

# Universidad de Sevilla

Escuela Técnica Superior de Ingenieros  
Departamento de Mecánica de Medios Continuos,  
Teoría de Estructuras e Ingeniería del Terreno  
Grupo de Elasticidad y Resistencia de Materiales

Trabajo de Fin de Master

**“Caracterización de Esquinas Bimateriales Elásticas Usando el Método de los Elementos de Contorno”**

Realizada por  
Eric Alcalá Mendoza

Dirigida por  
Dr. Enrique Graciani  
Dr. Alberto Barroso

Sevilla, Diciembre 2008

# INDICE

1. Introducción y objetivos.....	2
2. Integrales Independientes de Camino en Esquinas Elásticas.....	5
2.1. Caracterización de esquinas.....	5
2.2. El Factor de Intensificación de Tensiones Generalizado.....	7
2.3. Integrales Independientes del Camino.....	8
2.3.1. Integral conjugada de Buckner.....	9
3. El Cálculo de la Integral H en el Método de los Elementos de Contorno.....	11
3.1. Formulación general.....	11
3.2. Integración numérica.....	13
3.2.1. Regla de Simpson.....	13
3.2.2. Cuadratura de Gauss.....	15
4. Implementación.....	16
4.1. Diagramas de Flujo del Programa CFIT2D.....	16
4.2. Descripción del Programa CFIT2D.....	33
4.2.1. Bloque 1: Lectura y puntos de Gauss.....	33
4.2.2. Bloque 2: Evaluación de los desplazamientos y las tensiones e Integración Numérica .....	33
4.2.3. Bloque 3: Escritura de los archivos de Salida.....	35
4.3. Fichero de Entrada del Programa CFIT2D.....	35
4.4. Modelo Numérico .....	37
4.4.1. Características Generales del Programa CBEMD.....	37
4.4.2. Fichero de Entrada del Programa de CBEMD.....	39
5. Ejemplos Numéricos de Calculo del FIT en Esquinas Mediante el Método de los Elementos de Contorno y la Integral H.....	42
5.1. Verificación del Código.....	42
5.2. Aplicación del método desarrollado en el Método de los Elementos de Contorno.....	45
5.2.1. Esquina Homogénea de 270°.....	45
5.2.2. Esquina Bimaterial Isótropa.....	48
6. Conclusiones y desarrollos futuros.....	54
7. Bibliografía.....	55
8. Apéndice.....	59
8.1. Código Desarrollado.....	60

# Capítulo 1

## **Introducción y Objetivos**

Existen muchas aplicaciones ingenieriles que involucran la unión de materiales distintos, por ejemplo en los días actuales los automóviles y aviones contienen un gran número de componentes unidos desde cerámicos-metales hasta compuestos - metales. Una tensión no acotada podría desarrollarse en la esquina donde la interfaz entre los materiales unidos intersecta las superficies libres, cuando las uniones están sometidas a cargas mecánicas y/o térmicas. El tipo de singularidad depende de la geometría de la unión elegida y de las propiedades térmicas y elásticas de los materiales. Los fallos mayormente se inician en un punto singular, es por tanto importante caracterizar apropiadamente las singularidades de tensiones y los desplazamientos asociados, para que la geometría de la unión y la combinación de materiales pueden ser escogidas apropiadamente para retrasar el origen del fallo.

Las esquinas, se presentan como concentradoras de tensiones, convirtiéndolos en lugares potenciales para la aparición de defectos tales como pequeñas grietas que dañarían la unión al propagarse. Las interfaces unidas cerca de los bordes libres o cerca de la unión de materiales distintos sufren altos gradientes de tensiones debido a la presencia de diferencias térmicas y de rigidez de los materiales unidos.

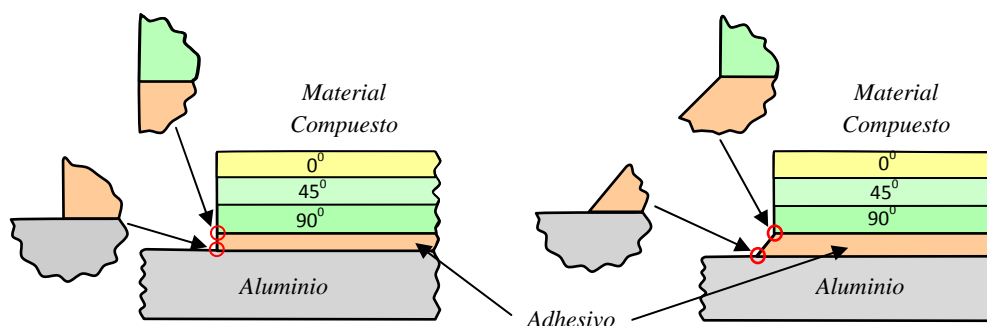


Figura 1.1: Diferentes tipos de uniones que presentan esquinas a lo largo de una interfaz de materiales distintos.

Hay muchos estudios dedicados al análisis de las tensiones singulares en esquinas y en fisuras desde el trabajo de Williams [1952] una de las primeras publicaciones trascendentes en este campo. Haciendo uso de la función de Airy y la separación de variables ( $r, \theta$ ), Williams analiza las singularidades de tensión que aparecen en una esquina de un material homogéneo con diferentes condiciones de contorno (siempre homogéneas). De las condiciones de contorno consideradas (bordes libres, empotrados o una combinación de ambos) encuentra que sólo las esquinas con condiciones de contorno mixtas (libre-empotrado) presentan singularidades para esquinas con un ángulo sólido (el que forma el material) por debajo de  $180^\circ$  (concretamente aparecen singularidades a partir de un ángulo de  $63^\circ$ ). Saltándonos el orden cronológico, es interesante mencionar aquí el trabajo de Vasilopoulos [1988] quien analizando el mismo problema, presenta con una precisión de 12 dígitos los órdenes de singularidad del problema de Williams, incluyendo términos de orden superior.

El cálculo del Factor de Intensificación de Tensiones (FIT) para grietas ha sido uno de los principales objetivos de la mecánica de la fractura elástica lineal desde que Williams [1957] e Irwin [1957] demostraron que este parámetro controla el campo de tensiones cerca del origen de la grieta. Experimentalmente se ha demostrado que existe un valor crítico del FIT que controla la iniciación de fractura. En la integridad estructural de componentes mecánicos el concepto del FIT es hoy en día ampliamente usado.

En referencia a las esquinas, inicialmente muchos estudios se enfocaron en la influencia del ángulo de la cuña y las propiedades elásticas del material en el orden de la singularidad. Cambiando luego esto hacia la determinación del campo de tensiones cerca de la esquina de la unión de materiales distintos, incluyendo la evaluación del factor de la intensificación de tensiones generalizado (FITG) asociado a la esquina.

En la actualidad existe abundante evidencia de que el factor de la intensificación de tensiones crítico de una cuña puede también ser utilizada de igual manera como en grietas para predecir el inicio del fallo en esquinas

agudas y en las interfaces de las esquinas ver por ejemplo Seweryn [1994] y Labossiere y Dunn [1998,1999] para esquinas en medios homogéneos y bimateriales, respectivamente.

La propuesta del trabajo que según se presenta consiste en usar el concepto del FITG para caracterizar esquinas bimateriales elásticas isótropas basándose en la expresión de la integral (independiente de camino)  $H$  desarrollada por Qian y Akisanya [1999], se implementó en un algoritmo, en el cual se utilizó como campo real la solución del método de los elementos de contorno en las tensiones y desplazamientos y como campo auxiliar las soluciones en términos de desplazamientos y tensiones desarrolladas por Qian y Akisanya [1999]. Para evaluar la integración numéricamente se ha utilizado en este trabajo dos métodos la regla de Simpson y la Cuadratura de Gauss.

# Capítulo 2

## ***Integrales Independientes de Camino en Esquinas Elásticas***

### **2.1. CARACTERIZACIÓN DE LAS ESQUINAS**

La caracterización completa de los campos de tensiones y desplazamientos en una esquina se obtiene mediante los órdenes de singularidad en tensiones y los correspondientes FITG. Está aceptado que estos FITG juegan un papel importante en criterios de fallo que gobiernan los procesos de fractura de estas esquinas.

El campo asintótico de tensiones es función de dos parámetros:

- El orden de singularidad de tensiones es  $(1-\lambda)$  que depende de la configuración local del problema es decir de las propiedades del material, de la geometría local del problema y del tipo de condiciones de contorno locales del problema, independientemente de la configuración lejana del problema, es decir de la geometría y distribución de cargas lejanas.
- El FITG que es el coeficiente en cada término de expansión asintótica, que depende de la configuración global del problema, del estado de cargas y de la geometría del problema y mide el peso que tiene el correspondiente a un modo singular o regular.

Existen numerosos trabajos que han abordado el cálculo de los  $\lambda$  del campo de tensiones en el entorno de las esquinas, con uno o varios materiales. En resumen, que no pretende ser exhaustivo, nos podemos remontar al trabajo de Williams [1952] , que obtuvo mediante el uso de la función de Airy los  $\lambda$  para un sector de material isótropo, Bogy [1971] usando la transformada de Mellin hizo lo propio para un bimaternal isótropo. Dempsey y Sinclair [1979, 1981], en un complejo trabajo para una esquina con n-materiales isótropos y múltiples condiciones de contorno (incluyendo fricción), posteriormente Vasilopoulos [1988] estudio el problema concreto de una entalla reentrante con diferentes

ángulos sacando unas tablas con los primeros ordenes de singularidad y con los modos simétricos y antisimétricos a los que esta asociados.

En general las tensiones y los desplazamientos cerca del vértice de la esquina pueden ser expresados como:

$$\sigma_{ij}^m(r, \theta) = \sum_{k=1}^N K_k r^{\lambda_k-1} f_{ijk}^m(\theta) + \sigma_{ij0}^m(r, \theta) \quad (2.1)$$

$$u_i^m(r, \theta) = \sum_{k=1}^N K_k r^{\lambda_k} g_{ik}^m(\theta) + u_{i0}^m(r, \theta)$$

Donde:

$i, j = r, \theta$ . Son coordenadas polares en la interfaz de la esquina.

$m = 1, 2$ . Es el número de materiales.

$\lambda_k (k=1, \dots, N)$ . Son autovalores del problema.

$f_{ijk}$  y  $g_{ik}$  son funciones que caracterizan la evolución angular adimensionales en función de:

- Las propiedades elásticas locales de los materiales
- La geometría local dependiente de los ángulos  $\theta_1$  y  $\theta_2$ ,
- Tipos de condiciones de contorno

$K_k$  es el factor de intensificación de tensiones generalizado de la esquina asociado con el autovalor  $\lambda_k$ .

Los segundos términos en (2.1) es decir:  $\sigma_{ij0}^m$  y  $u_{i0}^m$  son campos de tensiones y de los desplazamientos mas regulares (acotados y suaves) que puedan corresponder a cargas remotas y a condiciones de contorno no homogéneas asociados al vértice de la esquina, estos términos se eliminaran para cargas mecánicas remotas y son finitos para cargas térmicas y tracciones en la superficie cerca de interfaz de la esquina

Las funciones adimensionales  $f_{ijk}$  y  $g_{ik}$  se determinan analíticamente mientras que los valores de  $\lambda_k$  para una configuración local dada se obtienen mediante un procedimiento analítico con resolución numérico utilizando un código en Mathematca (a nivel de usuario) desarrollado por el Grupo de Elasticidad y

Resistencia de Materiales Barroso et al. [2007]. Este código recibe por fichero la geometría y propiedades de los materiales y evalúa analíticamente los valores de  $\lambda$ . El autovalor  $\lambda_k$  puede ser positivo real, negativo o conjugado. La expresión dada en (2.1) es para valores positivos reales de  $\lambda_k > 0$  o complejos con parte Real  $\lambda_k > 0$ .

## 2.2. FACTOR DE INTENSIFICACIÓN DE TENSIONES

La importancia de los Factores de Intensificación de Tensiones (FITs en adelante) en la Mecánica de la Fractura Elástica Lineal es innegable desde que Williams [1957] e Irwin [1957] demuestran que dichos factores controlan el estado tensional singular en el entorno de la punta de la grieta. Esta importancia se refuerza ante la evidencia experimental de la existencia de unos valores críticos de los FITs (la tenacidad a fractura) que controlan el inicio del fallo y su uso extensivo y fructífero (durante los últimos 50 años) en el diseño de componentes mecánicos.

Aunque resulte básico a la altura en la que nos encontramos, es importante remarcar que a diferencia de los *factores de concentración de tensiones*, los FITs son magnitudes dimensionales, cuya dimensión depende, como veremos más adelante en el caso de las esquinas, del exponente característico asociado a dicha esquina. Por lo tanto, la comparación de los valores de los FITs carece completamente de sentido físico cuando las magnitudes asociadas a dichos FITs no son iguales. El carácter adimensional de los factores de concentración de tensiones sí hace que sean comparables entre sí, y permiten, en los casos en los cuales su uso es apropiado (estados tensionales no singulares), comparar la severidad del estado tensional. En la comparación de la severidad del estado tensional en el caso de estados tensionales singulares, como el definido por (2.1), intervienen tanto los exponentes característicos  $\lambda_k$  como los FITs  $K_k$ .

Si bien, en presencia de grietas, el sentido físico de los FITs queda completamente definido a través de su equivalencia (en problemas de materiales con comportamiento elástico lineal) con la tasa de liberación de energía de deformación ( $G$ ), dicha relación deja de ser estrictamente válida en



problemas de esquinas, sin la presencia de grietas. Este hecho ha motivado que se denomine Factores de Intensificación de Tensiones Generalizados (FITGs) a los coeficientes del desarrollo en serie que define el estado tensional (2.1) cuando se trate de configuraciones de esquinas (sin grieta) en contraposición con su denominación en el caso de grietas.

Numerosos autores: Seweryn [1994], Dunn *et al.* [2001] y Yosibash *et al.* [2004] entre otros, defienden propuestas de criterios de fallo para materiales con comportamiento frágil, basados en el concepto de FITGs en configuraciones de esquinas multimateriales.

### **2.3. LAS INTEGRALES INDEPENDIENTES DEL CAMINO**

En cuanto al cálculo del FIT, la mayoría de los métodos existentes para este propósito pueden ser clasificados como métodos locales o globales, dependiendo si el método está basado en variables definidas cerca del origen de la grieta o si está basado en variables asociadas a todo el dominio. Los métodos locales usualmente son muy sensibles y su precisión depende fuertemente sobre todo del refinamiento cerca del origen de la grieta en el modelo discretizado. Entre estas técnicas destacan referencias clásicas como por ejemplo la extrapolación de tensiones y desplazamientos Chan *et al.* [1970].

En cambio, los métodos globales están basados en integrales de conservación como la integral  $J$ ,  $M$ ,  $L$  y  $H$ . Los procedimientos para el cálculo del FIT vía integrales de conservación constituyen el procedimiento más robusto debido a que estos evitan la necesidad de reproducir de forma muy precisa el campo singular de tensiones cerca del fondo de la grieta.

Probablemente, la integral  $J$  introducida por Rice, [1968]

$$J = \int_{\Gamma} (w dy - T^n \frac{\partial u}{\partial x}) ds \quad (2.2)$$

Es la integral independiente del camino más empleada para el cálculo de los FITs en el caso de grietas. En modos mixtos es necesario desacoplar los

modos I, II y III para obtener los correspondientes FITs, ver Huber *et al.* [1993]. En (2.2)  $w$  representa la densidad de energía de deformación, la dirección y es normal a la grieta y  $T^n$  y  $u$  son respectivamente el vector tensión y el vector de desplazamientos. Se puede demostrar, que para comportamiento elástico lineal la integral  $J$  se identifica con la tasa de liberación de energía de deformación  $G$ , representando la integral  $J$  la variación de energía potencial ( $\Pi$ ) con el crecimiento de grieta ( $a$  = tamaño de la grieta)  $J = -\partial\Pi / \partial a$ .

La integral  $M$ , introducida originalmente por Knowles y Sternberg [1978]

$$M = \int_{\Gamma} (wx_a n_a - T_a^n u_{k,a} x_a) ds \quad (2.3)$$

Representa una generalización de la integral  $J$  y ha sido adaptada y utilizada para la evaluación de FITGs por Im y Kim [2000] en esquinas 2D y por Lee y Im [2003] para vértices 3D.

Ejemplos de la utilización de la denominada integral  $L$  se pueden encontrar en Choi y Earmme [1992] para grietas circulares.

No todos los métodos para el cálculo del FIT en grietas basados en integrales de conservación pueden extenderse para el cálculo del FITG en esquinas, y aunque algunos puedan extenderse no necesariamente permiten calcular los FITG asociados a términos de tensión de orden superior Chen y Lu[2004].

### 2.3.1. Integral conjugada de Bueckner

La integral- $H$  también conocida como la integral conjugada de Bueckner [1973] se deriva a partir del segundo teorema de reciprocidad de Betti y permite calcular el FITG en esquinas 2-D basándose en la condición de biortogonalidad entre las familias de autovalores (exponentes de singularidades) positivos y negativos. Este método ha sido aplicado para el cálculo del FITG en esquinas en medios homogéneos Sinclair y Griffin [1984] o bimateriales Qian y Akisanya [1999].

Todas las metodologías basadas en las integrales conservativas necesariamente requieren de uso de un campo auxiliar. Por su parte, la integral de Bueckner requiere solamente de la evaluación de variables naturales como desplazamientos y tensiones, lo cual representa una inherente ventaja de este método y a su vez puede implicar una mejor precisión en comparación con otras integrales de conservación que requieren gradientes de desplazamientos y/o de tensiones.

La expresión general de la integral de Bueckner tiene la forma:

$$K = \int_{\Gamma} [(\sigma_{ij} u'_i - \sigma'_{ij} u_i)] d\Gamma \quad (2.4)$$

Un ejemplo del camino de integración  $\Gamma$  se muestra en la Figura 2.

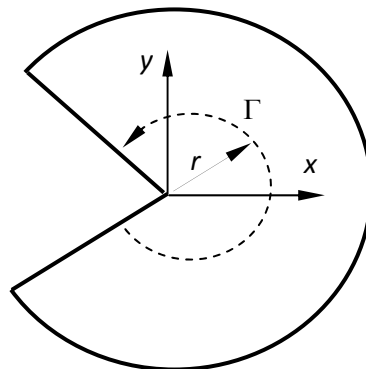


Figura 2.1: Camino de una integral de Bueckner

# Capítulo 3

## ***El Cálculo de la Integral H en el Método de los Elementos de Contorno***

### **3.1. FORMULACIÓN GENERAL**

El método de la integral de camino utilizado es sencillo y permite la evaluación de múltiples FITGs de forma simultánea. El método de la integral H es análogo a la utilizada por Qian y Akisanya [1999], pero adaptada a un modelo numérico del Método de Elementos de Contorno (MEC) en vez del Método de Elementos Finitos (MEF).

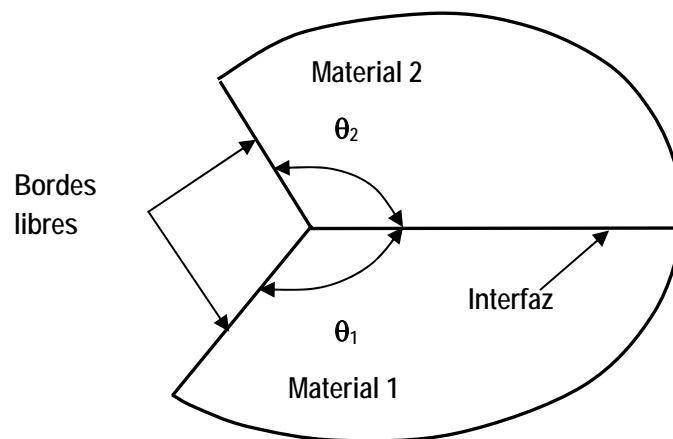


Figura 3.1 Configuración general de una esquina bimaterial

En la Figura 3.1 se muestra la configuración geométrica de una esquina caracterizada por los ángulos  $\theta_1$  y  $\theta_2$ , que corresponde a la superficie libre del material 1 y del material 2, respectivamente. El material en la región  $0 \leq \theta \leq \theta_2$  es referido como el Material 2 mientras que el Material 1 ocupa la región  $-\theta_1 \leq \theta \leq 0$ . Ambos materiales son elásticos lineales, isótropos y homogéneos. Al considerarse que la geometría tiene bordes libres cerca del vértice de la esquina, el campo de tensiones y desplazamientos vienen dados por la ecuación (2.1).

El método de la integral de camino de Bueckner es utilizado para evaluar el factor de la intensificación de tensiones  $K_K$  (o  $H_k$ ) de una esquina asociada a cada autovalor  $\lambda_k$ .

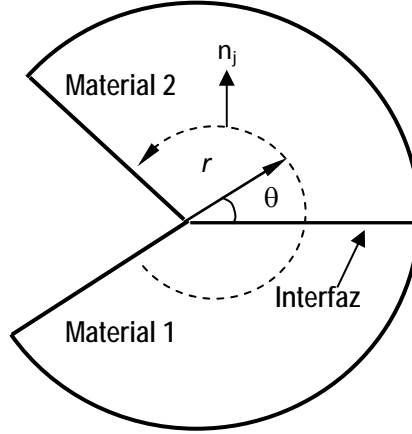


Figura 3.2 Un camino de integración  $\Gamma$  en una esquina bimaternal

La evaluación de la integral a lo largo del camino  $\Gamma$  en la ecuación (3.1), el campo de tensiones y desplazamientos real lo denominaremos campo actual  $(\sigma_{ij}^{act}, u_i^{act})$  es tomado como término asociado a un autovalor de interés  $\lambda_k$ , el campo auxiliar  $(\sigma_{ij}^*, u_i^*)$  es escogido como campo con el termino asociado con el autovalor opuesto al autovalor de interés  $\lambda_k^* = -\lambda_k$  y satisface las mismas condiciones de contorno que el campo actual, de donde  $Q_k$  es una constante, la cual depende de las propiedades de los materiales y de la apertura de la esquina; donde la magnitud de  $Q_k$  no depende de la distancia radial de  $\Gamma$ .

$$\oint_{\Gamma} (\sigma_{ij}^{act} u_i^* - \sigma_{ij}^* u_i^{act}) n_j ds = Q_k \quad (3.1)$$

$$\oint_{\Gamma} (\sigma_{ij}^{BEM} u_i^* - \sigma_{ij}^* u_i^{BEM}) n_j ds = K_k \quad (3.2)$$

En la determinación del factor de intensificación de tensiones  $K_k$  ( $H_k$ ) (3.2), la solución de elementos de contorno  $(\sigma_{ij}^{BEM}, u_i^{BEM})$  se utiliza para evaluar la integral a lo largo de  $\Gamma$  en la ecuación (3.2) el campo auxiliar se escoge como termino en (2.1) asociado con  $\lambda_k$ , pero con el factor de intensificación  $H_k$

remplazado por  $H_k^*$ , y el autovalor  $\lambda_k$  remplazado por  $\lambda_k^* = -\lambda_k$ . El campo auxiliar para la evaluación de la intensificación de tensiones asociado a la esquina con autovalor  $\lambda_k$  es entonces dado por:

$$\sigma_{ij}^*(\lambda_k) = H_k^* r^{-\lambda_k - 1} f_{ijk}^*; \quad u_i^*(\lambda_k) = H_k^* r^{-\lambda_k} g_{ik}^* \quad (3.3)$$

donde  $f_{ijk}^* = f_{ijk}(-\lambda_k)$  y  $g_{ik}^* = g_{jk}(-\lambda_k)$  es bien conocido que si  $\lambda_k$  es un autovalor para el problema también lo es  $-\lambda_k$ .

y el campo auxiliar se escoge con la estructura de (3.1) pero con la intensificación  $H_k^* = 1/Q_k$ . Esto asegura el valor de (3.2) dando el valor del factor de intensificación de tensiones de la esquina  $K_k$ . Este proceso puede ser repetido para evaluar la magnitud de la intensificación de tensiones asociada a otro autovalor, pero solo cambiando el campo auxiliar y el campo actual para las tensiones y desplazamientos para ese particular autovalor.

## 3.2. INTEGRACIÓN NUMÉRICA

Para evaluar la integración numéricamente se ha utilizado en este trabajo dos métodos: la regla de Simpson y la Cuadratura de Gauss. El código quedó implementado con la cuadratura de Gauss debido a su mayor exactitud y a su rapidez al converger con una menor cantidad de puntos de integración.

### 3.2.1. Regla de Simpson

La regla de Simpson se basa en el reemplazo de una función complicada o datos tabulados con una función aproximada que sea fácil de integrar. En la figura 3.3, se presenta la aproximación de una integral como el área bajo una parábola por medio de un polinomio de orden dos. Para este polinomio, se requiere conocer tres datos:  $a$ ,  $b$ , y un punto intermedio (a la mitad) entre  $a$  y  $b$ .

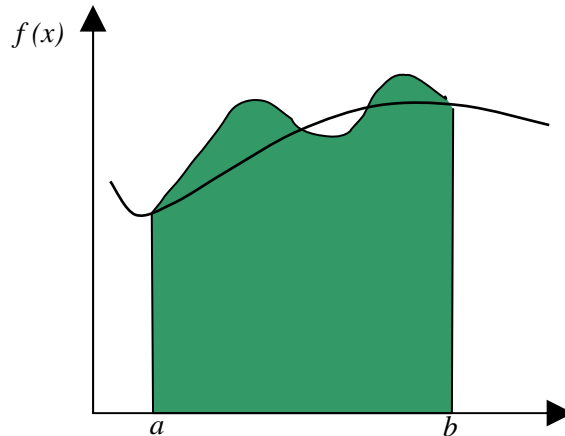


Figura 3.3 Aproximación de una integral como el área bajo una parábola.

$$I = \int_a^b f(x)dx \cong \int_a^b f_2(x)dx \quad (3.4)$$

Si a y b se toman como  $x_0$  y  $x_2$  respectivamente y  $f_2(x)$  se representa por un polinomio de Lagrange de orden dos la integral se transforma.

$$I = \int_{x_0}^{x_2} \left[ \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2) \right] dx \quad (3.5)$$

La integración de la forma anterior y la ordenación da como resultado.

$$I = (b-a) \frac{f(x_0) + 4f(x_1) + f(x_2)}{6} \quad (3.6)$$

Donde  $a=x_0$ ,  $b=x_2$  y  $x_1=(b+a)/2$ .

