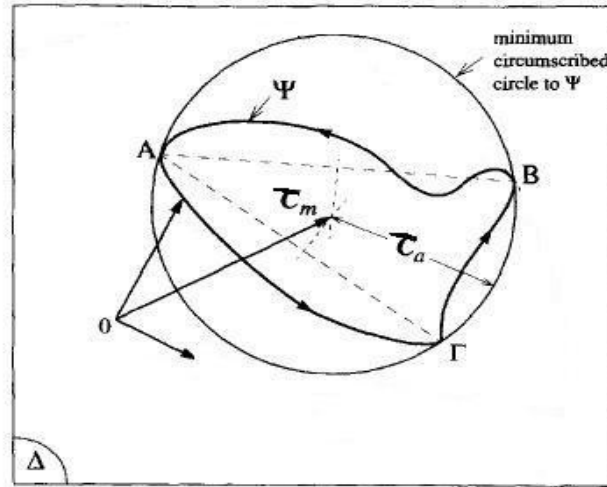


Fig.6 Criterio del círculo. Ψ representa la trayectoria dibujada por τ_{nt} [PAP 1997]

En el programa se calcula el círculo más pequeño que contenga todos los puntos entre las dos posibilidades. Ya que todos los puntos pertenecen al mismo plano, el círculo se calcula como la intersección entre la mínima esfera que contiene a todos los puntos y el plano en cuestión. Para calcular esta esfera se siguen los criterios para calcular el círculo descritos anteriormente.

En primer lugar se calcula la esfera que cumple la primera condición. Es inmediato deducir que la única esfera posible es la aquella cuyo diámetro sea igual a la distancia entre los dos puntos más separados. La esfera así definida no tiene porqué contener a todos los puntos (Fig 7), por lo que hay que comprobar que todos los puntos estén en el interior de la esfera, es decir, que se cumple:

$$(x_i - x_{centro})^2 + (y_i - y_{centro})^2 + (z_i - z_{centro})^2 \leq R^2 \quad (18)$$

donde:

$i = 1, \dots, \text{número de muestras}$

$(x_{centro}, y_{centro}, z_{centro}) = \text{punto medio entre los dos puntos más separados}$

$R = \text{radio, mitad de la distancia entre los dos puntos más separados}$

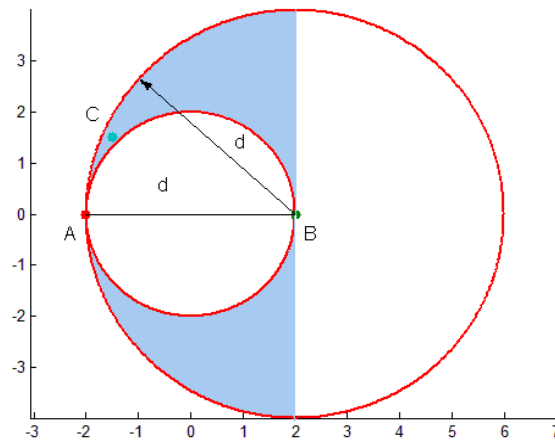


Fig.7 Los puntos de zona coloreada están más cerca de B que el punto A, pero no pertenecen al círculo definido por A y B

A continuación se calculan todas las esferas definidas por tres puntos. Si se denomina A, B y C a dichos puntos, puede definirse una esfera cuyo centro estará en la intersección de tres planos:

- El plano que contiene los puntos
- El plano de normal \vec{AB} y que pasa por el punto medio de \vec{AB}
- El plano de normal \vec{BC} y que pasa por el punto medio de \vec{BC}

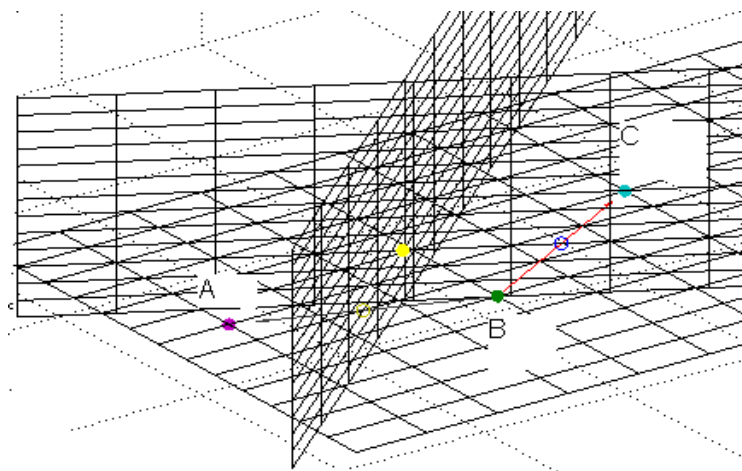


Fig. 8 Obtención del centro de la esfera a partir de tres puntos

El radio será la distancia entre el centro y cualquiera de los tres puntos, tomando el A:

$$R = \sqrt{((A_x - x_{centro})^2 + (A_y - y_{centro})^2 + (A_z - z_{centro})^2)} \quad (19)$$

donde A_i , $i = x, y, z$ representa las coordenadas del punto A

Una vez obtenidas todas las esferas se selecciona entre ellas la de mínimo radio que contenga todos los puntos y se compara con la obtenida con dos puntos. Se calcula el círculo de radio mínimo como la intersección entre la esfera más pequeña entre las dos anteriores y el plano dado.

Para calcular la elipse se parte del círculo obtenido: el semieje mayor de la elipse será igual al radio del círculo y el centro de la elipse será el mismo que el del círculo. Por lo tanto, bastará con calcular el semieje menor b . En este caso la tensión cortante media también será la distancia entre el centro de la elipse y el origen de coordenadas, pero la tensión cortante máxima se calculará según:

$$\tau_{r\ MAX} = \sqrt{a^2 + b^2} \quad (20)$$

siendo a y b los semiejes de la elipse.

En primer lugar se realiza un cambio de coordenadas para que el eje z coincida con el vector normal del plano y los ejes x e y pertenezcan al plano. A continuación se giran los ejes alrededor de z para que los ejes coordenados y los ejes de la elipse sean paralelos. Ahora ya puede calcularse el semieje menor a partir de la ecuación de la elipse:

$$\left(\frac{x_i - x_{centro}}{a}\right)^2 + \left(\frac{y_i - y_{centro}}{b}\right)^2 = 1 \quad (21)$$

donde:

$i = 1, \dots, \text{número de muestras}$

$(x_{centro}, y_{centro}, z_{centro}) = \text{centro del círculo}$

$a = \text{semieje mayor de la elipse, igual al radio del círculo}$

De todos los valores de b obtenidos hay que seleccionar, al igual que en el caso del círculo, el que defina la mínima elipse que contenga todos los puntos.

Hay que hacer notar que la elipse y el círculo sólo difieren cuando el mínimo círculo pasa por dos puntos, en el caso de que el círculo venga definido por tres puntos, la elipse se convierte en circunferencia. Eso sí, a pesar de que ambas curvas coinciden, el valor de la tensión cortante variable obtenido es diferente, cumpliéndose la relación $\tau_{elipse} = \sqrt{2} \tau_{circulo}$

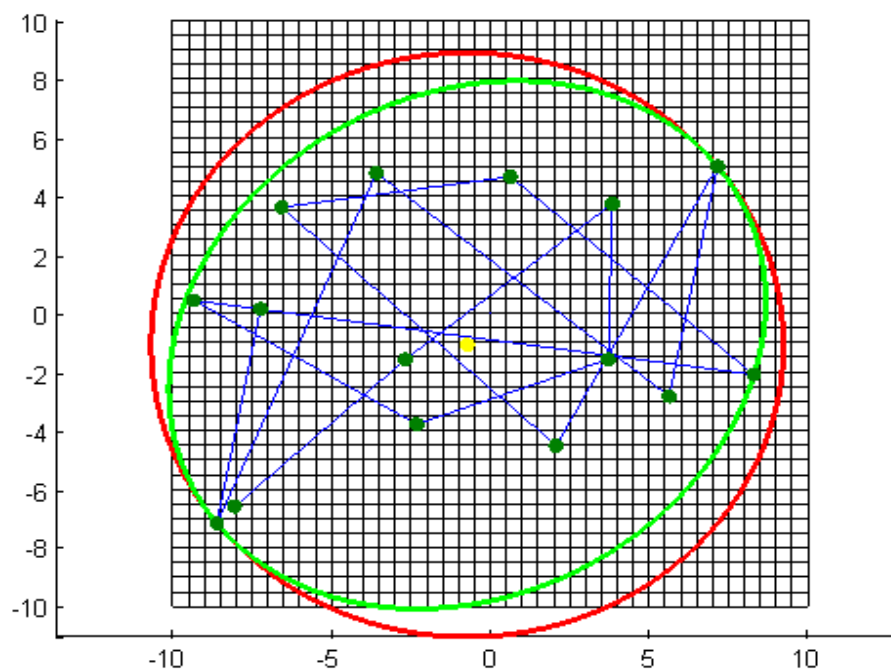


Fig.9 Criterios del círculo y de la elipse

4.5 Criterios de plano crítico:

Esta función realiza los mismo cálculos descritos en el apartado 4.4 pero no se limita a un único plano. Además aplica los criterios de plano crítico de Findley, Matake, McDiarmid y Dang Van e indica si se producirá fallo según ellos.

