

Proyecto Fin de Máster Ingeniería Aeronáutica

Estudio mediante elementos finitos de los
coeficientes de rozamientos mínimos para el arrastre
en laminación en frío.

Autor: Ángel Caballero Bazán

Tutor: Luis Valentín Bohórquez Jiménez

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

Sevilla, 2016



Proyecto Fin de Carrera
Máster en Ingeniería Aeronáutica

**Estudio mediante elementos finitos de los
coeficientes de rozamientos mínimos para el
arrastre en laminación en frío**

Autor:

Ángel Caballero Bazán

Tutor:

Luis Valentin Bohórquez Jiménez

Profesor Contratado Doctor

Dep. Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Proyecto Fin de Carrera: Estudio mediante elementos finitos de los coeficientes de rozamientos mínimos para el arrastre en laminación en frío

Autor: Ángel Caballero Bazán

Tutor: Luis Valentín Bohórquez Jiménez

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

A mi familia

A mi Cloti

Agradecimientos

En primer lugar agradecerle a mi familia el apoyo dado durante todos estos años, y su dedicación e implicación ya que sin ellos no podría haberlo conseguido.

Y una especial mención a Claudia, mi novia, puesto que es la persona que más me apoya y me anima incluso cuando no puede hacerlo consigo misma. Sin ella nada habría sido posible.

Ángel Caballero Bazán

Sevilla, 2016

El proceso de laminación consiste en hacer pasar preformas de material obtenidas según algún proceso anterior a través de unos rodillos con una apertura de sección constante. El mecanismo con el que se consiguen hacer pasar dichas preformas a través de los rodillos es el rozamiento, y es por eso que se quiere realizar un estudio variando todos los parámetros que influyen en este, para su estimación mediante un modelo de elementos finitos y posterior comparación con los resultados dados por la teoría clásica.

Para ello se van a realizar dos modelos, uno para la estimación del coeficiente de arrastre natural, y otro para la estimación del coeficiente de arrastre forzado, y variando algunos parámetros de interés, se comparara la veracidad de los resultados teóricos.

Dichos modelo se realizarán primero con Abaqus y posteriormente con Python, de manera que este proyecto será una buena forma de aprender más herramientas ingenieriles, por lo que será muy enriquecedor.

Finalmente, se compararán todos los resultados y se intentarán dar conclusiones lógicas si se encuentran variaciones, lo cual no será fácil puesto que solo se disponen de los resultados numéricos obtenidos, ya que debido a la difícil estimación del coeficiente de rozamiento hoy en día, no se disponen de muchos resultados experimentales con los que contrastar la veracidad de los resultados.

Abstract

This Project has consisted in doing finite element simulations of cold Rolling of some Steel plates using Abaqus 6.12 in order to calculate the drags coefficients, or what it is the same, the friction between the plate and the roller.

Cold Rolling process lies in two roller spinning with a determinated angular velocity. That rollers apply a pressure to a Steel plate, forcing it to pass between them. The mechanism which finally cause the lamination of the plates is the friction, which change its value depending of some parameters.

There are two kind of drag coeficientes, the natural one and the forced one, which are matched with two diferents geometry models. So once we calculate the friction coefficient to each model with Abaqus, we are going to compare them with the theoretical ones.

To sum up, what we are going to do at this Project is a parametric study of the drag coefficients of cold Rolling process, changing the parameters which affect to his coefficients. In order to do this study, the FEM software that we will use is Abaqus 6.12 and Python, which is the script's programming lenguaje of this software.

Objetivos

El objetivo de este Trabajo Fin de Máster es calcular el valor del rozamiento que da lugar a la laminación en diferentes casos de laminación en frío mediante el modelo de elementos finitos y compararlo con los valores de coeficientes de arrastre según los cuales la teoría afirma que ha de producirse la laminación.

Por lo tanto se van a realizar dos modelos genéricos con el programa de elementos finitos Abaqus, uno para calcular el coeficiente de arrastre natural y otro para el coeficiente de arrastre forzado. Posteriormente se programarán dicho modelos con el lenguaje de alto nivel Python con el fin de realizar un estudio paramétrico variando los parámetros principales del proceso de laminación.

Finalmente se representarán los resultados comprobándose si se cumplen o no las teorías, y en caso de no cumplirse, buscar una respuesta razonable que explique el por qué de dicha discrepancia.

Adicionalmente se mostrará sin entrar en cálculos como puede comprobarse si los valores de rozamiento calculados son adecuados, y además como podría realizarse el cálculo de las fuerzas y momentos de laminación.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Objetivos	xv
Índice	xvi
Índice de Tablas	xviii
Índice de Figuras	xix
Notación	xxii
1 TEORÍA DE LAMINACIÓN	23
1.1. <i>Análisis simple Del laminado Plano.</i>	23
1.2. <i>Condiciones de Arrastre.</i>	25
1.2.1. Arrastre natural.	25
1.2.2. Arrastre forzado	28
2 INTRODUCCIÓN AL ENTORNO DE ABAQUS.	30
2.1. <i>Entorno de Abaqus</i>	30
2.1.1 Generación de archivo de entrada mediante el interfaz gráfico.	30
3 INTRODUCCION AL ENTORNO DE PYTHON.	33
3.1. <i>Herramientas necesarias para la realización Del proyecto.</i>	33
3.1.1 Python Como calculadora.	33
3.1.2 Operaciones con vectores.	36
3.1.3 Inicios a la programación.	36
4 DEFINICIÓN DEL MODELO EN ABAQUS y DEL PROBLEMA.	39
4.1. <i>Materiales</i>	39
4.2. <i>Parámetros</i>	40
4.3. <i>Modelo.</i>	41
4.3.1. Arrastre natural.	42
4.3.2. Mallado del modelo.	45
4.3.3. Modelo de arrastre forzado. Diferencias principales.	48
5. MODELO EN PYTHON.	50
5.1. <i>Crear un Model Database (mdb)</i>	50
5.2. <i>Crear un Output Database (odb).</i>	57
6 RESULTADOS.	59
6.1 <i>Arrastre Natural</i>	60
6.1.1 Aluminio.	61
6.1.2 Acero	69
6.2 <i>Arrastre Forzado</i>	72
6.2.1 Aluminio.	72
6.2.2 Acero.	79
6.3. <i>Conclusiones.</i>	80

6.3.1. Posibles soluciones de mejora.	80
ANEXO A. Modelo paso a paso..	83
<i>A.1 Arrastre natural</i>	83
A.1.1. Modelo y Geometría.	83
A.1.2. Material y Sección.	85
A.1.3. Assembly.	87
A.1.4. Superficies y Sets.	89
A.1.5. Step.	90
A.1.6. Contact.	92
A.1.7. Condiciones de contorno.	94
A.1.8. Mallado.	96
A.1.9. Análisis.	98
A.1.10. Visualización.	98
<i>A.2. Arrastre forzado.</i>	98
A.2.1 Geometría.	99
A.2.2. Sets and Surfaces:	100
A.2.3. Assembly	101
A.2.4. Resto del modelo.	102
ANEXO B. Análisis explícito Vs Implícito.	103
<i>B.1. Metodo de Resolución.</i>	103
B.1.1. Integración directa implícita	103
B.1.2. Integración directa explícita.	103
B.1.3. Aplicación al modelo realizado.	104
ANEXO C. Malla adaptativa, Mass Scaling	105
<i>C.1. Malla adaptativa.</i>	105
<i>C.2. Mass Scaling.</i>	106
ANEXO D. Leyes de comportamiento de los materiales.	107
<i>D.1. Tipos de relaciones esfuerzo-deformación.</i>	108
<i>D.2. Esfuerzo de fluencia.</i>	109
ANEXO E. Visualización de algunos de los casos estudiados.	111
<i>E.1 Arrastre Natural. Aluminio, $R = 0.2 \text{ m}$, $h_0 = 0.04 \text{ m}$, $hf = 0.625 * h_0 \text{ m}$.</i>	111
<i>E.2. Arrastre Natural. Acero, $R = 0.5 \text{ m}$, $h_0 = 0.01 \text{ m}$. $hf = 0.5 * h_0 \text{ m}$.</i>	112
<i>E.3. Arrastre Forzado. Aluminio, $R = 0.5 \text{ m}$, $h_0 = 0.08 \text{ m}$. $hf = 0.375 * h_0 \text{ m}$.</i>	113
<i>E.4 Arrastre Forzado. Acero, $R = 0.2 \text{ m}$, $h_0 = 0.01 \text{ m}$, $hf = 0.125 * h_0 \text{ m}$.</i>	114
ANEXO F. Cálculo de fuerzas y momentos de laminación. Teoría.	115
ANEXO G. Cálculo de fuerzas y momentos de laminación. Un ejemplo con Abaqus.	117
ANEXO H. Criterios de convergencia.	119
Referencias	120

ÍNDICE DE TABLAS

Tabla 1. Estudio Paramétrico a realizar.	41
Tabla 2. Condiciones de contorno.	43
Tabla 3. Resultados Arrastre Natural para $R=0.2$ m	63
Tabla 4. Resultados Arrastre Natural para $R=0.5$ m.	64
Tabla 5. Propiedades Plásticas del Aluminio 7075	86
Tabla 6. Condiciones de Contorno.	94

ÍNDICE DE FIGURAS

Ilustración 1. Preformas	23
Ilustración 2. Proceso de Laminación.	24
Ilustración 3. Laminación de una Placa.	25
Ilustración 4. Equilibrio de Fuerzas.	26
Ilustración 5. Parámetros y relaciones geométricas.	27
Ilustración 6. Arrastre forzado.	28
Ilustración 7. Entorno de Abaqus.	31
Ilustración 8. Esquema del procesado de datos de Abaqus.	32
Ilustración 9. Modelo completo con dos rodillos.	42
Ilustración 10. Modelo con simetría.	43
Ilustración 11. Modelo con simetría y un solo arco de 90°.	44
Ilustración 12. Algoritmo de la evolución del modelo.	44
Ilustración 13. Modelo definitivo con Datum Point	45
Ilustración 14. Hipótesis deformación plana en Abaqus.	46
Ilustración 15. Mallado del modelo.	47
Ilustración 16. Diagrama de flujo del método utilizado para la obtención del rozamiento.	48
Ilustración 17. Modelo de Arrastre Forzado.	49
Ilustración 18. Creación de un sistema de coordenadas.	59
Ilustración 19. Arrastre.	60
Ilustración 20. No se produce arrastre.	60
Ilustración 21. AN- Al -1-: μ frente a Δh	61
Ilustración 22. AN- Al -2-: μ frente a Δh	61
Ilustración 23. AN - Al -3-: μ frente a Δh	62
Ilustración 24. AN- Al -4-: μ frente a Δh	64
Ilustración 25. AN - Al -5-: μ frente a Δh	65
Ilustración 26. AN - Al -6-: μ frente a Δh	65
Ilustración 27. AN - Al -7-: μ frente a $L/hmed$	66
Ilustración 28. AN - Al -8-: μ frente a $L/hmed$	66
Ilustración 29. AN - Al -9-: μ frente a $L/hmed$	67
Ilustración 30. AN - Al -10-: μ frente a θ	68
Ilustración 31. AN - Al -11-: Efecto del Mallado Adaptativo.	68
Ilustración 32. AN - AC -1-: μ frente a Δh	69
Ilustración 33. AN - AC -2-: μ frente a Δh	69
Ilustración 34. AN - AC -3-: μ frente a Δh	70

Ilustración 35. AN - AC -4-: μ frente a Δh	70
Ilustración 36. AN - AC -5-: μ frente a Δh	71
Ilustración 37. AN - AC -6-: μ frente a Δh	71
Ilustración 38. AF - AL -1-: μ frente a Δh	72
Ilustración 39. AF - AL -2-: μ frente a Δh	73
Ilustración 40. AF - AL -3-: μ frente a Δh	73
Ilustración 41. AF - AL -4-: Comparación con y sin Malla Adaptativa.	74
Ilustración 42. Modelo “Real 2” con dos <i>Steps</i> y dos rozamientos.	75
Ilustración 43. AF - AL -5-: μ frente a Δh	75
Ilustración 44. AF - AL -6-: μ frente a Δh	76
Ilustración 45. AF - AL -7-: μ frente a Δh	76
Ilustración 46. AF - AL -8-: μ frente a Δh	77
Ilustración 47. AF - AL -9-: μ frente a Δh	77
Ilustración 48. AF - AL -10-: μ frente a Δh	78
Ilustración 49. AF - AC -1-: μ frente a Δh	79
Ilustración 50. Parámetros de remallado.	81
Ilustración 51. Parámetros de la técnica ALE de Abaqus.	81
Ilustración 52. Interfaz de Abaqus.	84
Ilustración 53. Creación de la geometría de partida.	85
Ilustración 54. Creación del arco que conforma el rodillo.	85
Ilustración 55. Definición de las propiedades del material en Abaqus.	87
Ilustración 56. Datum Point.	88
Ilustración 57. Ensamblado del modelo.	89
Ilustración 58. Creación de los Sets.	89
Ilustración 59. History Output.	91
Ilustración 60. Definición de la malla adaptativa.	92
Ilustración 61. Introducción del valor del rozamiento.	93
Ilustración 62. Contacto entre rodillo y placa.	94
Ilustración 63. Condiciones de contorno.	95
Ilustración 64. Velocidad angular de los rodillos.	96
Ilustración 65. Tipo de elemento escogido.	97
Ilustración 66. Estructura de la malla.	97
Ilustración 67. Superficie <i>Top</i> en Arrastre forzado.	101
Ilustración 68. Superficie <i>Top</i> en Arrastre Natural.	101
Ilustración 69. Ensamblado del model de Arrastre Forzado.	101
Ilustración 70. Curva esfuerzo-deformación.	107
Ilustración 71. Zona plástica de la curva esfuerzo-deformación	108
Ilustración 72. Material perfectamente plástico.	108
Ilustración 73. Material elástico y perfectamente plástico.	109

Ilustración 74. Material elástico y con endurecimiento por deformación	109
Ilustración 75. Curva de fluencia.	110
Ilustración 76. Representación gráfica proceso laminación-1	111
Ilustración 77. Representación gráfica proceso laminación-2	112
Ilustración 78. Representación gráfica proceso laminación-3	113
Ilustración 79. Representación gráfica proceso laminación-4	114
Ilustración 80. Tensión de un elemento.	117
Ilustración 81. Deformación de un elemento.	118
Ilustración 82. Reacción en el eje del rodillo.	118

Notación

h_0	Altura/Espesor inicial de la placa
h_f	Altura/Espesor final de la placa
Δh	Incremento de altura/espesor de la placa
\bar{h}	Valor medio de la suma de alturas inicial y final.
l_0	Longitud inicial de la placa
w_0	Ancho inicial de la placa
w_f	Ancho final de la placa
v_0	Velocidad a la entrada de los rodillos
v_f	Velocidad a la salida de los rodillos
L	Longitud de contacto
$\overline{\sigma_{ys}}$	Tensión de fluencia media del proceso
σ_o	Tensión de tracción a la entrada del proceso
σ_f	Tensión de tracción a la salida del proceso
F	Fuerza de laminación
cos	Función coseno
tg	Función tangente
arctg	Función arco tangente
sen	Función seno
:	Tal que
<	Menor o igual
>	Mayor o igual
\cong	Aproximadamente igual que
m	Metro
mm	Milímetro.