La ecuación (3.6) tiene la forma trapezoidal  $I=\text{ancho} \times \text{altura promedio}$ .

Si se divide el intervalo de integración  $[a, b]$  en  $n$  segmentos iguales y se aplica a cada uno de ellos la ecuación (3.6) se obtiene:

$$I = (b-a) \frac{f(x_0) + 4 \sum_{i=1,3,5}^{n-1} f(x_i) + 2 \sum_{j=2,4,6}^{n-2} f(x_j) + f(x_n)}{3n} \quad (3.7)$$

### 3.2.2. Cuadratura de Gauss

En lugar de fijar los puntos donde el integrando será evaluado, como se hizo en la aproximación de Simpson, estos puntos se dejan libres entonces se tendrá con los correspondientes coeficientes de peso,  $2n$  incógnitas, lo cual origina una función que será reproducida por medio de un polinomio de grado  $2n-1$ . Dando una mejor aproximación.

La integral se calcula de la siguiente manera:

$$\int_b^a f(x)dx = \sum_{i=1}^n c_i f(x_i) \quad (3.8)$$

Se aplica esta formula en el intervalo  $(-1,+1)$  para cualquier intervalo  $(a, b)$  que será obtenido mediante la transformación:

$$\frac{\xi - 1}{\xi + 1} = \frac{x - a}{x - b} \quad (3.9)$$

La integral que se obtiene entonces será:

$$\int_{-1}^1 g(\xi)d\xi = \sum_{i=1}^n \omega_i g(\xi_i) \quad (3.10)$$

Siendo  $n$  el numero de puntos de integración,  $\xi_i$  las coordenadas del punto de integración  $i$  y  $\omega_i$  la función de peso asociado a estos puntos. Las funciones  $g(\xi)$  se aproximan mediante los Polinomios de Legendre  $P_n(\xi)$ :

$$P_n(\xi) = \frac{1}{2^n} \sum_{k=0}^{Int(n/2)} \frac{(-1)^k (2n-2k)!}{k!(n-k)!(n-2k)!} \xi^{n-2k} \quad (3.11)$$

$\xi_i$  se toma como el valor obtenido de la condición:

$$P_n(\xi)=0$$

Los valores de pesos se obtienen a partir de:

$$\omega_i = \frac{2}{(1 - \xi_i^2) \left[ \frac{dP_n(\xi)}{d\xi} \right]_{\xi=\xi_i}^2} \quad (3.12)$$



# Capítulo 4

## ***Implementación del Código de Calculo de FITG en Esquinas Bimateriales***

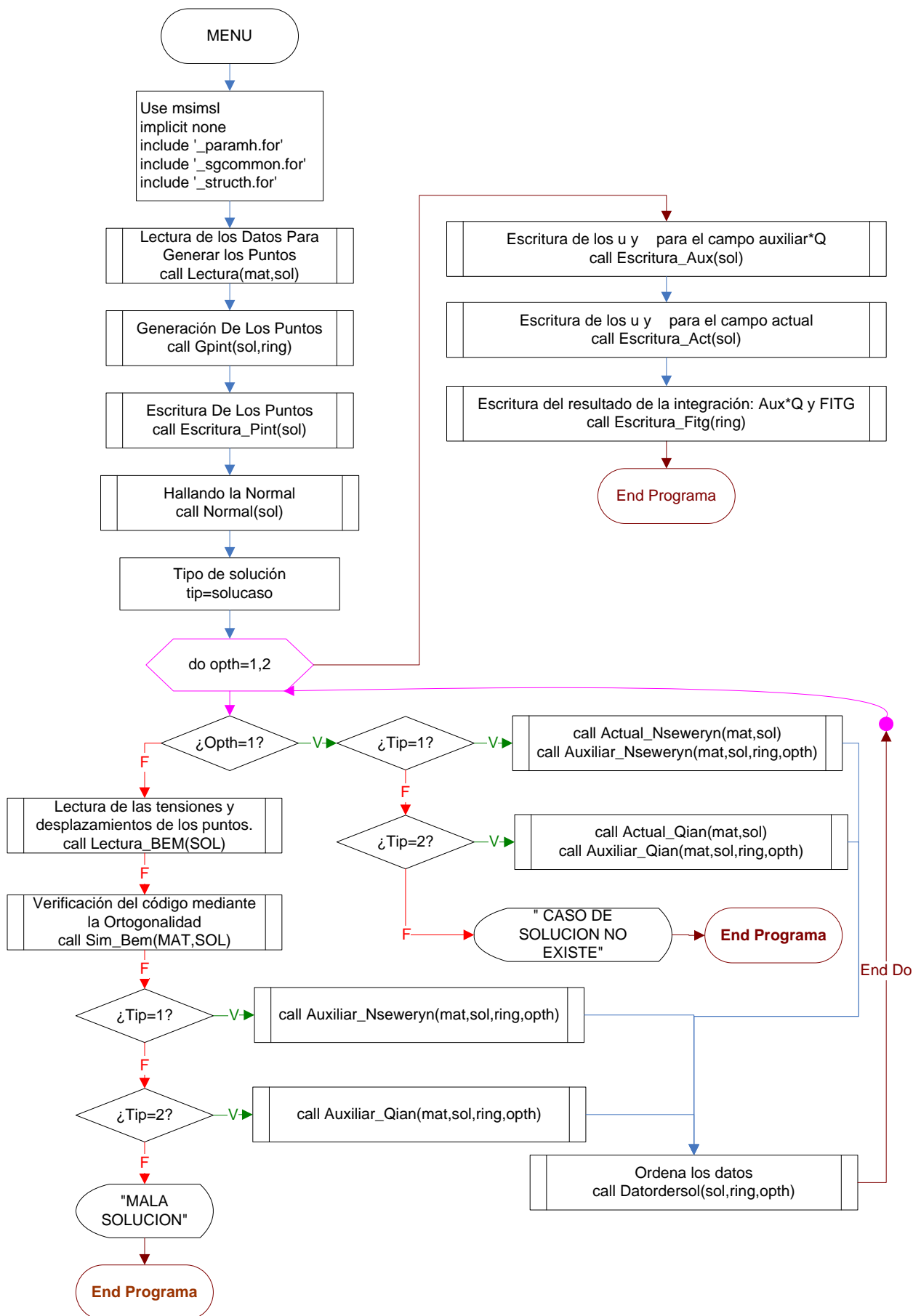
Para hallar el factor de intensificación de tensiones se ha implementado un programa escrito en Fortran 90 denominado CFITG2D. En este apartado describiremos brevemente cómo se ha llevado a cabo dicha implementación, cómo se realiza la entrada de datos del problema y que resultados proporciona dicho programa.

### **4.1. DIAGRAMAS DE FLUJO DEL CÓDIGO**

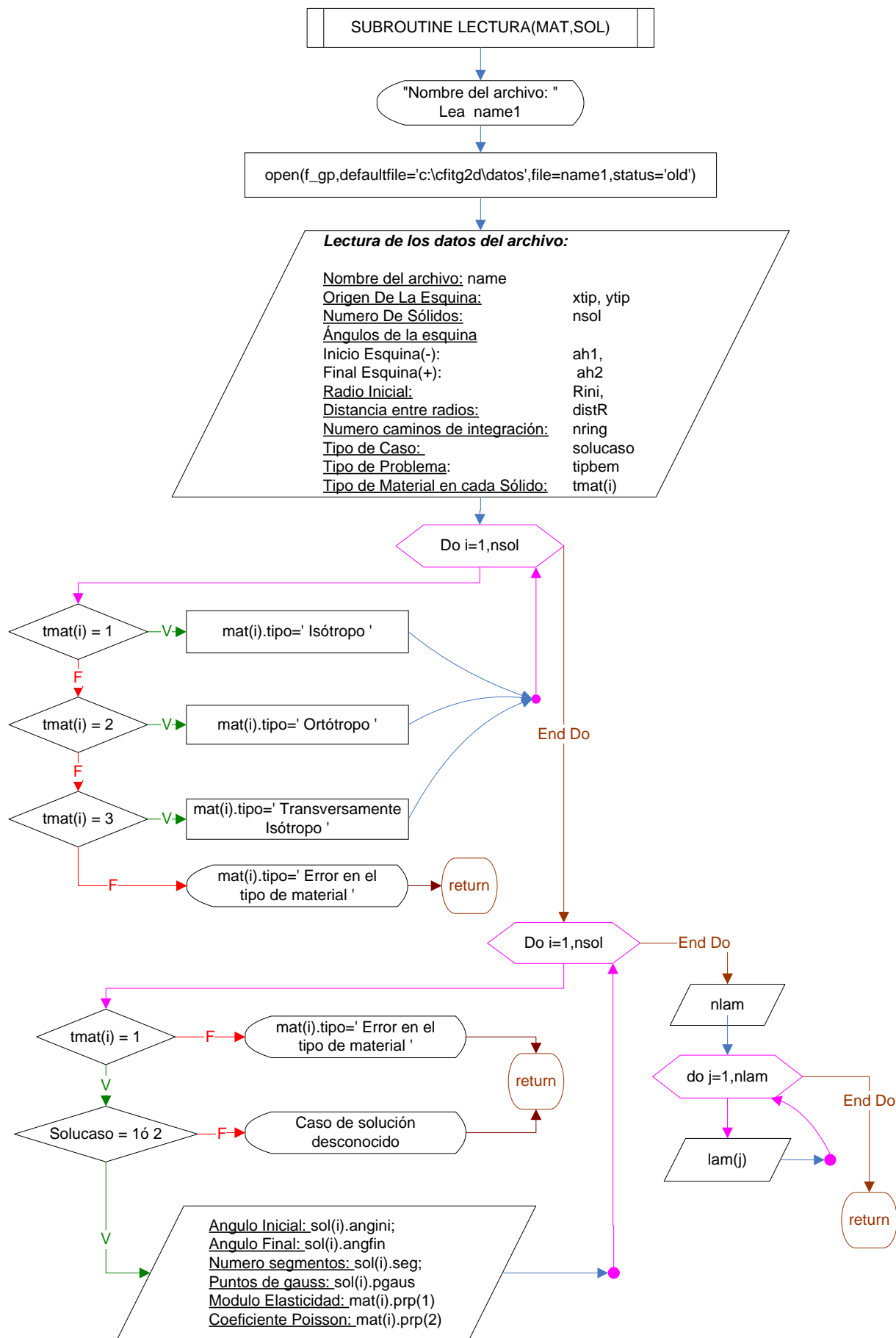
Para una mejor comprensión del programa se ha escrito los diagramas de flujo del este, en que por motivos de simplificación se han omitido las variables tanto locales como externas, así como formatos de escritura centrándonos en como se desarrolló el programa. Sin embargo estos datos se aprecian en el código del programa que se encuentra en el Apéndice A1. El programa consta de las siguientes subrutinas:

- Menú principal del programa
- Lectura de Datos de Entrada
- Generador de Puntos Internos
- Cuadratura de Gauss
- Normal de los puntos
- Campo Actual con funciones de Seweryn
- Campo Auxiliar con funciones de Seweryn
- Campo Actual con funciones de Qian y Akisanya
- Campo Auxiliar con funciones de Qian y Akisanya
- Lectura de Tensiones y Desplazamientos de BEM
- Verificación del Código (Comprobando Ortogonalidad)
- Escritura Puntos Internos Formato CBEMD
- Escritura del Campo Auxiliar (Tensiones y Desplazamientos)
- Escritura del Campo Actual (Tensiones y Desplazamientos)
- Escritura de Q (Factor de normalización) y FITG

# Menú principal del programa

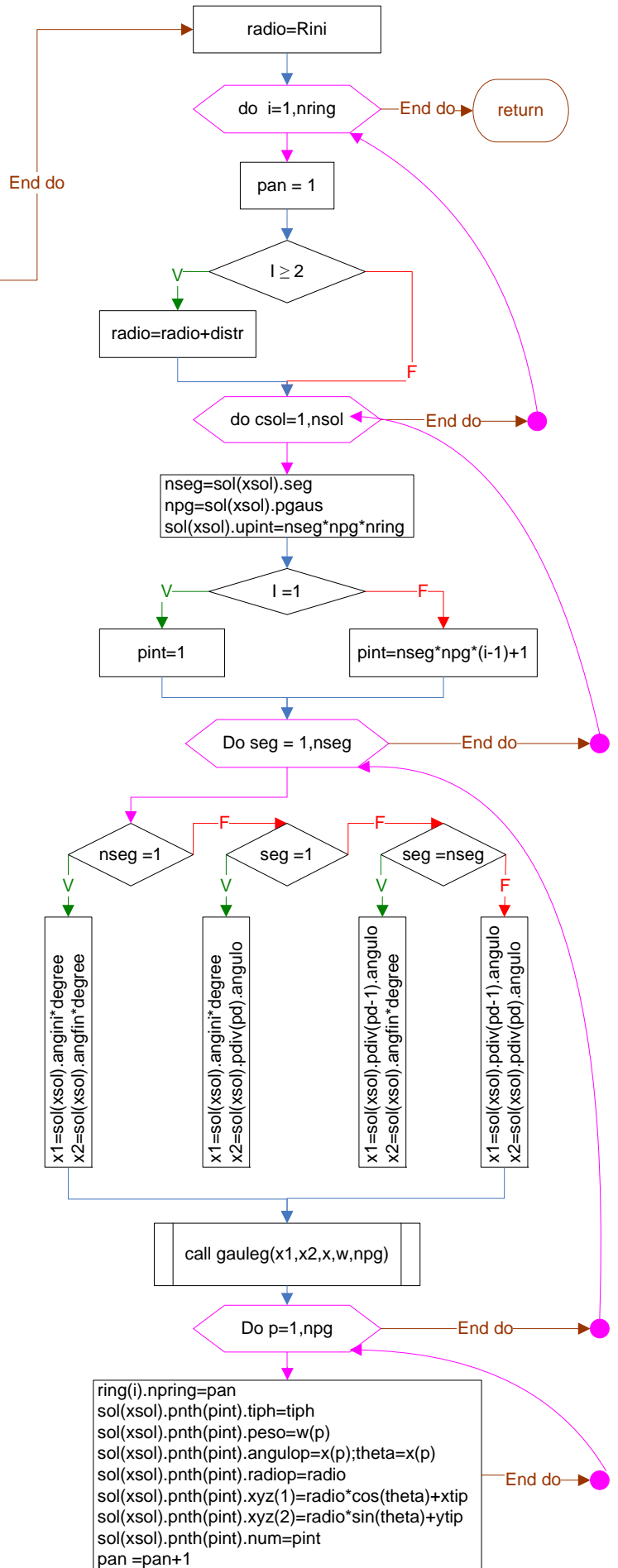
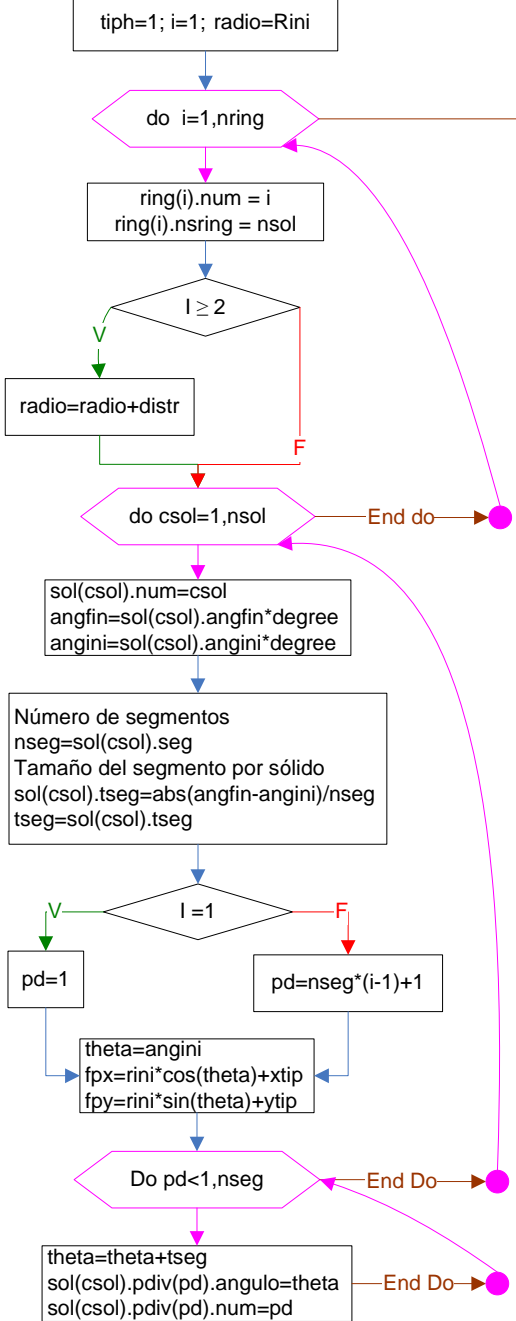


# Lectura de Datos de Entrada

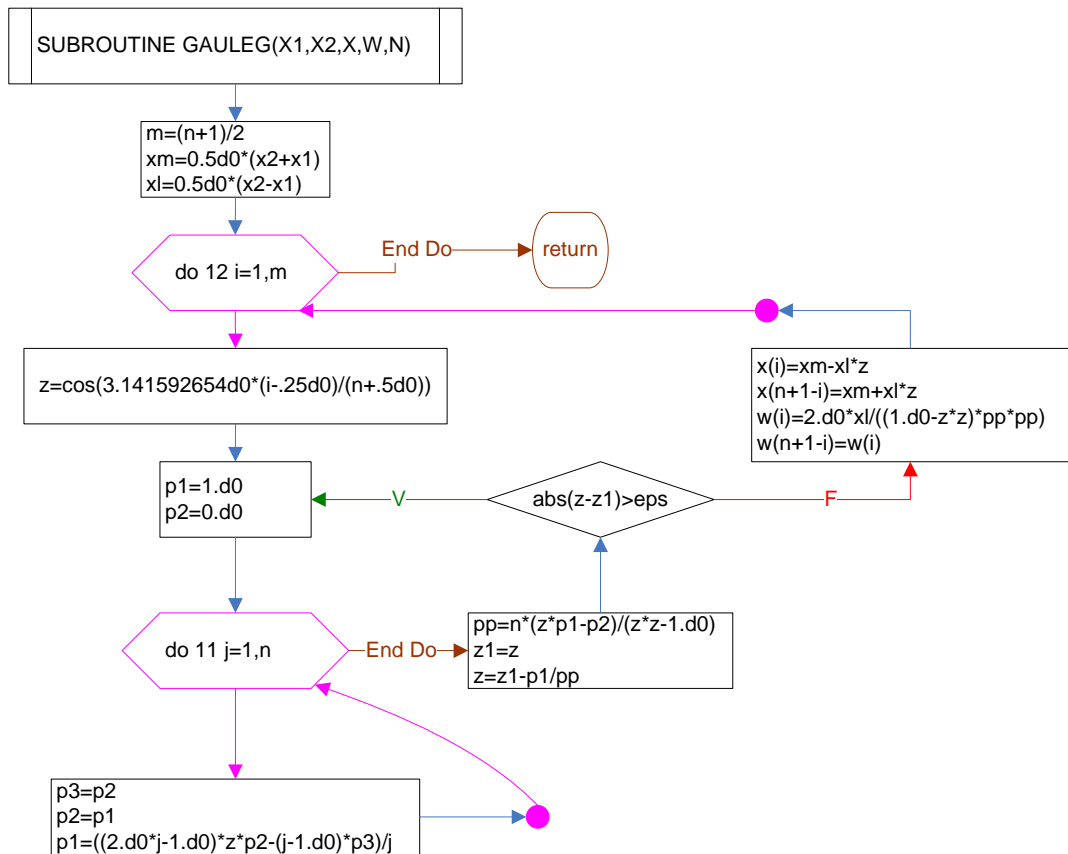


# Generador de Puntos Internos

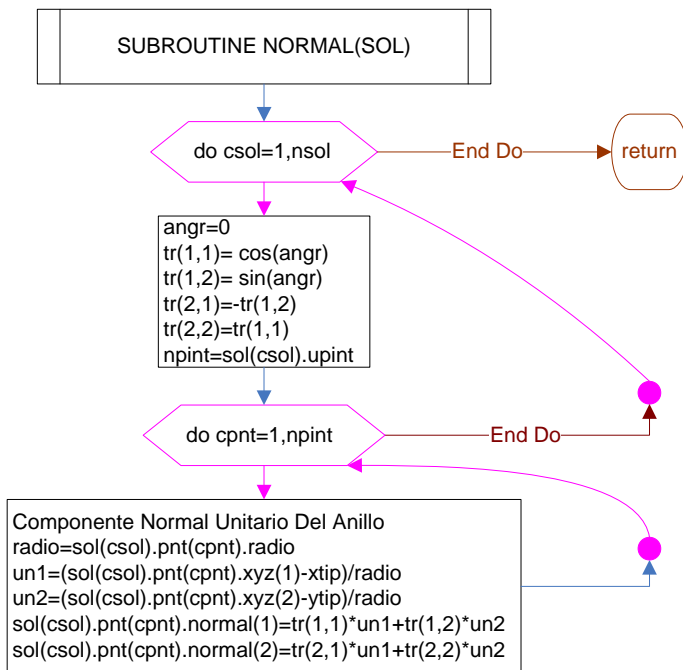
SUBROUTINE GPINT(SOL,RING)



## Cuadratura de Gauss



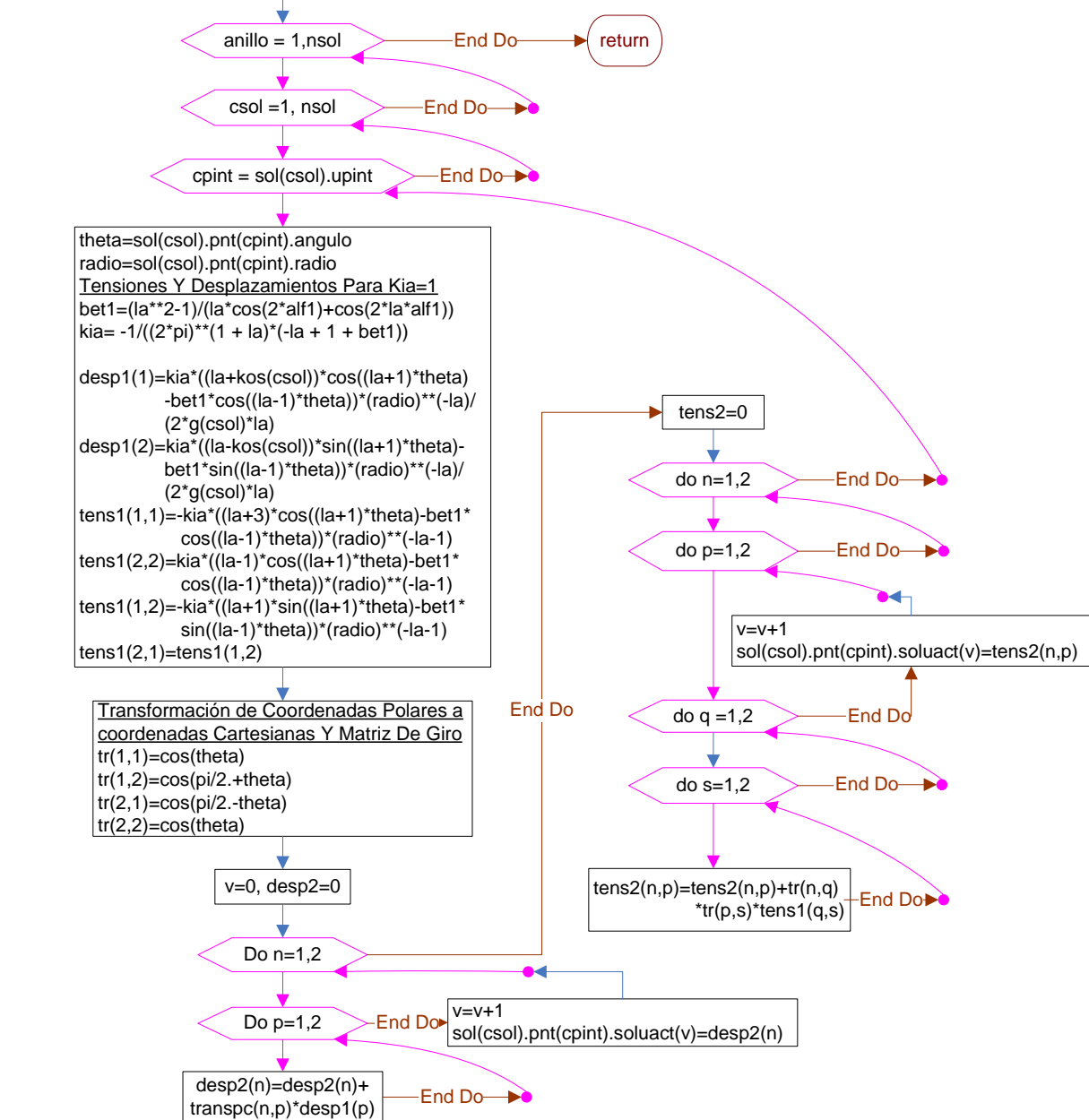
## Normal de los puntos



# Campo Actual con funciones de Seweryn

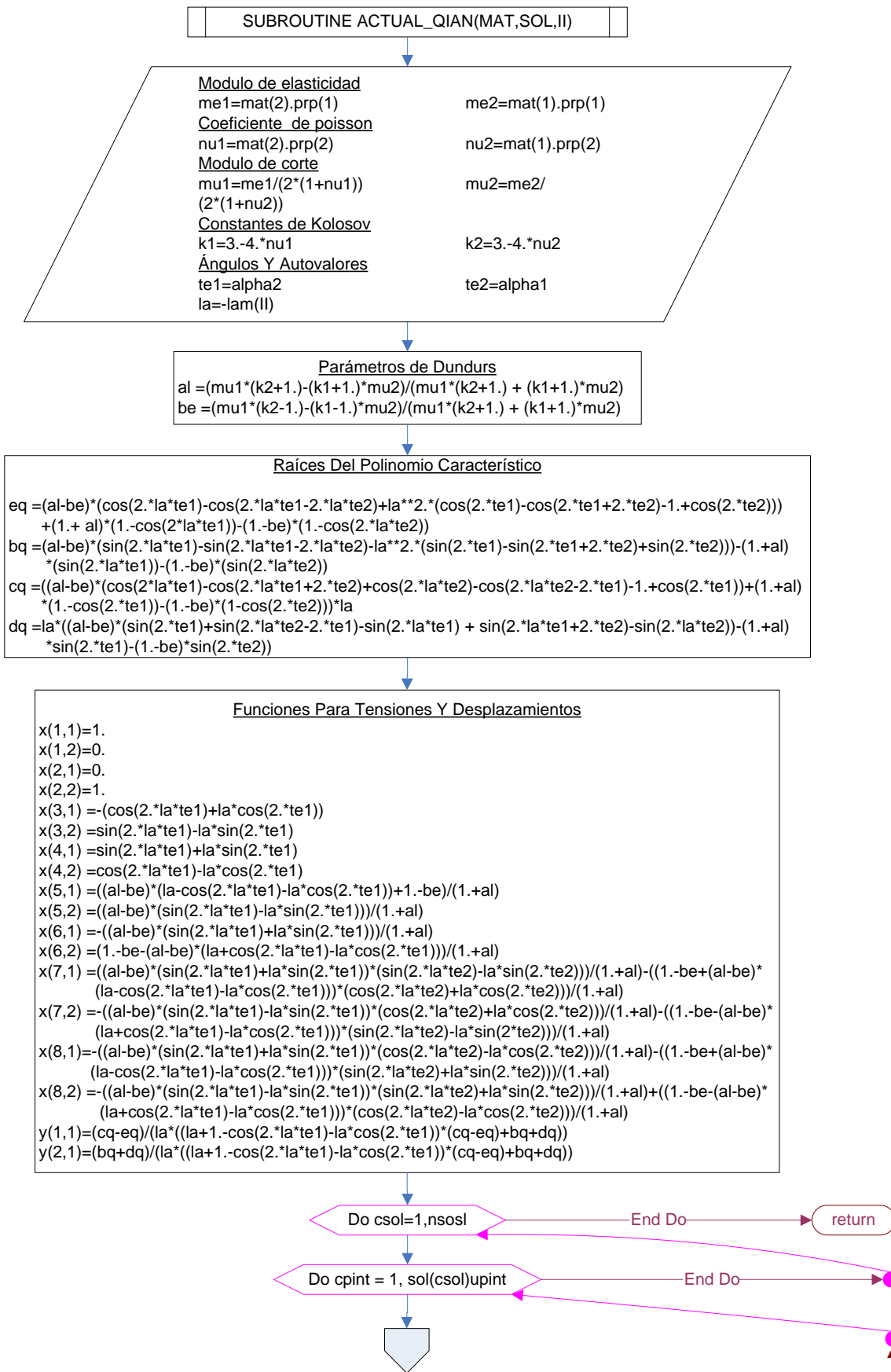
SUBROUTINE ACTUAL\_NSEWERYN(MAT,SOL)