El procedimiento es el siguiente:

- Se importan los datos del tensor de tensiones y del material
- Se definen los planos de referencia $[0, -1, 0]$, $[0, -0.7071, 0.7071]$, $[-0.7071, 0.7071, 0]$ y $[0.5754, 0.5754, 0.5754]$

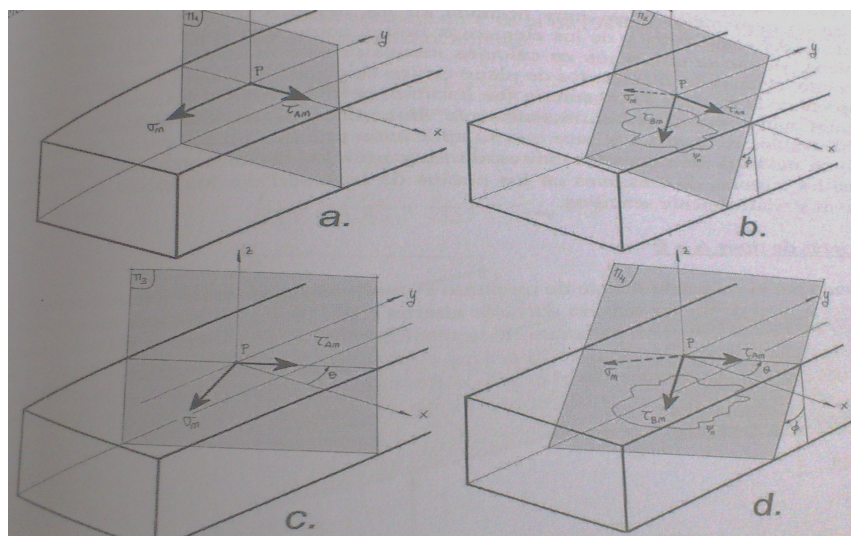


Fig 10 Planos de referencia [AVI 2012]

y se giran alrededor de los tres ejes coordenados con un paso elegido por el usuario.

- Una vez obtenidos los valores de la tensión en los planos se aplican los criterios de plano crítico.

4.6 Ejemplo de aplicación:

En este apartado se describe paso a paso un ejemplo de aplicación del programa. Se ha elegido el caso de tensiones aleatorias y la representación gráfica se realizará en el plano perpendicular al eje coordenado y , es decir, en definido por el vector normal $\{010\}^T$.

En primer lugar se han creado los archivos XLS que contienen las tensiones aplicadas, en este caso tensiones aleatorias en las que las seis componentes varían entre -100 y 100 Mpa de nombre *aleatorio.xls*, y el que contiene las constantes del material llamado *ctes.xls*.

Name	Value	Min	Max
CTES	[700,404.1452,1400,700]	404.14...	1400
MUESTRAS	21	21	21
a	[21,1]	1	21
excel	'aleatorio.xls'		
novalido	0		
opcion	1	1	1
sigma_xx	<21x1 double>	-96.56...	97.9431
sigma_yy	<21x1 double>	-79.18...	90.6189
sigma_zz	<21x1 double>	-95.42...	97.8271
tau_xy	<21x1 double>	-66.11...	93.5242
tau_xz	<21x1 double>	-93.80...	97.7722
tau_yz	<21x1 double>	-72.99...	78.6072
tiempo	<21x1 double>	0	1.0000

Fig.11 Datos importados

Si se ejecuta el script *Multiaxial* y se importan los datos se obtiene la siguiente pantalla, donde se le da al usuario diferentes opciones entre las que debe elegir. Cada una de estas opciones corresponde con los apartados 4.2 – 4.5.

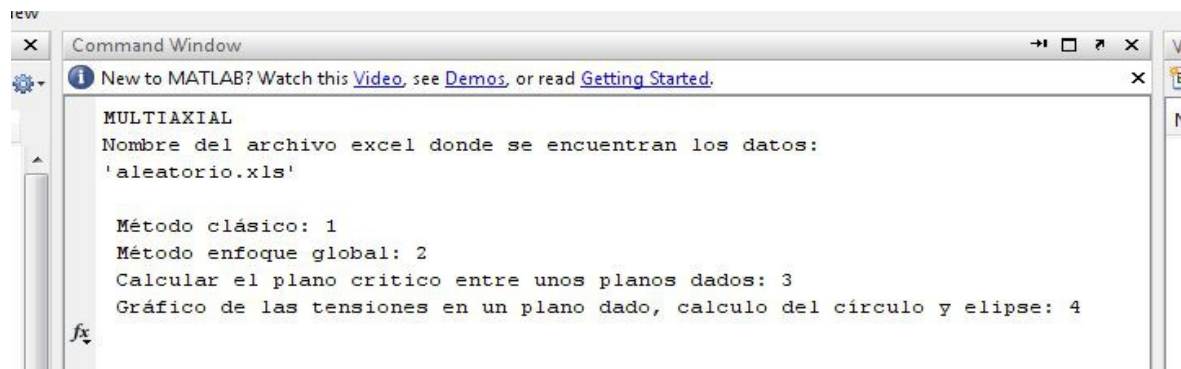


Fig 12 Opciones de cálculo

Las tres primeras opciones dan las tensiones equivalentes según los criterios presentados anteriormente. Si, por ejemplo, se elige la tercera opción, se obtiene el vector normal del plano de máxima tensión según los diferentes criterios de plano crítico y el plano en el que se da:

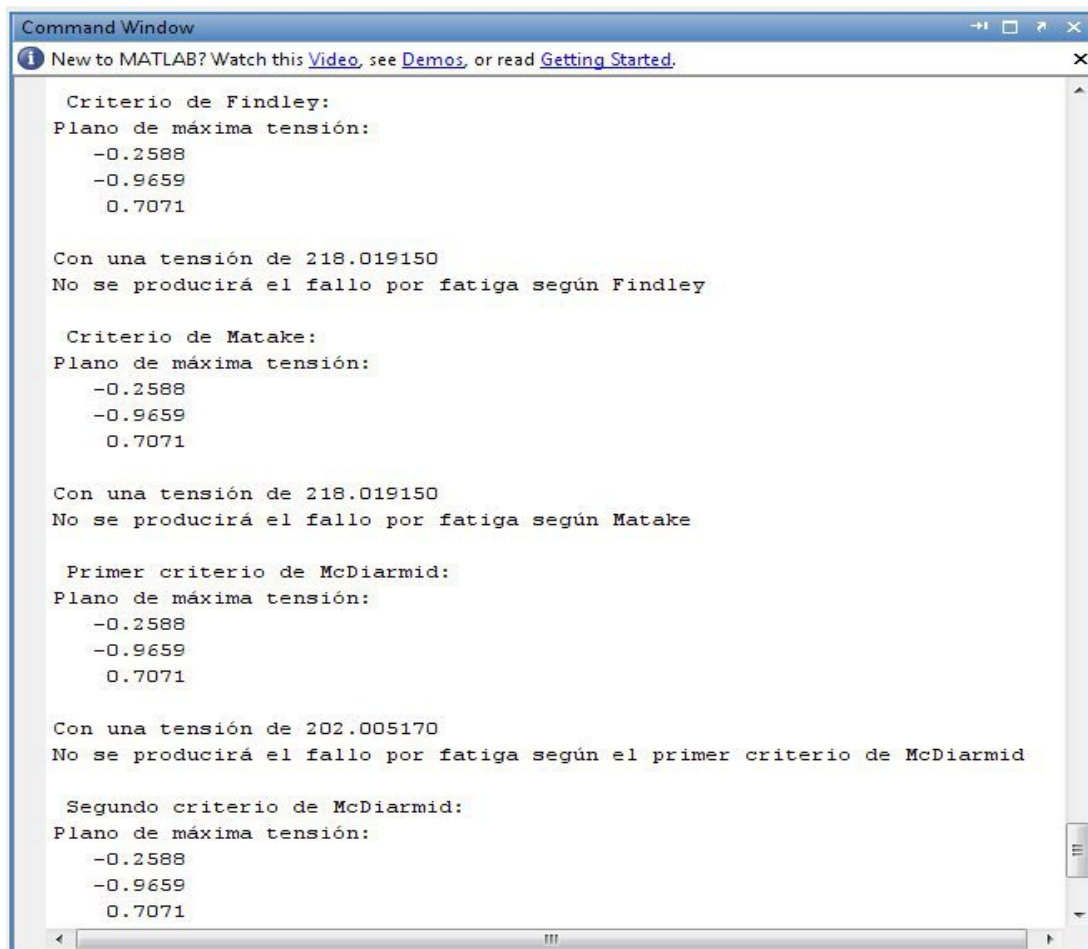


Fig 13 Cálculo según los criterios de plano crítico

En este caso, por ser las tensiones aleatorias, elegir la primera o segunda opción no nos llevaría a resultados válidos.