1 TEORÍA DE LAMINACIÓN

El proceso de laminación es un proceso de conformado en el cual se obliga a un material a pasar entre la abertura formada por dos rodillos, la cual se mantiene constante, teniendo el material una sección distinta a dicha abertura y pasando entre estos debido al arrastre producido por el giro de los rodillos. Una vez se produce el arrastre, la sección del material adquiere la forma de la abertura de los rodillos.

Es decir, la laminación se puede resumir en que es un proceso de conformado plástico en el cual se reduce el espesor de una chapa mediante fuerzas de compresión aplicadas por rodillos que giran a velocidad angular constante.

Se puede pensar que en el proceso de laminación solo se obtienen laminas, sin embargo, según la preforma usada, la forma de los rodillos, así como la configuración de los mismos, se pueden obtener diversas geometrías:

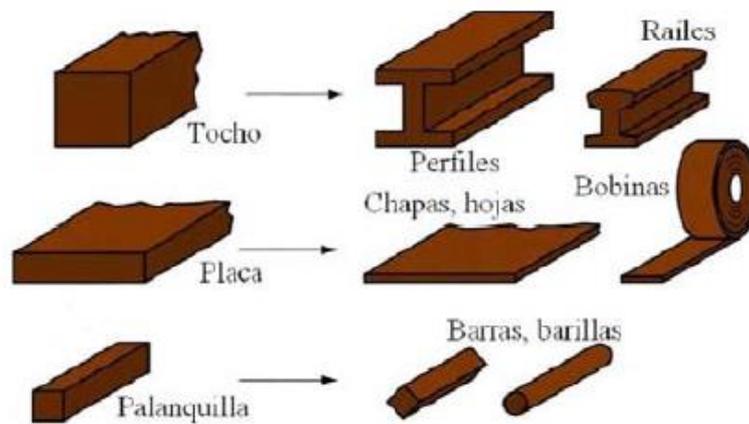


Ilustración 1. Preformas

Donde todas presentan la misma característica común que como ya se explicó consiste en que tienen una sección constante, que es la de la abertura entre rodillos, a lo largo de una dirección (la dirección de laminación), y la preforma de origen presenta una dimensión mucho mayor que las otras dos.

Aun así, en este proyecto se va a tratar el tema de laminación de placas, y todo el razonamiento teórico a seguir se va a basar en dicha geometría.

1.1. Análisis simple Del laminado Plano.

Supongamos que se quiere laminar una placa de dimensiones $h_o \times w_o \times l_o$, haciéndola pasar por unos rodillos:

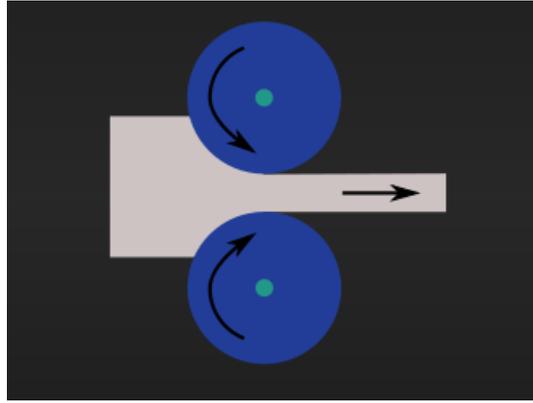


Ilustración 2. Proceso de Laminación.

Por conservación de volumen y caudal se obtiene que siempre debe cumplirse la siguiente relación:

$$h_o w_o v_o = h_f w_f v_f \quad (1 - 1)$$

En el laminado de chapas y láminas es habitual que el ancho de las mismas (w_o) sea mucho mayor que el espesor de las mismas. Ello da lugar a que para que se produzca la deformación del material entre los rodillos según el eje de los mismos, es decir, para que se produzca un aumento en el ancho de las placas, tendrían que rozar a lo largo de todo su ancho. Sin embargo en el sentido de la laminación la longitud de contacto es mucho menor, por lo que la resistencia que ofrece el material a pasar entre los rodillos es inferior, produciéndose la laminación en dicho sentido.

Este fenómeno da lugar a un estado de deformación plana, que da lugar a que la anchura de la lámina se mantenga constante, produciéndose por lo tanto una variación de altura y de longitud de las mismas. Por lo que se obtiene la siguiente igualdad:

$$\text{si } h_o \ll w_o \rightarrow L \ll w_o \rightarrow w_o \cong w_f \rightarrow h_o v_o = h_f v_f \quad (1 - 2)$$

Por lo que hasta ahora queda claro que el cambio de forma de las preformas se obtiene obligando al material a fluir por compresión entre los rodillos, y puesto que no se produce variación en el ancho de las placas, y se acorta el espesor, dicho incremento de altura lleva asociado un aumento de longitud de las placas.

Se podría por lo tanto pensar que es posible obtener el mismo resultado traccionando las placas longitudinalmente. Sin embargo, ello no es posible ya que la tracción conlleva fenómenos de inestabilidad muy elevados, y se produciría la estricción y posterior rotura de las láminas. Lo que si sería posible es añadir una tracción horizontal a la compresión que experimenta la lámina por parte de los rodillos para facilitar la fluencia del material.

Este fenómeno se puede observar en la expresión obtenida para la fuerza de los rodillos (ANEXO F), que es:

$$F = \left(\overline{\sigma}_{ys} - \frac{\sigma_f + \sigma_o}{2} \right) L w \quad (1 - 3)$$

Donde:

- $\overline{\sigma}_{ys}$: Tensión media de fluencia en el proceso.
- σ_o : Tensión de tracción a la entrada del proceso
- σ_f : Tensión de tracción a la salida del proceso.
- L : Longitud de contacto
- w : Anchura de la chapa.

Donde la expresión de la longitud de contacto se obtendrá posteriormente

Cabe destacar que en el término de tensión media de fluencia queda reflejado lo explicado anteriormente en lo que respecta a que la laminación se da gracias a esfuerzos de compresión, los cuales pueden verse aliviados con

fuerzas de tracción longitudinal.

Por lo que una vez explicado los fenómenos que intervienen en la laminación, el último aspecto a comentar es que mecanismo hace posible la transmisión de esfuerzos entre rodillos y placa para que se produzca la laminación, y dicho mecanismo no es otro que el rozamiento.

De manera que para el objetivo de este proyecto es interesante conocer las condiciones que tienen lugar para que se produzca la entrada de la preforma entre los rodillos: Las condiciones de arrastre.

1.2. Condiciones de Arrastre.

Como se ha explicado la laminación se basa en las fuerzas de fricción ejercida por rodillos sobre la preforma que se desea laminar. Estas fuerzas, que son de compresión (aunque pueden verse ayudadas por fuerzas de tracción como se explicó anteriormente), introducen la pieza entre los rodillos, lo que provoca una reducción de espesor de la misma. Además el mecanismo de transmisión de esfuerzos entre rodillo y placa es el rozamiento, y es el que permite que se produzca la laminación, ya que sin la existencia de este los rodillos patinarían sobre la chapa y sería imposible laminarla.

Por lo tanto, es interesante estimar el mínimo rozamiento necesario para llevar a cabo una determinada reducción de espesor Δh , con unos rodillos de radio R , o bien formulando el problema de otra forma, la máxima reducción de espesor que se puede obtener para un radio y un rozamiento determinado. Dicho valor mínimo de rozamiento es conocido como condición de arrastre, y existen dos tipos: Arrastre Natural y Arrastre Forzado.

Para analizar las condiciones de arrastre se va a partir de una configuración de deformación plana, dada por una placa de altura inicial h_0 que se introduce a una velocidad v_0 entre unos rodillos de radio R , para obtener una placa de altura final h_f y velocidad de salida v_f (la cual según la expresión [1-2] es mayor que la velocidad de entrada).

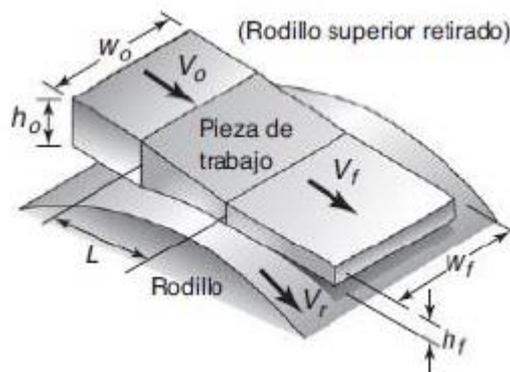


Ilustración 3. Laminación de una Placa.

1.2.1. Arrastre natural.

La condición de arrastre natural se obtiene del equilibrio de fuerzas en el primer punto de contacto entre la placa y el rodillo, como se puede observar en la siguiente figura:

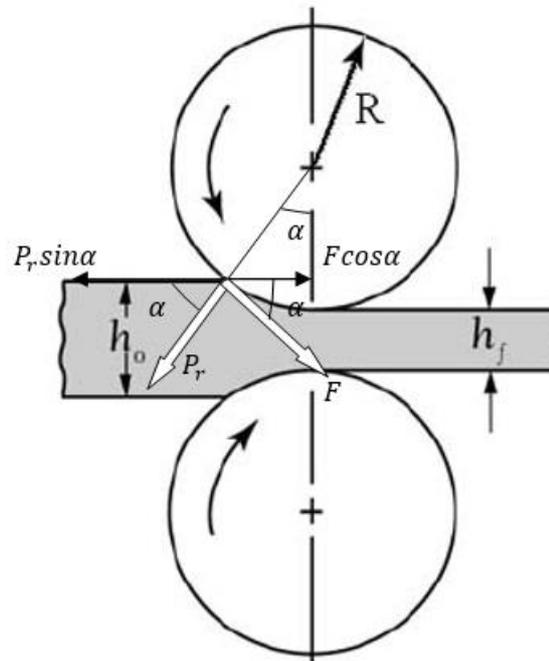


Ilustración 4. Equilibrio de Fuerzas.

Donde para que el material fluya entre los rodillos debe cumplirse que:

$$F \cos \alpha \geq P_r \sin \alpha \quad (1 - 4)$$

Reagrupando se obtiene:

$$\frac{F}{P_r} \geq \frac{\sin \alpha}{\cos \alpha} = \tan \alpha \quad (1 - 5)$$

Y teniendo en cuenta que la fuerza de fricción puede expresarse como $F = \mu P_r$ se obtiene:

$$\mu \geq \tan \alpha \quad (1 - 6)$$

Por lo que para que se dé la condición de arrastre natural debe cumplirse que el coeficiente de rozamiento sea mayor que la tangente del ángulo que forman el punto de contacto, con el radial de 270° del rodillo.

Puesto que el objetivo del proyecto es realizar un análisis paramétrico en el que se trabaje con diferentes configuraciones de h_0 , h_f , y radios, parece conveniente expresar la expresión anterior de alguna manera que involucre dichos parámetros.

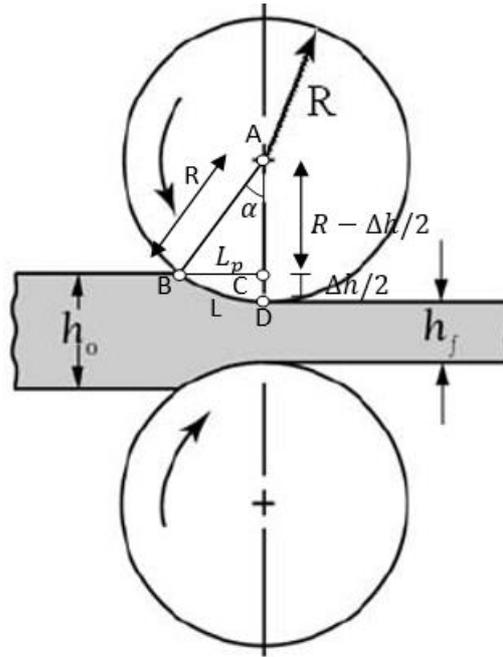


Ilustración 5. Parámetros y relaciones geométricas.

Como se observa en la imagen, del triángulo ABC se obtiene:

$$R^2 = L_p^2 + \left(R - \frac{\Delta h}{2}\right)^2 \quad (1 - 7)$$

$$L_p^2 = R^2 - \left(R^2 + \frac{\Delta h^2}{4} - R\Delta h\right) = R\Delta h - \frac{\Delta h^2}{4} \quad (1 - 8)$$

Como se puede observar en la figura, en los procesos de laminación de laminas el radio de los rodillos suele ser mucho mayor que la reducción de espesor que se va a llevar a cabo, por lo que:

$$R \gg \frac{\Delta h}{2} \rightarrow L_p = \sqrt{R\Delta h} \quad (1 - 9)$$

Donde

$$\Delta h = h_o - h_f \quad (1 - 10)$$

Por lo que la condición de arrastre natural quedaría de la siguiente forma:

$$\mu \geq \tan \alpha = \frac{L_p}{R - \frac{\Delta h}{2}} \approx \frac{\sqrt{R\Delta h}}{R - \frac{\Delta h}{2}} = \sqrt{\frac{\Delta h}{R}} \quad (1 - 11)$$

O reordenando términos

$$\mu^2 R \geq \Delta h \quad (1 - 12)$$

Por lo que es fácil estimar el coeficiente de rozamiento necesario para que se produzca al arrastre dados el radio del rodillo y la reducción de espesor deseado, o el problema inverso, obtener la reducción de espesor dado un coeficiente de rozamiento y unos rodillos de radio R.