Modulo de elasticidad  
 $e(1)=mat(1).prp(1)$                        $e(2)=mat(2).prp(1)$   
Coefficiente Poisson  
 $rnu(1)=mat(1).prp(2)$                        $rnu(2)=mat(2).prp(2)$   
Modulo De Corte  
 $g(1)=e(1)/(2*(1+rnu(1)))$                        $g(2)=e(2)/(2*(1+rnu(2)))$   
Constante Kolosov  
 $kos(1)=3.-4.*mat(1).prp(2)$                        $kos(2)=3.-4.*mat(2).prp(2)$   
Autovalores: Lambdas  
 $la=-(-lam(1))$   
Angulo de la esquina  
 $alf1=alpha2$

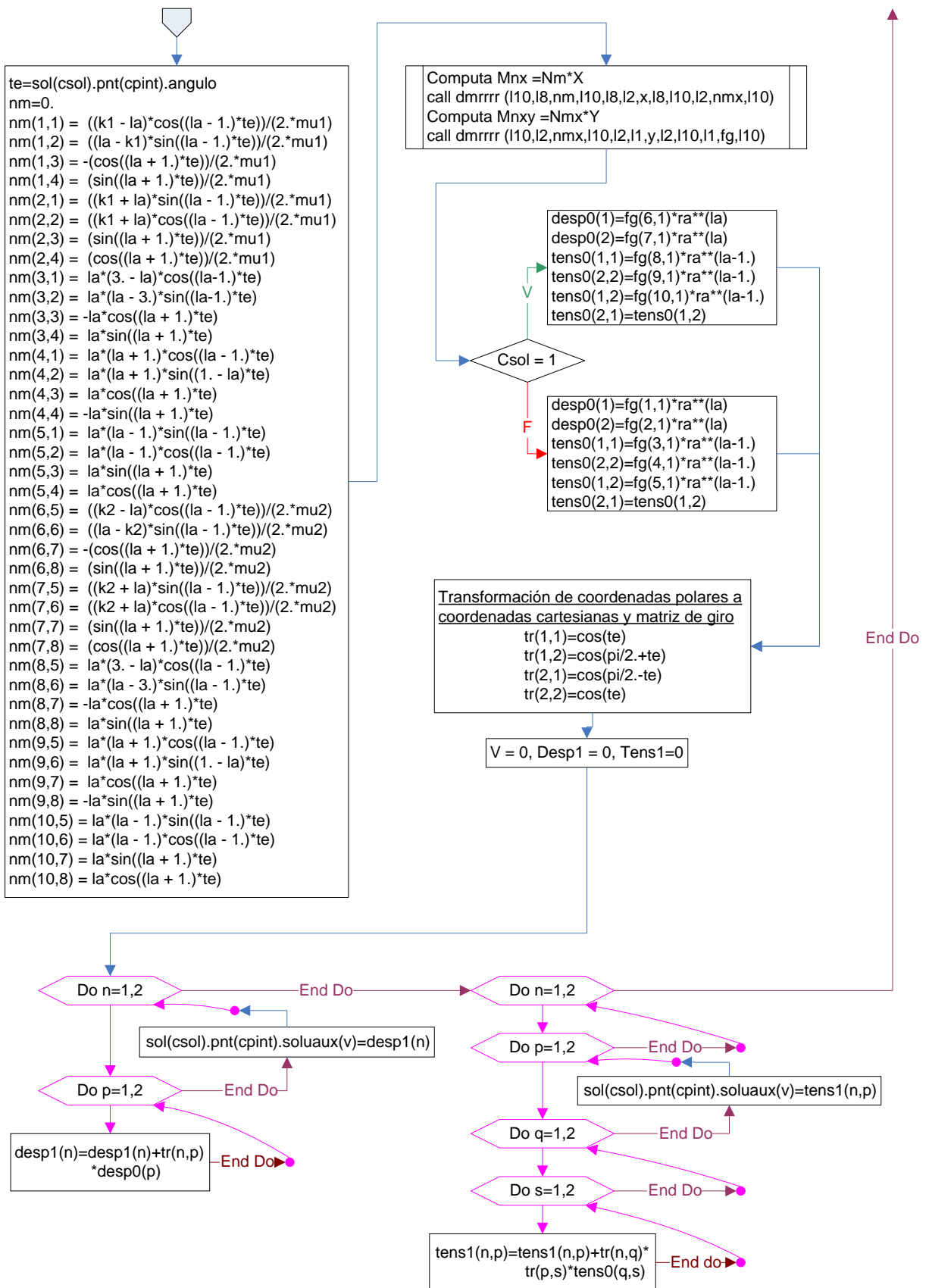




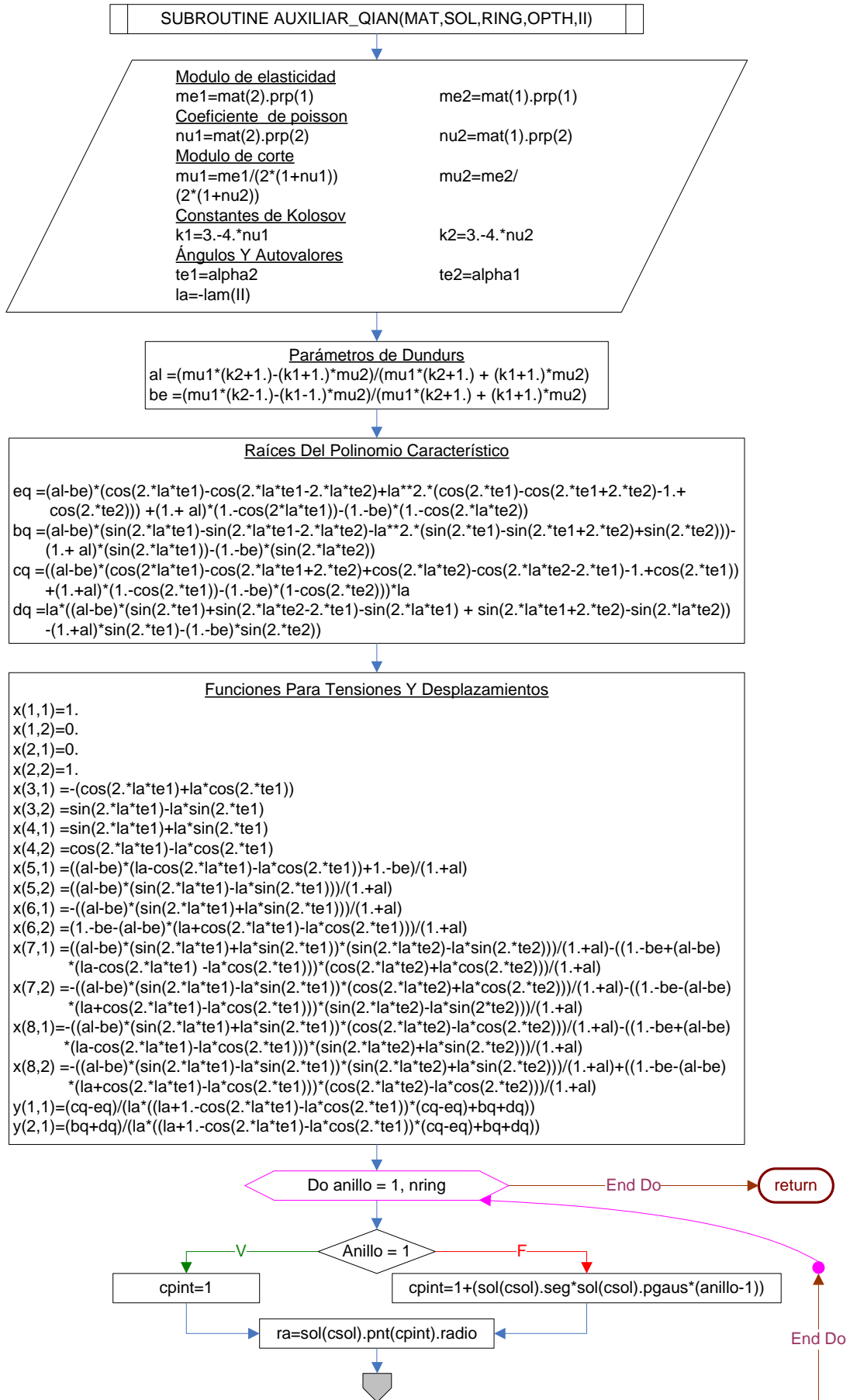
## Campo Actual con funciones de Qian y Akisanya

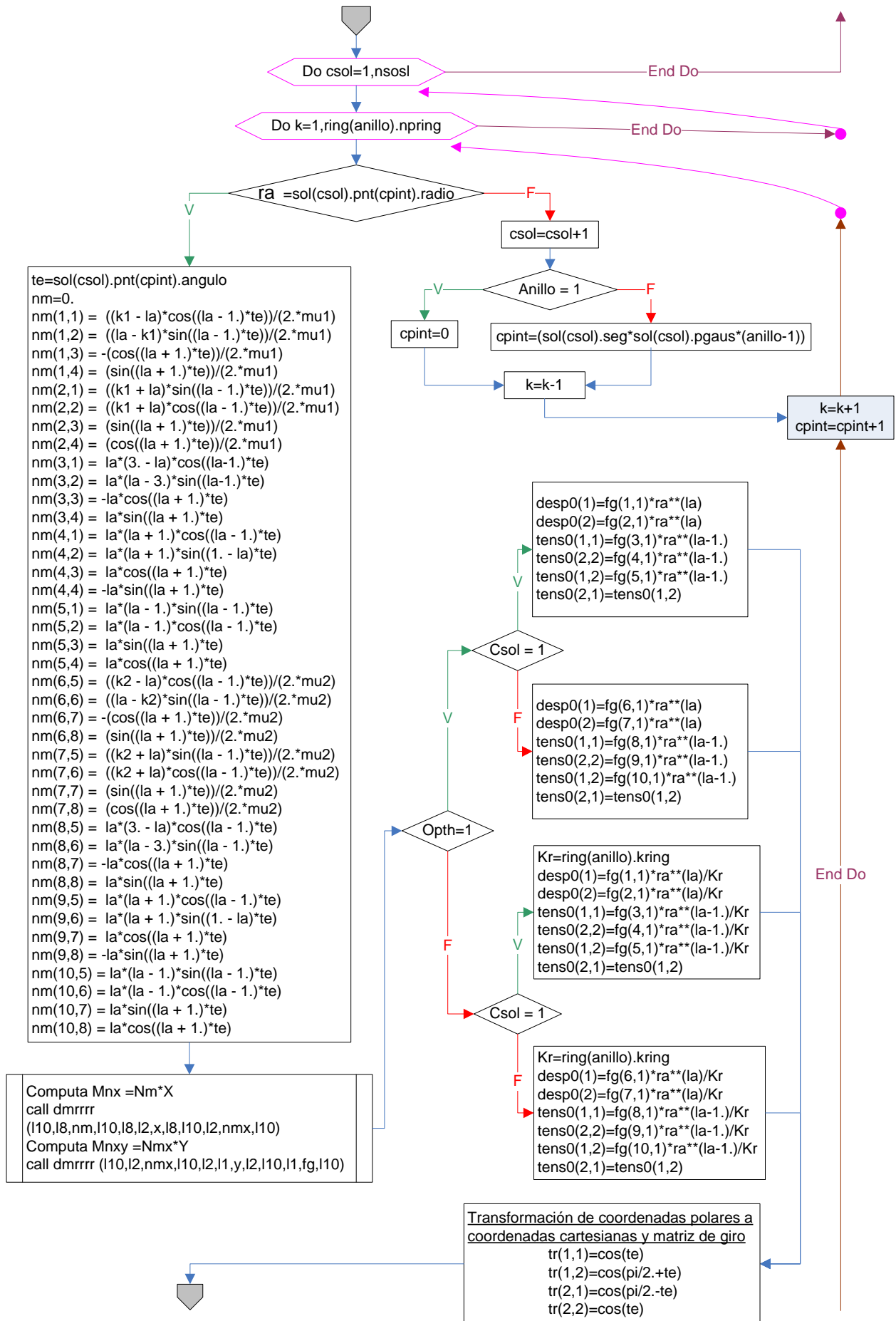


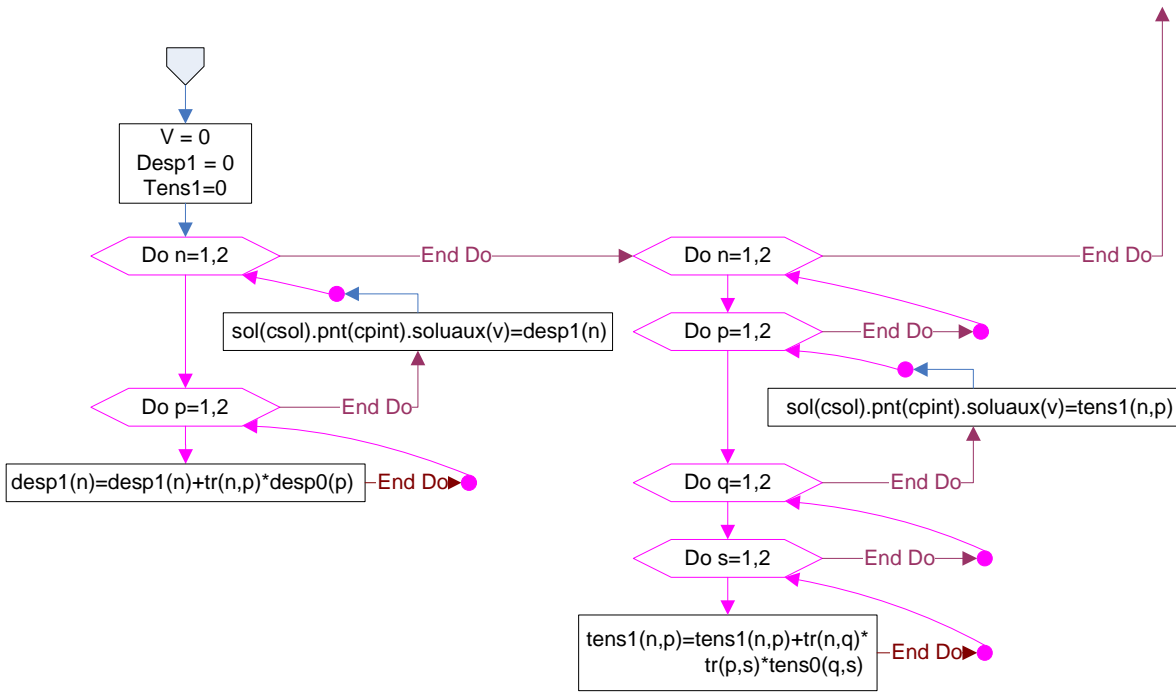




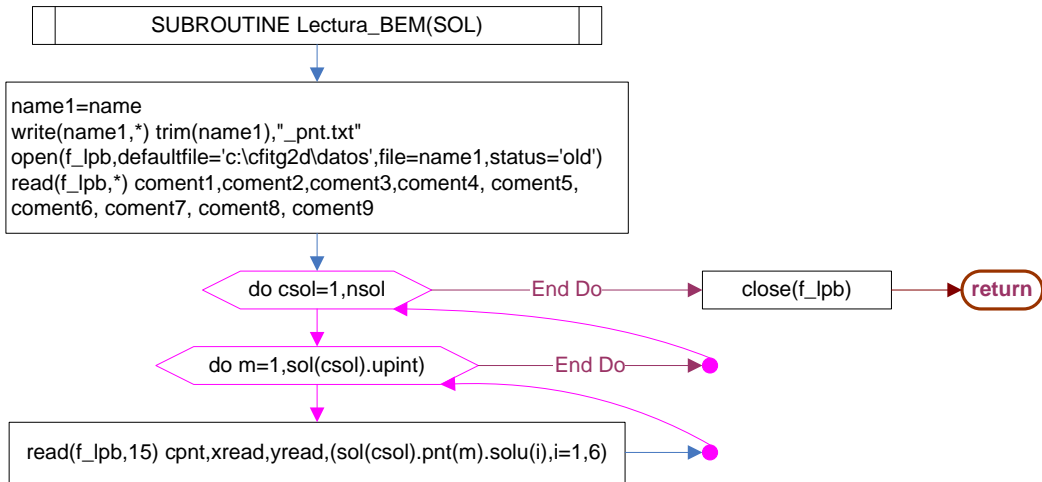
# Campo Auxiliar con funciones de Qian y Akisanya



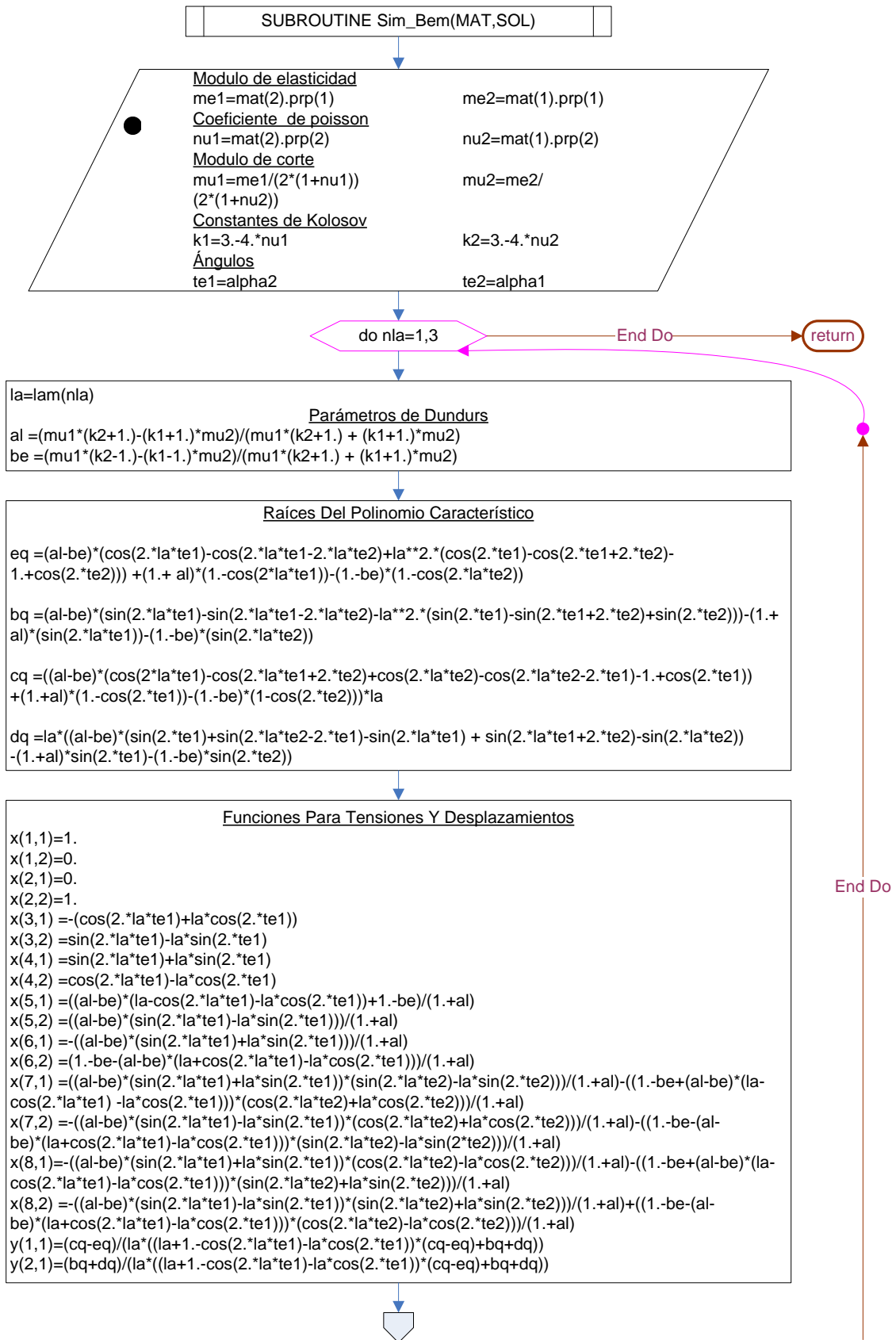


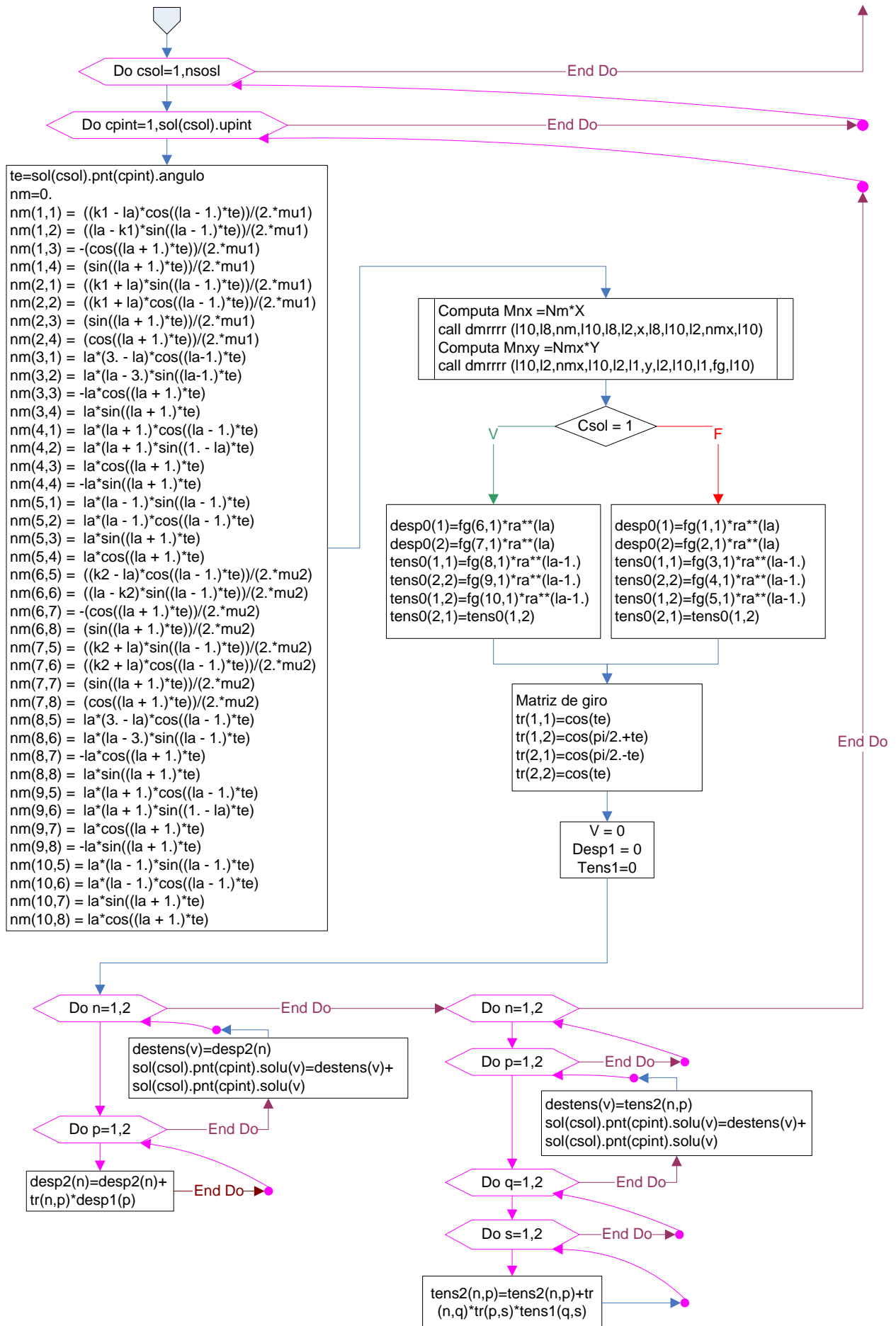


## Lectura de Tensiones y Desplazamientos de BEM



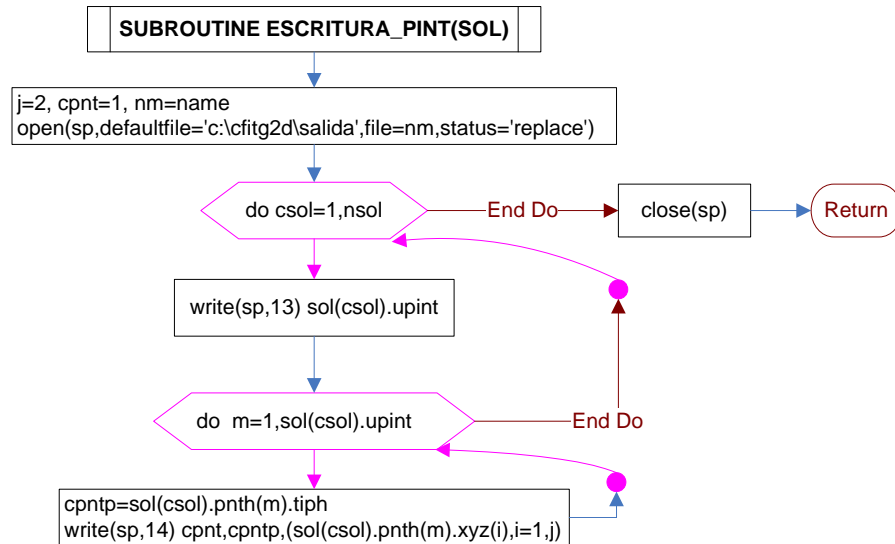
## Verificación del Código (Comprobando Ortogonalidad)