Por último, si se elige la opción número cuatro, se le pide al usuario el vector normal del plano para el que se quieren representar los vectores de tensión. Para el vector $[0;1;0]$, por ejemplo, se reproduce una animación donde se ve el movimiento de los vectores en el tiempo y, finalmente, se tiene:

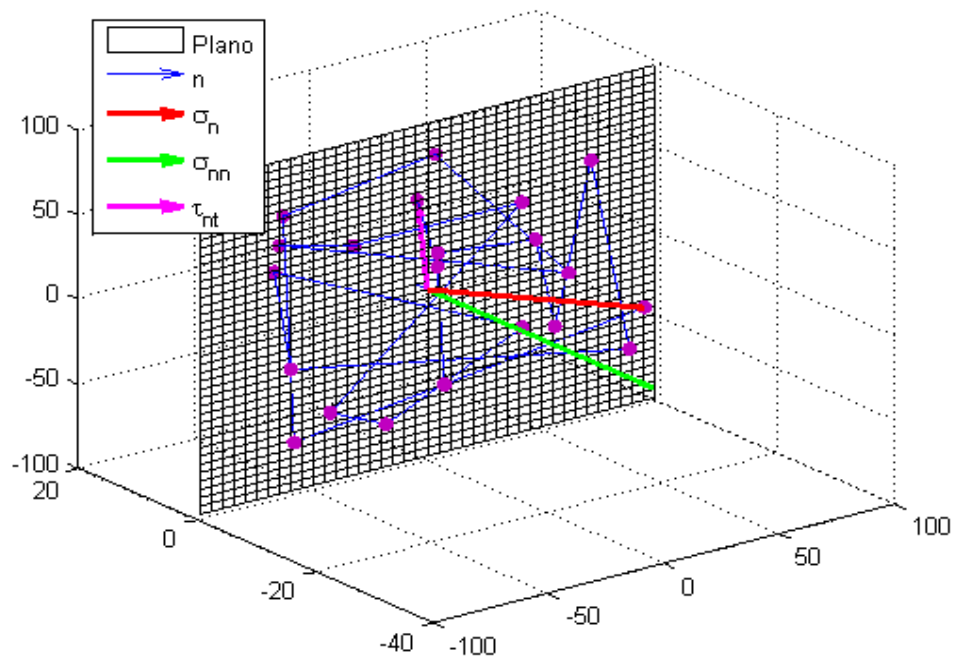


Fig.14 Representación de los vectores de tensión en el plano $[0;1;0]$

Donde se ven los puntos de la trayectoria dibujada por el vector de tensión cortante y los vectores σ_n , σ_{nn} y τ_{nt} en el último instante de tiempo

También se calcula el radio del círculo más pequeño y los ejes de la elipse que contienen los puntos de la trayectoria:

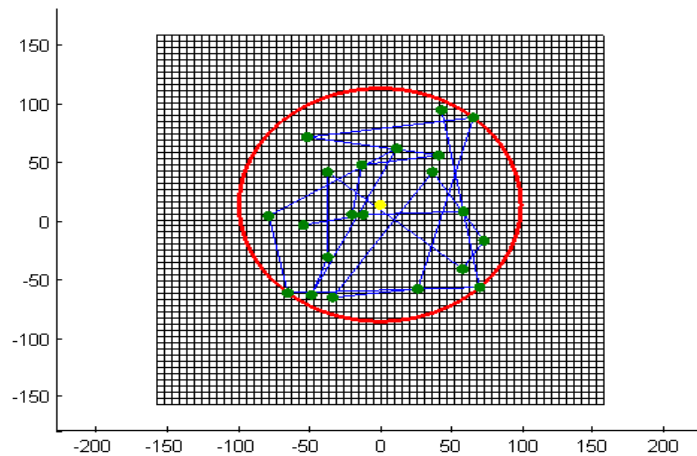


Fig 15 Aplicación del criterio del círculo

En este caso el círculo y la elipse coinciden como se puede ver en:

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Introduzca el vector normal del plano a analizar (vector columna):
[0;1;0]
El círculo y la elipse coinciden
Tension cortante media: 12.7462
Tensión cortante variable (según criterio del círculo): 99.2134
Tensión cortante variable (según criterio de la elipse): 140.3089