1.2.2. Arrastre forzado

En esta ocasión la placa se ha introducido entre los rodillos debido a varias posibilidades: fuerza aplicada inicialmente, arrastre natural hasta dicha configuración, borde con mordida, etc.

Por lo tanto no conocemos puntualmente el equilibrio de fuerza que debe darse en cualquier punto de la zona de contacto para que se produzca el arrastre de la placa. La configuración explicada se muestra en la siguiente figura:

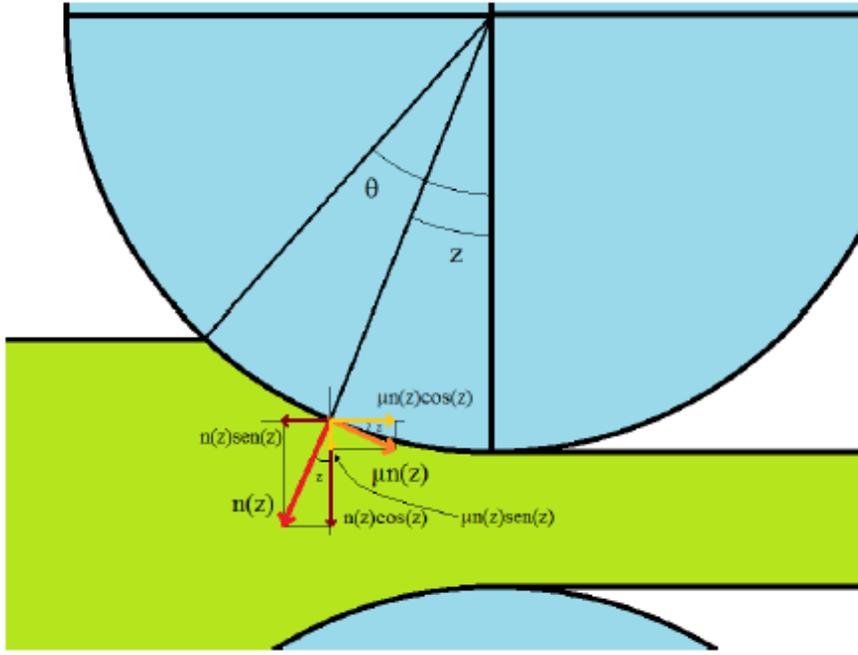


Ilustración 6. Arrastre forzado.

Donde se tienen las siguientes expresiones para definir las fuerzas implicadas en la determinación de la condición de arrastre forzado:

$$\text{Fuerza de arrastre: } F_a = \int_0^\theta \mu n(z) \cos z w R dz - \int_0^\theta n(z) \sin z w R dz \quad (1 - 13)$$

$$\text{Fuerza de compresión: } F_c = \int_0^\theta \mu n(z) \sin z w R dz + \int_0^\theta n(z) \cos z w R dz \quad (1 - 14)$$

Suponemos que la tensión de compresión es constante. Esto puede estar medio justificado si suponemos que el material comienza a fluir al empezar la laminación, por lo que las tensiones normales verticales deberán estar próximas al límite elástico del material. También sería aproximadamente cierto durante el desarrollo de la laminación si el material fuese elasto-plástico perfecto.

$$dF_c = \mu n(z) \sin z w R dz + n(z) \cos z w R dz \quad (1 - 15)$$

El área diferencial sobre el que actúa la fuerza de compresión será:

$$dA = w R \cos z dz \quad (1 - 16)$$

La tensión de compresión será entonces:

$$\sigma = \mu n(z) \tan z + n(z) = n(z) (\mu \tan z + 1) \approx \sigma_y \quad (1 - 17)$$

Por lo tanto:

$$F_a \approx \int_0^\theta \frac{\mu \cos z}{\mu \tan z + 1} \sigma_y R w dz - \int_0^\theta \frac{\sin z}{\mu \tan z + 1} \sigma_y R w dz \quad (1 - 18)$$

El mínimo coeficiente de rozamiento será aquél que haga la fuerza de arrastre cero. Por lo tanto:

$$\int_0^\theta \frac{\mu \cos z}{\mu \tan z + 1} dz - \int_0^\theta \frac{\sin z}{\mu \tan z + 1} dz = 0 \quad (1 - 19)$$

La solución de la ecuación integral anterior es complicada pero se puede obtener una solución aproximada para ángulos de contacto pequeños. Para ello basta con hacer $dA = wRdz$, es decir suponer que en esa expresión $\cos z$ es prácticamente uno. La ecuación se transforma entonces en

$$\int_0^\theta \frac{\mu \cos z}{\mu \sin z + \cos z} dz - \int_0^\theta \frac{\sin z}{\mu \sin z + \cos z} dz = 0 \quad (1 - 20)$$

Que integrando queda en:

$$[\text{Log}(\cos z + \mu \sin z)]_0^\theta = 0 \quad (1 - 21)$$

Que a su vez, simplificando, da:

$$\text{Log}(\cos \theta + \mu \sin \theta) = 0 \Rightarrow \mu = \frac{1 - \cos \theta}{\sin \theta} = \tan \frac{\theta}{2} \quad (1 - 22)$$

G.W. Rowe (Principles of Industrial Metalworking Processes) llega al mismo resultado suponiendo linealidad de las fuerzas de contacto, cosa que no justifica. En ese caso las fuerzas medias se producen en el ángulo medio y por tanto el resultado será el mismo que el correspondiente a arrastre natural pero considerando la mitad del ángulo de contacto.

2 INTRODUCCIÓN AL ENTORNO DE ABAQUS.

Abaqus es un programa informático que utiliza el método de los elementos finitos para resolver problemas de muchos ámbitos de la ciencia, y principalmente de la ingeniería. Con dicho programa se pueden resolver una gran variedad de problemas, independientemente de la geometría o no linealidades que aparezcan. Además su gran compatibilidad con el programa CAD/CAM de diseño CATIA ofrece un gran abanico de posibilidades a la hora de diseñar geometrías de gran complejidad de una manera muy sencilla y eficiente.

Cabe destacar que el programa posee una librería muy amplia de elementos finitos de manera que estos se adapten de la manera más eficiente y real posible al problema que se quiera tratar.

También posee una gran variedad de modelos de solución o *solvers* que permiten al usuario escoger el que mejor se adapte al problema que quiera resolver.

Además Abaqus posee una extensa guía de usuario, dónde se puede encontrar como resolver prácticamente cualquier problema que pueda surgir, y que herramientas son más indicadas en cada caso, todo ello acompañado de numerosos ejemplos resueltos que permiten al usuario que el aprendizaje sea más sencillo y cómodo.

Por todo ello se decidió usar dicho programa de elementos finitos para abarcar la resolución del problema que se ha planteado en el presente trabajo.

2.1. Entorno de Abaqus

A la hora de trabajar con ABAQUS hay que tener en cuenta que la comunicación con el programa se puede efectuar de dos maneras posibles: mediante scripts escritos en un lenguaje de programación compatible con el programa o utilizando el preprocesador gráfico del mismo.

Cabe destacar que la comunicación mediante editores de texto con Abaqus conlleva un amplio conocimiento del lenguaje de programación PYTHON en el que debe escribirse el código, el cual está sujeto a normas de sintaxis muy rígidas.

Sin embargo, trabajar mediante dicho encriptado ofrece un amplio abanico de posibilidades que el entorno gráfico de ABAQUS no permite, como es por ejemplo, la parametrización de problemas, que es lo que se pretende realizar en este trabajo.

Es por eso que en este proyecto se ha optado en primer lugar por usar el preprocesador gráfico de ABAQUS para realizar los modelos, y una vez realizados estos y entendido el funcionamiento de ABAQUS, se ha decidido utilizar Scripts que permitan recrear lo realizado con el interfaz, facilitando así el estudio paramétrico del proceso de laminación.

A continuación se muestran algunas características generales de ABAQUS sin introducimos mucho en todas las herramientas que posee, ya que estas se ampliarán a medida que se desarrolle el trabajo.

2.1.1 Generación de archivo de entrada mediante el interfaz gráfico.

El archivo de entrada de Abaqus debe poseer todos los datos necesarios para el posterior análisis. Es por eso que antes de iniciar el análisis, abaqus comprobará todos estos datos, y en caso de error, no permitirá que se inicie el análisis del mismo.

Todos los datos necesarios para realizar el modelo se pueden introducir usando el interfaz gráfico de abaqus, mediante todos los módulos que ofrece el preprocesador. El preprocesador tiene la siguiente estructura:

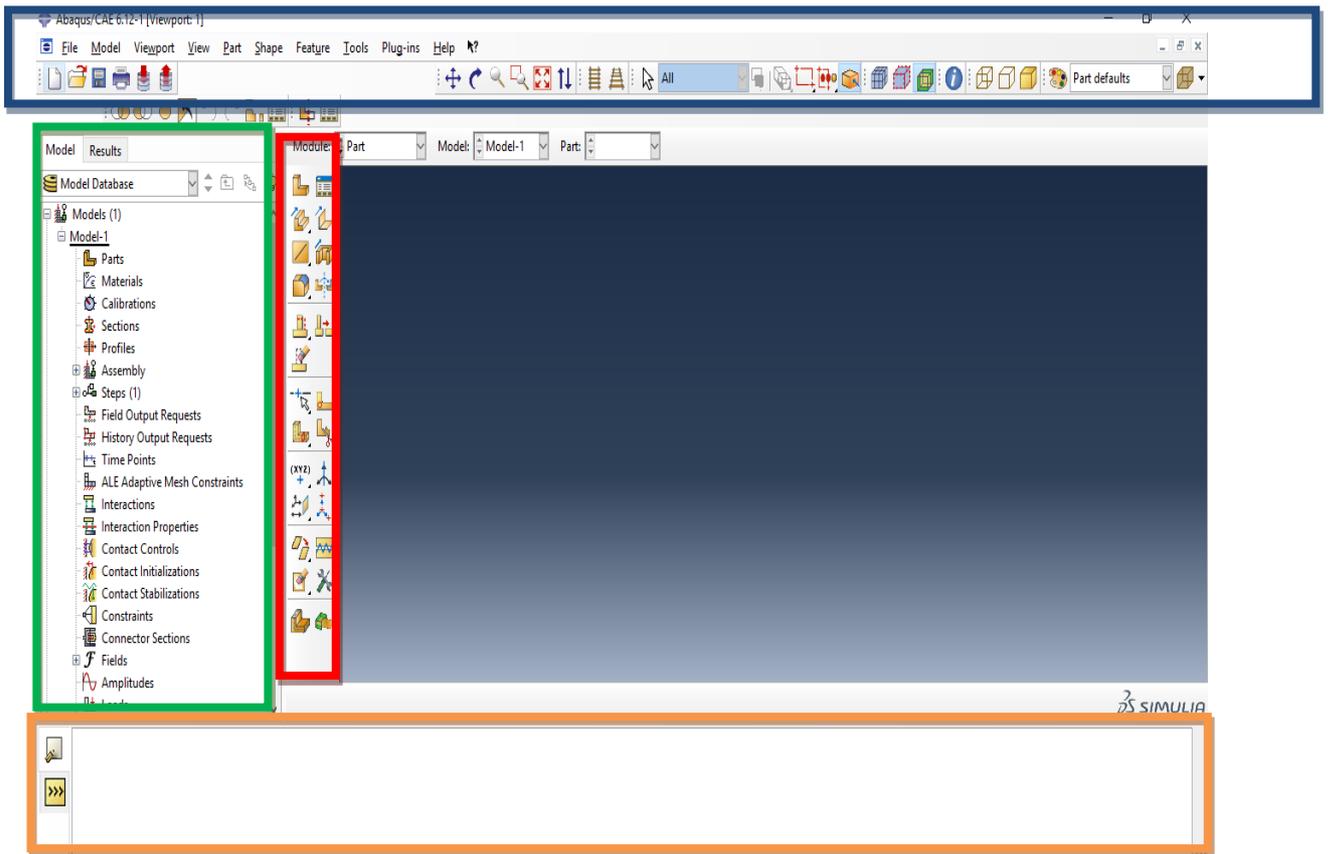


Ilustración 7. Entorno de Abaqus.

En la parte superior, en color azul, se puede observar el menú desplegable de Abaqus, donde se encuentran todas las herramientas que se pueden utilizar en cada módulo, y que cambia según en cual nos encontremos.

Otra forma de trabajar es utilizando el menú de iconos que se observa en el recuadro rojo, el cual también posee todas las herramientas propias de cada módulo pero ya desplegadas en forma de iconos, lo cual facilita mucho el aprendizaje a los nuevos usuarios.

También podemos comunicarnos con el programa mediante el árbol del modelo, encuadrado en verde. Esta es quizás la forma más eficiente de trabajo ya que contiene replegados en las diferentes pestañas que lo componen todos los módulos y herramientas de cada módulo, de manera que permite visualizar el modelo de una manera más global, y usar herramientas de un módulo u otro de una manera más eficaz y rápida.

Finalmente, encuadrado en naranja, se observa la línea de comandos del programa. Aquí se pueden dar las mismas órdenes que en los casos anteriores pero escritos manualmente en un lenguaje de programación entendible por el programa (PYTHON).

Además en dicha ventana se obtienen mensajes de errores, órdenes de proceso, órdenes ejecutadas, etc, emitidas por el programa para la correcta comunicación usuario-programa.