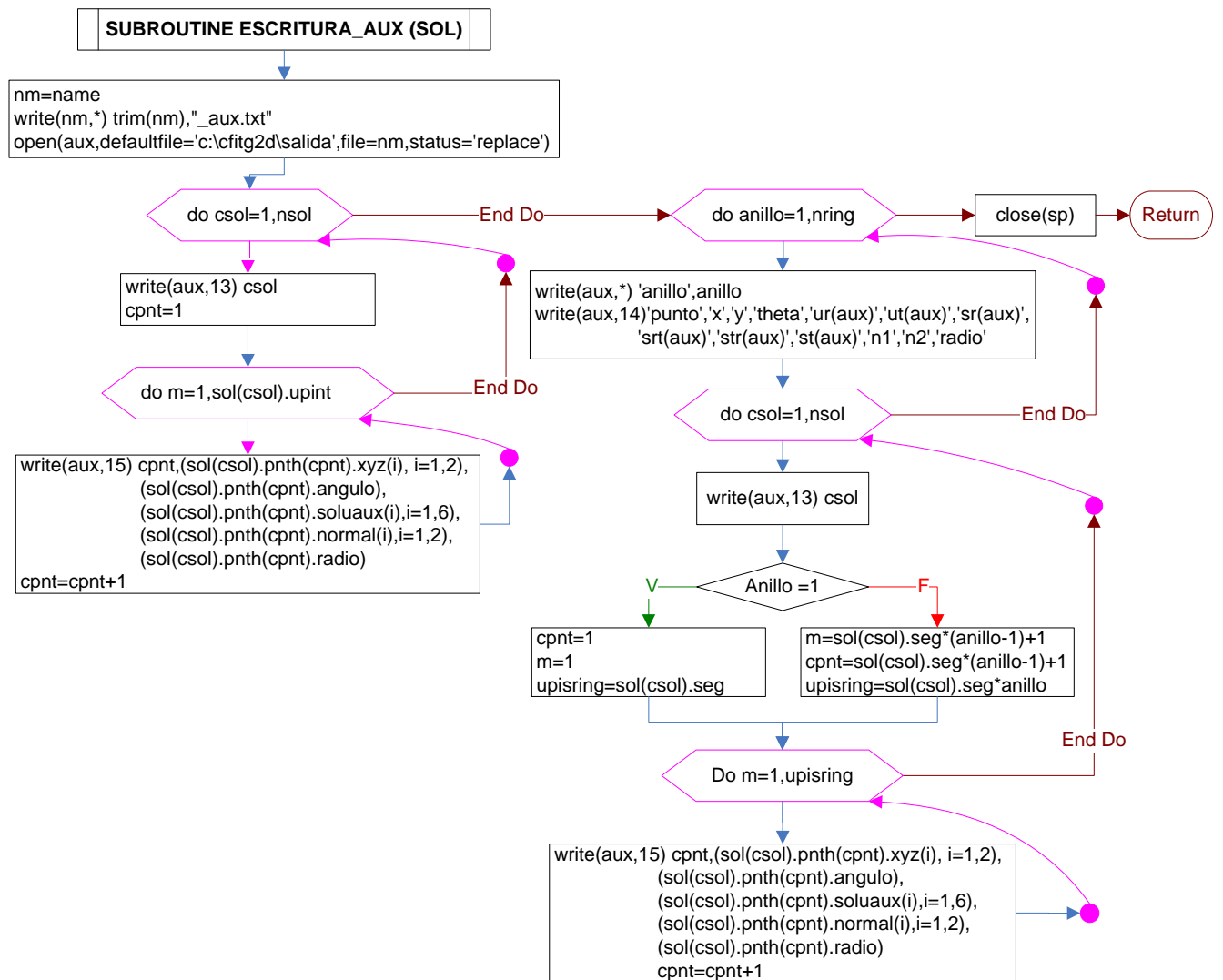




## Escritura Puntos Internos

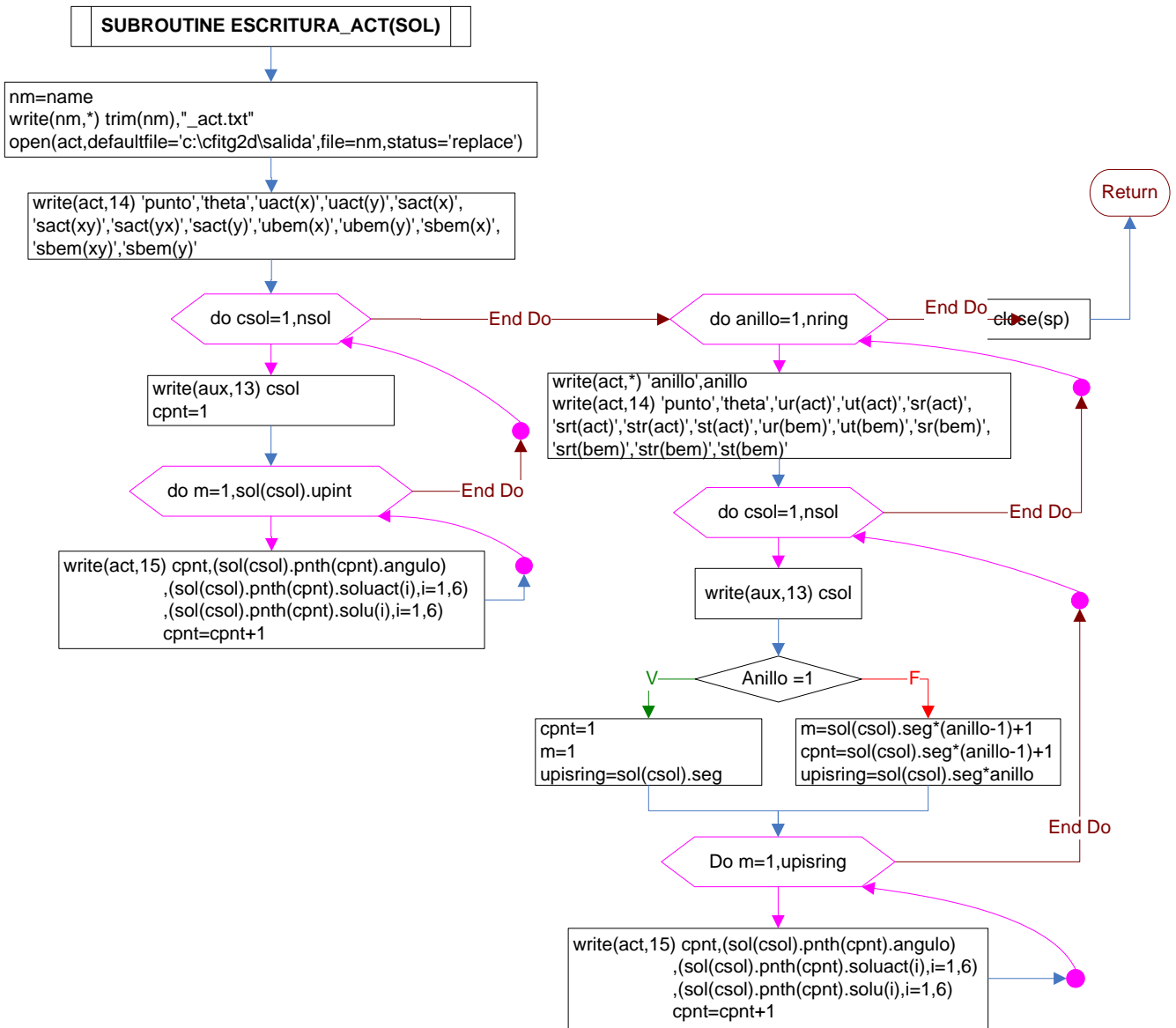


## Escritura Campo Auxiliar

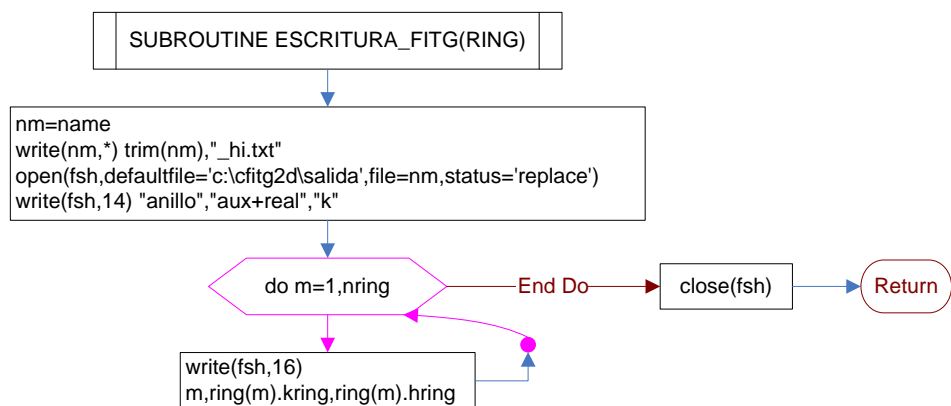




## Escritura Campo Actual



## Escritura de los FITGS



## **4.2. DESCRIPCION DEL PROGRAMA CFIT2D**

A continuación se da una descripción de las principales características del programa utilizado para hallar los FITGs.

### **4.2.1. Bloque 1: Lectura y puntos de Gauss**

- **Lectura de los datos de entrada.**

En este bloque se leen las variables básicas del programa como son, el ángulo y las coordenadas de la esquina, el numero de caminos a integrar, el radio y la distancias de separación entre cada camino.

En este programa se han implementado las funciones utilizadas por Seweryn [1994] para hallar el campo de tensiones y desplazamientos de un material homogéneo, y las funciones utilizadas por Qian y Akisanya [1999] para esquinas bimateriales.

Luego se definirán los ángulos entre los cuales esta comprendido cada material, el tipo de material, el numero de segmentos en que se dividirán los materiales con la cantidad de puntos de Gauss que tendrá cada segmento; donde los puntos de Gauss son los puntos internos del modelo que se evaluará mediante el Método de los Elementos de Contorno con el código desarrollado por Graciani [2006] CBEM2D.

Por ultimo se ingresan las constantes elásticas de cada material y los autovalores asociados a la esquina los cuales se analizaran con el código de Barroso (2007) el cual fue utilizado a nivel de usuario.

- **Generación de los puntos de Gauss**

En esta parte del código se realiza la generación de los puntos de gauss de manera que estén distribuidos formando arcos circunferenciales dentro de cada material (sólidos), los puntos de Gauss están dispuestos dentro de segmentos que dividen los arcos de los materiales. En esta parte se evalúan los pesos de cada punto y se almacena en un vector para su posterior uso en la integración numérica.

- **Escritura**

El código da un fichero de salida donde se presenta el número de puntos de Gauss por material y sus coordenadas cartesianas; el modo de presentación del fichero está diseñado de tal manera que se pueda añadir al fichero de entrada del código de CBEMD donde los puntos de Gauss son los puntos internos del modelo.

#### **4.2.2. Bloque 2: Evaluación de los desplazamientos y las tensiones e Integración Numérica.**

- **Evaluación de los desplazamientos y las tensiones del campo asintótico**

Se calcula en cada punto de Gauss (puntos internos) la normal y conforme a la ecuación (3.1) las tensiones y desplazamientos se evalúan para un campo actual asociados a un autovalor  $\lambda_k$ , luego se evalúan las tensiones y desplazamientos para el campo auxiliar asociado a su correspondiente  $(-\lambda_k)$  mediante las funciones desarrolladas por Qian y Akisanya [1999].

- **Integración Numérica.**

En un primer momento se la integración numérica se realizó mediante la Regla de Simpson y se observó que era necesaria una gran cantidad de puntos integración para obtener errores relativos pequeños. Posteriormente se utilizó la Cuadratura de Gauss que nos permite una mayor exactitud del código, esta integración se realiza contra las agujas del reloj, mediante este método se convierte una integral de línea en una integral de dominio vía el teorema de Gauss.

- **El FITG**

Una vez obtenidos las tensiones y desplazamientos mediante el código de elementos de contorno CBEMD estos son leídos por el programa CFITG2D para solucionar la ecuación (3.2) y así hallar el FITG asociado a su correspondiente autovalor.

- **La Ortogonalidad:**

Se creó una subrutina la cual nos permite verificar el código mediante la ortogonalidad de las funciones Szabó, y Babuška [1991]. Para permitir el funcionamiento de esta subrutina se tiene que poner los valores constantes manualmente y solo funciona para las funciones de Qian y Akisanya [1999]

#### 4.2.3. Bloque 3: Escritura de los archivos de Salida

El código tiene 3 archivos de salida en los cuales se escriben las tensiones y desplazamientos para los campos actuales y auxiliares (el utilizado para la ecuación 3.2) y el FITG para cada camino de integración.

#### 4.3. FICHERO DE ENTRADA DEL PROGRAMA CFIT2D

A continuación presentamos de manera esquemática el archivo de entrada del programa CFITG2D; donde cada punto representa una línea de entrada del fichero.

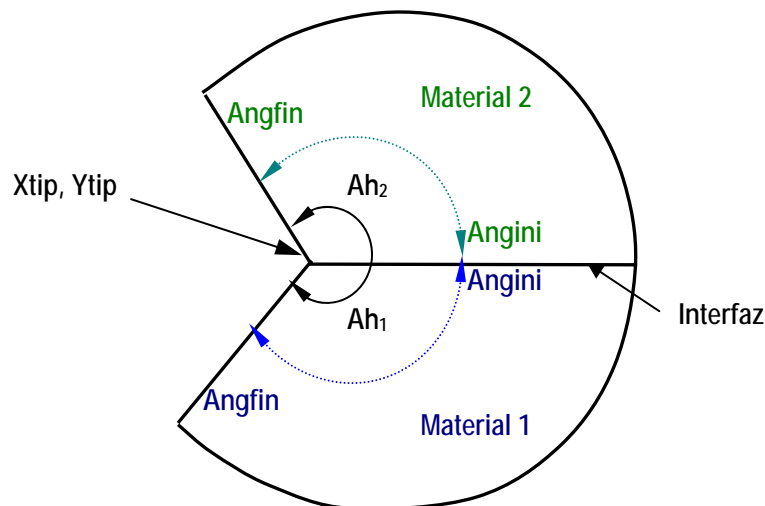


Figura 4.1 Esquina bimaterial donde se muestran los ángulos que se ingresan en el archivo de datos.

NAME: Nombre del fichero

### **Coordenadas del centro de la esquina**

Xtip, Ytip, NSol

Xtip: Coordenada X

Ytip: Coordenada Y

NSol: Numero de sólidos

### **Ángulos de la esquina**

Ah1, Ah2

Ah1: Angulo de inicio de la esquina.

Ah2: Angulo final de la esquina.

### **Datos de los anillos**

Nring, Rini, DistR

Nring: Numero de anillos

Rini: Radio

DistR: Distancia entre radios.

### **Tipo de Caso**

Solucaso:

(1) Homogéneo (Funciones de Seweryn [1994])

(2) Bimaterial (Funciones de Qian y Akisanya [1999])

### **Tipo de Problema**

Tprob:

(1) Lectura de las Tensiones y Desplazamientos de BEM

(2) Para evaluar la ortogonalidad de las funciones.

### **Tipos de Material en cada Sólido**

*TMAT(i)*

(1) Isótropo

### **Datos para cada sólido:**

*Angini, Angfin*

Angini: Angulo inicial del sólido

Angfin: Angulo final del sólido  
*Nseg, Npgauss*  
Nseg: Numero de Segmentos  
Npgauss: Numero de Puntos de Gauss

*EXX, NU*

Exx: Modulo de elasticidad  
Nu: Coeficiente de Poison

#### **Datos de los Autovalores:**

Nla: Numero de autovalores  
La(l): Listados de autovalores

### **4.4. MODELO NUMERICO**

Para las simulaciones numéricas realizadas en esta Tesis, se ha empleado el Método de los Elementos de Contorno, París y Cañas (1997), implementado en un código desarrollado por Graciani (2006). Dicho código se ha utilizado exclusivamente a nivel de usuario, sin que haya surgido la necesidad de desarrollar nuevas subrutinas y por tanto modificar el código original. La intención de este apartado es simplemente mostrar las capacidades de la herramienta y su aplicación a los problemas bajo estudio.

#### **4.4.1. Las características principales del programa:**

**Sólidos:** El programa admite la presencia de múltiples sólidos simultáneamente. Se admiten leyes de comportamiento lineales elásticas isótropas, ortótropas y anisótropas.

**Elementos:** El programa implementa exclusivamente elementos lineales continuos, siendo, por tanto la aproximación de la geometría y de todas las variables del problema lineal.

**Contornos:** El programa admite la definición de distintos tipos de contorno en tres categorías:

- **Contorno sin posibilidad de contacto:** Son zonas del contorno cuyos nodos interiores no tienen posibilidad de entrar en contacto con otros contornos. Utilizaremos este tipo de contorno en todos los contornos que no sean de la zona de pegado.
- **Contorno de interfaz:** Zona del contorno común entre dos sólidos, sin posibilidad de separación ni deslizamiento. Esta será la condición en los contornos comunes en las zonas de pegado, ya que asumimos pegado perfecto entre los materiales.
- **Contorno de contacto:** Zona del contorno que tiene posibilidad de contactar con el contorno de otro sólido, sin restricciones para separarse o deslizarse, salvo las impuestas por las condiciones de contacto con fricción entre dichos contornos.

Puntos de colocación: Además de los puntos del contorno discretizado, donde se sitúan puntos de colocación (y se imponen las ecuaciones de colocación) el programa está preparado para poner puntos de colocación adicionales en el interior de los elementos que contienen algún nodo con sobreabundancia de incógnitas.

Puntos internos: El programa, a través de la solución obtenida en los puntos del contorno, permite la evaluación de los desplazamientos y tensiones en puntos internos del sólido, sin más que proporcionar sus coordenadas en el fichero de entrada. La salida de resultados de estos puntos, se presentan además de en el fichero de resultados principal, y por comodidad para el usuario, en un fichero de resultados aparte (siendo este fichero el utilizado como dato de entrada del código de CFIT2D).

#### **4.4.2. Fichero de entrada del programa CBEMD:**

De manera muy esquemática vamos a presentar la estructura del fichero de entrada del programa, a través de los cuales podemos también entrever las posibilidades del código (cada punto representa una línea del fichero).

##### **Título del problema**

TP TF NSOL TI NPG

- TP: Tipo de problema (1=elástico axisimétrico, sólo para isótropos, 31=tensión plana generalizada, 32=deformación plana).
- Dado que la solución asintótica se ha obtenido bajo deformación plana generalizada, debemos utilizar 32.
- TF: Tipo de formulación (11=solución directa, 12=algoritmo incremental). En nuestro caso, sin zonas de contacto, siempre 11.
- NSOL: Número de sólidos en el problema.
- TI: Tipo de integración (2= integración analítica).
- NPG: Número de puntos de Gauss (si los hay).

##### **Datos para cada sólido:**

n-SOL NNOD NCNT NPINT NAPO

- n-SOL: Número del sólido.
- NNOD: Número de nodos del sólido.
- NCNT: Número de contornos.
- NPINT: Número de puntos internos.
- NAPO: Número de apoyo para impedir el movimiento como sólido rígido.

##### **Datos de los apoyos:**

ANOD DIR (1)...DIR(n) TMSR MSR(1)...MSR(n)

- ANOD: Número del nodo donde se aplican las condiciones de apoyo como sólido rígido.
- DIR (i): Dirección del desplazamiento impedido.
- TMSR: Tipo de movimiento como sólido rígido que se impide (1=desplazamiento, 2=giro).



- MSR (i): Dirección del desplazamiento impedido (para TMSR(i)=1) o centro de rotación (para TMSR(i)=2).

**Datos para cada Material:**

TM

- TM: Tipo de material (1=isótropo, 2=ortótropo, 3=ortótropo fuera de ejes, 4=anisótropo con constantes ingenieriles, 5=anisótropos con constantes  $\beta_{ij}$ ).

PRP(1)...PRP(n)

- Constantes elásticas del material, siendo su número función del tipo de material TM.

**Datos para cada nodo del sólido se introduce:**

n-NOD X Y

- n-NOD: número del nodo
- X: Coordenada x.
- Y: Coordenada y.

**Para cada contorno del sólido se introduce:**

n-CNT TC NELT

- n-CNT: Número del contorno.
- TC: Tipo de contorno (1=sin contacto, 2=interfase, 3=contacto).
- NELT: Número de elemento en el contorno.

**Para cada elemento del sólido se introduce:**

n-ELT: TE NG(ini) NG(final) NI(ini) NI(final)

- n-ELT: Número del elemento.
- TE: Tipo de elemento (sólo tipo 1=lineal continuo).
- NG (i): Nodos geométricos.
- NI (i): Nodos de integración.

**Condiciones de contorno en cada sólido:**

n-NOD CdC(1) CdC(1) CdC(1) V(1)...V(n)

- n-NOD: Número del nodo donde se impone la CC.

- CdC(1): Condición elástica. Consta de un par de dígitos que fijan respectivamente el valor de las variables, antes (1er dígito) y después (2º dígito) del nodo, sobre las que se imponen las condiciones de contorno (1=las dos componentes del vector tensión, 2=las dos componentes del vector de desplazamientos, 3= el desplazamiento normal y la componente tangencial del vector tensión, 4= la componente normal del vector tensión y el desplazamiento tangente a la cara).
- CdC(2): Condición de continuidad y condiciones especiales (0=normal, 1=impone continuidad del vector tensión, para casos CdC(1)=22, 33 ó 44, 2=condiciones “especiales” de contacto).
- CdC(3): Código de contacto (0=no hay contacto, 1=nodo libre (no entrará nunca en contacto), 2=nodo en adhesión,  $\pm 3$ =nodo en deslizamiento).
- V(i)=Valores de la condición de contorno, en nuestro caso siempre son 4 y el orden es siempre:  $u_{na}$ ,  $u_{ta}$ ,  $u_{nd}$ ,  $u_{td}$ ,  $t_{na}$ ,  $t_{ta}$ ,  $t_{nd}$ ,  $t_{td}$ . Donde u y t son los desplazamientos y el vector tensión y los subíndices n y t son. Normal y tangencial, y los subíndices a y d son antes y después del nodo.

**Datos para cada punto interno del sólido se introduce:**

n-PINT TPI X Y

- n-PINT: número del punto interno.
- TPI: Tipo de punto interno (siempre 1)
- X: Coordenada x.
- Y: Coordenada y.

Después de la información de todos los sólidos, se introduce la información de las zonas de contacto entre sólidos:

n-Z: TZ A B MU

- n-Z: número de la zona de contacto.
- TZ: Tipo de la zona de contacto (11=Interfase, 2=contacto).
- A: Contorno 1 en contacto.
- B: Contorno 2 en contacto.
- MU: Coeficiente de fricción (si procede).