Criterio de Findley:
Plano de máxima tensión:
    0
    1
    0

Con una tensión de 113.232187
No se producirá el fallo por fatiga según Findley

Criterio de Matake:
Plano de máxima tensión:
    0
    1
    0

Con una tensión de 113.232187
No se producirá el fallo por fatiga según Matake

Primer criterio de McDiarmid:
Plano de máxima tensión:
    0
    1
    0

Con una tensión de 99.213394
    
```

Fig. 16 Resultados de la cuarta opción del programa

5. Comparación de métodos:

Se han aplicado los diferentes los criterios de fallos descritos a los siguientes casos:

- Tensión alterna uniaxial de componente media nula
- Tensión alterna uniaxial de componente media no nula
- Torsión
- Carga estática
- Tensión biaxial síncrona
- Tensión aleatoria

A continuación se presentan los resultados obtenidos. Todos los cálculos se han

realizado para un acero de $\sigma_{ut}=1400\text{ MPa}$, $\sigma_n'=700\text{ MPa}$ y $\tau_n=\frac{\sigma_n'}{\sqrt{3}}$

5.1 Tensión alterna uniaxial de componente media nula:

Si se aplica una tensión de valor $\sigma_{xx}=100\sin(2\pi t)$, se obtienen los siguientes resultados:

Criterio	Tensión equivalente (MPa)	Plano Crítico
Von Mises - Goodman	100	-
Goodman – Von Mises	100	-
Sines	50	-
Crossland	50	-
Findley	58.111053	[0.7887, 0.2113, 0.5774]
Matake	57.735027	[-0.7071, 0.7071, 0]
McDiarmid 1	50	[-0.7071, 0.7071, 0]
McDiarmid 2	50	[-0.7071, 0.7071, 0]
Dang Van	62.922305	[0.7887, 0.2113, 0.5774]

Tabla 2 Resultados para el caso de tensión uniaxial con componente media nula

Teniendo en cuenta que, como en este caso la tensión tangencial máxima es igual a la mitad de la tensión normal, su valor es de 50 MPa y que se da en los planos a 45° del plano en el que se aplica la tensión, los resultados obtenidos son coherentes.

El plano crítico calculado según Findley y Dang Van difiere de los demás porque la definición que éstos hacen, ya que para la búsqueda del plano crítico utilizan una combinación de la tensión normal y cortante.

5.2 Tensión alterna uniaxial de componente media no nula:

Para una tensión de $\sigma_{xx} = 100 + 50 \sin(2\pi t)$, se obtienen unas tensiones equivalentes de:

Criterio	Tensión equivalente (MPa)	Plano Crítico
Von Mises - Goodman	53.846154	-
Goodman – Von Mises	53.846154	-
Sines	47.1405	-
Crossland	28.8675	-
Findley	39.054446	[-0.8660, 0.5000, 0]
Matake	36.602540	[-0.7071, 0, 0.7071]
McDiarmid 1	25	[-0.7071, 0, 0.7071]
McDiarmid 2	25	[0.7071, 0.7071, 0]
Dang Van	47.756351	[0.7887, 0.2113, 0.5774]

Tabla 3 Resultados para el caso de tensión uniaxial con componente media no nula

En este caso la tensión cortante variable máxima es de 25 MPa y se da, igual que en el caso anterior en los planos a 45°. Era de esperar que los criterios que definen como plano crítico aquel en el que la tensión variable es máxima, Matake y

McDiarmid, indentificaran el plano de normal $\left\{\frac{-1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right\}$ como el plano critico.

5.3 Torsión:

Si se aplica un momento torsor que provoque una tensión cortante de $\tau_{xy} = 100 \sin(2\pi t)$ se obtiene:

Criterio	Tensión equivalente (MPa)	Plano Crítico
Von Mises - Goodman	173,2	-
Goodman – Von Mises	173,2	-
Sines	81.6496	-
Crossland	100	-
Findley	100	[1 0 0]
Matake	100	[0 1 0]
McDiarmid 1	100	[0 1 0]
McDiarmid 2	100	[0 1 0]
Dang Van	100	[1 0 0]

Tabla 4 Resultados para el caso de torsión

En este caso la máxima tensión cortante se da en los planos perpendiculares a x [1, 0, 0] e y [0 1 0], porque son los planos donde se aplica $\tau_{xy} = \tau_{yx}$. Por lo tanto, los resultados obtenidos con los criterios de plano crítico son los esperados.

Para el caso de Sines y de Crossland, se tiene una tensión normal equivalente de 173.2 Mpa, que equivale a $100 \sqrt{3}$ igual que la calculada con el método clásico y que es lo esperado para torsión.

Hay que hacer notar que este es un caso singular para el criterio de signos presentado, así que ha habido que corregir los resultados del programa.

5.4 Carga estática:

Si la tensión aplicada es de $\sigma_{xx} = 100 + 0.1 \sin(2\pi t)$, lo que puede considerarse una carga estática por ser $\sigma_m \gg \sigma_r$.

Criterio	Tensión equivalente (MPa)	Plano Crítico
Von Mises - Goodman	0.107692	-
Goodman – Von Mises	0.107692	-
Sines	23.6174	-
Crossland	0.057735	-
Findley	15.485524	[1, 0, 0]
Matake	5.855484	[-0.6124, -0.3536, 0.7071]
McDiarmid 1	0.048412	[-0.6124, -0.3536, 0.7071]
McDiarmid 2	0.048412	[-0.6124, -0.3536, 0.7071]
Dang Van	23.228286	[1, 0, 0]

Tabla 5 Resultados para el caso de carga estática

En este caso, al tratarse de una carga estática, no debe producirse el fallo por fatiga, por lo que la tensión equivalente debería ser cero. Von Mises – Goodman y Crossland dan el resultado esperado. En el caso de Crossland es así porque debido a la relación entre los datos del material $\alpha_c = 0$, así que no tiene en cuenta el efecto de la tensión hidrostática. El valor que se obtiene usando el criterio de Sines

viene de que la tensión hidrostática es de 33.33 MPa multiplicado por $\alpha_s = \frac{1}{\sqrt{2}}$.

Para los criterios de plano crítico se tiene que, tanto para Findley como para Dang Van, que definen el plano crítico como el de máxima combinación de tensión normal y cortante, el plano crítico es aquel cuya normal coincide con la dirección de aplicación de la carga.

En el caso de McDiarmid y Matake, no se descarta que la elección de ese plano se deba a limitaciones de memoria por las que el mínimo paso entre dos posiciones de los planos de referencia es de 30°.

5.5 Tensión biaxial síncrona:

En un caso biaxial síncrono en el que las tensiones aplicadas tengan un valor de

$$\sigma_{xx} = 100 \sin(2\pi t)$$

$$\sigma_{yy} = 10 \sin(2\pi t)$$

Criterio	Tensión equivalente (MPa)	Plano Crítico
Von Mises - Goodman	95.393920	-
Goodman – Von Mises	95.393920	-
Sines	44.9691	-
Crossland	55.0757	-
Findley	58.111053	[-0.7071, 0, 0.7071]
Matake	57.735027	[-0.7071, 0, 0.7071]
McDiarmid 1	50	[-0.7071, 0, 0.7071]
McDiarmid 2	50	[-0.7071, 0, 0.7071]
Dang Van	62.922305	[-0.7887, -0.2113, 0.5774]

Tabla 6 Resultados para el caso de tensión biaxial síncrona

Es el mismo caso que torsión pero con ambas tensiones principales positivas y de valores diferentes. En este caso la tensión cortante máxima tiene un valor de 50 MPa, similar a la obtenida con los métodos de plano crítico, dependiendo de la importancia que cada uno de ellos le da a la tensión normal.

La tensión equivalente de Crossland y de Von Mises – Goodman tienen una relación de $\sqrt{3}$, la misma que existe entre la tensión octaédrica que utiliza Von Mises – Goodman en sus cálculos y el segundo invariante de la tensión necesario

para el criterio de Crossland. Ambos resultados son equivalentes si se tiene en cuenta que en este caso la tensión hidrostática es nula.

Lo mismo puede decirse para la tensión equivalente de Sines, cuya relación con la tensión de Von Mises – Goodman es de $\frac{3}{\sqrt{2}}$, que coincide con la relación entre la tensión octaédrica y la tensión cortante octaédrica.

5.6 Tensión aleatoria:

En este apartado se trata el caso más general, el de una tensión aleatoria cuyos seis componentes varían entre -100 y 100 MPa. Se obtienen las siguientes tensiones equivalentes:

Criterio	Tensión equivalente (MPa)	Plano Crítico
Von Mises - Goodman	38.280262	-
Goodman – Von Mises	38.280262	-
Sines	96,64	-
Crossland	121.1663	-
Findley	190.997025	[-0.7887, -0.2113, 0.5774]
Matake	190.997025	[-0.7887, -0.2113, 0.5774]
McDiarmid 1	176.806257	[-0.7887, -0.2113, 0.5774]
McDiarmid 2	176.806257	[-0.7887, -0.2113, 0.5774]
Dang Van	198.092409	[-0.7887, -0.2113, 0.5774]

Tabla 7 Resultados para el caso de tensión aleatoria

Los resultados de los criterios de plano crítico son bastante coherentes entre sí, siendo el de Dang Van el más conservador de todos. Además se ve que los criterios de enfoque global y clásicos no son válidos para este tipo de problema, porque no son capaces de detectar los cambios de las direcciones principales.

6. Conclusiones:

- A pesar del tiempo y esfuerzo invertidos, no existe un único criterio para realizar los cálculos de fatiga multiaxial. Existen multitud de métodos, cada uno con su campo de aplicación.
- Los métodos clásicos son sencillos de aplicar, pero no detectan estado de tensiones cuasihidroestático o el cambio de direcciones principales.
- Los métodos de enfoque global son igual de fáciles de aplicar que los clásicos y evitan el problema de los estados de tensión cuasihidroestáticos
- Los métodos de plano crítico, aunque no se les escapa ningún fallo tienen un coste computacional mucho mayor, por ejemplo para el caso uniaxial se ha tardado 0.016221 s para realizar los cálculos para enfoque global y 1.133840 s para plano crítico, 70 veces más.
- En los casos cuyo resultado se conoce como el uniaxial con tensiones medias nulas y no nulas, torsión y tensión biaxial síncrona los resultados obtenidos con los métodos analizados son similares.
- Entre los métodos de plano crítico Dang Van es el más conservador, obteniéndose en los casos estudiados siempre una tensión mayor
- La falta de precisión del programa puede afectar a los resultados.

7. Referencias:

- [AVI 2005] Análisis de fatiga en máquinas. Rafael Avilés. Paraninfo 2005
- [AVI 2012] Notas sobre análisis de fatiga en máquinas. Rafael Avilés. Publicaciones – ETS Ingeniería Bilbao 2012
- [CHA 1994] Fatigue life prediction of components: an analysis of the different working steps. M. Chaudonneret. Société Française de Metallurgie et de Matériaux (France), pp. 479-488, 1994
- [CROS 1959] Effect of Hydrostatic Pressure on the Direct-Stress Fatigue Strength of an Alloy Steel D. J. White, B. Crossland, J. L. M. Morrison, Journal of Mechanical Engineering Science. June 1959 vol. 1 no. 1 39-49
- [DAN 1993] Macro - micro approach in high-cycle multiaxial fatigue K. Dang-Van ASTM Special Technical Publication, 1993
- [DEP 1994] Probabilistic design methods for multiaxial high cycle fatigue A. Deperrois, A. Bignonnet, P. Merrien, Fourth International Conference on Biaxial/Multiaxial Fatigue (Vol. I), St. Germain en Laye, France; 31 May-3 June 1994. pp. 381-392. 1994
- [FAT 2011] Multiaxial fatigue: An overview and some approximation models for life estimation. A. Fatemi, N. Shamsaei. International Journal of Fatigue vol 33, pp 948–958. Elsevier 2011
- [FIN 1957] Fatigue of metals under combination of stresses. Transactions of the ASME vol. 79 pp. 1337 – 1348, 1957
- [MAT 1977] An Explanation on Fatigue Limit under Combined Stress T. Mataka. Bulletin of JSME, vol 20 nº 141, p.257-263, 1977

- [MCD 1991] A general criterion for high cycle multiaxial fatigue failure. D. L. McDiarmid. *Fatigue & Fracture of Engineering Materials & Structures*. vol14, nº 4, pp 429–453, April 1991
- [SIN 1981] Fatigue criteria under combined stresses or strains G Sines, G Ohgi - *Journal of engineering materials and technology*, 1981
- [PAP 1997] A comparative study of multiaxial high-cycle fatigue criteria for metals I. V. Papadopoulos, P. Davoli, C. Gorla, M Filippini, A Bernasconi *International Journal of Fatigue* Vol. 19, Nº. 3, pp. 219-235, 1997 Elsevier.
- [WAN 1994] Multiaxial random load fatigue: life prediction techniques and experiments C. H. Wang, M. W. Brown, *Fourth International Conference on Biaxial/Multiaxial Fatigue (Vol. I)*, St. Germain en Laye, France; 31 May-3 June 1994. pp. 367-380. 1994
- [YIN 2004] Evaluation and comparison of several multiaxial fatigue criteria Ying-Yu Wang, Wei-Xing Yao, *International Journal of Fatigue* vol 26 pp 17–25. Elsevier 2004

ANEXO: PROGRAMA

Se anexa a continuación el código del programa de Matlab desarrollado.

En primer lugar el script principal que importa los datos necesario y da al usuario a elegir entre las diferentes opciones de cálculo:

```
% MULTIAXIAL

% Se importan los datos de un archivo excel con formato 'nombre.xls'
% En este archivo las componentes de la tensión son columnas, empezando
% por la segunda columna, y en el orden sigma_xx, sigma_yy, sigma_zz,
% tau_xy, tau_xz, tau_yz
%
% Calcula la tensión equivalente de VonMises, de Sines o de Crossland,
% el plano critico entre varios planos dados (llama a 'calculo') o
% representa
% gráficamente las tensiones en un plano en el tiempo y calcula el mínimo
% círculo y la mínima elipse que circunscriben la trayectoria de la
% tensión
% cortante (llama a 'planograf')

clear
clc
close('all')

disp('MULTIAXIAL')

% ***** Importar datos*****

excel = input('Nombre del archivo excel donde se encuentran los datos:
\n');
importfile(excel)

tiempo    = data(:,1); % No se utiliza para los cálculos pero es útil para
representación gráfica
sigma_xx  = data(:,2);
sigma_yy  = data(:,3);
sigma_zz  = data(:,4);
tau_xy    = data(:,5);
tau_xz    = data(:,6);
tau_yz    = data(:,7);

clear('data')
clear('textdata')
clear('colheaders')

% Número de muestras

a = size(sigma_xx);

MUESTRAS = a(1);
```

```
%*****

% IMPORTAR DATOS DEL MATERIAL

% Se importan de un excel el límite de fatiga en el ensayo de probeta
% rotatoria, el tau limite de probeta rotatoria, la tensión de rotura
% ingenieril y límite de fatiga en el ensayo de tracción alterna

importfile('ctes.xls')

CTES = data;
clear('data')
clear('textdata')
clear('colheaders')

novalido = true;

while novalido

    opcion = input('\n Método clásico: 1 \n Método enfoque global: 2 \n
        Calcular el plano critico entre unos planos dados: 3 \n
        Gráfico de las tensiones en un plano dado, calculo del
        círculo y elipse: 4 \n');

    switch opcion

        case 1

            % Von Mises
            VonMises(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
                MUESTRAS,CTES)
            novalido = ~novalido;

        case 2

            % Sines - Crossland
            EnfoqueGlobal(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz,
                tau_yz, MUESTRAS, CTES)
            novalido= ~novalido;

        case 3

            %Plano crítico
            Calculo(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
                MUESTRAS, CTES)
            novalido= ~novalido;

        case 4

            %Representación gráfica
            Planograf(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz,
                tau_yz, MUESTRAS, CTES)
            novalido= ~novalido;

        otherwise

            fprintf('\n Opción no válida')
```

```
end  
end
```

Si el usuario elige la opción 1 el programa llama a la siguiente función que calcula las tensiones equivalentes de Goodman – Von Mises y Von Mises – Goodman:

```
function VonMises(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,  
MUESTRAS, CTES)  
  
% VonMises(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,  
% MUESTRAS, CTES)  
% Calcula la tension equivalente de VonMises-Goodman y GoodmanVonMises  
% necesita los datos del tensor de tensiones para los instantes de  
% tiempo, el número de muestras y los datos del material  
  