Por lo que los pasos a seguir para resolver cualquier problema usando el preprocesador de Abaqus consiste en realizar un modelo introduciendo los datos de interés en cada uno de los módulos que conforman el programa y utilizando el gran abanico de posibilidades que aporta Abaqus para comunicarnos.

Los principales módulos son los siguientes:

- *Parts.*
- *Property.*
- *Assembly.*

- *Step.*
- *Interaction.*
- *Load.*
- *Mesh.*
- *Optimitation.*
- *Job*

De manera que una vez se introduzcan los datos de interés en cada uno de dichos módulos, se generará un archivo de entrada que Abaqus leerá, y si no hay ninguna incidencia, se puede proceder al análisis.

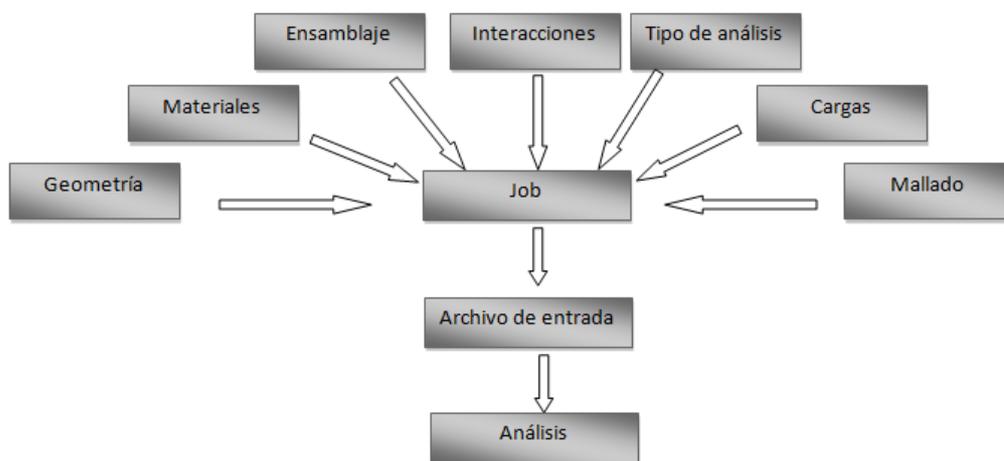


Ilustración 8. Esquema del procesado de datos de Abaqus.

3 INTRODUCCION AL ENTORNO DE PYTHON.

Python es un lenguaje de programación sencillo de aprender y potente. Presenta eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz, lo cual facilita mucho su aprendizaje.

Además usa una sintaxis muy sencilla y elegante, goza de una gestión de tipos dinámica y su naturaleza interpretada hace de él un lenguaje ideal para guiones (scripts) y desarrollo rápido de aplicaciones, en muchas áreas y en la mayoría de las plataformas.

Es por eso que cuando se inicio el proyecto y se vio la posibilidad de usar dicho lenguaje de programación debido a su compatibilidad con Abaqus, se decidió utilizar dicha potente herramienta para desarrollar el proyecto de una manera más eficiente.

Cabe destacar que el intérprete de Python y la extensa biblioteca estándar están disponible libremente y al alcance de todos, por lo que esto hace más sencillo su aprendizaje. Además se puede ampliar el intérprete de Python con nuevas funciones y tipos de datos implementados en C y C++, por lo que para el lector usuario habitual de dichos lenguajes, esta breve introducción puede ser ofrecer una gran posibilidad para iniciarse en el mundo de la programación en PYTHON así como al desarrollo de aplicaciones.

Además PYTHON presente una ventaja muy clara respecto a otros lenguajes de programación como pueden ser C o C++, y es que sus códigos son muy compactos y legibles. Ello se debe principalmente a varios motivos:

- Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola sentencia.
- El agrupamiento de sentencias se realiza mediante sangrado (indentación) en lugar de begin/end o llaves.
- No es necesario declarar los argumentos ni las variables.

Por lo que se obtiene aquí una razón más para iniciarse en el mundo de la programación en PYTHON.

3.1. Herramientas necesarias para la realización Del proyecto.

Puesto que el entorno de PYTHON es muy extenso y sus aplicaciones son demasiado amplias para tratarlas o introducirlas todas, se va a explicar a continuación las herramientas generales que han sido necesarias para la realización del proyecto, y posteriormente, se explicarán las herramientas particulares de Python en el entorno de Abaqus que se han utilizado.

Por ello se van a explicar principalmente el funcionamiento de Abaqus como calculadora, y la forma de programar algunas sentencias sencillas como son los bucles for y while o el condicional if.

3.1.1 Python Como calculadora.

En primer lugar cabe destacar que para obtener toda la funcionalidad de Python en el script que se desarrollará posteriormente para Abaqus, es conveniente importar todas las herramientas matemáticas que posee dicho lenguaje. Para ello se utiliza la siguiente línea de código:

```
Import math
```

Una vez a nuestra disposición dichas herramientas, se va a explicar cómo realizar operaciones sencillas con Python como son la asignación de variables, suma, resta, multiplicación, logaritmos, exponenciales, trigonometría etc.

3.1.1.1 Asignación de variables.

Al igual que en otros lenguajes de programación como C, la igualdad se utiliza para asignar un valor a una variable:

```
>>> Ah=0.005
>>>R=0.2
```

Y además se puede asignar un valor simultáneamente a varias variables:

```
>>> x1=x2=x3=0
```

O asignación múltiple de las mismas:

```
>>> a,b=1,2 ### Es igual que asignar por separado el valor a cada variable a=1;a=2.
```

Donde “#” se utiliza para introducir comentarios, es decir, cuando el programa lee el símbolo de almohadilla pasa a la **siguiente línea**.

3.1.1.2 Aritmética.

Para sumar, restar, multiplicar y dividir se utilizan los mismos operadores que en otros lenguajes: “+,-,*,/”. De igual forma se utilizan paréntesis para agrupar las operaciones:

```
>>> 2+2
4
>>>(Ah*R-2)/4
0.49975
```

Además cabe destacar que para rescatar el último resultado impreso en pantalla, y utilizarlo para un cálculo posterior, se utiliza la variable “_”, que en modo interactivo es donde Python almacena el valor.

```
>>>2+2
4
>>>_+3
7
```

Finalmente, hay que destacar que el operando usado por Python para el cálculo de las exponenciales difiere de otros lenguajes de programación, en los que se usa el operador “^”. En el entorno de Python se utiliza el operando “**”

```
>>>2**3
8
>>>3**3
27
```

Por lo que para calcular raíces, solo hay que recordar que estas se pueden poner de forma exponencial. Por ejemplo para calcular la raíz de 4

```
>>>4**1/2
2
```

3.1.1.3 Trigonometría.

A continuación se va a explicar cómo utilizar las funciones trigonométricas en Python dado que van a ser de gran utilidad en el desarrollo del proyecto. Como se ha descrito al principio del texto lo primero que hay que hacer para importar todas las funciones matemáticas de Python en un script es escribir la línea de código “Import Math”

```
Import Math
```

Las funciones seno, coseno, y tangente se declaran directamente como sin, cos y tan:

```
>>>sin(0.0)
0
>>>cos(0.0)
1
```

O bien desde un script como

```
math.cos(0.0)
1
```

Cabe destacar que Python espera que introduzcamos el valor de los ángulos en radianes, por lo que es de utilidad saber que se dispone de la constante pi

```
>>>math.pi
3.141592653589793

>>> math.sin(math.pi/3)
0.8660254037844386
```

Por otro lado, las funciones inversas se obtienen respectivamente como asin(), acos() y atan():

```
>>>math.acos(1)
0.0
```

También cabe destacar que, se puede pasar de radianes a grados de la siguiente forma:

```
>>>math.atan(1)
0.7853981633974483
>>>math.degrees(0.7853981633974483)
45.0
```

Y la operación inversa, es decir, pasar de grados a radianes con la función radians()

```
>>>Math.radians(90)
1.5707963267948966
```

Finalmente se acaba explicando cómo obtener la hipotenusa de un triángulo rectángulo, lo cual nos puede resultar de gran utilidad en el proyecto:

```
>>>Math.hypot(3,5)
5.0
```

3.1.2 Operaciones con vectores.

Las operaciones con vectores están englobadas dentro de un tipo de datos compuestos conocido como “listas”. Dicho tipo es el más versátil puesto que permite agrupar elementos de distinto tipo (números y letras).

La forma de definir una lista es mediante corchetes, con sus elementos separados con comas. Así que si se quiere crear un vector:

```
>>> a=[1,3,5,8]
```

Donde se puede acceder a sus componentes, así como cambiar alguno de ellos de la siguiente forma:

```
>>> a[0]
1
>>> a[3]
8
>>>a[1]=3
>>> print(a)
a=[3,3,5,8]
```

Finalmente se muestra como definir listas con diferentes tipos de elementos, así como el acceso y modificación de los mismos en este caso:

```
>>> b=["Proyecto",10,"TFM","Angel",[10,9,8]]
>>>b[0]
Proyecto
### Para acceder a un elemento del vector [10,9,8] se utiliza:
>>>b[[4][0]]
10
>>>b[[4][2]]
8
## y para modificar algún elemento análogamente al caso anterior
>>>b[0]="Rolling"
>>>Print (b)
b=["Rolling",10,"TFM","Angel",[10,9,8]]
```

3.1.3 Inicios a la programación.

3.1.3.1 Funciones.

Se van a introducir finalmente como definir funciones sencillas en Python, así como las principales y más conocidas herramientas de control de flujo.

Para definir funciones simples basta con usar las siguientes líneas de código:

```
Def FUNCION:  
Return RESULTADO DE LA FUNCIÓN
```

Con el siguiente ejemplo se ilustra de mejor manera:

```
Import math  
Def sen (grados):  
Return math.sin(math.radians(grados))  
>> sen (30)  
0.4999999999999994
```

Donde se ha creado una función para el seno que admita como argumento de entrada el valor del ángulo en grados.

En lo que respecta a las herramientas de control de flujo, se van a explicar con ejemplos las principales y más conocidas, que son las estructuras `while`, `for` e `if`.

3.1.3.2 While.

La estructura del bucle `while` es la que se muestra a continuación:

```
>>>b=1  
While b<10  
Print b  
b=b+1
```

Al igual que en otros lenguajes de programación, es necesario definir una variable contador, que es la que se va incrementando en cada bucle y la que nos indica cuando se sale del mismo. En este caso el bucle “while” se ejecuta mientras la condición sea cierta.

Los operadores de comparación estándar se escriben de la misma forma que en otros lenguajes de programación como C: `<` (menor que), `>` (mayor que), `==` (igual a), `<=` (menor o igual que), `>=` (mayor o igual que), y `!=` (diferente de).

3.1.3.3 For

La estructura del bucle `for` se puede resumir en:

```
For variable in elemento iterable (lista, cadena, range, etc.):  
Cuerpo del bucle
```

La construcción `for` es un poco diferente a lo acostumbrado en otros lenguajes de programación como C, MATLAB, etc. Ello se debe a que en lugar de recorrer una progresión aritmética o dejar libertad de inicialización, salto de paso, comprobación, etc, el `for` de Python recorre los elementos de una secuencia (como por ejemplo una lista) en el orden que aparecen en la secuencia.

Por lo demás la estructura es la misma, donde bloque de instrucciones que se repite se suele llamar cuerpo del bucle y cada repetición iteración.

Otra cosa a destacar es que, a diferencia de otros lenguajes, no es necesario definir la variable de control del bucle o inicializarla antes de definir este. A continuación se muestran algunos ejemplos para entender mejor el

funcionamiento:

```
>>>For i in [0,1,2]:
    Print ("Hola",end="")
    Print ()
Hola Hola Hola
```

```
>>> For i in [3,4,5]
    a=i
    b=i**2
    print(a,b)
3,9
4,16
5,25
```

3.1.3.4 if

La estructura del condicional if se va a ilustrar con el siguiente ejemplo:

```
>>> x=int(raw_input("introduce un numero:"))
If x<0
X=0
Printf 'Negativo cambiado a cero'
Elif x==0;
Print 'cero'
Elif x==1:
Print 'uno'
Else print 'Más'
```

Cabe destacar que puede haber ninguna, o más partes elif, que es una abreviatura de else if usado en otros lenguajes de programación.

Además la sentencia else (si no) es opcional.

4 DEFINICIÓN DEL MODELO EN ABAQUS Y DEL PROBLEMA.

El objetivo de este capítulo es que una vez introducida la teoría necesaria referente al proceso de laminación, conocidos los parámetros del mismo que se quieren hallar, y habiéndonos familiarizado con el entorno de ABAQUS y PYTHON, definir el estudio paramétrico que se va a llevar a cabo, definiendo los distintos parámetros a barrer así como la evolución del modelo.

Se van a realizar dos modelos diferentes: uno de ellos para estimar el coeficiente de arrastre natural, y otro para el de arrastre forzado. Sin embargo los modelos son muy similares por lo que se explicará en primer lugar el correspondiente con la estimación del coeficiente de arrastre natural, y posteriormente se ilustrarán las principales diferencias que presenta con el de arrastre forzado.

En cuanto a los parámetros que se van a estudiar, van a ser aquellos que influyan en los coeficientes de rozamientos que determinan el arrastre natural y arrastre forzado:

$$\text{- Arrastre natural: } \mu = \tan\theta = \sqrt{\frac{\Delta h}{R}} = \sqrt{\frac{h_0 - h_f}{R}} \quad (4 - 1)$$

$$\text{- Arrastre forzado: } \mu = \tan\left(\frac{\theta}{2}\right) = \tan\left(\tan^{-1}\left(\sqrt{\frac{\Delta h}{R}}\right) / 2\right) \quad (4 - 2)$$

Por lo que como se puede observar, los parámetros implicados son h_0 , h_f , y R . Por otro lado, también se cree conveniente estudiar el problema para distintos tipos de material y observar si se producen variaciones en los resultados o no para los mismos tipos de geometría. Los materiales que se van a estudiar son Aluminio 70-75 y Acero al carbono dulce, de cuyas propiedades nos interesa principalmente densidad, las propiedades elásticas (Modulo de elasticidad, coeficiente de Poisson, y límite elástico), y plásticas (Ley de comportamiento plástica).