# Capítulo 5

## ***Ejemplos Numéricos de Calculo del FIT en Esquinas Mediante el Método de los Elementos de Contorno y la Integral H***

### **5.1. VERIFICACIÓN DEL CÓDIGO**

En este apartado se evaluará la exactitud que tiene el código para hallar los FITGS, para lo cual se ha creado una subrutina en la que se verificara la ortogonalidad de las funciones. Para su evaluación se reemplazó la solución del campo de tensiones y desplazamientos hallados mediante el Método de los Elementos de Contorno (campo real) por una solución que consiste en la suma de las tensiones y desplazamientos asociados a los autovalores que están asociadas a la esquina.

El problema que se utilizo para verificar el código es el de una esquina con dos materiales de comportamiento isótropo elástico lineal en deformación plana que se muestra en la Figura 5.1, libres de tensión en el borde izquierdo, con apoyos de bolas en las caras inferior y derecha y con una tracción uniforme en la cara superior, con los siguientes parámetros de Dundurs del conjunto bimaterial:  $\alpha=0.8$  y  $\beta=0.2$ .

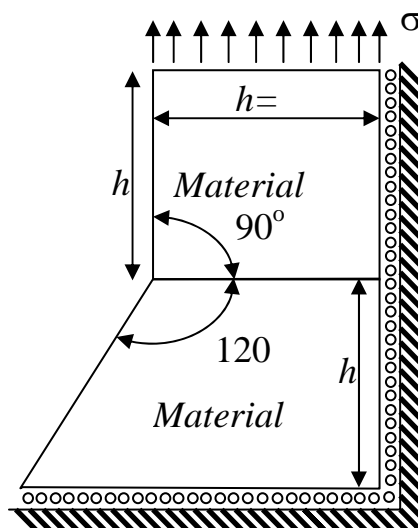


Figura 5.1 La esquina bimaterial isótropa (Qian y Akisanya, 1999)

El modelo de elementos de contorno cuyas características principales son: tamaño de los elementos alejados del vértice:  $0.025h$ , tamaño en el vértice:  $10^{-8}h$  con elementos lineales y un factor de crecimiento de 1.5 en las aristas que confluyen al vértice. Se ha tomado  $\sigma=1$ ,  $h=1$  y  $r =0.0217h$  (radio). El sólido situado en la parte superior de la figura tiene un total de 212 elementos, mientras que el sólido situado en la parte inferior tiene un total de 241 elementos. Los valores constantes a hallar los son  $K_1=2$ ,  $K_2=1$  y  $K_3=5$ .

Se utilizo para la integración en un primer momento la Regla de Simpson debido a su facilidad para ser implementada, obteniendo buenos resultados para esquinas homogéneas pero para esquinas bimateriales nos encontramos con la necesidad de poner una gran cantidad de puntos internos para poder obtener un resultado satisfactorio. Para la colocación de los puntos donde se integro se tuvo en cuenta que estén a distancias semejantes en ambos materiales. A continuación se presenta figuras donde se muestra la verificación de la Ortogonalidad del programa.

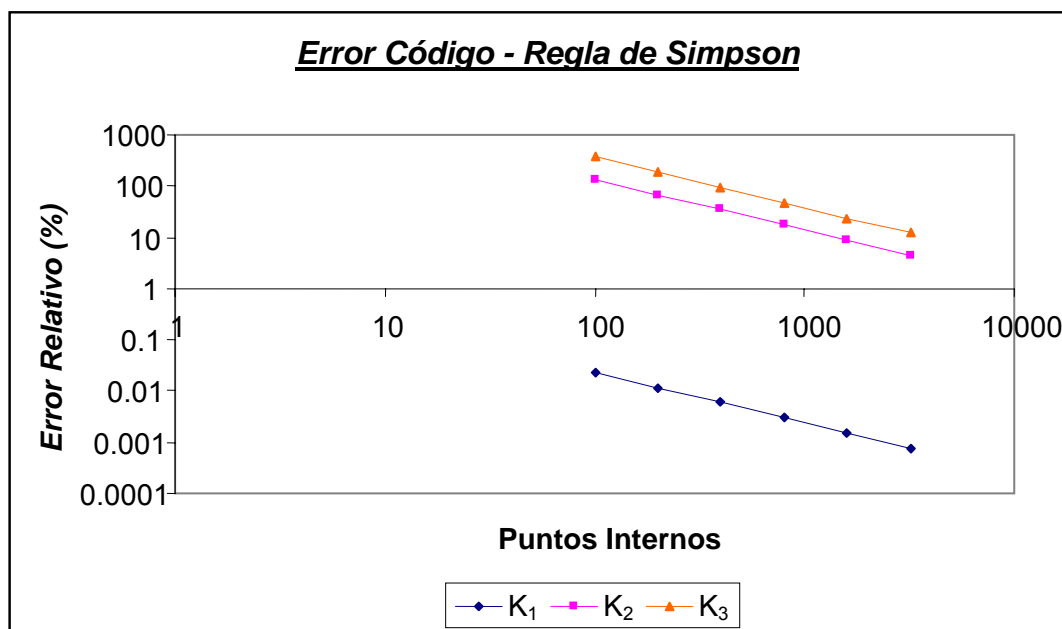


Figura 5.2: Error del Código utilizando la Regla de Simpson para el procedimiento de integración. Donde se ha elegido los siguientes valores de FITG.  $K_1=2$ ,  $K_2=1$ ,  $K_3=5$

En la Figura 5.2 se muestra que para obtener el FIT Modo I es necesario tan solo 100 puntos internos para obtener un porcentaje menor al 1%; y en cambio para obtener el modo 2 y 3 es necesario más de 1000 puntos internos para que el procedimiento de integración tenga un error aproximado al 5%.

Al utilizar como procedimiento de integración de la Cuadratura de Gauss el código alcanzó una buena exactitud siendo necesarios tan solo 50 puntos de Gauss. Los materiales fueron divididos en segmentos de tal modo que los puntos de integración se encontraran a distancias semejantes en ambos materiales, y cada segmento tenía un máximo de 20 puntos de Gauss.

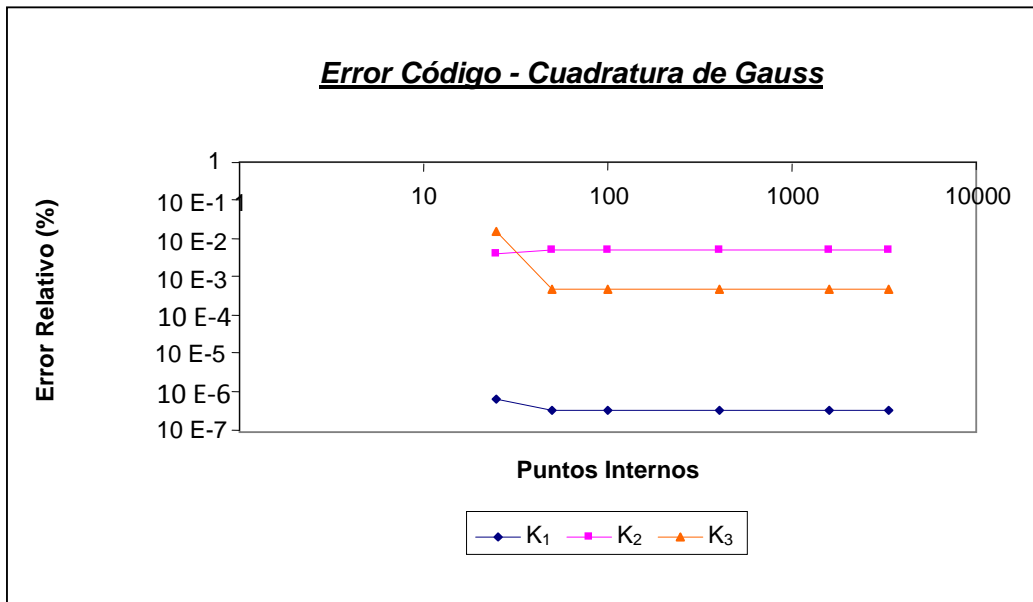


Figura 5.3: Error del Código utilizando la Cuadratura de Gauss para el procedimiento de integración. Donde se ha elegido los siguientes valores de FITG se  $K_1=2$ ,  $K_2=1$ ,  $K_3=5$

Al utilizar como procedimiento de integración de la Cuadratura de Gauss el código alcanzó una buena exactitud siendo necesarios tan solo 50 puntos de Gauss para obtener el FIT Modo I un error menor al  $1 \times 10^{-6}\%$  y para el Modo II y III un error menor al 0.1%

## 5.2. APLICACIONES DEL MÉTODO DESARROLLADO EN EL MÉTODO DE LOS ELEMENTOS DE CONTORNO

### 5.2.1. Esquina Homogénea de 270° Sometida a Tracción

Se ha analizado el problema de una esquina re-entrante de un material homogéneo a tracción uniforme con una entalla lateral de 90°. El material de la placa es homogéneo y con comportamiento isótropo, elástico y lineal. Este es un problema clásico que dispone de una solución de referencia bastante fiable y con un grado de precisión muy elevado. La analizaremos utilizando la solución de Seweryn [1994].

La geometría modelada se ha tomado con  $w=10$  (por tanto:  $a=5$  y  $h=20$ ). El modelo de elementos de contorno tiene un total de 709 nodos y elementos lineales con un tamaño de 0.1 en las caras horizontales y verticales. En las caras inclinadas se realiza un refinado progresivo (utilizando un factor de 1.5) hacia la esquina en la cual el último elemento tiene un tamaño de  $10^{-8}$  teniendo dichas caras un total de 100 nodos.

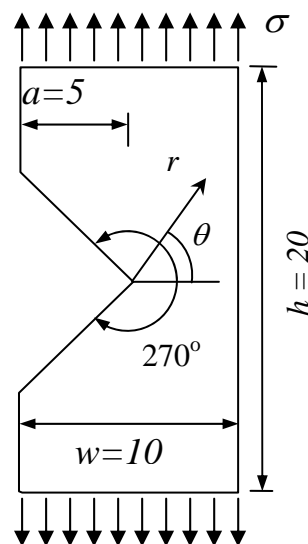


Figura 5.4 El problema de tracción con entalla de Helsing y Jonsson, 2002)

El estado de cargas exteriores corresponde a una tensión uniforme de tracción en las dos caras horizontales que genera en el entorno del vértice de la entalla un Modo I de apertura. Este problema ha sido analizado en detalle en la

literatura. Concretamente Helsing y Jonsson [2002], además de recopilar los resultados que sobre esta geometría particular existen en publicaciones precedentes, presentan sus propios resultados con una precisión de 8 cifras significativas. El valor tomado como referencia para este problema es  $K_I=154.948529$

En este problema se realiza una aproximación de los FITGs mediante ajustes que involucren caminos de integración muy alejados del entorno de la esquina. Existiendo errores, ya no asociados a la discretización, sino a la propia validez de la representación asintótica lejos del vértice de la esquina. Se trata, en primer lugar, de ver si podemos encontrar una zona lo suficientemente alejada del vértice de la entalla, para evitar los errores asociados a la discretización, pero sin alejarnos tanto que perdamos la representatividad de la solución asintótica.

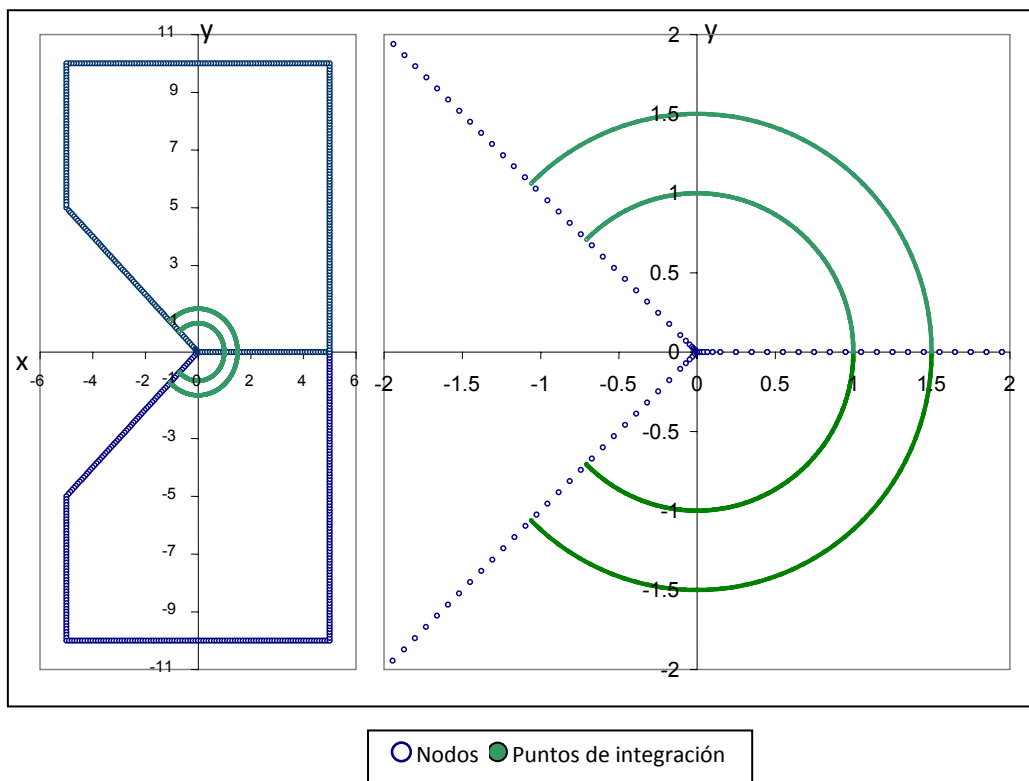


Figura 5.5: Malla del modelo de la esquina y detalle de los puntos internos colocados a  $r/a = 0.2$  y  $0.3$ .

Se han probado diversas mallas en las que se ha hecho variar el tamaño del elemento del vértice, manteniendo el resto de parámetros del modelo

invariantes (tamaño máximo de elementos lejanos = 0.1, factor de crecimiento = 1.5), entre  $10^{-5}$  y  $10^{-14}$  verificándose, a modo de análisis de convergencia, que los resultados son estables y que basta, a efectos prácticos con la discretización realizada. Por pequeño que sea el elemento del vértice, se deben evitar, por errores numéricos asociados a la discretización los elementos hasta una distancia dos órdenes de magnitud superiores al tamaño del elemento del vértice.

Para hallar el FITG de se ha utilizado la Regla de Simpson y la Cuadratura de Gauss para realizar una comparación. Para la regla de Simpson se utilizo una cantidad de 1000 puntos internos por camino de integración utilizándose radios iguales a 0.5, 1, 1.5, y 2 es decir una relación de  $r/a = 0.1, 0.2, 0.3, 0.4$ .

En la tabla 5.1 y 5.2 se muestran los FITGs obtenidos con un error respecto al valor dado por Helsing de 0.14% para el método de integración de la regla de Simpson y de 0.17% para el método de integración de Gauss.

Tabla 5.1: FITG obtenidos para una esquina homogénea de  $270^\circ$  método de integración Simpson.

Radio	1	2	3	4
$K_1$	154.721129	154.725431	154.726527	154.728339

Mientras para la Cuadratura de Gauss se utilizo una cantidad de 800 puntos internos por camino de integración utilizándose radios iguales a 0.5, 1, 1.5 y 2 es decir una relación de  $r/a = 0.1, 0.2, 0.3, 0.4$ .

Tabla 5.2: FITG obtenidos para una esquina homogénea de  $270^\circ$  método de integración Cuadratura de Gauss.

Radio	1	2	3	4
$K_1$	154.683988	154.684646	154.684827	154.685473



## 5.2.2. La esquina bimaterial isótropa

El problema es el que se muestra en la Figura 5.1, con dos materiales de comportamiento isótropo libres de tensión en el borde izquierdo, con apoyos de bolas en las caras inferior y derecha y con una tracción uniforme en la cara superior. El problema particular resuelto por Qian y Akisanya [1999], se obtuvo con los siguientes parámetros de Dundurs del conjunto bimaterial:  $\alpha=0.8$  y  $\beta=0.2$ , lo que hace que el material de la parte superior sea unas 10 veces más rígido que el situado en la parte inferior. Las expresiones utilizadas por Qian y Akisanya [1999] para las tensiones y desplazamientos son:

$$\sigma_{ij}^m = \sum_{k=1}^N H_k r^{\lambda_{k-1}} f_{ijk}^m + \sigma_{ij0}^m \quad u_i^m = \sum_{k=1}^N H_k r^{\lambda_k} g_{ik}^m + u_{i0}^m \quad (5.1)$$

Donde:

$$H_k = \sigma h^{1-\lambda_k} a_k(\alpha, \beta, \lambda_k, \theta_1, \theta_2) \quad (5.2)$$

En (5.1) y (5.2)  $m$  es el material,  $H_k$  es el FITG asociado a  $\lambda_k$ ,  $a_k$  es el FITG adimensionalizado (que se iguala a  $H_k$  si  $h=1$  y  $\sigma=1$ ),  $f(\theta)$  y  $g(\theta)$  son funciones conocidas (cuyas expresiones explícitas ocupan dos páginas completas, ver Qian y Akisanya, [1999]),  $\theta_1$  y  $\theta_2$  los ángulos sólidos ( $90^\circ$  y  $120^\circ$  respectivamente) y  $\alpha$  y  $\beta$  son los parámetros de Dundurs.

$$\alpha = \frac{\mu_1(\kappa_2 + 1) - \mu_2(\kappa_1 + 1)}{\mu_1(\kappa_2 + 1) + \mu_2(\kappa_1 + 1)} \quad \beta = \frac{\mu_1(\kappa_2 - 1) - \mu_2(\kappa_1 - 1)}{\mu_1(\kappa_2 + 1) + \mu_2(\kappa_1 + 1)} \quad (5.3)$$

donde  $\mu_1, \mu_2$  son los módulos de cortadura de los materiales y  $\kappa_1, \kappa_2$  son las constantes de Kolosov. Siendo  $\kappa_i = 3 - 4\nu_i$  ( $i=1,2$ ) para deformación plana y  $\kappa_i = (3-\nu_i)/(1+\nu_i)$  ( $i=1,2$ ) para tensión plana generalizada, y  $\nu_i$  es el coeficiente de Poisson del material  $i$ .

La esquina bimaterial de la Fig. 5.2 sólo presenta un modo singular. Qian y Akisanya [1999] proporcionan los valores de los exponentes característicos de

dicho modo singular y de los dos modos, no singulares ( $\lambda_k > 1$ ), siendo los siguientes:  $\lambda_1=0.6747$ ,  $\lambda_2=1.1637$  y  $\lambda_3=1.5938$ .

Los resultados para los tres primeros exponentes característicos, evaluados mediante la herramienta desarrollada por Barroso en su Tesis(2007) varían muy ligeramente de los valores anteriores, siendo los valores de dichos exponentes obtenidos por el método analítico:  $\lambda_1=0.673473$  (0.18% de diferencia),  $\lambda_2=1.167477$  (0.32% de diferencia) y  $\lambda_3=1.589147$  (0.29% de diferencia), que se considera la mejor precisión.

Los valores de los FITGs obtenidos por Qian y Akisanya [1999], mediante una integral de contorno independiente del camino (finalmente transformada en una integral de dominio, mediante el teorema de Gauss), muestran efectivamente una razonable independencia del camino al ser evaluadas numéricamente dichas integrales en tres dominios a diferentes distancias del vértice y los valores medios (en los tres dominios analizados) de los coeficientes  $a_k$  expresados en forma adimensional se muestran en la Tabla 5.3.

Tabla 5.3: FITGS obtenidos por Qian y Akisanya

Radio	0.0053h-0.0063h	0.0217h-0.0255h	0.0869h-0.11h
$K_1$	0.6301	0.6301	0.6300
$K_2$	-0.3666	-0.3673	-0.3675
$K_3$	0.5407.	0.5430.	0.5491.

En la Tesis de Barroso (2007) se analizó también este problema utilizando un método de mínimos cuadrados en desplazamiento para hallar el FITG, utilizando las dos componentes de los desplazamientos,  $u_r$  y  $u_\theta$ , de las tres aristas que llegan al vértice ( $\theta=-120^\circ$ ,  $0^\circ$  y  $90^\circ$ ), los valores de  $a_1$ ,  $a_2$ , y  $a_3$  (el FITG adimensionalizado). Sabiendo que los desplazamientos son de la forma representada en la ecuación 5.4

$$u_i(r, \theta) \propto \sum_{k=1}^N \frac{K_k}{r^{1-\lambda_k}} u_{ik}(\theta) \quad (5.4)$$

siendo  $i$  la componente del desplazamiento, y  $N$  el numero de términos seleccionados del desarrollo asintótico. Al tener el conocimiento previo de los  $\lambda_k$  de la estructura anterior (esquina 90 - 120) se conocería todo excepto los FITGs,  $K_k$  por lo que de la función  $J_u$

$$J_u = (K_1, \dots, K_k) = \sum_{i=1}^A \sum_{j=1}^B \sum_{n=1}^C [u_i^{MEC}(r_n, \theta_j) - u_i^{ana}(r_n, \theta_j)]^2 \quad (5.5)$$

donde  $u_i^{MEC}$  son los desplazamientos del modelo numérico y  $u_i^{ana}$  son los desplazamientos obtenidos analíticamente, siendo  $i$  la componente del desplazamiento,  $j$  el numero de caras involucradas en la esquina y  $n$  el numero de nodos considerado en cada cara.

En la expresión  $J_u$  se consideran tres sumatorios: donde  $A$  representa el numero de componentes de desplazamientos,  $B$  define el numero de materiales usados en la evaluación del error y finalmente  $C$  define el numero de nodos usados a lo largo de la cara para la evaluación de  $J_u$ .

Para obtener los  $K_k$  se realiza un ajuste por mínimos cuadrados (se busca el minimo de  $J_u$ ) que resulta en el sistema de ecuaciones siguiente:

$$\frac{\partial J_u}{\partial K_k} = 0 \quad \text{para cada } k=1, \dots, N \quad (5.6)$$

De la ecuación anterior se obtienen los valores de  $K_k$  tales que la función de error  $J_u$  sea mínima.

Tabla 5.4: Valores de FITG para esquina bimaterial obtenidos por Barroso.

Rango	$5.818585 \times 10^{-6}h - 0.262099h$
$K_1$	0.6776
$K_2$	0.3192
$K_3$	0.1668

De los resultados presentados por Barroso [2007] se aprecia que los valores obtenidos para los términos no singulares son:  $K_2=0.3192$  y  $K_3=0.1668$ , que son muy diferentes (incluso con cambio de signo) de los obtenidos por Qian y Akisanya[1999].

Para evaluar los FITGs se ha generado un modelo de elementos de contorno cuyas características principales son: tamaño de los elementos alejados del vértice:  $0.025h$ , tamaño en el vértice:  $10^{-8}h$  con elementos lineales y un factor de crecimiento de 1.5 en las aristas que confluyen al vértice. Se ha tomado  $\sigma=1$  y  $h=1$  de forma que  $K_k$ . El sólido situado en la parte superior de la figura tiene un total de 212 elementos, mientras que el sólido situado en la parte inferior tiene un total de 241 elementos

En la Figura 5.5 se muestran los distintos caminos de integración tomados. Donde para hallar el FITG se utilizó para el sólido situado en la parte superior 72 segmentos con 20 puntos de Gauss por segmento dando un total de 1440 puntos de Gauss y para sólido inferior se utilizó 96 segmentos con 20 puntos de Gauss dando un total de 1920 puntos internos, lo cual nos da un total 3360 puntos para cada camino de integración, donde los radios utilizados fueron los mismos que los de Qian y Akisanya [1999].

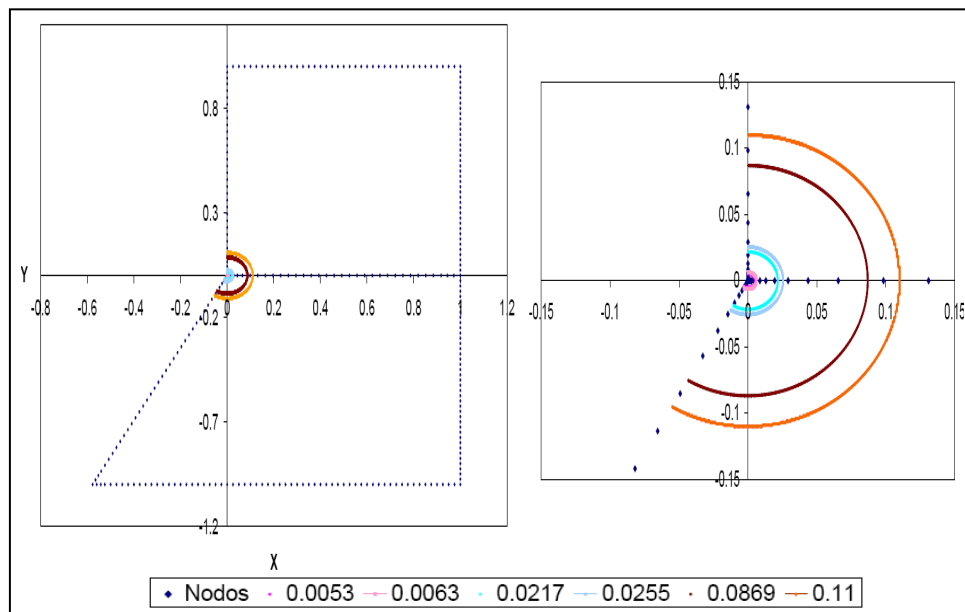


Figura 5.6: Malla del modelo de la esquina bimaterial y detalle de los puntos internos colocados a diferentes radios.

Los valores de FITG obtenidos son:

Tabla 5.5: Valores de FTIG obtenidos para la esquina bimaterial isótropa

Radio	0.0053h	0.0063h	0.0217h	0.0255h	0.0869h	0.11h
$K_1$	0.6784	0.6781	0.6788	0.6788	0.6795	0.6792
$K_2$	-1.4746	-1.5006	-1.5585	-1.5581	-1.5686	-1.5683
$K_3$	-0.0174	0.3026	0.7553	0.7433	0.7942	0.7999

De los resultados se aprecia que los valores obtenidos para el Modo I tiene un error de 0.17% respecto a los valores obtenidos por Barroso [2007] y de 7.73% respecto a los valores obtenidos por Qian y Akisanya [1999], para el Modo II y Modo III los errores son muy diferentes y altos lo cual demuestra la dificultad para hallar dichos modos.