% Tensión de rotura ingenieril  
  
sigma_ut = CTES(3);  
  
% Inicializar matrices:  
  
tension_princ = zeros(3, MUESTRAS); % matriz cuyas columnas son las  
tensiones principales en cada instante de tiempo  
  
% MATLAB ORDENA LAS TENSIONES DE MENOR A MAYOR, POR LO QUE EN UN CASO  
% UNIAXIAL UNA TENSIÓN EMPIEZA SIENDO SIGMA_3 Y PASA A SER SIGMA_1  
% Y CALCULA MAL LAS TENSIONES MEDIAS Y VARIABLES  
  
% Para evitarlo:  
  
% Como las tensiones principales no pueden variar para que enfoque  
% global funcione, se calcula para el primer punto la tensiones  
% principales y las direcciones principales y con estos vectores  
% principales se calculan el resto de tensiones principales.  
  
sigma= [sigma_xx(2) tau_xy(2) tau_xz(2); tau_xy(2) sigma_yy(2)  
tau_yz(2); tau_xz(2) tau_yz(2) sigma_zz(2)];  
  
[vectores_princ tens_princ]= eig(sigma);  
  
for j=1:MUESTRAS  
  
% calcular tensiones principales  
  
sigma= [sigma_xx(j) tau_xy(j) tau_xz(j); tau_xy(j) sigma_yy(j)  
tau_yz(j); tau_xz(j) tau_yz(j) sigma_zz(j)];  
  
tens_princ = vectores_princ' * sigma* vectores_princ;  
  
tens_princ = [tens_princ(1,1); tens_princ(2,2); tens_princ(3,3)];  
  
tension_princ(:,j)= tens_princ;  
  
end
```

```

% Tensión principal 1:

sigma_1_m = (max(tension_princ(1,:))+ min(tension_princ(1,:)))/2 ;
sigma_1_r = (max(tension_princ(1,:))- min(tension_princ(1,:)))/2 ;

% Tensión principal 2:

sigma_2_m = (max(tension_princ(2,:))+ min(tension_princ(2,:)))/2 ;
sigma_2_r = (max(tension_princ(2,:))- min(tension_princ(2,:)))/2 ;

% Tensión principal 3:

sigma_3_m = (max(tension_princ(3,:))+ min(tension_princ(3,:)))/2 ;
sigma_3_r = (max(tension_princ(3,:))- min(tension_princ(3,:)))/2 ;

% GOODMAN - VON MISES:

% Goodman: sigma_eq = sigma_r / (1 - sigma_m / sigma_ut)

sigma_1_eq = sigma_1_r / (1 - sigma_1_m / sigma_ut) ;
sigma_2_eq = sigma_2_r / (1 - sigma_2_m / sigma_ut) ;
sigma_3_eq = sigma_3_r / (1 - sigma_3_m / sigma_ut) ;

% Tensión equivalente de Von Mises

sigma_GM = sqrt(1/2*((sigma_1_eq - sigma_2_eq)^2 + (sigma_1_eq -
    sigma_3_eq)^2 + (sigma_2_eq -sigma_3_eq)^2));

% Criterio de signos Von Mises:
% si sigma_1 + sigma_2 + sigma_3 < 0 -> sigma_eq = -sigma_eq

if (sigma_1_eq + sigma_2_eq + sigma_3_eq < 0)

    sigma_GM = - sigma_GM ;

end

% VON MISES - GOODMAN:

% Von Mises

sigma_eq_m = sqrt(1/2*((sigma_1_m - sigma_2_m)^2 + (sigma_1_m -
    sigma_3_m)^2 + (sigma_2_m -sigma_3_m)^2));

if (sigma_1_m + sigma_2_m + sigma_3_m < 0)

    sigma_eq_m = - sigma_eq_m ;

end

sigma_eq_r = sqrt(1/2*((sigma_1_r - sigma_2_r)^2 + (sigma_1_r -
    sigma_3_r)^2 + (sigma_2_r -sigma_3_r)^2));

if (sigma_1_r + sigma_2_r + sigma_3_r < 0)

    sigma_eq_r = - sigma_eq_r ;

```



```
end

% Goodman

sigma_MG = sigma_eq_r / (1 - sigma_eq_m / sigma_ut) ;

fprintf('\n Tensión equivalente Von Mises - Goodman %f \n', sigma_MG)
fprintf('Tensión equivalente Goodman - Von Mises %f \n', sigma_GM)

end
```

En el caso de que se elija la segunda opción, la función requerida es:

```
function EnfoqueGlobal(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz,
    tau_yz, MUESTRAS, CTES)

% EnfoqueGlobal(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
    MUESTRAS, CTES)
% Calcula las tensiones equivalentes de Sines y Crossland. Necesita
% los valores del tensor de tensiones en cada instante de tiempo, el
% número de muestras y los datos del material

% CTES

% Sines

lim_probeta_rotatoria = CTES(1);

sigma_ut = CTES(3);

alfa_s = sqrt(2)*(lim_probeta_rotatoria / sigma_ut);
beta_s = sqrt(2)/3 * lim_probeta_rotatoria;

% Crossland -> pag 142

lim_tau_torsion = CTES(2);

alfa_c = 3* lim_tau_torsion / lim_probeta_rotatoria - sqrt(3) ;
beta_c = lim_tau_torsion ;

% Inicializar matrices:

tension_princ = zeros(3, MUESTRAS); % matriz cuyas columnas son las
tensiones principales en cada instante de tiempo

sigma= [sigma_xx(2) tau_xy(2) tau_xz(2); tau_xy(2) sigma_yy(2)
    tau_yz(2); tau_xz(2) tau_yz(2) sigma_zz(2)];

[vectores_princ tens_princ]= eig(sigma);

for j=1:MUESTRAS
```

```

% calcular tensiones principales

sigma= [sigma_xx(j)  tau_xy(j)  tau_xz(j); tau_xy(j)  sigma_yy(j)
        tau_yz(j); tau_xz(j)  tau_yz(j)  sigma_zz(j)];

tens_princ = vectores_princ' * sigma* vectores_princ;

tens_princ = [tens_princ(1,1); tens_princ(2,2); tens_princ(3,3)];

tension_princ(:,j)= tens_princ;

end

% Tensiones octaedricas en los instantes de tiempo (vector 1xMUESTRAS)

tau_octaedrica= zeros(1,MUESTRAS);

for j=1: MUESTRAS

    tau_octaedrica(j) = 1/3 * ((tension_princ(1,j)-
                                tension_princ(2,j))^2 + (tension_princ(1,j)-
                                tension_princ(3,j))^2 + (tension_princ(2,j)-
                                tension_princ(3,j))^2).^0.5;

    % CRITERIO DE SIGNOS A TAU_OCTAEDRICA

    if (tension_princ(1,j)+ tension_princ(2,j)+ tension_princ(3,j)<0);

        tau_octaedrica(j)= -tau_octaedrica(j);

    end

end

% Tensiones hidrostáticas en los instantes de tiempo (vector
1xMUESTRAS)
sigma_hidrost = (tension_princ(1,:)+ tension_princ(2,:)+
                 tension_princ(3,:))/3 ;

% Tension hidrostática media

sigma_hidrost_m = (min(sigma_hidrost) + max(sigma_hidrost))/2;

%Tension octaedrica variable

tau_octaedrica_r = (max(tau_octaedrica)-min(tau_octaedrica)) /2;
sigma_octaedrica_r = 3*tau_octaedrica_r/sqrt(2);
j_2_r = sqrt(3/2)*tau_octaedrica_r;  % segundo invariante

% SINES

Sines = tau_octaedrica_r + alfa_s * sigma_hidrost_m ;

```

```
fprintf('\n Tensión equivalente de Sines: \n')
disp(num2str(Sines))

if Sines < beta_s

    fprintf('\n No se produce fallo según Sines \n')

else

    fprintf('\n Se produce fallo según Sines \n')

end

% -----

% CROSSLAND

% sigma_eq_c = tau_octaedrica + alfa_c * sigma_hidrost_maxima > beta_c

Crossland = j_2_r + alfa_c * max(sigma_hidrost);

fprintf('\n Tensión equivalente de Crossland: \n')
disp(num2str(Crossland))

if Crossland < beta_c

    fprintf('\n No se produce fallo según Crossland \n')

else

    fprintf('\n Se produce fallo según Crossland \n')

end

end
```

Para la opción tres se llama a:

```
function Calculo(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
    MUESTRAS, CTES)

% TENSOR DE TENSIONES

%sigma= [sigma_x tau_xy tau_xz; tau_xy sigma_y tau_yz; tau_xz
tau_yz
%sigma_z];

% N: matriz cuyas columnas son los diferentes vectores n

% TIENE PROBLEMAS NUMÉRICOS EN LOS CASOS PARTICULARES

fprintf('CÁLCULO DEL PLANO CRÍTICO \n ') % poner planos por defecto
```

```

angulos= [90 180 90 ; 90 135 45; 45 135 90; -45 -135 45]; % Cada fila
representa los angulos que forma la normal del plano con los ejes
x,y,z

% los cuatro planos de referencia

angulos= angulos'; % Para que los vectores normales sean columnas

N = cosd(angulos);

% Número de vectores normales

NUMVEC = 4;
DELTA = 30; % delta_theta: paso entre dos posiciones -> CON MENOS DE
30° OUT OF MEMORY

posiciones = 180/DELTA * 3;