4.1. Materiales

Cabe destacar que para ambos materiales se va a definir una ley de comportamiento plástica parabólica, que tiene la siguiente forma:

$$\sigma_p = \sigma_e + K\epsilon_p^n \quad (4 - 3)$$

Donde:

- K = Coeficiente de resistencia.
- n = Coeficiente de endurecimiento.
- σ_p = Tensión plástica.
- σ_e =Límite elástico.
- ϵ_p =Deformación plástica.

Finalmente destacar que para obtener las tensiones plásticas es necesario introducir unos valores de deformaciones plásticas, los cual se va a hacer a incrementos de un 5% hasta llegar a una deformación plástica $\epsilon = 1$ (equivalente a una deformación del 100%).

A continuación se definen los valores utilizados para cada material.

- Aluminio 70-75.
 - $\rho = 2810 \text{ kg/m}^3$
 - $E = 72 \text{ Gpa}$
 - $\nu = 0.33$
 - $K = 318.96 \text{ Mpa}$
 - $n = 0.227$
 - $\sigma_e = 418.11 \text{ Mpa}$
 - $\epsilon_p = [0:0.05:1]$
 - $\sigma_p = 418.11 + 318.96 * \epsilon_p^{0.227} \text{ Mpa}$

- Acero al carbono
 - $\rho = 7850 \text{ kg/m}^3$
 - $E = 210 \text{ Gpa}$
 - $\nu = 0.3$
 - $K = 600 \text{ Mpa}$
 - $n = 0.21$
 - $\sigma_e = 400 \text{ Mpa}$
 - $\epsilon_p = [0:0.05:1]$
 - $\sigma_p = 400 + 600 * \epsilon_p^{0.21} \text{ Mpa}$

4.2. Parámetros

Los parámetros que se van a variar son altura inicial de la placa, altura final, y radio del rodillo para cada uno de los materiales descritos. El procedimiento a seguir va a consistir en fijar radio de rodillo y altura inicial, y variar la altura final a incrementos de 12.5 % de 0 a 100 de manera que se obtengan configuraciones de deformación homogénea y no homogénea. Dicho procedimiento se va a llevar a cabo para los distintos radios y alturas iniciales que se van a analizar.

En cuanto a los rodillos se van a modelar de la manera más realista posible, por lo que se considerarán radios de 0.2 y 0.5 metros.

Finalmente se van a laminar placas de tres alturas iniciales de 1, 4, y 8 centímetros. Por lo que el estudio paramétrico que se va a realizar queda de la siguiente manera:

Material	Radio [m]	h0 [m]	Hf [m]
Aluminio 70-75	0.2	0.01	[0.125:0.125:0.875] * h ₀
		0.04	
		0.08	
	0.5	0.01	
		0.04	
		0.08	
Acero al carbono	0.2	0.01	[0.125:0.125:0.875] * h ₀
		0.04	
		0.08	
	0.5	0.01	
		0.04	
		0.08	

Tabla 1. Estudio Paramétrico a realizar.

Es decir, para cada combinación de material, Radio, y h_0 , se obtendrá el coeficiente de rozamiento que da lugar al arrastre natural y forzado para los siguientes valores de h_f :

- $0.125 * h_0$
- $0.25 * h_0$
- $0.375 * h_0$
- $0.5 * h_0$
- $0.625 * h_0$
- $0.75 * h_0$
- $0.875 * h_0$

De manera que se van a realizar un total de 168 análisis (la mitad para estimar el coeficiente de arrastre natural y la otra mitad para el forzado). Dicho esto se puede observar la importancia que tiene simplificar el modelo lo máximo posible ya que cada análisis con la malla definitiva tarda una media de 1-2 horas como se mostrará más adelante.

4.3. Modelo.

Se va explicar a continuación como se ha llevado a cabo el desarrollo del modelo numérico en el programa de elementos finitos ABAQUS/CAE.

Puesto que los fenómenos que se van a estudiar (arrastre natural y arrastre forzado) difieren únicamente en la geometría y posición inicial de la placa de material que se va a extruir, se va a mostrar cómo se ha llevado a cabo el modelo de arrastre natural, y posteriormente se mostrarán las diferencias que presente respecto a este el de arrastre forzado.

Cabe destacar que en este capítulo no se va a entrar en detalle en el diseño del modelo puesto que este aparece reflejado en el ANEXO 1 del texto.

4.3.1. Arrastre natural.

Como se explico en el capítulo 1, el proceso de laminación es un sencillo proceso de fabricación que consiste en extruir placas de material haciéndolas pasar por rodillos que giran a una cierta velocidad angular.

Es por eso que se trató de modelarlo al máximo detalle y con el menor número de hipótesis y simplificaciones, salvo la de modelar el problema en dos dimensiones debido a que la teoría del proceso de laminación admite dicha hipótesis obteniéndose buenos resultados. Además debido a la complejidad computacional que supondría hacer el modelo en tres dimensiones, dicha opción no tendría cabida en un proyecto del ámbito que se está realizando.

Todo esto dio lugar a que el primer modelo que se realizara constara de dos rodillos perfectamente modelados, y una placa de material h_0 que se introduciría entre dichos rodillos.

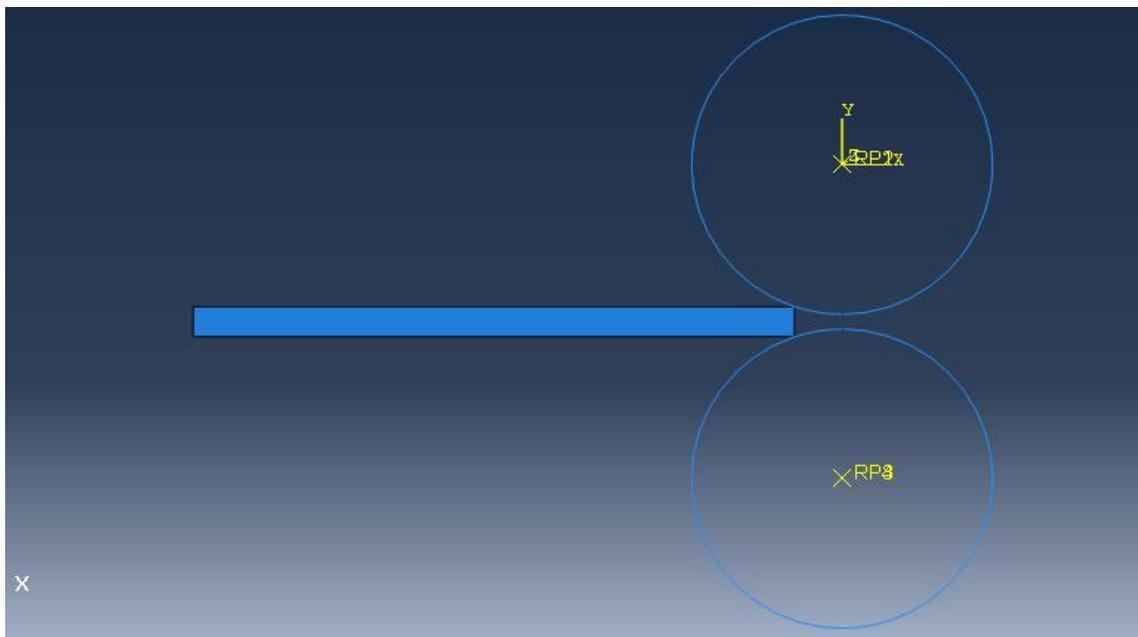


Ilustración 9. Modelo completo con dos rodillos.

Para crear dicho modelo se creó un *Part deformable*, que es la placa de material, y dos rodillos formados por 5 part de tipo *analytical Rigid* cada uno, tres de 90 grados, y el que antes entra en contacto con la placa que se dividía a su vez en dos arcos unidos en el punto de contacto en el que la teoría de laminación afirma que se produce el arrastre natural. La elección de geometría de tipo *analytical Rigid* se debe a que este tipo de geometría es rígida, y puesto que no nos interesan las tensiones que aparecen en los rodillos, era el tipo que mejor encajaba con nuestro modelo. No se utilizó el tipo *Discrete Rigid* puesto que ese tipo de geometría está destinado a análisis **implícitos**, y como se explicará más adelante, en este proyecto se va a realizar un análisis de tipo dinámico explícito.

Una vez definida la geometría del modelo y las características del material y sección que conforma la placa de material que se va a extruir, se ensambló el modelo.

Para ello se trasladó la placa de material de manera que sus esquinas coincidieran con el punto de transición entre los dos arcos que conforman el primer arco de 90°.

Llegados a este punto se procede a crear el caso de estudio que se va llevar a cabo. Debido a que el proceso de laminación es un proceso dinámico, se va a realizar un análisis de tipo *dinamic explicit* en lugar de uno estático.

Sin embargo, las bajas velocidades que caracterizan el modelo (~1m/s), hacen que las inercias características

del proceso no adquieran valores importantes, por lo que este se puede considerar como un proceso quasi-estático. Ello abre la posibilidad de realizar el análisis de manera estática, o de usar otras herramientas propias del análisis explícito como lo son el escalado en tiempo o masa, que se explican en el ANEXO 2. Sin embargo para el estudio en cuestión se ha decidido realizar un análisis dinámico de tipo explícito y no se han usado dichas simplificaciones nombradas anteriormente.

Una vez creado el caso de estudio, se crean las interacciones entre rodillos y placa de material, así como las condiciones de contorno de cada elemento que compone el modelo:

Elemento	Condicion de contorno
Arcos que componen el rodillo	$U1=U2=0$; $VR3=6.28$ rad/s
Placa	$V_{inicial}=1.2$ m/s

Tabla 2. Condiciones de contorno.

Donde la condición inicial de velocidad de la placa se introdujo para reducir el salto en el campo de velocidades que sufre la placa de material al entrar entre los rodillos del material, comprobándose previamente que no afectaba a los resultados y que reducía el coste computacional del modelo.

Cabe destacar que puesto que hay un total de diez arcos de circunferencia conformando los dos rodillos, hay que aplicar un total de 20 condiciones de contorno para restringir y simular el movimiento de los mismos, lo que supone una carga computacional elevada.

Es por ello que tras realizar el mallado del modelo y comprobar que reflejaba correctamente lo que postula la teoría de laminación, se procedió a simplificar el modelo.

La primera simplificación que se llevó a cabo es la de aplicar una condición de simetría en nuestro modelo que nos permitiera obtener la mitad de interacciones, condiciones de contorno, y de elementos en el mallado.

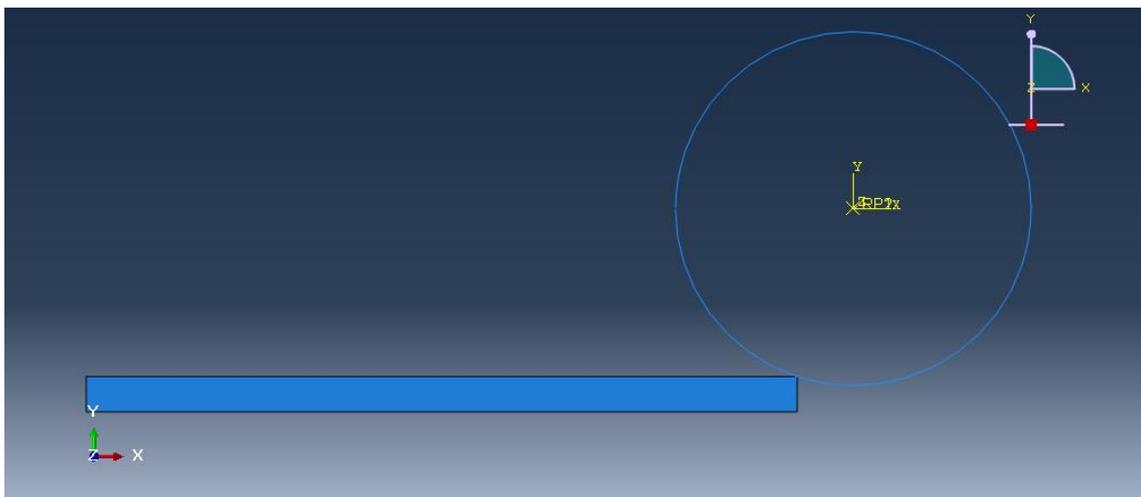


Ilustración 10. Modelo con simetría.

Se comprobaron los resultados y eran análogos a los obtenidos con el modelo completo (como era de esperar).

Tras varios análisis y comprobando que el coste computacional era aun elevado, se llevaron a cabo dos simplificaciones más:

1. Acortar las placas de material dado que realmente solo nos interesa demostrar que estas pasaran entre los rodillos sin aplicar ninguna fuerza cuando ese den las condiciones de arrastre natural. Finalmente se escogió una longitud igual a $1.2 \cdot L$, siendo L la longitud característica de contacto.
2. Debido al tamaño que tienen las placas tras la simplificación propuesta en el punto 1, la placa de material entra entre los rodillos sin que estos giren más de noventa grados. Por ello se decidió simplificar aún más el modelo, y se va a usar para realizar el análisis paramétrico un solo arco de 90 grados compuesto por dos arcos unidos en el punto en el que debe entrar en contacto la plancha de material para que se dé la condición de arrastre natural:

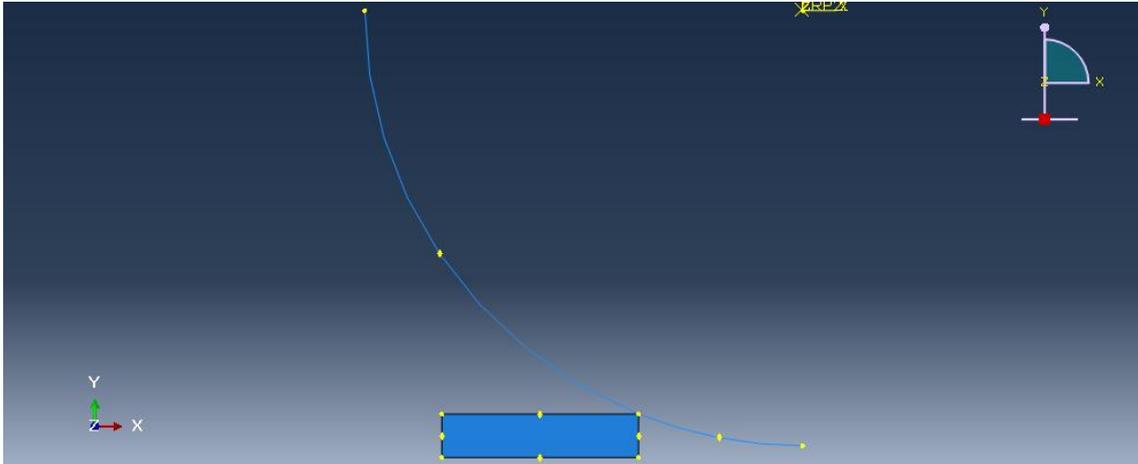


Ilustración 11. Modelo con simetría y un solo arco de 90°.