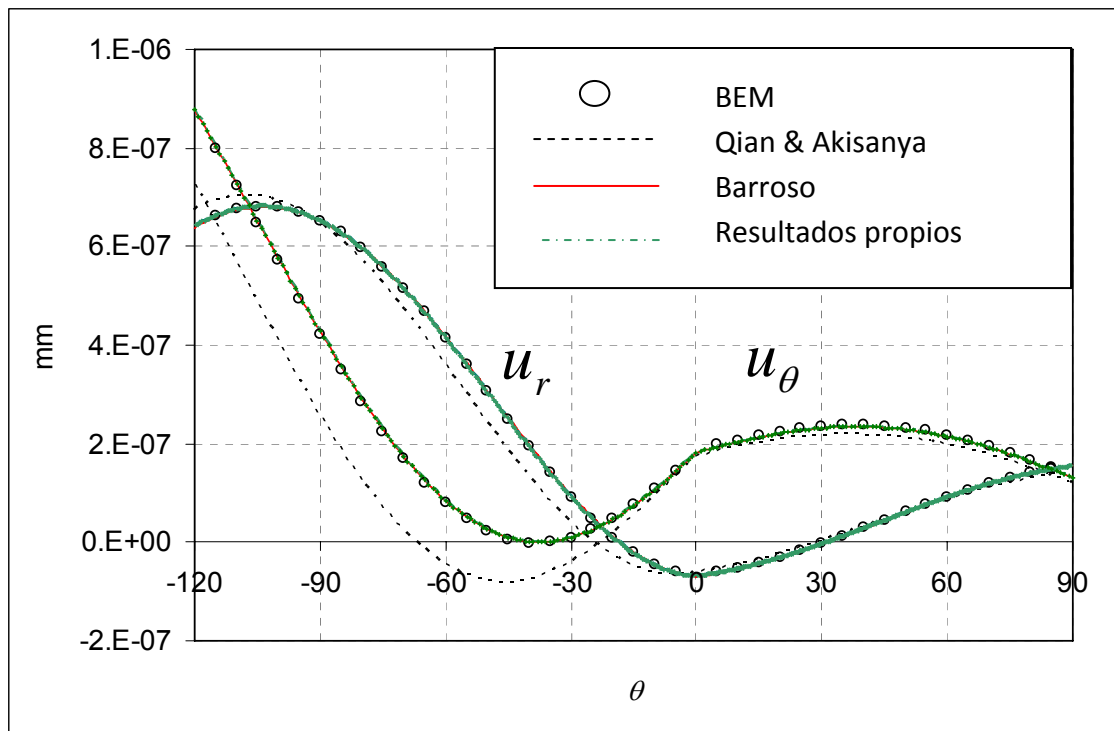


Figura 5.7 Comparación de los desplazamientos del problema de la esquina bimaterial.

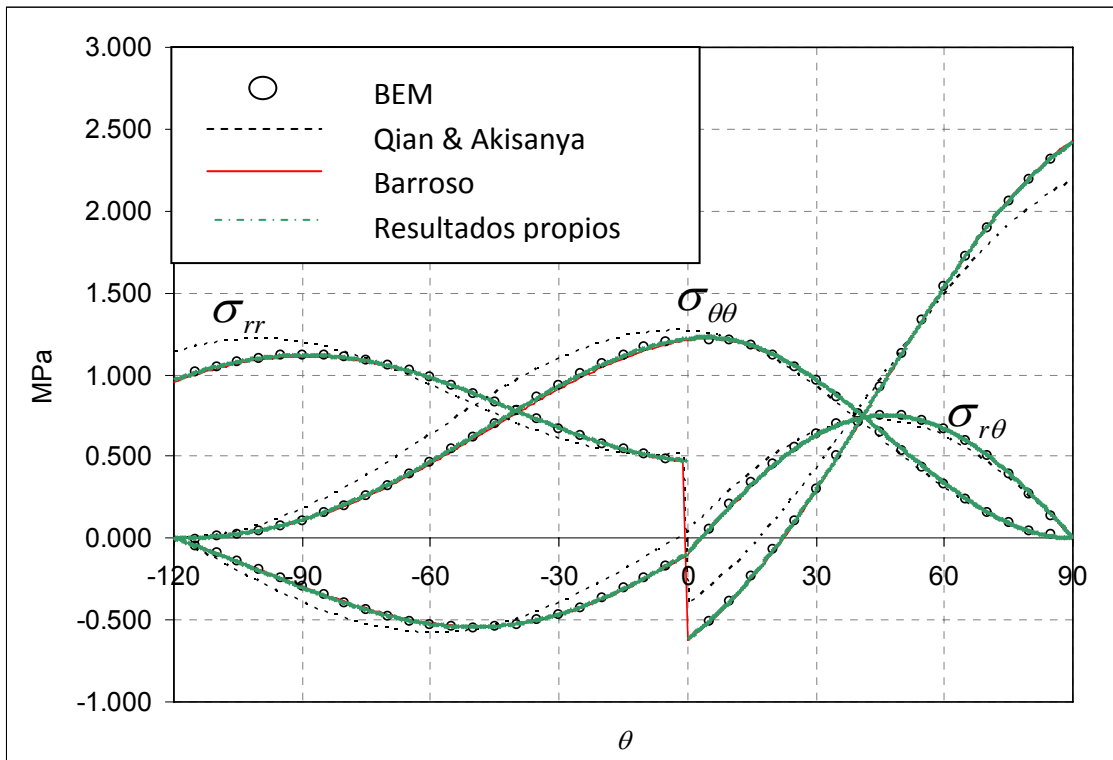


Figura 5.8 Comparación del estado tensional del problema de la esquina bimaterial.

En la figura 5.7 y 5.8 se muestra un excelente acuerdo entre los resultados del modelo numérico (*solución MEC*) que se indican con círculos y los resultados propios (*res. propios*) con línea continua, así como con los resultados presentados por Barroso [2007] para realizar las comparaciones se distribuyeron los puntos de integración en un  $r = 0.1h$ .

# Capítulo 6

## ***Conclusiones y desarrollos futuros***

En la presente tesis se ha puesto a punto un procedimiento para el cálculo del factor de intensidad de tensiones generalizado (FITG) que aparecen en esquinas bimateriales elásticas isótropas sometidos a un estado de tensión o deformación plana. El método consiste en la creación de puntos internos donde se evalúan las tensiones y los desplazamientos para un campo real (BEM) y un campo auxiliar que es escogido cuidadosamente como asociado a un término del desarrollo asintótico de la esquina. El campo auxiliar representa en realidad un término del desarrollo asintótico de la esquina con un exponente de singularidad negativa. Mediante la integral de línea de Bueckner se obtendrá los FITG para los diferentes términos asociados a la esquina.

En futura espera se espera desarrollar este método para materiales con comportamiento anisótropo para poder analizar esquinas en las que entran en juego los materiales compuestos y las uniones adhesivas; por lo cual se necesitara la aplicación del Formalismo de Stroh de elasticidad anisotropa.

# Capítulo 7

## **Bibliografía**

Banks-Sills L., A conservative integral for determining stress intensity factors of a bimaterial notch. *International Journal of Fracture*, pp: 115, 1-26, 2002.

Barroso A., Mantič V., and Paris. F., Singularity of anisotropic multimaterial corner. *International Journal of Fracture* 119, pp: 1-23, 2003.

Barroso A. Caracterización de Estados Singulares de Tensión en Esquinas Multimateriales Aplicación a Uniones Adhesivas con Materiales Compuestos, Tesis Doctoral, Universidad de Sevilla 2007.

Bogy D. B., Two edge-bonded elastic wedges of different materials and wedge angles under surface tractions, *Journal of Applied Mechanics* 38, pp: 377-386. 1971.

Brebbia C. and Dominguez J., *Boundary Elements – An Introductory Course*, MacGraw-Hill, 1989.

Bueckner H. F., *Mechanics of Fracture: Methods of Analysis and Solution of Crack Problems*, G. C. Sih (Ed.), Noordhoff, Leyden, pp. 239–314, 1973.

Chan S. K., Tuba I. S. and Wilson W. K., On the finite element method in linear fracture mechanics. *Engineering Fracture Mechanics* 2(1), pp: 1-17, 1970.

Chen Y. H. and Lu T. J., On the path dependence of the J-integral in notch problems. *International Journal of Solids and Structures* 41, pp: 607–618, 2004.

Choi N. Y. and Earmme Y. Y. “Evaluation of stress intensity factors in a circular arc-shaped interfacial crack using L-integral”, *Mechanics of Materials* 14, pp: 141-153, 1992.



Dempsey J. P. and Sinclair G. B., On the stress singularities in the plane elasticity of the composite wedge, *Journal of Elasticity* 9, No. 4, pp: 373-391, 1979.

Dempsey J. P. and Sinclair G. B., On the singular behaviour at the vertex of a bi-material wedge, *Journal of Elasticity* 11, No. 3, pp: 317-327, 1981.

Dunn M. L., Hui C. Y., Labossiere P. E. W. and Lin Y. Y., Small scale geometric and material features at geometric discontinuities and their role in fracture analysis, *International Journal of Fracture* 110, pp: 101-121, 2001.

Graciano E. Formulación e implementación del método de los elementos de contorno para problemas axisimétricos con contacto. Aplicación a la caracterización de la interfase fibra matriz en materiales compuestos, Tesis doctoral, Universidad de Sevilla 2006.

Helsing J. and Jonsson A., On the accuracy of benchmark tables and graphical results in the applied mechanics literature, *ASME Journal of Applied Mechanics* 69, pp: 88-90, 2002.

Huber, O., Nickel, J. and Kuhn, G. On the decomposition of the J integral for 3D crack problems, *International Journal of Fracture* 64, pp: 339-348, 1993.

Im S. and Kim S. K., An application of two-state M-integral for computing the intensity of the singular near-tip field for a generic wedge, *Journal of the Mechanics and Physics of Solids* 48, pp: 129-151, 2000.

Irwin G. Analysis of stresses and strains near the end of a crack traversing a plate, *Journal of Applied Mechanics*, ASME, 24, 361-364, 1957.

Knowles J. K. and Sternberg E., On a class of conservation laws in a linearized and finite elastostatics", *Archive for Rational Mechanics and Analysis* 44, pp: 187-211, 1978.

Labossiere P.E.W. and Dunn M.L., Fracture initiation at three-dimensional bimaterial interface corners, *Journal of the Mechanics and Physics of Solids* 49, pp: 609-634, 2001.

Lee Y. and Im S., On the computation of the near-tip stress intensities for three-dimensional wedges via two-state M-integral", *Journal of the Mechanics and Physics of Solids* 51, pp: 825-850, 2003.

París F. and Cañas J., Boundary Element Method: Fundamentals and Applications, Oxford University Press, 1997.

Qian Z. Q. and Akisanya A. R., An experimental investigation of failure initiation in bonded joints. *Acta Material* 46, 14, pp: 4895-4904, 1998.

Qian Z. Q. and Akisanya A. R., Wedge corner stress behaviour of bounded dissimilar materials. *Theoretical and Applied Fracture Mechanics* 32, pp: 209-222, 1999.

Rice J. R., A path independent integral and the approximate analysis of strain concentration by notches and cracks, *ASME Journal of Applied Mechanics* 35, pp: 379-386, 1968.

Seweryn A., Brittle fracture criterion for structures with sharp notches. *Engineering Fracture Mechanics* 47, pp: 673-681, 1994.

Sinclair G.B., Okajima M. and Griffin J.H., Path independent integrals for computing stress intensity factors at sharp notches in elastic plates. *International Journal for Numerical Methods in Engineering* 20, pp: 999-1008. 1984.

Szabó, B.A. and Babuška, I. *Finite Element Analysis*, John Wiley & Sons, New York 1991.

Vasilopoulos D., On the determination of higher order terms of singular elastic stress fields near corners, *Numerische Mathematik* 53, pp: 51-95, 1988.

Williams M. L., Stress singularities resulting from boundary conditions in angular corners of plates in extension. *Journal Applied Mechanics* 19, pp: 526-528, 1952.

Williams M. L., On the stress distribution at the base of a stationary crack. *Journal of Applied Mechanics* 24, pp:109-114, 1957.

Yosibash Z., Bussiba A. and Gilad I., Failure criteria for brittle elastic materials, *International Journal of Fracture* 125, pp: 307-333, 2004.

# APENDICE

## APENDICE A1: CODIGO DEL PROGRAMA

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
PROGRAM MENU
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Menu del programa
    use msimsl
    implicit none
    include './_paramh.for'
    include './_sgcommon.for'
    include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    record /s_material/ mat(n_max_sol)
    record /s_solido/ sol(n_max_sol)
    record /s_anillo/ ring(n_max_ring)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    integer tip,opth
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C  Paso1: Lectura y Generación de Coordenadas de los Puntos

C          Lectura de los Datos Para Generar los Puntos
          call Lectura(mat,sol)
C          Generación De Los Puntos
          call Gpint(sol,ring)
C          Escritura De Los Puntos
          call Escritura_Pint(sol)

C  Paso2: Realizando La Integral

C  Se halla la normal de los puntos
          call normal(sol)

C  Se hace un cambio de variable solo por comodidad, donde solucaso es
C  1: para materiales homogeneos (subroutine nseweryn) y
C  2: para bimateriales (subroutine Qian)

          tip=solucaso

C  La variable opth nos selecciona los campos a integrar.
C  1: Integra el campo actual y auxiliar
C  2: Integra BEM con el campo auxiliar*Q

          do opth=1,2
            if(opth.eq.1) then
              if(tip.eq.1)then
C                Un Solo Material Homogéneo
                call Actual_Nseweryn(mat,sol)
                call Auxiliar_Nseweryn(mat,sol,ring,opth)

              elseif(tip.eq.2)then
C                Bimaterial Isótopos
                call Actual_Qian(mat,sol)
                call Auxiliar_Qian(mat,sol,ring,opth)

            else
```

```

        write(*,*) " CASO DE SOLUCION NO EXISTE"
        stop
    endif
else
C      Lectura De Las Tensiones Y Desplazamientos De Los Puntos.
      call Lectura_Pint_BEM(sol)

C      Verificación Del Código Mediante La Ortogonalidad
C      call Qian_Sim_Bem(mat,sol)

C      Campos auxiliares multiplicado por Q
      if(tip.eq.1)then
          call Auxiliar_Nseweryn(mat,sol,ring,opth)
      elseif(tip.eq.2)then
          call Auxiliar_Qian(mat,sol,ring,opth)
      else
          write(*,*) "MALA SOLUCION"
          stop
      endif
    endif

C      Ordena los datos para su integracion.
      call Datordersol(sol,ring,opth)

    enddo

C      Paso3: Escritura De Los Archivos De Salida

C      Escritura en un archivo de las coordenadas del punto, radio, angulo;y
C      escritura de las tensiones y desplazamientos del campo auxiliar
      call Escritura_Aux(sol)

c      Escritura en un archivo de las tensiones y desplazamientos del campo actual
c      y de BEM.
      call Escritura_Act(sol)

c      Escritura del resultado de la integración del campo actual y el auxiliar y
c      escritura Factor de intensidad Generalizado
      call Escritura_Fitg(ring)

end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE LECTURA(MAT,SOL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c Lectura de datos para generar los puntos internos
  implicit none
  include './_paramh.for'
  include './_sgcommon.for'
  include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  record /s_material/ mat(n_max_sol)
  record /s_solido/ sol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  character*30 comentario, name1
  integer i,,j,nlam
  real*8 ah1,ah2
  real*8 wi,wd,exx,nu
  real*8 angini,angfin,nseg, npgauss
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  pi=4*atan(1.0)
  degree=pi/180.          !angulos a radianes
  write(*,*)
  write(*,*) "Nombre del archivo:   "
  read(*,*) name1
  open(f_gp,defaultfile='c:\cfitg2d\datos',file=name1,status='old')

C   Verifica el archivo de entrada
  read(f_gp,*) name
  write(*,*)
  write(*,*) "nombre del problema:   ",name
  write(*,*)
  read(f_gp,*) comentario
  read(f_gp,*) comentario
C   Origen De La Esquina Y Numero De Solidos
  read(f_gp,*) xtip,ytip,nsol
C   Ángulos de la esquina
  read(f_gp,*) comentario
C   Inicio Esquina(-), Final Esquina(+)
  read(f_gp,*) comentario
  read(f_gp,*) ah1,ah2
C   Radio Inicial, Distancia entre radios, Numero de caminos de integración
  read(f_gp,*) comentario
  read(f_gp,*) Rini,distR,nring
C   Tipo de Caso
  read(f_gp,*) comentario
  read(f_gp,*) solucaso
C   Tipo de Problema
  read(f_gp,*) comentario
  read(f_gp,*) tipbem
C   Tipo de Material en cada Sólido
  read(f_gp,*) comentario
  read(f_gp,*) (tmat(i),i=1,nsol)

C   Asigna un comentario al MAT(I)
  do i=1,nsol
    if(tmat(i).eq.1)then
      mat(i).tipo=' Isótropo '
    elseif(tmat(i).eq.2)then
      mat(i).tipo=' Ortótropo '

```

```

elseif(tmat(i).eq.3)then
mat(i).tipo=' Transversalmente Isótropo '
else
print *, "ERROR EN EL TIPO DE MATERIAL"
endif
enddo

C   Ángulos para cada (i) sólido y creación de matrices
read(f_gp,*)comentario
do i=1,nsol
  if(tmat(i).eq.1)then
C     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
C     Material Isótropo Caso Seweryn O Qian
C     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    if(Solucaso.Eq.1.Or.Solucaso.Eq.2)Then
C     Angulo Inicial, Angulo Final
      read(f_gp,*) comentario
      read(f_gp,*)angini,angfin
      sol(i).angini=angini
      sol(i).angfin=angfin
C     Numero de segmentos, numero de puntos de gauss
      read(f_gp,*) comentario
      read(f_gp,*) nseg, npgauss
      sol(i).seg=nseg
      sol(i).pgaus=npgauss
C     Propiedades E y Nu
      read(f_gp,*) comentario
      read(f_gp,*)exx,nu
      mat(i).prp(1)=exx
      mat(i).prp(2)=nu

      else
        write(*,*) "CASO DE SOLUCION DESCONOCIDO"
        stop
      endif
    else
      write(*,*) "MATERIAL",i," TIPO DESCONOCIDO"
      stop
    endif
  endif
enddo

C   Número de autovalores
read(f_gp,*) comentario
read(f_gp,*) nlam
C   Autovalores
read(f_gp,*) comentario
  do j=1,nlam
    read (f_gp,*) lam(j)
  end do
C   Ángulos alfas (sol1 y sol2) los utiliza Qian
alpha1=(-ah1*degree)
alpha2=(ah2*degree)
close(f_gp)
20  format(a50)
11  format(i2,2x,a20)
12  format(a,': ')
13  format(d22.16)
return
end

```



```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE GPINT(SOL,RING)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c Crea puntos de gauss alrededor de una esquina
  implicit none
  include './_paramh.for'
  include './_sgcommon.for'
  include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  record /s_solido/ sol(n_max_sol)
  record /s_anillo/ ring(n_max_ring)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  integer tiph,i,pan,csol,nseg,mseg, npg,seg,pd,pint,p
  real*8 radio,angfin,angini,tseg,theta, fpx, fpy
  real*8 x1,x2,x(n_max_pgauss),w(n_max_pgauss)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  tiph=1; i=1           ! tipo de punto interno = 1; contador del numero de anillos
  radio=Rini
  do while (i.le.nring)           !anillo
    ring(i).num=i                 !# del anillo
    ring(i).nsring=nsol           !# de sólidos por anillo
C   Distancia Entre Anillos
    if (i.ge.2) then
      radio=radio+distr
    end if
    csol=1
      do while (csol.le.nsol) !Para cada sólido
        sol(csol).num=csol
C   Ángulos final e inicial del sólido (radianes)
        angfin=sol(csol).angfin*degree
        angini=sol(csol).angini*degree
C   Generación De Los Segmentos Del Sólido
C   Número de segmentos
        nseg=sol(csol).seg
C   Tamaño del segmento por sólido
        sol(csol).tseg=abs(angfin-angini)/nseg
        tseg=sol(csol).tseg
C   Ángulos de división de los segmentos
        if (i.eq.1) then
          pd=1
        else
          pd=nseg*(i-1)+1
        endif
C   Coordenadas Polares:
C   Primer punto del sólido (rini, theta)
        theta=angini
        fpx=rini*cos(theta)+xtip
        fpy=rini*sin(theta)+ytip
C   Puntos que forman los segmentos
        do while (pd.lt.nseg)
          theta=theta+tseg
          sol(csol).pdiv(pd).angulo=theta
          sol(csol).pdiv(pd).num=pd
          pd=pd+1
        enddo
      csol=csol+1

```

```

        enddo
    i=i+1
    enddo
C   Generando Los Puntos Internos
    radio=rini
    i=1
    do while (i.le.nring)          !anillo
C       distancia entre anillos
        if (i.ge.2) then
            radio=radio+distr
        end if
    csol=1
    do while (csol.le.nsol) !solido
        seg=1;pd=1
        nseg=sol(csol).seg
        npg=sol(csol).pгаus
        sol(csol).upint=nseg*npg*nring
        if (i.eq.1) then
            pint=1
        else
            pint=nseg*npg*(i-1)+1
        endif
        do while (seg.le.nseg) !segmento
            if (nseg.eq.1) then
                x1=sol(csol).angini*degree
                x2=sol(csol).angfin*degree
            else
                if (seg.eq.1) then
                    x1=sol(csol).angini*degree
                    x2=sol(csol).pdiv(pd).angulo
                elseif (seg.eq.nseg) then
                    x1=sol(csol).pdiv(pd-1).angulo
                    x2=sol(csol).angfin*degree
                else
                    x1=sol(csol).pdiv(pd-1).angulo
                    x2=sol(csol).pdiv(pd).angulo
                endif
            endif
        endif
        call gauleg(x1,x2,x,w,npg)
        Creando los Puntos de Gauss
        do p=1,npg
            ring(i).npring=pan
            sol(csol).pnt(pint).tiph=tiph
            sol(csol).pnt(pint).peso=w(p)
            sol(csol).pnt(pint).angulo=x(p);theta=x(p)
            sol(csol).pnt(pint).radio=radio
            sol(csol).pnt(pint).xyz(1)=radio*cos(theta)+xtip
            sol(csol).pnt(pint).xyz(2)=radio*sin(theta)+ytip
            sol(csol).pnt(pint).num=pint
            pint=pint+1; pan=pan+1
        end do
        seg=seg+1; pd=pd+1
    enddo
    csol=csol+1
enddo
i=i+1
enddo
return
end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  SUBROUTINE GAULEG(X1,X2,X,W,N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c Evaluación de los puntos de gauss usando la cuadratura
c x1 y x2 dan el limite inferior y superior de la integración
c que devuelve las matrices x(1:n) y w(1:n) de
c longitud n que contiene las abscisas y los pesos de
c Gauss Legendre, para n puntos de la cuadratura
  implicit none
  include './_paramh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  integer n
  real*8 x1,x2,x(n_max_pgauss),w(n_max_pgauss)
  real*8 eps      !relativo a la precisión
  parameter (eps=3.d-14)
  integer i,j,m
  real*8 p1,p2,p3,pp,xl,xm,z,z1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

  m=(n+1)/2
  xm=0.5d0*(x2+x1)
  xl=0.5d0*(x2-x1)

  do 12 i=1,m
    z=cos(3.141592654d0*(i-.25d0)/(n+.5d0))
1    continue
    p1=1.d0
    p2=0.d0
    do 11 j=1,n
      p3=p2
      p2=p1
      p1=((2.d0*j-1.d0)*z*p2-(j-1.d0)*p3)/j
11   continue
      pp=n*(z*p1-p2)/(z*z-1.d0)
      z1=z
      z=z1-p1/pp

      if(abs(z-z1).gt.eps)goto 1
      x(i)=xm-xl*z
      x(n+1-i)=xm+xl*z
      w(i)=2.d0*xl/((1.d0-z*z)*pp*pp)
      w(n+1-i)=w(i)
12  continue

  return
  end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE NORMAL(SOL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    implicit none
    include './_paramh.for'
    include './_sgcommon.for'
    include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    record /s_solido/ sol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    integer csol,cpnt,npint
    real*8 angr,tr,radio,un1,un2
    dimension tr(2,2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    do csol=1,nsol

        angr=0
        tr(1,1)= cos(angr)
        tr(1,2)= sin(angr)
        tr(2,1)=-tr(1,2)
        tr(2,2)=tr(1,1)

C    Rotaciones De Los Datos Calculados Del Sistema De Referencia Global
        cpnt=1
        npint=sol(csol).upint
        do while (cpnt.le.npint)
C    Componente Normal Unitario Del Anillo
            radio=sol(csol).pnt(cpnt).radio
            un1=(sol(csol).pnt(cpnt).xyz(1)-xtip)/radio
            un2=(sol(csol).pnt(cpnt).xyz(2)-ytip)/radio
            sol(csol).pnt(cpnt).normal(1)=tr(1,1)*un1+tr(1,2)*un2
            sol(csol).pnt(cpnt).normal(2)=tr(2,1)*un1+tr(2,2)*un2
            cpnt=cpnt+1
        end do
    end do

    return
end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE ACTUAL_NSEWERYN(MAT,SOL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c      Solución de campo de Seweryn generado a partir de las
c      soluciones auxiliares de Sinclair
      implicit none
      include '.././_paramh.for'
      include '.././_sgcommon.for'
      include '.././_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      record /s_material/ mat(n_max_sol)
      record /s_solido/ sol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      real*8 radio,e(2),rnu(2),g(2),kos(2)
      real*8 theta,alf1,la,bet1,kia,transpc(2,2)
      integer anillo,csol,cpint,n,p,q,s,v
      real*8 tens1p(2,2),tens1(2,2),tens2(2,2)
      real*8 desp1p(2),desp1(2),desp2(2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Modulo de elasticidad
      e(1)=mat(1).prp(1)
      e(2)=mat(2).prp(1)