% Giro alrededor de z:

matriz_giro = Giro(deg2rad(DELTA),'z'); % matriz que gira DELTA grados
alrededor de z

for i = 1: 180/DELTA

    M = matriz_giro*N;
    N = [N M];

end

% Giro alrededor de x:

matriz_giro = Giro(deg2rad(DELTA),'x'); % matriz que gira DELTA grados
alrededor de z

for i = 1: 180/DELTA

    M = matriz_giro*N;
    N = [N M];

end

% Giro alrededor de y:

matriz_giro = Giro(deg2rad(DELTA),'y'); % matriz que gira DELTA grados
alrededor de z

for i = 1: 180/DELTA

    M = matriz_giro*N;
    N = [N M];

end

%*****Inicializar matrices *****

```

```

circuitos= zeros(NUMVEC * posiciones, 4);
tau_media= zeros(NUMVEC * posiciones ,1);

TAU= zeros(MUESTRAS,3);
SIGMAN= zeros(MUESTRAS,1);

SIGMAN_M = zeros(NUMVEC * posiciones, 1);
SIGMAN_R = zeros(NUMVEC * posiciones, 1);
EjesPeq = zeros(NUMVEC * posiciones, 1);

%*****

for i= 1 : NUMVEC*posiciones % número de columnas de N

    n = N(:,i);
    n = n/norm(N(:,i));

    for j=1:MUESTRAS

        %Matriz de tensiones importada

        sigma= [sigma_xx(j) tau_xy(j) tau_xz(j); tau_xy(j)
        sigma_yy(j) tau_yz(j); tau_xz(j) tau_yz(j) sigma_zz(j)];

        % tension normal

        sigma_n = sigma * n; % vector columna

        % componente perpendicular al plano

        sigma_nn = sigma_n' * n;
        vecsigma_nn=sigma_nn*n;

        %Guardar valor tensión normal (vector cuyos elementos son
        sigma_normal)

        SIGMAN(j)= sigma_nn;

        % Componente tangencial:

        tau_nt = sigma_n - vecsigma_nn;

        % Guardar valor de tau (matriz, cada fila es un tau):

        TAU (j,:) = tau_nt;

    end

    % TENSION NORMAL (para criterios)
```

```

sigma_max = max(SIGMAN);
sigma_min = min(SIGMAN);

SIGMAN_M(i) = (sigma_max(1) + sigma_min(1))/2; % tensión normal
media

SIGMAN_R(i) = (sigma_max(1) - sigma_min(1))/2; % tensión normal
variable

% CALCULAR TODOS LOS CÍRCULOS

% CÍRCULO

[pos1 pos2] = MaximoSegmento(TAU); % los dos puntos que
definen el eje mayor de la elipse

[xcentro ycentro zcentro radio dos] = MinimoCirculo(TAU, n',
pos1, pos2);

circulos(i,:) = [xcentro, ycentro, zcentro, radio];

% TAU MEDIA = DISTANCIA CENTRO DEL CÍRCULO/ELIPSE A CERO

tau_media(i,:) = norm([xcentro, ycentro, zcentro]);

% CALCULO DE LAS ELIPSES

if dos

    paso= cambioCoord(n');
    TAU_nuevo= paso * TAU'; %pasa los puntos al nuevo sistema de
    coordenadas (deberían tener todos z=0)
    TAU_nuevo=TAU_nuevo';
    centro_nuevo= paso* [xcentro; ycentro; zcentro];

    punto1 = TAU_nuevo(pos1,:);
    punto2 = TAU_nuevo(pos2,:);

    grados= atan((punto2(2)- punto1(2))/(punto2(1)- punto1(1)));
    % radianes
    paso= Giro(grados, 'z');

    TAU_nuevo = paso* TAU_nuevo';
    TAU_nuevo = TAU_nuevo'; % para que las coordenadas sean filas
    centro_elipse= paso* centro_nuevo;

    %Calcular eje pequeño de la elipse

    b = Elipse(TAU_nuevo, centro_elipse(1), centro_elipse(2),
        radio);

    EjesPeq(i)= b ;

end

end

```

```
fprintf('\n Tensión cortante variable máxima en los planos
(círculo) \n')

    radios=circulos(:,4);
    disp(num2str(radios))

fprintf('\n Tensión cortante variable máxima en los planos (elipse)
\n')

    EjesPeq(EjesPeq==0)= radios(EjesPeq==0); % si b=0 (el círculo pasa
por tres puntos), sustituir ese valor por el radio
    tau_r_elipse=(EjesPeq.^2 + radios.^2).^0.5;

    disp(tau_r_elipse)

fprintf('\n Tensión cortante media en los planos \n')

    disp(num2str(tau_media))

% TENSION NORMAL

fprintf('\n Tensión normal media en los planos \n')

    disp(num2str(SIGMAN_M))

fprintf('\n Tensión normal variable en los planos \n')

    disp(num2str(SIGMAN_R))

% CRITERIOS

    fprintf('\n  CRITERIOS \n')

% Plano de mayor tensión cortante variable (mayor radio del círculo)

    [tau_crit pos] = max(radios);

    fprintf('\n CÍRCULO \n El plano de mayor tensión cortante variable
        es el de normal:\n')

    n = N(:,pos);
    n = n/norm(n);

    disp(num2str(n));

    disp(['con una tensión cortante variable de ',num2str(tau_crit)])

% Elipse
% si elipse = círculo, valor de elipse = círculo * sqrt(2)

    [tau_crit pos] = max(tau_r_elipse);

    fprintf('\n ELIPSE \n El plano de mayor tensión cortante variable
```

```

                es el de normal:\n')

n = N(:,pos);
n = n/norm(n);

disp(num2str(n));

disp(['con una tensión cortante variable de ',num2str(tau_crit)])

% Criterios

Criterios(SIGMAN_M, SIGMAN_R, radios , N, CTES)

end

```

Esta función requiere a su vez que se llame a diferentes funciones:

- Giro (*ángulo, eje*) , que gira los ejes un ángulo dado alrededor del eje elegido
- MaximoSegmento(*coordenadas*), que calcula la posición en la matriz *coordenadas* de los puntos más alejados entre sí
- MinimoCírculo(*coordenadas, normal, pos1, pos2*) , que calcula el radio y el centro del círculo más pequeño que contiene todos los puntos de la matriz *coordenadas*, conociendo el plano en el que se encuentran esos puntos y la posición en la matriz de los puntos más alejados.
- CambioCoord(*normal*), que realiza un cambio de coordenadas a unos ejes adscrito a un plano del que se conoce la normal.
- Elipse(*coordenadas, centro_x, centro_y, centro_z, radio*), que calcula el eje pequeño de la menor elipse que contiene los puntos de la matriz *coordenadas*, conociendo el centro y el radio del círculo que contiene los puntos
- Criterios(σ_{nm} , σ_{nr} , τ_r , N, ctes), que aplica los criterios de plano crítico.

A continuación se adjuntan sus respectivos códigos:


```
function [matriz_paso]= Giro(grados, eje)

%GIRO: se gira el sistema de coordenadas alrededor del 'eje'

switch eje

    case 'x'
        matriz_paso = [1 0 0; 0 cos(grados) sin(grados); 0 -sin(grados)
                        cos(grados)];

    case 'y'
        matriz_paso = [cos(grados) -sin(grados) 0; 0 1 0; sin(grados)
                        cos(grados) 0];

    case 'z'
        matriz_paso = [cos(grados) sin(grados) 0; -sin(grados)
                        cos(grados) 0; 0 0 1];

end

end

*****

function [pos1 pos2]=MaximoSegmento(coordenadas)
% Calcula cuales son los dos puntos más alejados de una matriz cuyas filas
% son coordenadas de puntos

tamano=size(coordenadas);
maxlong=0;

%Inicializar (para evitar error si todas las coordenadas coinciden)

pos1=1;
pos2=1;

for i=1:(tamano(1)-1)

    for j=(i+1):tamano(1)

        segmento= coordenadas(j,:)-coordenadas(i,:);
        longitud= norm(segmento);

        if longitud > maxlong
            maxlong=longitud;
            pos1=i;
            pos2=j;
        end

    end

end

end

*****
```

```
function [xcentro ycentro zcentro radio dos] =
    MinimoCirculo(coordenadas,normal, pos1, pos2)

% Inicialización de matrices

centros_x = 0;
centros_y = 0;
centros_z = 0;
centros_r = 0;

% las coordenadas están en filas

dos= false; % Inicilizar valor. Si la circunferencia la definen 2
             puntos será true

tamano= size(coordenadas);

% Comprobación de que todas las coordendas no sean iguales

comp = zeros(tamano(1)-1 , 3);

for j = 2: tamano(1)
    comp(j-1,:) = (coordenadas(j,:)== coordenadas(1,:));
end

if (~all(all(comp)))

    switch tamano(1)

        case 1 % si solo hay un punto centro=punto r=0

            xcentro=coordenadas(1,1);
            ycentro=coordenadas(1,2);
            zcentro=coordenadas(1,3);
            radio=0;

        case 2 % si hay dos puntos centro en la mitad, r por pitagoras

            %radio= segmento que une dos puntos cualquiera

            a=coordenadas(pos1,:);
            b=coordenadas(pos2,:);
            [xcentro ycentro zcentro radio] = RadioSegmento(a,b);
            dos = true;

        case 3 % si hay tres puntos el centro estará donde se crucen
                las mediatrices de las rectas que se logran juntando
                los puntos 2 a 2

            a=coordenadas(1,:);
            b=coordenadas(2,:);
            c=coordenadas(3,:);
            [xcentro ycentro zcentro radio] =
                TresPuntos(a,b,c,normal);

        otherwise

            %mínimo círculo de entre todas las posibilidades 2 y 3
```

```
%2) calcular los círculos que se definen con un segmento cualquiera.
%el círculo mínimo será el de menor radio que contenga todos los
%puntos. Será el círculo que definan los dos puntos más separados.