Finalmente se presente un diagrama de flujo donde se muestra la evolución que ha seguido el modelo desde su concepción hasta su refinado final.

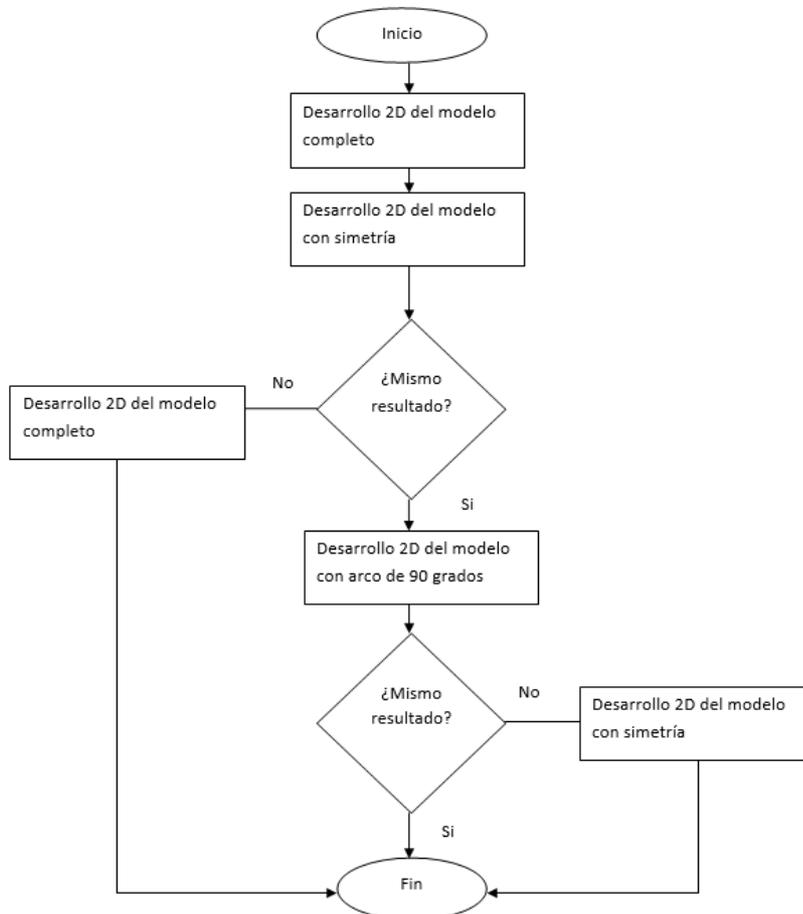


Ilustración 12. Algoritmo de la evolución del modelo.

Cabe destacar que existe otro diseño para el que se obtienen los mismos resultados que para el resultante tras el proceso de optimización desarrollado en los párrafos anteriores.

Este consiste en modelar el rodillo con un solo arco de 90 grados, y en lugar de dividir dicho arco en dos como se hizo anteriormente para definir el punto de contacto, usar una herramienta de ABAQUS denominada *Datum Point*, que permite crear sobre un eje un punto dadas ciertas coordenadas o parámetros, y el cual no forma realmente parte de la geometría, es decir, no lleva asociado propiedades y además no requiere de condiciones de contorno.

En el **ANEXO 1** se explicará cómo se diseñan ambos modelos, (arco de 90 grados dividido en dos o arco de 90 grados con Datum point), y se profundizará mucho más en el proceso llevado a cabo paso por paso. Esto es debido a que se van a usar ambos modelos para comprobar la fiabilidad de los resultados obtenidos en el proyecto de investigación que se está llevando a cabo.

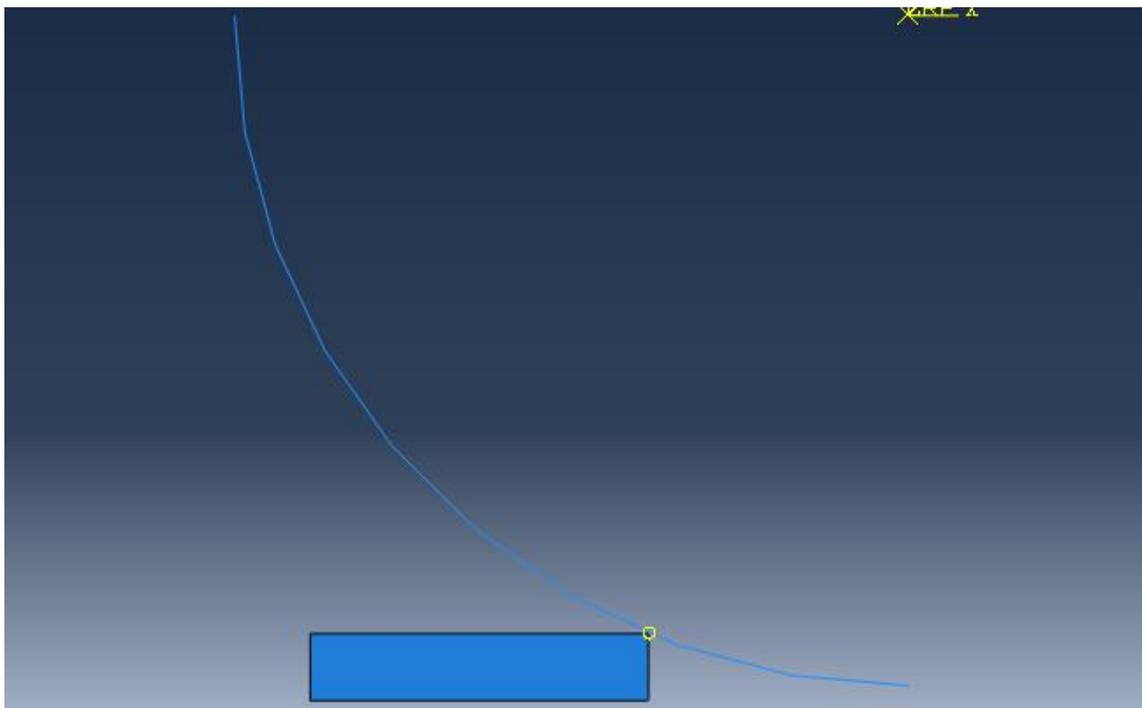


Ilustración 13. Modelo definitivo con Datum Point

4.3.2. Mallado del modelo.

Se procede a continuación a explicar qué mallado se ha llevado a cabo y el proceso de optimización que ha sufrido el mismo. Se va a dedicar una sección de la memoria a dicho apartado puesto que es uno de los puntos más críticos en lo que respecta a la consecución de resultados los más exactos posibles. Para ello hay que tener en cuenta algunos aspectos del proceso de laminación que han determinado los tipos de malla que se van a utilizar.

El proceso de laminación es un proceso de fabricación en el que tienen lugar grandes deformaciones en los materiales que se van a extruir. Además, a pesar de que la velocidad del proceso no es excesivamente elevada, y que se podría considerar un proceso cuasiestático debido al efecto que tienen las inercias en el mismo, en este trabajo se ha optado por no despreciar dichas inercias y realizar un caso de estudio dinámico.

Otro aspecto a tener en cuenta es que en el proceso de laminación se produce el contacto entre dos superficies, produciéndose un impacto inicial, que aunque de poca intensidad e importancia, tiene gran influencia en el tipo de mallado a utilizar. De hecho, la ayuda de ABAQUS aconseja utilizar un tipo de malla llamado adaptativo para dichos casos. ¿Y en que consiste un mallado adaptativo? Dicho tipo de mallado consiste simplemente en un remallado continuo de la malla fijado principalmente cada cierto intervalo de tiempo.

Por lo que recordando que se ha mencionado que el proceso que se está estudiando es característico por las grandes deformaciones sufridas en el material, parece ser que dicho tipo de mallado es el más conveniente, dado que si solo mallamos la placa de estudio al inicio del análisis del caso, quizás la geometría sufra modificaciones o deformaciones que dicho tipo de malla no sea capaz de reflejar. Sin embargo si se produce un remallado cada cierto número de iteraciones sobre la geometría ya deformada, los elementos de dicha malla reflejarán y se adaptarán mucho mejor a lo que le está sucediendo a la placa en realidad.

Una vez escogido el mallado adaptativo para nuestro modelo, había que pensar que está sucediendo en el modelo en lo que respecta a tensiones y deformaciones para escoger un tipo de elemento que se adapte bien al caso de estudio. Por lo que teniendo en cuenta que en el **capítulo 1** se explicó que el proceso de laminación era característico por tener un estado de deformación plana, se van a escoger elementos cuadrados de deformación plana, que se corresponde con el elemento tipo CPE4R (con elementos cuadrados) y CPE3 (triangulares).

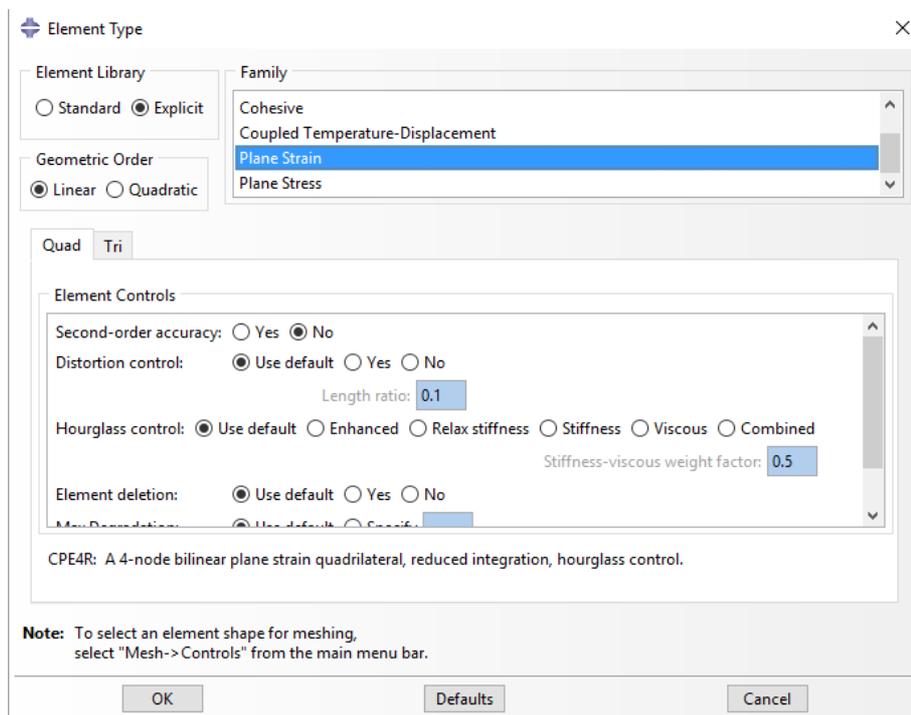


Ilustración 14. Hipótesis deformación plana en Abaqus.

Finalmente, para eliminar la posibilidad de que los elementos se distorsionen más de lo deseado y por lo tanto que la masa asociada a los mismos sea prácticamente nula, se debería coger una malla estructurada en lugar de libre. Sin embargo el mallado adaptativo solo es aplicable para mallas de distribución libre, por lo que se va a escoger dicha estructura.

Una vez definidos el tipo de malla, organización de la misma, y tipo de elementos que la componen, es hora de proceder a determinar el número de elementos de la malla así como su distribución y concentración en la geometría de las placas que se quieren laminar.

Puesto que la geometrías que se van a estudiar tienen un espesor muy pequeño (10,40 y 80 milímetros), se ha optado por escoger un número de elementos que garantice una discretización tal que permita obtener resultados en desplazamientos del orden de magnitud de $1E-3$, es decir, los elementos tendrán como máximo una longitud de un milímetro.

Por lo que teniendo en cuenta la condición de simetría, la semilla de mallado de partida para cada tipo de geometría vendrá dada por:

$$N^{\circ}elementos = (h_0/2) * 1000$$

Donde el número de elementos se definen a lo largo de la altura, y tomando el tamaño asociado a dicho elemento

se asigna a lo largo de la longitud de la placa.

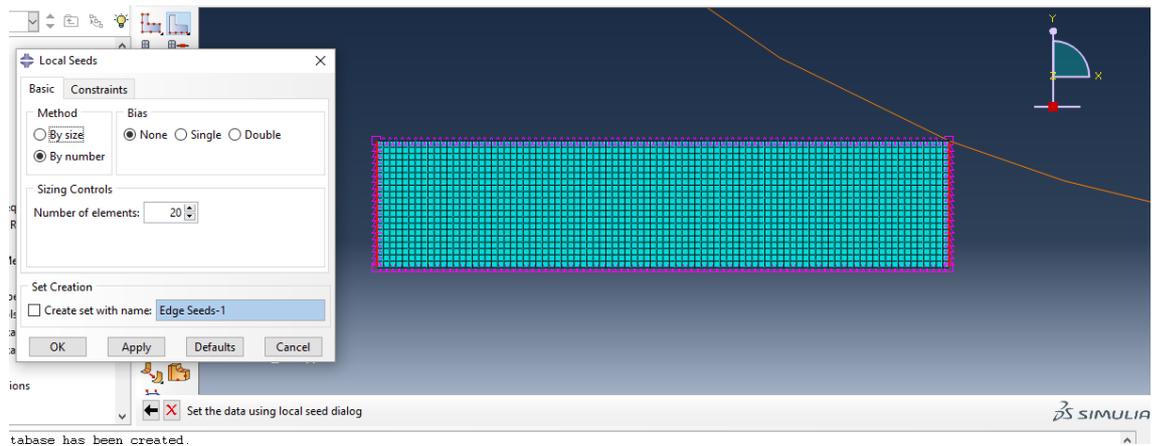


Ilustración 15. Mallado del modelo.