C      Coeficiente Poisson
      rnu(1)=mat(1).prp(2)
      rnu(2)=mat(2).prp(2)

C      Modulo De Corte
      g(1)=e(1)/(2*(1+rnu(1)))
      g(2)=e(2)/(2*(1+rnu(2)))

C      Constante Kolosov
      kos(1)=3.-4.*mat(1).prp(2)
      kos(2)=3.-4.*mat(2).prp(2)

C      Autovalores: Lambdas
      la=(-lam(1))

C      Angulo De La Esquina
      alf1=alpha2

      anillo=1
      do while (anillo.le.nring)
      csol=1
          do while (csol.le.nsol)
              cpint=1
                  do while (cpint.le.sol(csol).upint)
                      theta=sol(csol).pnt(cpint).angulo
                      radio=sol(csol).pnt(cpint).radio
                  enddo
              enddo
          enddo
      enddo

C      Tensiones Y Desplazamientos Para Kia=1
      bet1=(la**2-1)/(la*cos(2*alf1)+cos(2*la*alf1))
      kia= -1/((2*pi)**(1 + la)*(-la + 1 + bet1))

      desp1(1)=kia*((la+kos(csol))*cos((la+1)*theta)-bet1*
@      cos((la-1)*theta))*(radio)**(-la)/(2*g(csol)*la)

```

```

desp1(2)=kia*((la-kos(csol))*sin((la+1)*theta)-bet1*
@      sin((la-1)*theta))*(radio)**(-la)/(2*g(csol)*la)
tens1(1,1)=-kia*((la+3)*cos((la+1)*theta)-bet1*cos((la-1)*
@      theta))*(radio)**(-la-1)
tens1(2,2)=kia*((la-1)*cos((la+1)*theta)-bet1*cos((la-1)*
@      theta))*(radio)**(-la-1)
tens1(1,2)=-kia*((la+1)*sin((la+1)*theta)-bet1*sin((la-1)*
@      theta))*(radio)**(-la-1)
tens1(2,1)=tens1(1,2)

```

C Transformación de Coordenadas Polares a Coordenadas  
C Cartesianas Y Matriz De Giro

```

transpc(1,1)=cos(theta)
transpc(1,2)=cos(pi/2.+theta) !-sen
transpc(2,1)=cos(pi/2.-theta) !sen
transpc(2,2)=cos(theta)

v=0
desp2=0
do n=1,2
do p=1,2
desp2(n)=desp2(n)+transpc(n,p)*desp1(p)
enddo
v=v+1
sol(csol).pnt(cpint).soluact(v)=desp2(n)
enddo

tens2=0
do n=1,2
do p=1,2
do q=1,2
do s=1,2
@ tens2(n,p)=tens2(n,p)+transpc(n,q)*transpc(p,s)*
      tens1(q,s)
enddo
enddo
v=v+1
sol(csol).pnt(cpint).soluact(v)=tens2(n,p)
enddo
enddo
cpint=cpint+1
end do
      csol=csol+1
end do
anillo=anillo+1
end do

return
end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE AUXILIAR_NSEWERYN(MAT,SOL,RING,OPTH)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c Solución auxiliar de Sinclair: se calculan las componentes de tensión y
c de desplazamiento afectadas por un conveniente Q (factor de normalización)
c en coordenadas polares y se transforman a coordenadas cartesianas 2d
  implicit none
  include '.././_paramh.for'
  include '.././_sgcommon.for'
  include '.././_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  record /s_material/ mat(n_max_sol)
    record /s_solido/ sol(n_max_sol)
    record /s_anillo/ ring(n_max_ring)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  real*8 radio,e(2),rnu(2),g(2),kos(2)
  real*8 theta,alf1,la,bet1,kia,transpc(2,2)
  integer anillo,csol,cpint,n,p,q,s,v,opth,k
  real*8 tens1p(2,2),tens1(2,2),desp1p(2),desp1(2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Propiedades Material
C  Modulo de elasticidad
  e(1)=mat(1).prp(1)
  e(2)=mat(2).prp(1)
C  Coeficiente Poisson
  rnu(1)=mat(1).prp(2)
  rnu(2)=mat(2).prp(2)
C  Modulo De Corte
  g(1)=e(1)/(2*(1+rnu(1)))
  g(2)=e(2)/(2*(1+rnu(2)))
C  Constante Kolosov
  kos(1)=3.-4.*mat(1).prp(2)
  kos(2)=3.-4.*mat(2).prp(2)
C  Autovalores: Lambdas
  la=-(-lam(1))
C  Angulo de la esquina
  alf1=alpha2
  anillo=1
  do while (anillo.le.nring)
    csol=1
    if (anillo.eq.1) then
      cpint=1
    else
      cpint=1+(sol(csol).seg*sol(csol).pgaus*(anillo-1))
    end if
    radio=sol(csol).pnt(cpint).radio
    do while (csol.le.nsol)
      k=1
      do while (k.le.ring(anillo).npring)
        if (radio.eq.sol(csol).pnt(cpint).radio) then
          theta=sol(csol).pnt(cpint).angulo
          Tensiones Y Desplazamientos Para Kia=1
          bet1=(la**2-1)/(la*cos(2*alf1)+cos(2*la*alf1))
          if (opth.eq.1) then
            kia= -1/((2*pi)**(1 + la)*(-la + 1 + bet1))
          else

```

```

        kia= -1/(((2*pi)**(1 + la)*(-la + 1 + bet1))
@         *ring(anillo).kring)
        endif
@         desp1p(1)=kia*((la+kos(csol))*cos((la+1)*theta)-bet1*
@         cos((la-1)*theta))*(radio)**(-la)/(2*g(csol)*la)
@         desp1p(2)=kia*((la-kos(csol))*sin((la+1)*theta)-bet1*
@         sin((la-1)*theta))*(radio)**(-la)/(2*g(csol)*la)
@         tens1p(1,1)=-kia*((la+3)*cos((la+1)*theta)-bet1*
@         cos((la-1)*theta))*(radio)**(-la-1)
@         tens1p(2,2)=kia*((la-1)*cos((la+1)*theta)-bet1*
@         cos((la-1)*theta))*(radio)**(-la-1)
@         tens1p(1,2)=-kia*((la+1)*sin((la+1)*theta)-bet1*
@         sin((la-1)*theta))*(radio)**(-la-1)
        tens1p(2,1)=tens1p(1,2)

```

C Transformación De Coordenadas Polares A Coordenadas Cartesianas y Matriz de Giro

```

        transpc(1,1)=cos(theta)
        transpc(1,2)=cos(pi/2.+theta) !-sen
        transpc(2,1)=cos(pi/2.-theta) !sen
        transpc(2,2)=cos(theta)
        v=0
        desp1=0
        do n=1,2
        do p=1,2
        desp1(n)=desp1(n)+transpc(n,p)*desp1p(p)
        enddo
        v=v+1
        sol(csol).pnt(cpint).soluaux(v)=desp1(n)
        enddo
        tens1=0
        do n=1,2
        do p=1,2
        do q=1,2
        do s=1,2
        tens1(n,p)=tens1(n,p)+transpc(n,q)*transpc(p,s)
@         *tens1p(q,s)
        enddo
        enddo
        v=v+1
        sol(csol).pnt(cpint).soluaux(v)=tens1(n,p)
        enddo
        enddo
        else
        csol=csol+1
        if (anillo.eq.1) then
        cpint=0
        else
        cpint=(sol(csol).seg*sol(csol).pgaus*(anillo-1))
        end if
        k=k-1
        end if
        cpint=cpint+1
        k=k+1
        end do
        csol=csol+1
        end do
        anillo=anillo+1
        end do
        return
        end

```



```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE ACTUAL_QIAN(MAT,SOL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c   Solucion De Campo De Qian_&_Akisanya
    implicit none
    include '../_paramh.for'
    include '../_sgcommon.for'
    include '../_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    record /s_material/ mat(n_max_sol)
    record /s_solido/ sol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    integer csol,cpint,v,n,p,q,s,ll
    real*8 me1,me2,nu1,nu2,mu1,mu2,k1,k2,te1,te2,la
    real*8 al,be,a,b,c,d,e,ra,te
    real*8 desp1(2),tens1(2,2),desp2(2),tens2(2,2)
    real*8 transpc(2,2)
c   Subrutina Dmrrrr (Producto De Matrices)
    integer l10, l8, l2, l1
    parameter (l10=10, l8=8, l2=2,l1=1)
    real*8 nm(10,8),x(8,2),y(2,1),nmx(10,2),fg(10,1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C   Propiedades Del Material

C   Modulo elástico
    me1=mat(2).prp(1)
    me2=mat(1).prp(1)

C   Coeficiente de Poisson
    nu1=mat(2).prp(2)
    nu2=mat(1).prp(2)

C   Modulo de corte
    mu1=me1/(2*(1+nu1))
    mu2=me2/(2*(1+nu2))

C   Constantes de Kolosov
    k1=3.-4.*nu1
    k2=3.-4.*nu2

C   Ángulos Y Autovalores
    te1=alpha2 ! Sólido1
    te2=alpha1
    la=lam(ll)

C   Parámetros de Dundurs
    al=(mu1*(k2+1.)-(k1+1.)*mu2)/(mu1*(k2+1.)+(k1+1.)*mu2)
    be=(mu1*(k2-1.)-(k1-1.)*mu2)/(mu1*(k2+1.)+(k1+1.)*mu2)

C   Raíces del polinomio característico

    e=(al-be)*(cos(2.*la*te1)-cos(2.*la*te1-2.*la*te2)+
@la**2.*(cos(2.*te1)-cos(2.*te1+2.*te2)-1.+cos(2.*te2)))+(1.+
@al)*(1.-cos(2*la*te1))-(1.-be)*(1.-cos(2.*la*te2))

```

$$b = (a-b)e^{i2t_1} - \sin(2t_1 - 2t_2) - a^2(\sin(2t_1) - \sin(2t_1 + 2t_2) + \sin(2t_2)) - (1+a)(\sin(2t_1) - (1-b)e^{i2t_2})$$

$$c = ((a-b)(\cos(2t_1) - \cos(2t_1 + 2t_2) + \cos(2t_2)) - \cos(2t_2 - 2t_1) - 1 + \cos(2t_1)) + (1+a)(1 - \cos(2t_1)) - (1-b)(1 - \cos(2t_2))$$

$$d = a((a-b)(\sin(2t_1) + \sin(2t_2 - 2t_1) - \sin(2t_1)) + \sin(2t_1 + 2t_2) - \sin(2t_2)) - (1+a)\sin(2t_1) - (1-b)\sin(2t_2)$$

### C Funciones Para Tensiones Y Desplazamientos

$$\begin{aligned} x(1,1) &= 1. \\ x(1,2) &= 0. \\ x(2,1) &= 0. \\ x(2,2) &= 1. \end{aligned}$$

$$\begin{aligned} x(3,1) &= -(\cos(2t_1) + a\cos(2t_1)) \\ x(3,2) &= \sin(2t_1) - a\sin(2t_1) \\ x(4,1) &= \sin(2t_1) + a\sin(2t_1) \\ x(4,2) &= \cos(2t_1) - a\cos(2t_1) \\ x(5,1) &= ((a-b)(a\cos(2t_1) - a\cos(2t_1)) + 1 - b) / (1+a) \\ x(5,2) &= ((a-b)(\sin(2t_1) - a\sin(2t_1))) / (1+a) \\ x(6,1) &= -((a-b)(\sin(2t_1) + a\sin(2t_1))) / (1+a) \\ x(6,2) &= (1 - b - (a-b)(a\cos(2t_1) - a\cos(2t_1))) / (1+a) \end{aligned}$$

$$\begin{aligned} x(7,1) &= ((a-b)(\sin(2t_1) + a\sin(2t_1))(\sin(2t_2) - a\sin(2t_2))) / (1+a) - ((1-b + (a-b)(a\cos(2t_1) - a\cos(2t_1))) \cos(2t_2) + a\cos(2t_2)) / (1+a) \\ x(7,2) &= -((a-b)(\sin(2t_1) - a\sin(2t_1))(\cos(2t_2) + a\cos(2t_2))) / (1+a) - ((1-b - (a-b)(a\cos(2t_1) - a\cos(2t_1))) \sin(2t_2) - a\sin(2t_2)) / (1+a) \\ x(8,1) &= -((a-b)(\sin(2t_1) + a\sin(2t_1))(\cos(2t_2) - a\cos(2t_2))) / (1+a) - ((1-b + (a-b)(a\cos(2t_1) - a\cos(2t_1))) \sin(2t_2) + a\sin(2t_2)) / (1+a) \\ x(8,2) &= -((a-b)(\sin(2t_1) - a\sin(2t_1))(\sin(2t_2) + a\sin(2t_2))) / (1+a) + ((1-b - (a-b)(a\cos(2t_1) - a\cos(2t_1))) \cos(2t_2) - a\cos(2t_2)) / (1+a) \end{aligned}$$

$$\begin{aligned} y(1,1) &= (c-e) / (a((1 - \cos(2t_1)) - a\cos(2t_1)) + (c-e) + b + d) \\ y(2,1) &= (b+d) / (a((1 - \cos(2t_1)) - a\cos(2t_1)) + (c-e) + b + d) \end{aligned}$$

csol=1

do while (csol.le.nsol)  
cpint=1

do while (cpint.le.sol(csol).upint)  
ra=sol(csol).pnt(cpint).radio  
te=sol(csol).pnt(cpint).angulo

nm=0.  
nm(1,1) = ((k1 - la)\*cos((la - 1.)\*te))/(2.\*mu1)  
nm(1,2) = ((la - k1)\*sin((la - 1.)\*te))/(2.\*mu1)  
nm(1,3) = -(cos((la + 1.)\*te))/(2.\*mu1)  
nm(1,4) = (sin((la + 1.)\*te))/(2.\*mu1)  
nm(2,1) = ((k1 + la)\*sin((la - 1.)\*te))/(2.\*mu1)

```

nm(2,2) = ((k1 + la)*cos((la - 1.)*te))/(2.*mu1)
nm(2,3) = (sin((la + 1.)*te))/(2.*mu1)
nm(2,4) = (cos((la + 1.)*te))/(2.*mu1)
nm(3,1) = la*(3.-la)*cos((la-1.)*te)
nm(3,2) = la*(la - 3.)*sin((la-1.)*te)
nm(3,3) = -la*cos((la + 1.)*te)
nm(3,4) = la*sin((la + 1.)*te)
nm(4,1) = la*(la + 1.)*cos((la - 1.)*te)
nm(4,2) = la*(la + 1.)*sin((1. - la)*te)
nm(4,3) = la*cos((la + 1.)*te)
nm(4,4) = -la*sin((la + 1.)*te)
nm(5,1) = la*(la - 1.)*sin((la - 1.)*te)
nm(5,2) = la*(la - 1.)*cos((la - 1.)*te)
nm(5,3) = la*sin((la + 1.)*te)
nm(5,4) = la*cos((la + 1.)*te)

```

```

nm(6,5) = ((k2 - la)*cos((la - 1.)*te))/(2.*mu2)
nm(6,6) = ((la - k2)*sin((la - 1.)*te))/(2.*mu2)
nm(6,7) = -(cos((la + 1.)*te))/(2.*mu2)
nm(6,8) = (sin((la + 1.)*te))/(2.*mu2)
nm(7,5) = ((k2 + la)*sin((la - 1.)*te))/(2.*mu2)
nm(7,6) = ((k2 + la)*cos((la - 1.)*te))/(2.*mu2)
nm(7,7) = (sin((la + 1.)*te))/(2.*mu2)
nm(7,8) = (cos((la + 1.)*te))/(2.*mu2)
nm(8,5) = la*(3. - la)*cos((la - 1.)*te)
nm(8,6) = la*(la - 3.)*sin((la - 1.)*te)
nm(8,7) = -la*cos((la + 1.)*te)
nm(8,8) = la*sin((la + 1.)*te)
nm(9,5) = la*(la + 1.)*cos((la - 1.)*te)
nm(9,6) = la*(la + 1.)*sin((1. - la)*te)
nm(9,7) = la*cos((la + 1.)*te)
nm(9,8) = -la*sin((la + 1.)*te)
nm(10,5) = la*(la - 1.)*sin((la - 1.)*te)
nm(10,6) = la*(la - 1.)*cos((la - 1.)*te)
nm(10,7) = la*sin((la + 1.)*te)
nm(10,8) = la*cos((la + 1.)*te)

```

- C           Calcula Mnx =Nm\*X  
call dmrrrr (l10,l8,nm,l10,l8,l2,x,l8,l10,l2,nmx,l10)
- C           Calcula MnxY =NmX\*Y  
call dmrrrr (l10,l2,nmx,l10,l2,l1,y,l2,l10,l1,fg,l10)
- C           Solido1 = material 2 de Qian & Akisanya  
if(csol.eq.1)then  
    desp1(1)=fg(6,1)\*ra\*\*(la)  
    desp1(2)=fg(7,1)\*ra\*\*(la)  
    tens1(1,1)=fg(8,1)\*ra\*\*(la-1.)  
    tens1(2,2)=fg(9,1)\*ra\*\*(la-1.)  
    tens1(1,2)=fg(10,1)\*ra\*\*(la-1.)  
    tens1(2,1)=tens1(1,2)
- C           Solido2 = material 1 de Qian & Akisanya  
else  
    desp1(1)=fg(1,1)\*ra\*\*(la)  
    desp1(2)=fg(2,1)\*ra\*\*(la)  
    tens1(1,1)=fg(3,1)\*ra\*\*(la-1.)  
    tens1(2,2)=fg(4,1)\*ra\*\*(la-1.)  
    tens1(1,2)=fg(5,1)\*ra\*\*(la-1.)  
    tens1(2,1)=tens1(1,2)
- endif

C  
C

Transformación de coordenadas polares a coordenadas cartesianas

matriz de giro

transpc(1,1)=cos(te)

transpc(1,2)=cos(pi/2.+te) !-sen

transpc(2,1)=cos(pi/2.-te) !sen

transpc(2,2)=cos(te)

v=0

desp2=0

do n=1,2

do p=1,2

desp2(n)=desp2(n)+transpc(n,p)\*desp1(p)

enddo

v=v+1

sol(csol).pnt(cpint).soluact(v)=desp2(n)

enddo

tens2=0

do n=1,2

do p=1,2

do q=1,2

do s=1,2

tens2(n,p)=tens2(n,p)+transpc(n,q)\*transpc(p,s)\*tens1(q,s)

enddo

enddo

v=v+1

sol(csol).pnt(cpint).soluact(v)=tens2(n,p)

enddo

enddo

cpint=cpint+1

end do

csol=csol+1

end do

return

end

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE AUXILIAR_QIAN(MAT,SOL,RING,OPTH)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Solucion De Campo De Qian_&_Akisanya
    implicit none
    include '../_paramh.for'
    include '../_sgcommon.for'
    include '../_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    record /s_material/ mat(n_max_sol)
    record /s_solido/ sol(n_max_sol)
    record /s_anillo/ ring(n_max_ring)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    integer anillo,csol,cpint,k,opth,v,n,p,q,s,ll
    real*8 me1,me2,nu1,nu2,mu1,mu2,k1,k2,te1,te2,la
    real*8 al,be,eq,bq,cq,dq,ra,te
    real*8 desp0(2),tens0(2,2),desp1(2),tens1(2,2)
    real*8 transpc(2,2)
C   Subrutina Dmrrrr (Producto De Matrices)
    integer l10, l8, l2, l1
    parameter (l10=10, l8=8, l2=2,l1=1)
    real*8 nm(10,8),x(8,2),y(2,1),nmx(10,2),fg(10,1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Propiedades Del Material
C   Modulo de elasticidad
    me1=mat(2).prp(1)
    me2=mat(1).prp(1)

C   Coeficiente de poisson
    nu1=mat(2).prp(2)
    nu2=mat(1).prp(2)

C   Modulo de corte
    mu1=me1/(2*(1+nu1))
    mu2=me2/(2*(1+nu2))

C   Constantes de Kolosov
    k1=3.-4.*nu1
    k2=3.-4.*nu2

C   Ángulos Y Autovalores
    te1=alpha2
    te2=alpha1
    la=-lam(ll)
                                     ! Sólido 1

C   Parámetros de Dundurs
    al=(mu1*(k2+1.)-(k1+1.)*mu2)/(mu1*(k2+1.) + (k1+1.)*mu2)
    be=(mu1*(k2-1.)-(k1-1.)*mu2)/(mu1*(k2+1.) + (k1+1.)*mu2)

C   Raices Del Polinomio Característico

    eq=(al-be)*(cos(2.*la*te1)-cos(2.*la*te1-2.*la*te2)+
@la**2.*(cos(2.*te1)-cos(2.*te1+2.*te2)-1.+cos(2.*te2)))+(1.+
@al)*(1.-cos(2.*la*te1))-(1.-be)*(1.-cos(2.*la*te2))

    bq=(al-be)*(sin(2.*la*te1)-sin(2.*la*te1-2.*la*te2)-

```

$$\begin{aligned} & @la^{*2} \cdot (\sin(2 \cdot te1) - \sin(2 \cdot te1 + 2 \cdot te2) + \sin(2 \cdot te2)) - (1 + \\ & @al) \cdot (\sin(2 \cdot la \cdot te1)) - (1 - be) \cdot (\sin(2 \cdot la \cdot te2)) \end{aligned}$$

$$\begin{aligned} cq = & ((al - be) \cdot (\cos(2 \cdot la \cdot te1) - \cos(2 \cdot la \cdot te1 + 2 \cdot te2) + \cos(2 \cdot la \cdot te2)) - \\ & @\cos(2 \cdot la \cdot te2 - 2 \cdot te1) - 1 + \cos(2 \cdot te1)) \\ & @+(1 + al) \cdot (1 - \cos(2 \cdot te1)) - (1 - be) \cdot (1 - \cos(2 \cdot te2))) \cdot la \end{aligned}$$

$$\begin{aligned} dq = & la \cdot ((al - be) \cdot (\sin(2 \cdot te1) + \sin(2 \cdot la \cdot te2 - 2 \cdot te1) - \sin(2 \cdot la \cdot te1) + \\ & @\sin(2 \cdot la \cdot te1 + 2 \cdot te2) - \sin(2 \cdot la \cdot te2)) \\ & @-(1 + al) \cdot \sin(2 \cdot te1) - (1 - be) \cdot \sin(2 \cdot te2)) \end{aligned}$$