[xcentro ycentro zcentro radio] =
RadioSegmento(coordenadas(pos1,:), coordenadas(pos2,:));

%comprobar que todos los puntos están dentro (si están all(distancias)==1)

distancias = (((coordenadas(:,1)-xcentro)./radio).^2 +
((coordenadas(:,2)-ycentro)./radio).^2+((coordenadas(:,3)-
zcentro)./radio).^2) <= 1;

%3) calcular los círculos que se definen con tres puntos cualquiera.
%el círculo mínimo será el de menor radio que contenga todos los
%puntos

dos = true;

cont=1;

if ~all(distancias)

    for i=1:(tamano(1)-2)
        a=coordenadas(i,:);

        for j=(i+1):(tamano(1)-1)
            b=coordenadas(j,:);

            for k=(j+1):tamano(1)

                c=coordenadas(k,:);

                [xc yc zc rc] = TresPuntos(a,b,c,normal);

                %comprobar si todos los puntos están dentro del círculo:

distancias= ((coordenadas(:,1)-xc).^2 +
(coordenadas(:,2)-yc).^2 +
(coordenadas(:,3)-zc).^2) <= (rc^2 + eps);

                if all(distancias)

                    centros_x(cont)= xc;
                    centros_y(cont)= yc;
                    centros_z(cont)= zc;
                    centros_r(cont)= rc;

                    cont=cont+1;

                end

            end
        end
    end

end
```

```

                                end
                            end
                        end

                        [minimo posicion]= min(centros_r);

                        xcentro=centros_x(posicion);
                        ycentro=centros_y(posicion);
                        zcentro=centros_z(posicion);

                        radio=minimo;

                        dos=false;

                        end

                    end

                else

                    a=coordenadas(pos1,:);
                    b=coordenadas(pos2,:);
                    [xcentro ycentro zcentro radio] = RadioSegmento(a,b);

                end

            end

        end

        *****

function [xcentro ycentro zcentro radio] = RadioSegmento(a,b)

% Calcula el radio y el centro de una circunferencia que pasa por dos
puntos

    xcentro=(b(1,1)-a(1,1))/2 + a(1,1);
    ycentro=(b(1,2)-a(1,2))/2 + a(1,2);
    zcentro=(b(1,3)-a(1,3))/2 + a(1,3);
    radio= sqrt((b(1,1)-xcentro)^2+(b(1,2)-ycentro)^2+(b(1,3)-
        zcentro)^2);

end

        *****

function [xcentro ycentro zcentro radio] = TresPuntos(a,b,c,normal)

% Calcula el centro y el radio de una circunferencia que pase por tres
% puntos
% a, b, c y normal son vectores fila

    vec1= b - a;
    mitad1=vec1/2 + a;

```

```

D=- (vec1(1)*mitad1(1)+vec1(2)*mitad1(2)+vec1(3)*mitad1(3));

vec2= c - a;
mitad2=vec2/2 + a;
H=- (vec2(1)*mitad2(1)+vec2(2)*mitad2(2)+vec2(3)*mitad2(3));

A=[vec1; vec2;normal];

if rcond(A)> eps*5
    B=[-D;-H;0];

    centro=A\B;

    xcentro=centro(1);
    ycentro=centro(2);
    zcentro=centro(3);

    radio= a - centro';
    radio= norm(radio);
else
    xcentro=0;
    ycentro=0;
    zcentro=0;
    radio =0;
end

end

*****

function [matriz_paso]= cambioCoord(n)

%CAMBIO DE COORDENADAS: Se pasa de un sistema de coordenadas XYZ a otro
%donde z' = normal al plano,
%      x' y' pertenecen al plano.
%Datos: vector normal del plano (vector fila) -> n

n=n/norm(n); % Convertir n en unitario

a=[1 0 0]; % a -> cualquier vector (si se multiplica vectorialmente n
por cualquier vector se consigue un vector que pertenece al plano)

if all(a == n) % si a = n se coge otro a
    a = [0 0 1];
end

u= cross(n,a); % se consigue un primer vector que pertenezca al plano
u= u/norm(u); % unitario

v= cross(u,n); % un segundo vector que pertenezca al plano
v= v/norm(v); % unitario

matriz_paso= [u; v; n];

*****

```

```

function [eje_peq] = Elipse(coordenadas, centrox, centroy, radio)

% Calcula el eje pequeño de una elipse que contiene los puntos de la
matriz
% coordenadas (cada fila representa las coordenadas de un punto), sabiendo
% el centro y el radio del círculo que contiene dichos puntos.
% Los ejes de la elipse son paralelos a los ejes coordenados
% El centro de la elipse coincide con el del mínimo círculo que engloba
% todos los puntos y su eje mayor es igual al radio de dicho círculo

    eje_peq = radio;

    b = (coordenadas(:,2)- centroy)./ (1-((coordenadas(:,1)-
centrox)/radio).^2).^0.5; % parte positiva de la elipse
    b1 = (coordenadas(:,2)- centroy)./ -(1-((coordenadas(:,1)-
centrox)/radio).^2).^0.5; % parte negativa de la elipse

    B = [b; b1];

    numero=size(B);

    %elegir el b más pequeño para el que todos los puntos estén dentro

        for i=1:numero(1)

            distancias= ( ((coordenadas(:,1)-centrox)./radio).^2 +
((coordenadas(:,2)-centroy)./B(i)).^2 ) <= 1 ;

                if all (distancias) && (abs(B(i)) < eje_peq)

                    eje_peq = abs(B(i));

                end

            end

        end

    end

*****

function Criterios(Sigma_nm, Sigma_nr, Tau_r, Planos, CTES)

% Criterios(Sigma_nm, Sigma_nr, Tau_r, Planos, CTES)
% Se calculan las tensiones equivalentes y se dice si se produce o no
fallo
% según Findley, Mataka, McDiarmid y Dang Van.
%
% Datos: Sigma_nm (tensión normal media en cada instante), Sigma_nr
(normal
% variable), Tau_r (cortante variable), Planos(planos a analizar), CTES
% (datos del material)

    lim_traccion_alterna = CTES(4);
    lim_tau_torsion = CTES(2);
    sigma_ut = CTES(3);

```

```
if (lim_tau_torsion/lim_traccion_alterna > 0.5)

    % Findley: tau_nr + alfa*(sigma_nm + sigma_nr)
    % Plano crítico el que maximice la función anterior

    alfa = 2*(lim_tau_torsion/lim_traccion_alterna) - 1;
    beta = lim_tau_torsion;

    fprintf('\n Criterio de Findley: \n')

    Findley = Tau_r + alfa*(Sigma_nm + Sigma_nr);
    [max_tension max_plano]= max(Findley);

    fprintf('Plano de máxima tensión: \n')
    n = Planos(:,max_plano);
    n=n/norm(n);
    disp(n)
    fprintf('Con una tensión de %f \n', max_tension)

    if max_tension > beta

        disp('Se producirá el fallo por fatiga según Findley')

    else
        disp('No se producirá el fallo por fatiga según Findley')
    end

    % Mataka: tau_r + alfa*(sigma_m + sigma_r)< beta (mismos alfa y
    % beta)
    % Plano crítico: el de tensión de cortadura máxima

    fprintf('\n Criterio de Mataka: \n')
    [max_tension max_plano]= max(Tau_r);
    fprintf('Plano de máxima tensión: \n')
    n = Planos(:,max_plano);
    n=n/norm(n);
    disp(n)

    Mataka = Tau_r(max_plano)+ alfa*(Sigma_nm(max_plano)+
        Sigma_nr(max_plano));

    fprintf('Con una tensión de %f \n', Mataka)

    if Mataka > beta

        disp('Se producirá el fallo por fatiga según Mataka')

    else
        disp('No se producirá el fallo por fatiga según Mataka')
    end

    % McDiarmid -> dos criterios
    % Plano crítico (en ambos casos): el de tensión de cortadura
    máxima
```

```

fprintf('\n Primer criterio de McDiarmid: \n')

% Ctes

A = lim_tau_torsion;
B = (lim_tau_torsion + 0.5*lim_traccion_alterna)/
    (0.5*lim_traccion_alterna)^1.5;

McDiarmid = (max_tension + B*(Sigma_nr).^1.5)./A;

fprintf('Plano de máxima tensión: \n')
n = Planos(:,max_plano);
n=n/norm(n);
disp(n)
fprintf('Con una tensión de %f \n', max_tension)

if all(McDiarmid >1)

    disp('Se producirá el fallo por fatiga según el primer
        criterio de McDiarmid')

else
    disp('No se producirá el fallo por fatiga según el primer
        criterio de McDiarmid')
end

if all(Sigma_nm < (0.5*sigma_ut))

    McDiarmid2 = ((1 - 2*Sigma_nm./sigma_ut).^(-0.5) .*
        max_tension + B*Sigma_nr.^1.5)./A;
    fprintf('\n Segundo criterio de McDiarmid: \n')

    fprintf('Plano de máxima tensión: \n')
    n = Planos(:,max_plano);
    n=n/norm(n);
    disp(n)
    fprintf('Con una tensión de %f \n', max_tension)

    if all(McDiarmid2 > 1)

        disp('Se producirá el fallo por fatiga según el segundo
            criterio de McDiarmid')

    else
        disp('No se producirá el fallo por fatiga según el segundo
            criterio de McDiarmid')
    end
end

% Dang Van (macroscopico)

fprintf('\n Criterio de Dang Van: \n')

alfa = 3*(lim_tau_torsion/lim_traccion_alterna - 0.5);
beta = lim_tau_torsion;

```



```

DangVan = (Tau_r + alfa*(Sigma_nm + Sigma_nr))./beta;
[max_tension max_plano]= max(DangVan);

fprintf('Plano de máxima tensión: \n')
n = Planos(:,max_plano);
n=n/norm(n);
disp(n)
fprintf('Con una tensión de %f \n', max_tension*lim_tau_torsion)

if max_tension > 1

    disp('Se producirá el fallo por fatiga según Dang Van')

else
    disp('No se producirá el fallo por fatiga según Dang Van')
end

else

    fprintf('El criterio de Findley, el de Mataka, el de McDiarmid y
           el de Dang Van no son válidos: tau_n / sigma_n < 0.5')

end

end

```

Si se elige la opción número cuatro se llama a la siguiente función, que a su vez llama a:

- MaximoSegmento(), MínimoCírculo(), CambioCoord(), Elipse(), Giro() y Criterios(), ya citadas
- DibujaPlano(*normal*, *limites*), que dibuja un plano conociendo su normal y los límites del mismo
- DibujaElipseParam(centro_x, centro_y, a, b, angulo), que dibuja una elipse conociendo las coordenadas de su centro, sus dos ejes y el ángulo que forma su eje mayor con el eje x.