En cuanto a la fiabilidad de los resultados obtenidos, se ha llevado a cabo el análisis de sensibilidad de mallado que se muestra en el diagrama de flujo de la **figura 16**, el cual además muestra el proceso seguido para obtener el valor del coeficiente de arrastre natural del proceso de laminación.

Ambos procesos mencionados van de la mano puesto que lo que se quiere demostrar es que el valor del coeficiente de rozamiento para el que se produce el arrastre natural en el proceso de laminación es el mismo sea cual sea el mallado que se utilice, lo cual otorga una gran fiabilidad a los resultados que se obtengan de dicho análisis.

Por ello el proceso seguido ha consistido básicamente en plantear una malla, y obtener el valor del coeficiente de rozamiento necesario para que se de el fenómeno de arrastre natural partiendo del valor teórico. Una vez obtenido, se aumenta el número de elementos de la malla, y se comprueba que sigue cumpliendo dicho valor. Si no se cumple, se vuelve al principio pero esta vez con una malla más discreta, y se repite el procedimiento de estimación del coeficiente. Si se cumple, se sigue aumentando el número de elementos de la malla y comprobando la veracidad del valor obtenido hasta que se considere oportuno.

Cabe destacar que en el trabajo realizado se ha decidido realizar mallas de cómo máximo 5000 elementos. Ello se debe a que el coste computacional de realizar un análisis explícito con Abaqus es elevado, y a ello hay que sumar que la potencia del ordenador con el que se ha trabajado no es la mejor de las posibles.

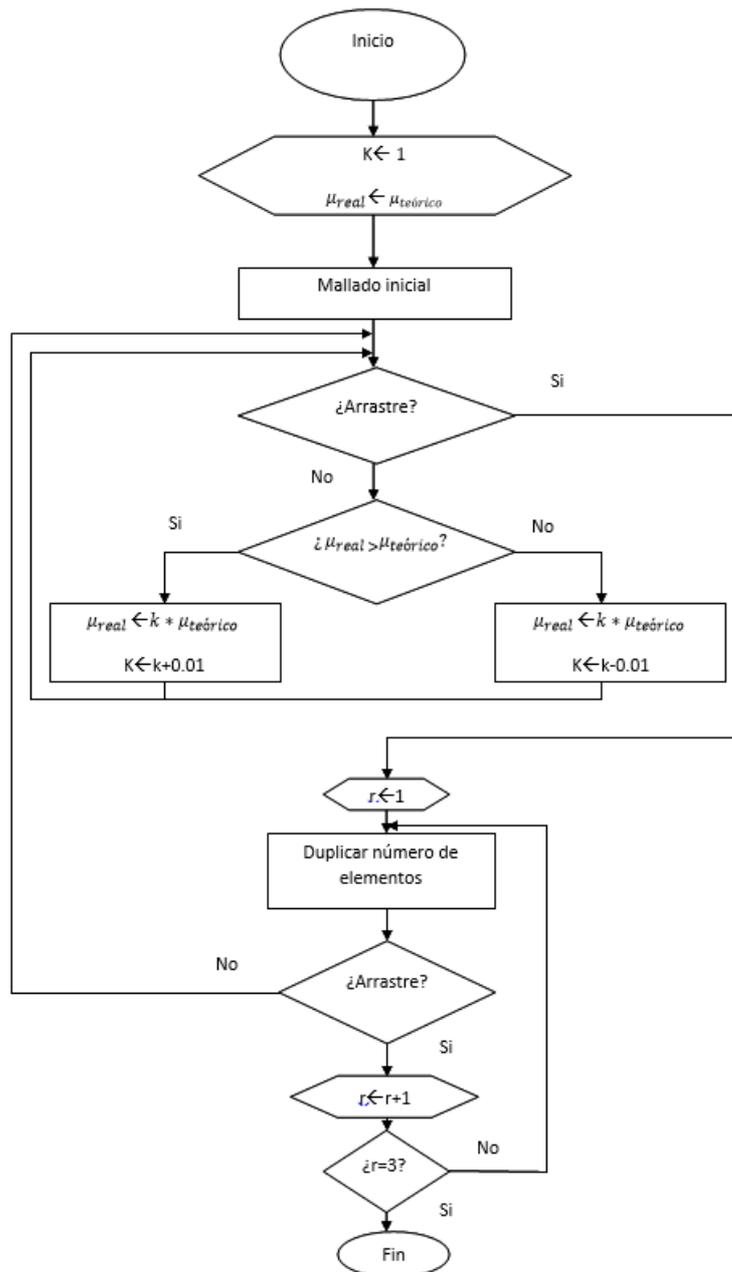


Ilustración 16. Diagrama de flujo del método utilizado para la obtención del rozamiento.

4.3.3. Modelo de arrastre forzado. Diferencias principales.

En lo que respecta al modelo realizado para simular las condiciones de arrastre forzado, la única diferencia respecto al modelo anterior es la geometría inicial.

Ello se debe a que la condición de arrastre forzado se da cuando la placa se ha introducido entre los rodillos ya sea aplicando una fuerza inicial que la obligue a ello, debido a mordida de las placas., etc.

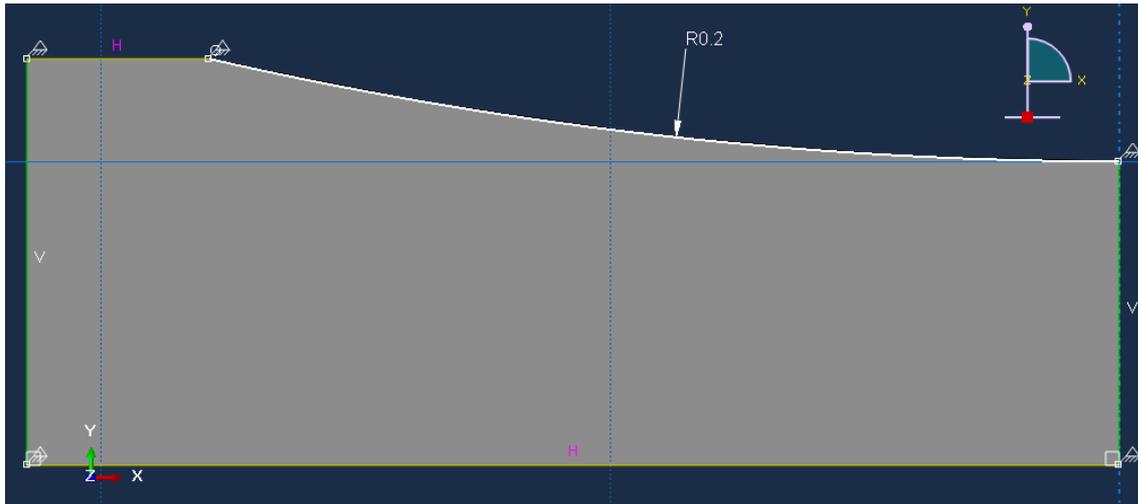


Ilustración 17. Modelo de Arrastre Forzado.

Por lo que la principal diferencia es la geometría de partida, que como se puede observar en esta ocasión no es un rectángulo, sino que en este caso tenemos en la cara superior un arco de circunferencia que se corresponde con el rodillo, y el cual viene definido por las longitudes características del proceso de laminación L y $\Delta h/2$.

Por otro lado debido a dicho cambio en la geometría, cambia la definición de la superficie “top” que se definirá posteriormente, ya que en ese caso está formada por el arco y el segmento horizontal que unido a el conforman la cara superior (En lugar de estar formado por la cara superior y la lateral derecha como en el caso del arrastre natural).

Finalmente destacar que en el **ANEXO A** se mostrará en más detalle dicho modelo.

5. MODELO EN PYTHON.

Una vez explicado todo lo referente al proceso de fabricación que se va a estudiar, y a los modelos y casos de estudio que se van a llevar a cabo con ABAQUS, es hora de introducir el código en PYTHON que se ha usado para realizar el estudio paramétrico que se pretende realizar con el presente proyecto.

En primer lugar destacar que la idea de usar PYTHON viene dada por el hecho de que se van a variar 4 parámetros característicos del proceso de laminación, dando lugar a un total de 168 casos de estudio. El proceso llevado a cabo para modelar el problema directamente con ABAQUS, así como su refinamiento y optimización ha sido costoso y duradero, ya que no se tenían nociones acerca del uso de dicho programa anteriormente.

Es cierto que una vez realizado este modelo por primera vez, se puede repetir en cuestión de algo más de media hora, por lo que parece un tiempo asumible. Sin embargo se vio la oportunidad de estudiar una nueva herramienta que permitiera realizarlo de manera automática solo introduciendo algunos parámetros de entrada, y por eso se decidió volver a realizar los modelos desarrollados, pero esta vez usando el lenguaje de programación PYTHON.

La programación con PYTHON es una herramienta muy poderosa, ya que no solo nos permite resolver problemas del tipo que se está tratando, en la que todos los parámetros están relacionados por lo que permiten una parametrización no muy compleja, sino que además permiten combinar la gran funcionalidad del entorno gráfico de ABAQUS con la gran potencia que posee el lenguaje de programación PYTHON.

A continuación se va a explicar de manera general y también aplicada a nuestro modelo, los siguientes aspectos:

- Crear un Model Database (mdb)
- Crear un Output Database (odb).

Además se van a explicar paso por paso en el mismo orden en el que se han llevado a cabo los modelos usando el interfaz gráfico de ABAQUS (Se puede observar en el ANEXO 1).

5.1. Crear un Model Database (mdb)

Puesto que el entorno de PYTHON no contiene por defecto toda la funcionalidad de la que goza ABAQUS, lo primero que se debe hacer es importar todas las herramientas y módulos de abaqus que se vayan a usar mediante las siguientes líneas de código:

```
from part import *
from material import *
from section import *
from optimization import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
```

```
import math
```

Donde la última secuencia de caracteres sirve para activar todas las operaciones matemáticas que PYTHON permite realizar y que serán de gran utilidad a la hora de definir algunas características del modelo.

A continuación se va seguir paso por paso el procedimiento seguido en el anexo 1 para la realización del modelo:

1. **Crear un Part:** 2D plano, deformable, tipo Shell, y de tamaño aproximado 0.5 m. Crear un Sketch rectangular a través de punto inicial y punto final.

```
mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
sheetSize=4.0)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(x0, y0),
point2=(x6, y6))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='plate',
type=
DEFORMABLE_BODY)
mdb.models['Model-1'].parts['plate'].BaseShell(sketch=
mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
```

Como se puede observar la estructura de las líneas de código de Python dentro del entorno de Abaqus son siempre iguales:

```
Mdb.models['Nombre del modelo'].Módulo o entidad a la que pertenece la
operación a realizar ['nombre asociado a la entidad'].Operación(argumentos
de entrada necesarios).
```

En el caso en que sea una operación directa o independiente de un módulo u objeto en cuestión se usa:

```
Mdb.models['Nombre del modelo'].Operación(Argumentos de entrada).
```

Destacar que entre los argumentos de entrada es típico encontrarse el argumento “Type”, e cual te permite escoger entre las distintas opciones que se escoge manualmente en el interaz gráfico de Abaqus.

2. **Crear un Part:** 2D plano, analítico rígido, Tipo Shell, y de tamaño aproximado 0.5 m. Crear un Sketch: Arco a través de punto central, inicial y final.

```
mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
sheetSize=4.0)
mdb.models['Model-
1'].sketches['__profile__'].ArcByCenterEnds(center=(0.0, 0.0)
, direction=CLOCKWISE, point1=(0.0, -R), point2=(-R, 0))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='roller1',
type=
ANALYTIC_RIGID_SURFACE)
mdb.models['Model-1'].parts['roller1'].AnalyticRigidSurf2DPlanar(sketch=
```

```

mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']

```

3. Crear Material. Lineal elásto-plástico.

Al igual que se hizo con el preprocesador gráfico, hay que crear el material definiendo la densidad, propiedades elásticas, y propiedades plásticas.

```

mdb.models['Model-1'].Material(name='steal')
mdb.models['Model-1'].materials['steal'].Density(table=((rho, ), ))
mdb.models['Model-1'].materials['steal'].Elastic(table=((E, nu),
))
mdb.models['Model-1'].materials['steal'].Plastic(table=((o1, e1), (
o2,e2), (o3, e3), (o4,e4), (o5,
e5), (o6, e6), (o7, e7), (o8, e8), (
o9, e9), (o10,e10), (o11, e11),(o12,e12), (o13, e13), (o14,e14), (o15,
e15), (o16, e16), (o17, e17), (o18, e18), (
o19, e19), (o20,e20), (o21, e21)))

```

4. Crear Sección. Solida, homogénea.

Para crear una sección y asociarla a la geometría se procede de la siguiente manera:

```

mdb.models['Model-1'].HomogeneousSolidSection(material='steal', name=
'Section-1', thickness=None)
mdb.models['Model-1'].parts['plate'].Set(faces=
mdb.models['Model-1'].parts['plate'].faces.getSequenceFromMask(['#1  ]',
),
), name='Set-1')

mdb.models['Model-1'].parts['plate'].SectionAssignment(offset=0.0,
offsetField=
', offsetType=MIDDLE_SURFACE, region=
mdb.models['Model-1'].parts['plate'].sets['Set-1'], sectionName='Section-
1'
, thicknessAssignment=FROM_SECTION)

```

Donde se puede observar que la asociación se hace indicando a que Part (plate) se le quiere otorgar dicha sección.