### C Funciones Para Tensiones Y Desplazamientos

$$\begin{aligned} x(1,1) &= 1. \\ x(1,2) &= 0. \\ x(2,1) &= 0. \\ x(2,2) &= 1. \end{aligned}$$

$$\begin{aligned} x(3,1) &= -(\cos(2 \cdot la \cdot te1) + la \cdot \cos(2 \cdot te1)) \\ x(3,2) &= \sin(2 \cdot la \cdot te1) - la \cdot \sin(2 \cdot te1) \\ x(4,1) &= \sin(2 \cdot la \cdot te1) + la \cdot \sin(2 \cdot te1) \\ x(4,2) &= \cos(2 \cdot la \cdot te1) - la \cdot \cos(2 \cdot te1) \end{aligned}$$

$$\begin{aligned} x(5,1) &= ((al - be) \cdot (la - \cos(2 \cdot la \cdot te1) - la \cdot \cos(2 \cdot te1)) + 1 - be) / (1 + al) \\ x(5,2) &= ((al - be) \cdot (\sin(2 \cdot la \cdot te1) - la \cdot \sin(2 \cdot te1))) / (1 + al) \\ x(6,1) &= -((al - be) \cdot (\sin(2 \cdot la \cdot te1) + la \cdot \sin(2 \cdot te1))) / (1 + al) \\ x(6,2) &= (1 - be - (al - be) \cdot (la + \cos(2 \cdot la \cdot te1) - la \cdot \cos(2 \cdot te1))) / (1 + al) \end{aligned}$$

$$\begin{aligned} x(7,1) &= ((al - be) \cdot (\sin(2 \cdot la \cdot te1) + la \cdot \sin(2 \cdot te1)) \cdot (\sin(2 \cdot la \cdot te2) - \\ & @la \cdot \sin(2 \cdot te2))) / (1 + al) - ((1 - be + (al - be) \cdot (la - \cos(2 \cdot la \cdot te1) \\ & @- la \cdot \cos(2 \cdot te1))) \cdot (\cos(2 \cdot la \cdot te2) + la \cdot \cos(2 \cdot te2))) / (1 + al) \\ x(7,2) &= -((al - be) \cdot (\sin(2 \cdot la \cdot te1) - la \cdot \sin(2 \cdot te1)) \cdot (\cos(2 \cdot la \cdot te2) + \\ & @la \cdot \cos(2 \cdot te2))) / (1 + al) - ((1 - be - (al - be) \cdot (la + \cos(2 \cdot la \cdot te1) \\ & @- la \cdot \cos(2 \cdot te1))) \cdot (\sin(2 \cdot la \cdot te2) - la \cdot \sin(2 \cdot te2))) / (1 + al) \end{aligned}$$

$$\begin{aligned} x(8,1) &= -((al - be) \cdot (\sin(2 \cdot la \cdot te1) + la \cdot \sin(2 \cdot te1)) \cdot (\cos(2 \cdot la \cdot te2) - \\ & @la \cdot \cos(2 \cdot te2))) / (1 + al) - ((1 - be + (al - be) \cdot (la - \cos(2 \cdot la \cdot te1) - la \cdot \\ & @\cos(2 \cdot te1))) \cdot (\sin(2 \cdot la \cdot te2) + la \cdot \sin(2 \cdot te2))) / (1 + al) \\ x(8,2) &= -((al - be) \cdot (\sin(2 \cdot la \cdot te1) - la \cdot \sin(2 \cdot te1)) \cdot (\sin(2 \cdot la \cdot te2) + \\ & @la \cdot \sin(2 \cdot te2))) / (1 + al) + ((1 - be - (al - be) \cdot (la + \cos(2 \cdot la \cdot te1) - \\ & @la \cdot \cos(2 \cdot te1))) \cdot (\cos(2 \cdot la \cdot te2) - la \cdot \cos(2 \cdot te2))) / (1 + al) \end{aligned}$$

$$\begin{aligned} y(1,1) &= (cq - eq) / (la \cdot ((la + 1 - \cos(2 \cdot la \cdot te1)) - la \cdot \\ & @\cos(2 \cdot te1)) \cdot (cq - eq) + bq + dq) \\ y(2,1) &= (bq + dq) / (la \cdot ((la + 1 - \cos(2 \cdot la \cdot te1)) - la \cdot \\ & @\cos(2 \cdot te1)) \cdot (cq - eq) + bq + dq) \end{aligned}$$

anillo=1

do while (anillo.le.nring)  
csol=1

```
if (anillo.eq.1) then
  cpint=1
else
  cpint=1+(sol(csol).seg*sol(csol).pgaus*(anillo-1))
end if
ra=sol(csol).pnt(cpint).radio
```

```

do while (csol.le.nsol)
k=1
do while (k.le.ring(anillo).npring)

if (ra.eq.sol(csol).pnt(cpint).radio) then
te=sol(csol).pnt(cpint).angulo

nm=0.
nm(1,1) = ((k1 - la)*cos((la - 1.)*te))/(2.*mu1)
nm(1,2) = ((la - k1)*sin((la - 1.)*te))/(2.*mu1)
nm(1,3) = -(cos((la + 1.)*te))/(2.*mu1)
nm(1,4) = (sin((la + 1.)*te))/(2.*mu1)
nm(2,1) = ((k1 + la)*sin((la - 1.)*te))/(2.*mu1)
nm(2,2) = ((k1 + la)*cos((la - 1.)*te))/(2.*mu1)
nm(2,3) = (sin((la + 1.)*te))/(2.*mu1)
nm(2,4) = (cos((la + 1.)*te))/(2.*mu1)
nm(3,1) = la*(3. - la)*cos((la-1.)*te)
nm(3,2) = la*(la - 3.)*sin((la-1.)*te)
nm(3,3) = -la*cos((la + 1.)*te)
nm(3,4) = la*sin((la + 1.)*te)
nm(4,1) = la*(la + 1.)*cos((la - 1.)*te)
nm(4,2) = la*(la + 1.)*sin((1. - la)*te)
nm(4,3) = la*cos((la + 1.)*te)
nm(4,4) = -la*sin((la + 1.)*te)
nm(5,1) = la*(la - 1.)*sin((la - 1.)*te)
nm(5,2) = la*(la - 1.)*cos((la - 1.)*te)
nm(5,3) = la*sin((la + 1.)*te)
nm(5,4) = la*cos((la + 1.)*te)

nm(6,5) = ((k2 - la)*cos((la - 1.)*te))/(2.*mu2)
nm(6,6) = ((la - k2)*sin((la - 1.)*te))/(2.*mu2)
nm(6,7) = -(cos((la + 1.)*te))/(2.*mu2)
nm(6,8) = (sin((la + 1.)*te))/(2.*mu2)
nm(7,5) = ((k2 + la)*sin((la - 1.)*te))/(2.*mu2)
nm(7,6) = ((k2 + la)*cos((la - 1.)*te))/(2.*mu2)
nm(7,7) = (sin((la + 1.)*te))/(2.*mu2)
nm(7,8) = (cos((la + 1.)*te))/(2.*mu2)
nm(8,5) = la*(3. - la)*cos((la - 1.)*te)
nm(8,6) = la*(la - 3.)*sin((la - 1.)*te)
nm(8,7) = -la*cos((la + 1.)*te)
nm(8,8) = la*sin((la + 1.)*te)
nm(9,5) = la*(la + 1.)*cos((la - 1.)*te)
nm(9,6) = la*(la + 1.)*sin((1. - la)*te)
nm(9,7) = la*cos((la + 1.)*te)
nm(9,8) = -la*sin((la + 1.)*te)
nm(10,5) = la*(la - 1.)*sin((la - 1.)*te)
nm(10,6) = la*(la - 1.)*cos((la - 1.)*te)
nm(10,7) = la*sin((la + 1.)*te)
nm(10,8) = la*cos((la + 1.)*te)

C      Computa Mnx =Nm*X
call dmrrrr (l10,l8,nm,l10,l8,l2,x,l8,l10,l2,nmx,l10)
C      Computa MnxY =NmX*Y
call dmrrrr (l10,l2,nmx,l10,l2,l1,y,l2,l10,l1,fg,l10)

if (opth.eq.1) then
C      Solido1 = material 2 de Qian & Akisanya
      if(csol.eq.1)then
        desp0(1)=fg(6,1)*ra**(la)
        desp0(2)=fg(7,1)*ra**(la)

```

```

tens0(1,1)=fg(8,1)*ra**(la-1.)
tens0(2,2)=fg(9,1)*ra**(la-1.)
tens0(1,2)=fg(10,1)*ra**(la-1.)
tens0(2,1)=tens0(1,2)

```

```

C      Solido2 = material 1 de Qian & Akisanya
      else
      desp0(1)=fg(1,1)*ra**(la)
      desp0(2)=fg(2,1)*ra**(la)
      tens0(1,1)=fg(3,1)*ra**(la-1.) !srz1
      tens0(2,2)=fg(4,1)*ra**(la-1.) !stz1
      tens0(1,2)=fg(5,1)*ra**(la-1.) !srtz1
      tens0(2,1)=tens0(1,2)

      endif

```

```

C      else
      Solido1 = material 2 de Qian & Akisanya
      if(csol.eq.1)then
      desp0(1)=fg(6,1)*ra**(la)/ring(anillo).kring
      desp0(2)=fg(7,1)*ra**(la)/ring(anillo).kring
      tens0(1,1)=fg(8,1)*ra**(la-1.)/ring(anillo).kring
      tens0(2,2)=fg(9,1)*ra**(la-1.)/ring(anillo).kring
      tens0(1,2)=fg(10,1)*ra**(la-1.)/ring(anillo).kring
      tens0(2,1)=tens0(1,2)

```

```

C      Solido2 = material 1 de Qian & Akisanya
      else
      desp0(1)=fg(1,1)*ra**(la)/ring(anillo).kring
      desp0(2)=fg(2,1)*ra**(la)/ring(anillo).kring
      tens0(1,1)=fg(3,1)*ra**(la-1.)/ring(anillo).kring
      tens0(2,2)=fg(4,1)*ra**(la-1.)/ring(anillo).kring
      tens0(1,2)=fg(5,1)*ra**(la-1.)/ring(anillo).kring
      tens0(2,1)=tens0(1,2)

      endif
endif

```

```

C      Transformación de coordenadas polares a coordenadas cartesianas
C      matriz de giro

```

```

transpc(1,1)=cos(te)
transpc(1,2)=cos(pi/2.+te) !-sen
transpc(2,1)=cos(pi/2.-te) !sen
transpc(2,2)=cos(te)

v=0
desp1=0.
do n=1,2
do p=1,2
desp1(n)=desp1(n)+transpc(n,p)*desp0(p)
enddo
v=v+1
sol(csol).pnt(cpint).soluaux(v)=desp1(n)
enddo

tens1=0.
do n=1,2
do p=1,2
do q=1,2

```



```

do s=1,2
tens1(n,p)=tens1(n,p)+transpc(n,q)*transpc(p,s)*tens0(q,s)
enddo
enddo
v=v+1
sol(csol).pnt(cpint).soluaux(v)=tens1(n,p)
enddo
enddo

else
csol=csol+1
  if (anillo.eq.1) then
    cpint=0
  else
    cpint=(sol(csol).seg*sol(csol).pgaus*(anillo-1))
  end if
k=k-1
end if
cpint=cpint+1
k=k+1
end do
csol=csol+1
end do
anillo=anillo+1
end do

return
end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE DATORDERSOL(SOL,RING,OPTH)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c Ordena los datos para realizar el integrando
  implicit none
  include './_paramh.for'
  include './_sgcommon.for'
  include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  record /s_solido/ sol(n_max_sol)
  record /s_anillo/ ring(n_max_ring)
  record /suma/ sumsol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  integer anillo,csol,cpnt1,nseg,opth,i
  real*8 radio,angini,angfin,rad
  real*8 n(2),uaux(2),saux(2,2),taux(2)
  real*8 uact(2),sact(2,2),tact(2)
  real*8 ubem(2),sbem(2,2),tbem(2)
  real*8 u(2),t(2),rhi,s,wi
  real*8 sumring(n_max_ring)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C   Reporta el valor del integrando en cada punto interno
  open (f_hvst, file='hvstta.txt',status='replace')
  write (f_hvst,13)'cpnt1','angulo','rhi','longseg','sumsol','anillo'

C   Inicia a cero la suma del integrando
  anillo=1
  do while(anillo.le.nring)
    csol=1
    do while (csol.le.nsol)
      sumsol(csol).ring(anillo)=0
      csol=csol+1
    enddo
    anillo=anillo+1
  enddo

C   Adquisición De Datos

  csol=1
  do while (csol.le.nsol)

    anillo=1
    cpnt1=1
    radio=sol(csol).pnt(1).radio

    do while (cpnt1.le.sol(csol).upint)

      if (radio.eq.sol(csol).pnt(cpnt1).radio) then

```

```

C   Longitud Del Segementos Asociado Al Solido Y Anillo
      angini=sol(csol).angini*degree
      angfin=sol(csol).angfin*degree
      nseg=sol(csol).seg
      rad=sol(csol).pnt(cpnt1).radio
      sol(csol).lseg(anillo)=rad

```

```

C      Componentes Normales Unitarios Del Anillo
      n(1)=sol(csol).pnt(cpnt1).normal(1)
      n(2)=sol(csol).pnt(cpnt1).normal(2)

C      Desplazamientos Y Tensiones Del Campo Auxiliar
      uaux(1)=sol(csol).pnt(cpnt1).soluaux(1)
      uaux(2)=sol(csol).pnt(cpnt1).soluaux(2)
      saux(1,1)=sol(csol).pnt(cpnt1).soluaux(3)
      saux(2,2)=sol(csol).pnt(cpnt1).soluaux(6)
      saux(1,2)=sol(csol).pnt(cpnt1).soluaux(4)
      saux(2,1)=sol(csol).pnt(cpnt1).soluaux(5)
      taux(1)=(saux(1,1)*n(1)+saux(1,2)*n(2))
      taux(2)=(saux(2,1)*n(1)+saux(2,2)*n(2))

C      Desplazamientos Y Tensiones Del Campo Real
      uact(1)=sol(csol).pnt(cpnt1).soluact(1)
      uact(2)=sol(csol).pnt(cpnt1).soluact(2)
      sact(1,1)=sol(csol).pnt(cpnt1).soluact(3)
      sact(2,2)=sol(csol).pnt(cpnt1).soluact(6)
      sact(1,2)=sol(csol).pnt(cpnt1).soluact(4)
      sact(2,1)=sol(csol).pnt(cpnt1).soluact(5)
      tact(1)=(sact(1,1)*n(1)+sact(1,2)*n(2))
      tact(2)=(sact(2,1)*n(1)+sact(2,2)*n(2))

C      Desplazamientos Y Tensiones Elementos De Contorno
      ubem(1)=sol(csol).pnt(cpnt1).solu(1)
      ubem(2)=sol(csol).pnt(cpnt1).solu(2)
      sbem(1,1)=sol(csol).pnt(cpnt1).solu(3)
      sbem(2,2)=sol(csol).pnt(cpnt1).solu(6)
      sbem(1,2)=sol(csol).pnt(cpnt1).solu(4)
      sbem(2,1)=sol(csol).pnt(cpnt1).solu(5)
      tbem(1)=(sbem(1,1)*n(1)+sbem(1,2)*n(2))
      tbem(2)=(sbem(2,1)*n(1)+sbem(2,2)*n(2))

C      Peso De La Cuadratura De Gauss
      wi=sol(csol).pnt(cpnt1).peso

      if (opth.eq.1) then
          do i=1,2
              t(i)=tact(i)
              u(i)=uact(i)
          enddo
      else
          do i=1,2
              t(i)=tbem(i)
              u(i)=ubem(i)
          enddo
      endif

      call Hinteger(taux,uaux,t,u,rhi)

      rhi=rhi*wi
      sumsol(csol).ring(anillo)=sumsol(csol).ring(anillo)+rhi

      write(f_hvst,14) cpnt1,(sol(csol).pnt(cpnt1).angulo),rhi
@      ,sol(csol).lseg(anillo),sumsol(csol).ring(anillo),anillo,rad

      else
          radio =sol(csol).pnt(cpnt1).radio

```

```

        anillo =anillo+1
        cpnt1=cpnt1-1
    end if

    cpnt1=cpnt1+1
end do

csol=csol+1
end do

print 12, 'solido','sumsol','longseg'
anillo=1

do while(anillo.le.nring)
    sumring(anillo)=0
    csol=1
    do while (csol.le.nsol)
        s=sumsol(csol).ring(anillo)*sol(csol).lseg(anillo)
        sumring(anillo)=sumring(anillo)+s
        print *, csol,s
    csol=csol+1
    enddo

    if (opth.eq.1) then
C        Salida Campo Auxiliar Y Campo Real
        ring(anillo).kring=sumring(anillo)
        print *, anillo, "fh=",ring(anillo).kring
        print *, " "

    else
C        Salida Campo Auxiliar Normalizado Y Bem
        ring(anillo).hring=sumring(anillo)
        print *, anillo, "ih=",ring(anillo).hring
        print *, " "

    endif

    anillo=anillo+1
enddo

close(f_hvst)

return
12 format (6x,a,8x,a,16x,a)
13 format (20x,a,25x,a,25x,a,25x,a,25x,a,25x,a)
14 format (4x,i5,4(4x,d22.16),4x,i5,4x,d22.16)
15 format (6x,a)
end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE HINTEGER(TAUX,U AUX,T,U,RHI)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    implicit none
    include './_paramh.for'
    include './_sgcommon.for'
    include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    real*8 u(2),t(2),u aux(2),taux(2),rhi
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    rhi=0
    rhi=t(1)*u aux(1)+t(2)*u aux(2)-taux(1)*u(1)-taux(2)*u(2)
    return
    end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE ESCRITURA_PINT(SOL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c    Escribe los puntos internos para CBEM2D

    implicit none
    include './_paramh.for'
    include './_sgcommon.for'
    include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    record /s_solido/ sol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    character*40 nm
    integer csol,cpnt,m,i,j,k
    integer cpntp
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    j=2
    cpnt=1
    nm=name

    write(nm,*) trim(nm), "_gen.txt"
    open(sp,defaultfile='c:\cfitg2d\salida',file=nm,status='replace')

    do csol=1,nsol
        write(sp,13) sol(csol).upint

        do m=1,sol(csol).upint
            cpntp=sol(csol).pnt(m).tiph
            write(sp,14) cpnt,cpntp,(sol(csol).pnt(m).xyz(i),i=1,j)
            cpnt=cpnt+1
        end do
    end do

    close(sp)
    return
13    format(4x,i5)
14    format(4x,i5,4x,i5,4x,e22.16,4x,e22.16)
    end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE ESCRITURA_AUX (SOL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    implicit none
    include './_paramh.for'
    include './_sgcommon.for'
    include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    record /s_solido/ sol(n_max_sol)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    character*40 nm
    integer csol,cpnt,m,i,anillo,upisring
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    nm=name
    write(nm,*) trim(nm), "_aux.txt"
    open(aux,defaultfile='c:\cfitg2d\salida',file=nm,status='replace')

C    Titulo En Cartesianas
    write(aux,14) 'punto','x','y','theta','uaux(x)','uaux(y)',
@    'saux(x)','saux(xy)','saux(yx)','saux(y)','n1','n2','radio'

    do csol=1,nsol
        write(aux,13) csol
        cpnt=1
        do m=1,sol(csol).upint
            write(aux,15) cpnt,(sol(csol).pnt(cpnt).xyz(i), i=1,2),
@            (sol(csol).pnt(cpnt).angulo), (sol(csol).pnt(cpnt).soluaux(i),i=1,6),
@            (sol(csol).pnt(cpnt).normal(i),i=1,2), (sol(csol).pnt(cpnt).radio)
            cpnt=cpnt+1
        end do
    end do

    do anillo=1,nring
        write(aux,*) 'anillo',anillo
        write(aux,14) 'punto','x','y','theta','ur(aux)','ut(aux)',
@        'sr(aux)','srt(aux)','str(aux)','st(aux)','n1','n2','radio'
        do csol=1,nsol
            write(aux,13) csol
            if (anillo.eq.1) then
                cpnt=1; m=1
                upisring=sol(csol).seg
            else
                m=sol(csol).seg*(anillo-1)+1; cpnt=sol(csol).seg*(anillo-1)+1
                upisring=sol(csol).seg*anillo
            endif
            do while (m.le.upisring)
                write(aux,15) cpnt,(sol(csol).pnt(cpnt).xyz(i), i=1,2),
@                (sol(csol).pnt(cpnt).angulo), (sol(csol).pnt(cpnt).soluaux(i),i=1,6),
@                (sol(csol).pnt(cpnt).normal(i),i=1,2), (sol(csol).pnt(cpnt).radio)
                m=m+1; cpnt=cpnt+1
            end do
        end do
    end do

    close(aux)
    return

```

```

13   format(4x,i5)
14   format(4x,a,15x,a,25x,a,23x,a,8(20x,a),23x,a)
15   format(4x,i5,12(4x,e22.16))
     end

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

### SUBROUTINE ESCRITURA\_ACT(SOL)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

implicit none
include '../_paramh.for'
include '../_sgcommon.for'
include '../_structh.for'

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C   Variables Externas

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

record /s_solido/ sol(n_max_sol)

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C   Variables Internas

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

character*40 nm
integer csol,cpnt,m,i,upisring,anillo

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

nm=name
write(nm,*) trim(nm), "_act.txt"
open(act,defaultfile='c:\cfitg2d\salida',file=nm,status='replace')

```

```

C   Titulo En Cartesianas

```

```

write(act,14) 'punto','theta','uact(x)','uact(y)','sact(x)',
@'sact(xy)','sact(yx)','sact(y)','ubem(x)','ubem(y)','sbem(x)',
@'sbem(xy)','sbem(y)'

```

```

do csol=1,nsol

```

```

write(act,13) csol
cpnt=1
do m=1,sol(csol).upint
write(act,15) cpnt,(sol(csol).pnt(cpnt).angulo)
@,(sol(csol).pnt(cpnt).soluact(i),i=1,6)
@,(sol(csol).pnt(cpnt).solu(i),i=1,6)
cpnt=cpnt+1

```

```

end do

```

```

end do

```

```

do anillo=1,nring

```

```

write(act,*) 'anillo',anillo
write(act,14) 'punto','theta','ur(act)','ut(act)','sr(act)',
@'srt(act)','str(act)','st(act)','ur(bem)','ut(bem)','sr(bem)',
@'srt(bem)','str(bem)','st(bem)'

```

```

do csol=1,nsol

```

```

write(act,13) csol
if (anillo.eq.1) then
cpnt=1; m=1
upisring=sol(csol).seg
else
m=sol(csol).seg*(anillo-1)+1; cpnt=sol(csol).seg*(anillo-1)+1
upisring=sol(csol).seg*anillo
endif

```

```

do while (m.le.upisring)
write(act,15) cpnt,(sol(csol).pnt(cpnt).angulo)

```

```

@                                     ,(sol(csol).pnt(cpnt).soluact(i),i=1,6)
@                                     ,(sol(csol).pnt(cpnt).solu(i),i=1,6)
                                     m=m+1; cpnt=cpnt+1
                                     end do
                               end do
end do

close(act)
return
13  format(2x,i5)
14  format(2x,a,15x,a,13(18x,a))
15  format(2x,i5,13(2x,e22.16))
end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE ESCRITURA_FITG(RING)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
implicit none
include './_paramh.for'
include './_sgcommon.for'
include './_structh.for'
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Externas
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
record /s_anillo/ ring(n_max_ring)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   Variables Locales
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
integer m
character*40 nm
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

nm=name
write(nm,*) trim(nm), "_hi.txt"
open(fh,defaultfile='c:\cfitg2d\salida',file=nm,status='replace')

write(fh,14) "anillo", "aux+real", "k"
m=1
do while (m.le.nring)
write(fh,16) m,ring(m).kring,ring(m).hring
m=m+1
end do

close(fh)

return
14  format(4x,a,15x,a,25x,a)
16  format(4x,i5,4x,e22.16,4x,e22.16)
end

```



```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PARAMETROS GLOBALES DEL PROGRAMA
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Archivos De Entrada
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c      Archivo entrada para generar las coordenadas de los puntos internos
      integer      f_gp
      parameter (f_gp      =91)
c      Archivo entrada solucion de pnt internos en bem
      integer      f_lpb
      parameter (f_lpb      =92)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Archivos De Salida
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c      Archivo salida puntos internos generados
      integer      sp
      parameter (sp      =93)
c      Archivo salida punt inter solucion auxiliar
      integer      aux
      parameter (aux      =94)
c      Archivo salida punt inter solucion actual
      integer      act
      parameter (act      =95)
c      Archivo salida de la integral h
      integer      fsh
      parameter (fsh      =96)
c      Archivo salida integral H Vs Theta
      integer      f_hvst
      parameter (f_hvst      =97)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      Parametros No Modificables
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c      Esquinas
      integer      dnmat
      parameter (dnmat      =5)
c      Numero de raiz
      integer      nraiz
      parameter (nraiz      =20)
      integer      n_max_prpmat
      parameter (n_max_prpmat      =35)
      integer      n_max_seg
      parameter (n_max_seg      =97)
      integer      n_max_pint
      parameter (n_max_pint      =6500)
      integer      n_max_ring
      parameter (n_max_ring      =4)
      integer      n_max_sol
      parameter (n_max_sol      =2)
      integer      n_max_pintring
      parameter (n_max_pintring      =2000)
      integer      max_dimh
      parameter (max_dimh      =2)
      integer      n_max_pgauss
      parameter (n_max_pgauss      =50)

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
c          estructuras del programa  "_sructh"          c
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C          /s_pntint/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          structure /s_pntint/
c   num      : numero del punto interno
          integer num
c   tiph     : tipo de punto interno
          integer tiph
c   xyz()    : coordenadas del punto interno
          real*8 xyz(2)
c           : angulo al q se ubico el punto interno
          real*8 angulo
c           : radio al q se ubico el punto interno
          real*8 radio

          real*8 peso
c   solu     : solucion de los elementos de contorno p los puntos internos
          real*8 solu(6)
c   soluaux : solucion del campo auxiliar p los puntos internos
          real*8 soluaux(6)
c   soluact : solucion del campo real p los puntos internos
          real*8 soluact(6)
c   normal  : es la normal de los puntos internos
          real*8 normal(2)

C
C           integer numpr,numpg
C
          end structure
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C          /s_pdiv/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          structure /s_pdiv/
c   num      : numero del segmento
          integer num
c           : angulop: angulo al q se ubico el punto interno
          real*8 angulo
c           : radiop: radio al q se ubico el punto interno
          real*8 radio
c   xyz()    : coordenadas de la division
          real*8 xyz(2)
          end structure

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C          /s_anillo/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          structure /s_anillo/
c   num: numero del anillos
          integer num
c           nsring: numero de solidos por anillo
          integer nsring
c           npring: numero de puntos por anillo
          integer npring
c           kring: factor de normalización
          real*8 kring
c           hring: es el factor de intensidad de tensiones
          real*8 hring

```

```

c
                                end structure
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c                                /s_sol/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c
                                structure /s_solido/
c  num : numero del solido
integer num
c                                angini:   angulo inicial del material(sexagecimal)
                                real*8 angini
c                                angfin   angulo final del material(sexagecimal)
                                real*8 angfin
c  seg : numero de divisiones (/solido) son constantes
integer seg
c                                pgaus: numero de puntos de gauss para las divisiones (/solido)
integer pgaus
c                                tseg:   tamaño del segmento
                                real*8 tseg
c                                lseg: longitud del segmento
                                real*8 lseg(n_max_ring)
c  upint : ultimo punto internos del solido(resultado)
integer upint
c  npint: numero de puntos internos del solido(contador)
integer npint
c
                                record /s_pntint/pnt(n_max_pint)
c
                                record /s_pdiv/pdiv(n_max_seg)
                                end structure
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c                                /s_material/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
                                structure /s_material/
c  tip :                               tipo de material
character*16 tipo
c  prp :                               constantes del material
real*8 prp(n_max_prpmat)
c
                                end structure
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c                                /s_suma/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
                                structure /suma/
                                real*8 ring(n_max_ring)
                                end structure
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c           commons generacion de puntos internos circulares
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c           name:           nombre del problema
           character*40 name
c           solucaso:      caso de solucion(seweryn, qian, stroh),
           integer solucaso,tipbem,tmat
c
           common/datname/name,solucaso,tipbem,tmat(dnmat)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

c           nsolh:         numero de solidos,
           integer nsol
c           nring:         numero de anillos
           integer nring
c           pi=4*atan(1.0),   degree=pi/180.
           real*8 pi,degree
c           xtip, ytip:     coordenadas del centro de la esquina
           real*8 xtip,ytip
c           rh,dr:         radio y distancia entre radios
           real*8      rini,distr
c           alpha1,alpha2:  angulo de solido1, angulo de solido2
           real*8 alpha1,alpha2
c           lam(3):        numero lambdas, autovalores
           real*8 lam(3)

           common/datgenral/nsol,nring,pi,degree,
@           xtip,ytip,rini,distr,alpha1,alpha2,lam

```