```
function DibujaPlano(n , fig , limite)

    % Dibuja un plano conociendo su normal

    figure(fig)
    hold on

    paso = limite/ 30;

    if n(3)~=0

        [x,y] = meshgrid(-limite:paso:limite, -limite:paso:limite);
        z=(-n(1)*x-n(2)*y)/n(3);
        surf(x,y,z, 'FaceColor', 'none', 'FaceAlpha',0.5)

    else
        if n(1)~=0

            [y,z] = meshgrid(-limite:paso:limite,
            -limite:paso:limite);
            x=(-n(2)*y-n(3)*z)/n(1);
            surf(x,y,z, 'FaceColor', 'none', 'FaceAlpha',0.5)

        else

            [x,z] = meshgrid(-limite:paso:limite,
            -limite:paso:limite);
            y=(-n(1)*x-n(3)*z)/n(2);
            surf(x,y,z, 'FaceColor', 'none', 'FaceAlpha',0.5)

        end

    end

    quiver3(0,0,0,n(1),n(2),n(3))
    view(n)

    *****
```

```
function DibujaElipseParam(centrox, centroy, angulo, a, b)

    % Dibuja una elipse centrada en (centrox, centroy)

    t=[0:0.01:2*pi];

    x= centrox + a*cos(t)*cos(angulo) - b*sin(t)*sin(angulo);
    y= centroy + a*cos(t)*sin(angulo) + b*sin(t)*cos(angulo);

    z= zeros(size(x));

    axis equal

    if a == b

        color= 'r'; % Dibuja la circunferencia en rojo
```

```

else

    color= 'g'; % Dibuja la elipse en verde

end

hold on
plot3(x,y,z,'LineWidth',2,'Color', color)

end

*****

```

Por último, se anexa el archivo de ayuda al que se accede escribiendo **help nombre_de_la_carpeta** en el Command Window de Matlab:

```

% PLANO CRÍTICO
%
% El archivo principal es Multiaxial.m. Aquí se importa el archivo excel
% con los datos del tensor de tensiones y otro con los datos del material.
% Se elige entre:
%
% - calcular con VonMises (se llama a VonMises.m)
% - calcular con un método de enfoque global: Sines o Crossland (se llama
a
%   EnfoqueGlobal.m)
% - calcular el plano crítico entre varios dados (se llama a Calculo.m)
% - visualizar las componentes de la tensión en un plano y los criterios
%   del círculo y la elipse (se llama a Planograf.m)
%
% Nota: el archivo excel que contiene los datos de tensión debe tener el
% siguiente formato:
%   tiempo   sigma_xx   sigma_yy   sigma_zz   tau_xy   tau_xz   tau_yz
%   dato     dato      dato       dato      dato     dato     dato
%   .         .         .         .         .         .         .
%   .         .         .         .         .         .         .
%   .         .         .         .         .         .         .
% La columna tiempo no se utiliza para los cálculos pero es útil para
% representación gráfica
%
% Nota2: los datos del material requeridos son:
% lim_probeta_rot   lim_tau_probeta_rot   sigma_ut   lim_ensayo_axial
tau_ensayo_axial
%   dato             dato                 dato         dato
dato
%
% Descripción de funciones:
%
% VonMises.m
%
% VonMises(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
% MUESTRAS, CTES)
% Calcula la tension equivalente de VonMises-Goodman y Goodman-VonMises
% Entrada: datos del tensor de tensiones para los instantes de

```

```
%
% tiempo, número de muestras y datos del material
% Salida: tensiones equivalentes de MG y GM, en pantalla
% Llamada por: Multiaxial.m
%
% EnfoqueGlobal.m
%
% EnfoqueGlobal(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
% MUESTRAS, CTES)
% Calcula las tensiones equivalentes de Sines y Crossland
% Entrada: datos del tensor de tensiones para los instantes de
% tiempo, número de muestras y datos del material
% Salida: tensiones equivalentes de Sines y Crossland, en pantalla
% Llamada por: Multiaxial.m
%
% Calculo.m
%
% Calculo(sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz, MUESTRAS,
% CTES)
% Entrada: datos del tensor de tensiones para los instantes de
% tiempo, número de muestras y datos del material
% Salida : (en pantalla) cálculo del plano crítico según los criterios
% Datos adicionales : pide una matriz que contenga los vectores normales
de
% los planos a analizar (existe la opción de planos por defecto)
% Llamada por : Multiaxial.m
% Llama a : MaximoSegmento.m, MinimoCirculo.m, CambioCoord.m, Giro.m,
% Elipse.m, Criterios.m
%
% Planograf.m
%
% Planograf (sigma_xx, sigma_yy, sigma_zz, tau_xy, tau_xz, tau_yz,
MUESTRAS,
% CTES)
% Se representa gráficamente la evolución de los componentes de tensión en
% el tiempo. Se aplican y dibujan el criterio del círculo y de la elipse
% Entrada: datos del tensor de tensiones para los instantes de
% tiempo, número de muestras y datos del material
% Salida : (en pantalla) cálculo de las tensiones en el plano dado; 2
% figuras: animación del movimiento de las tensiones; criterios
% del círculo y elipse
% Datos adicionales : vector normal del plano (columna)
% Llamada por : Multiaxial.m
% Llama a : MaximoSegmento.m, MinimoCirculo.m, CambioCoord.m, Giro.m,
% Elipse.m, DibujaPlano.m, DibujaElipseParam.m, Criterios.m
%
% MinimoCirculo. m
%
% Calcula el mínimo círculo que engloba a todos los puntos dados
% Entrada : coordenadas(coordenadas de un conjunto de puntos, cada
% coordenada es una fila), normal (vector normal del plano,
fila),
% pos1, pos2 (las posiciones en la matriz coordenadas de los dos
% puntos más alejados, se obtienen con MaximoSegmento.m);
% Salida : xcentro ycentro zcentro (coordenadas del centro de la
% circunferencia), radio, dos (será true si la circunferencia
% calculada sólo pasa por dos de los puntos)
% Llamada por : planograf.m, calculo.m
% Llama a : RadioSegmento.m, TresPuntos.m
%
```

```
% MaximoSegmento.m
%
% Calcula los puntos que forman el máximo segmento entre los puntos dados
% Entrada : coordenadas (coordenadas de un conjunto de puntos,
%           cada coordenada es una fila)
% Salida  : pos1, pos2 (posición en la matriz coordenadas de los 2 puntos
%           más alejados)
% Llamada por : Planograf.m, Calculo.m
%
% TresPuntos.m
%
% Calcula la circunferencia (centro y radio) que pasa por tres puntos.
% Entrada : a,b,c (coordenadas de los puntos), normal del plano
% Salida  : xcentro, ycentro, zcentro y radio
% Llamada por : MinimoCirculo.m
%
% RadioSegmento.m
%
% Calcula el radio y el centro de una circunferencia que pasa por dos
puntos
% Entrada : a,b (coordenadas de los puntos)
% Salida  : xcentro, ycentro, zcentro y radio
% Llamada por : MinimoCirculo.m
%
% CambioCoord.m
%
% Se pasa a un sistema de coordenadas donde z'=normal, x' e y' pertenecen
% al plano
% Entrada : normal del plano
% Salida  : matriz de paso
% Llamada por : planograf.m, calculo.m
% Nota: la función CambioPlanoPerpendicular.m realiza la misma función
pero
% dibujando los planos
%
% Giro.m
%
% Se giran los ejes x e y un ángulo en sentido contrario de las agujas
% del reloj
% Entrada : grados que se gira
% Salida  : matriz de paso
% Llamada por : Planograf.m, Calculo.m
%
% Elipse.m
%
% Calcula el eje pequeño de una matriz que contiene todos los puntos dados
% Entrada : coordenadas, centrox, centroy, radio (datos de la
circunferencia)
% Salida  : eje pequeño de la elipse
% Llamada por : Planograf.m, Calculo.m
%
% DibujaPlano.m
%
% Dibuja un plano conociendo su normal
% Entrada : normal del plano, número de figura
% Salida  : figura
% Llamada por : planograf.m
%
% DibujaElipseParam.m
```

```
%  
% Dibuja una elipse  
% Entrada : centrox, centroy, angulo (de los ejes de la elipse respecto a  
% los ejes coordenados), a, b (semiejes de la elipse)  
% Salida : elipse en la figura actual  
% Llamada por : planograf.m  
% Nota1: si a=b dibuja circunferencias  
% Nota2: a pesar de que en el archivo Proyecto haya una función  
% DibujaElipse.m no se utiliza  
%  
% Criterios(Sigma_nm, Sigma_nr, Tau_r, Planos, CTES)  
% Se calculan las tensiones equivalentes y se dice si se produce o no  
fallo  
% según Findley, Mataka, McDiarmid y Dang Van  
% Entrada: Sigma_nm (tensión normal media en cada instante), Sigma_nr  
(normal variable), Tau_r (cortante variable), Planos(planos a analizar),  
% CTES (datos del material)  
% Salida: tensiones equivalentes en pantalla  
% Llamada por : Planograf.m
```