5. Crear Sets y superficies

En primer lugar se van a crear distintos *Sets*, herramienta que consiste asociar un nombre a distintas partes de la geometría de manera que se facilite la asociación de condiciones de contornos u otras opciones a dichas partes.

```
mdb.models['Model-1'].parts['plate'].Set(edges=
mdb.models['Model-1'].parts['plate'].edges.getSequenceFromMask(['#1  '],
),
), name='top')
mdb.models['Model-1'].parts['plate'].Set(edges=
mdb.models['Model-1'].parts['plate'].edges.getSequenceFromMask(['#4  '],
),
), name='bottom')
mdb.models['Model-1'].parts['plate'].Set(edges=
mdb.models['Model-1'].parts['plate'].edges.getSequenceFromMask(['#2  '],
),
), name='left')
mdb.models['Model-1'].parts['roller1'].ReferencePoint(point=
mdb.models['Model-1'].parts['roller1'].InterestingPoint(
mdb.models['Model-1'].parts['roller1'].edges[0], CENTER))
mdb.models['Model-1'].parts['roller1'].Set(name='refpt1',
referencePoints=(
mdb.models['Model-1'].parts['roller1'].referencePoints[2], ))
```

Se puede observar en las últimas líneas de comando que se ha creado un *referent Point*, que servirá de base para aplicar las condiciones de contorno del modelo, y al cual se le ha asociado un *Set*.

A continuación ahora se crean las superficies:

```
#mdb.models['Model-1'].parts['plate'].Surface(name='top', #side1Edges=
#mdb.models['Model-1'].parts['plate'].edges.getSequenceFromMask(['#1
'], ),
#))
mdb.models['Model-1'].parts['roller'].Surface(name='roller', side1Edges=
    mdb.models['Model-1'].parts['roller'].edges.getSequenceFromMask(['#9
'],
    ), ))
```

En este punto cabe destacar que existe una forma alternativa a la hora de seleccionar la geometría a la cual se quiere asignar una superficie o un set, y dicha alternativa viene dada por el comando *Findat*:

```
mdb.models['Model-1'].parts['plate'].Surface(name='top', side1Edges=
mdb.models['Model-1'].parts['plate'].edges.findAt(((x,y,0.0),)))
```

Como se puede observar la principal diferencia radica en que el comando *getSequenceFromMask* identifica el

eje al que se quiere asignar una superficie o *Set* mediante un sistema numérico asignado por abaqus (Por ejemplo, para un cuadrado, cada lado tendrá un número del 1 al 4, y el cuadrado al completo el número 5), y el comando *FindAt* busca el eje en el punto que se introduce como argumento.

6. Assembly.

En primer lugar se crea el Datum Point que servirá para definir el contacto entre rodillo y placa, donde como se puede observar se piden como argumentos el eje en el que se quiere crear y el valor parametrizado que nos da su posición.

```
mdb.models['Model-1'].rootAssembly.DatumPointByEdgeParam(edge=
    mdb.models['Model-1'].rootAssembly.instances['roller1-1'].edges[0],
    parameter=datapoint)
```

Por lo que ya se puede crear el conjunto, introduciendo los dos Parts implicados, cuyo posicionamiento y contacto en el Datum Point se hará de forma manual.

```
mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='plate-1',
part=
mdb.models['Model-1'].parts['plate'])
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='roller1-
1',
part=mdb.models['Model-1'].parts['roller1'])
```

7. Step.

Con las siguientes líneas de código se crea el *Step* dinámico explícito, así como las variables que se quieren obtener del mismo, donde como se puede observar se puede escoger fácilmente a que parte del modelo van a ir asociados (*Sets*), el número de puntos en los que se quieren representar, etc. Cabe destacar que se han creado dos tipos de historiales de resultados: *Field Output* y *History Output*.

```
mdb.models['Model-1'].ExplicitDynamicsStep(name='Step-1',
previous='Initial',
    timePeriod=0.1)

mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(numIntervals=
    100)

mdb.models['Model-1'].historyOutputRequests['H-Output-1'].setValues(
numIntervals=100, rebar=EXCLUDE, region=
mdb.models['Model-1'].rootAssembly.instances['roller1-1'].sets['refpt1'],
sectionPoints=DEFAULT, variables=('RF1', 'RF2', 'RF3', 'RM1', 'RM2',
'RM3',
```

```
'RT', 'RM', 'RWM', 'CF1', 'CF2', 'CF3', 'CM1', 'CM2', 'CM3', 'SF1', 'SF2',
'SF3', 'SM1', 'SM2', 'SM3', 'NFORC', 'NFORCSO', 'RBFOR', 'ALLAE', 'ALLCD',
'ALLDMD', 'ALLFD', 'ALLIE', 'ALLKE', 'ALLPD', 'ALLSE', 'ALLVD', 'ALLWK',
'ETOTAL'))
```

8. Contacto e interacciones.

Este es el apartado más importante del modelo desarrollado en Python puesto que es donde se define la propiedad asociada al rozamiento, que es en definitiva, el parámetro que se quiere obtener con el proyecto.

Los argumentos principales de entrada para crear dicha propiedad son el tipo de contacto, la formulación que se va a escoger para definirlo, y según sea esta, el valor del rozamiento que se va a introducir en el modelo.

```
mdb.models['Model-1'].ContactProperty('IntProp-1')
mdb.models['Model-1'].interactionProperties['IntProp-1'].TangentialBehavior(
dependencies=0, directionality=ISOTROPIC, elasticSlipStiffness=None,
formulation=PENALTY, fraction=0.005, maximumElasticSlip=FRACTION,
pressureDependency=OFF, shearStressLimit=None, slipRateDependency=OFF,
table=((mu, ), ), temperatureDependency=OFF)
```

En lo que respecta a las interacciones, los argumentos son el step en el que se va a crear, la propiedad que se quiere asociar a dicha interacción, así como las superficies implicadas.

```
mdb.models['Model-1'].SurfaceToSurfaceContactExp(clearanceRegion=None,
createStepName='Step-1', datumAxis=None, initialClearance=OMIT,
interactionProperty='IntProp-1', master=
mdb.models['Model-1'].rootAssembly-surfaces['roller1'],
mechanicalConstraint=KINEMATIC, name='Int-1', slave=
mdb.models['Model-1'].rootAssembly-surfaces['top'], sliding=FINITE)
```

10. Condiciones de contorno.

Las condiciones de contorno que se van a aplicar son en desplazamientos, y velocidad principalmente, y la forma de programarlas es muy sencilla e intuitiva:

- Fijar desplazamiento.

```
mdb.models['Model-1'].DisplacementBC(amplitude=UNSET,
createStepName='Initial',
distributionType=UNIFORM, fieldName='', localCsys=None, name='fixroll1',
region=
mdb.models['Model-1'].rootAssembly.instances['roller1-1'].sets['refpt1'],
u1=SET, u2=SET, ur3=UNSET)
```

- Velocidades.

```
mdb.models['Model-1'].VelocityBC(amplitude=UNSET, createStepName='Step-1',
distributionType=UNIFORM, fieldName='', localCsys=None, name='vroll1',
region=
mdb.models['Model-1'].rootAssembly.instances['roller1-1'].sets['refpt1'],
v1=UNSET, v2=UNSET, vr3=6.28)
```

- Simetría.

```
####symetry
mdb.models['Model-1'].YsymmBC(createStepName='Step-1', localCsys=None,
name=
'symetry', region=
mdb.models['Model-1'].rootAssembly.instances['plate-1'].sets['bottom'])
```

- Velocidad inicial.

```
mdb.models['Model-1'].Velocity(distributionType=MAGNITUDE, field='',
name=
'Predefined Field-1', omega=0.0, region=
mdb.models['Model-1'].rootAssembly.instances['plate-1'].sets['Set-4'],
velocity1=1.25, velocity2=0.0)
```

11. Mallado.

En lo que respecta al mallado, este se va a realizar una parte mediante el código de Python, y otra parte manualmente (Por comodidad debido al elevado número de casos de estudio que se tiene). Lo primero que hay que hacer es hacer ambas partes del conjunto placa-rodillo independientes, debido a que solo se va a mallar la placa (puesto que el rodillo es una superficie analítica y no se puede mallar).

```
mdb.models['Model-1'].rootAssembly.makeIndependent (instances=(
mdb.models['Model-1'].rootAssembly.instances['plate-1'], ))
mdb.models['Model-1'].rootAssembly.makeIndependent (instances=(
mdb.models['Model-1'].rootAssembly.instances['roller1-1'], ))
```

Y finalmente se escoge el tipo de elemento y la forma en la que se va a estructurar la malla, dejando la semilla de mallado y el mallado definitivo en manos del usuario, el cual deberá realizarlo de manera manual para usando

el preprocesador gráfico de Abaqus:

```
mdb.models['Model-1'].rootAssembly.setMeshControls(elemShape=TRI,
regions=
    mdb.models['Model-1'].rootAssembly.instances['plate-
1'].faces.getSequenceFromMask(
    ('[#1 ]', ), )
)

mdb.models['Model-1'].rootAssembly.setElementType(elemTypes=(ElemType(
    elemCode=CPE3, elemLibrary=EXPLICIT, secondOrderAccuracy=OFF,
    hourglassControl=DEFAULT, distortionControl=DEFAULT), ElemType(
    elemCode=CPE3, elemLibrary=EXPLICIT)), regions=(
    mdb.models['Model-1'].rootAssembly.instances['plate-
1'].faces.getSequenceFromMask(
    ('[#1 ]', ), ), )
)
```

Donde se puede observar que se puede escoger fácilmente la forma del elemento, el tipo, estructura de la malla (en este caso al no poner nada se está dejando una malla libre), etc.

12. Job.

En el proyecto que se está realizando la definición del trabajo así como el acceso a los resultados se ha realizado de manera manual. Ello se debe a que debido al gran número de modelos que se han elaborado, se necesitaba gozar de un control y una gestión de los trabajos o *Jobs*, y de los resultados, el cual se lleva mejor a cabo de manera manual.

Aun así se incluye como sería la sentencia a escribir en Python para hacerlo de manera automática (se va a hacer para un caso general puesto que como se ha dicho no se ha hecho a lo largo del proyecto).

```
Mdb.job(contactPrint=OFF,description='',echoPrint=OFF,
explicitPrecision=SINGLE,historyPrint=OFF,memory=90,
memoryUnits=PORCENTAJE,model='Model-
1',modelPrint=OFF,multiprocessingmodel=DEFAULT,name='Ejemplo-
Job',nodalOutputPrecision=SINGLE,numCpus=1,numDomains=1,parallelizatio
nMethodExplicit=DOMAIN,scratch='',type=ANALYSIS,userSubroutine='')
Mdb.jobs['Ejemplo-Job'].submit(ConsistencyChecking=OFF)
```

Dicha sentencia crea el job y hace que comience el análisis. Se pueden observar algunos argumentos de entrada que deben ser conocidos para un usuario habitual de Abaqus como son el tipo de precisión (Single, Double, etc), el uso o no uso de varios procesadores, el porcentaje de memoria Ram asociado, el uso o no de paralelización, etc.

5.2. Crear un Output Database (odb).

Una vez creado el modelo en Python, el siguiente paso es acceder a los resultados. Puesto que dicha opción se ha realizado a mano a lo largo del Proyecto, no se va a entrar en mayor profundidad, aunque cabe destacar que es de gran utilidad si se está trabajando con varios Scripts a la vez o con varios ordenadores, ya que puede suponer un ahorro de tiempo considerable.

De forma general, para acceder a los resultados lo primero que habría que hacer es guardar el archivo generado en la sección anterior, con el formato “.py. Y posteriormente incluir las siguientes líneas de código que abre algunos módulos, carga la ventana de resultados y abre el odb.

```
From abaqus import*
From abaqusConstants import*
Session.Viewport(name='Viewport: 1', origin=(0.0,0.0), width=270,
height=155)
Session.Viewports['Viewport: 1'].makeCurrent()
Session.viewports['Viewport: 1'].maximize()
executeOnCaeStartup()
o1=session.openOdb(name='/home/overveld/ScriptManual/Example.odb')
session.viewports['Viewport:1'].setValues(displayedObject=o1)
```

Y por ejemplo, si se quiere representar las tensiones sobre el contorno de la geometría del modelo deformada:

```
Session.viewports['Viewport:
1'].odbDisplay.display.setValues(plotstate=(CONTOURS_ON_DEF,))
Session.printToFile(filename='Ejemplo-
odb',format=TIFF,canvasObjects=(session.viewports['Viewport:1'],))
```

6 RESULTADOS.

Una vez definidos los modelos y el estudio paramétrico que se va a realizar, es la hora de realizar todos los análisis implicados, almacenar, ordenar datos y resultados para analizarlos detenidamente y obtener las conclusiones que sean posibles.

Puesto que el modelo se ha realizado en 2D, los resultados del análisis se obtendrán como tal. Sin embargo, para que visualmente quede más claro, se va a aplicar una extrusión en los resultados, así como una simetría, de manera que en lugar de observar el modelo 2D simétrico, se va a poder ver una representación 3D completa del problema (solo a fin de visualización, ya que los resultados representados se corresponden con los del modelo 2D, es decir, simplemente se extruyen los mismos en la dirección **Z**).

Para ello, dentro del módulo *Visualization*, se hace clic en *View* → *ODB display options*, y en las pestañas *Sweep/Extrude* y *Mirror/Pattern* se puede hacer tanto la extrusión como representar el problema simétrico. Para esta segunda visualización, es necesario crear en primer lugar el sistema de referencia respecto al que se quiere hacer la simetría. Para ello se crea un sistema coordenado cartesiano fijo cuyo eje X esté contenido en la base de la placa, el eje Y se dirija hacia la parte superior de la placa perpendicularmente, y el eje Z complete el triedro. Por lo que el plano de simetría es el plano XZ, y así se ha de seleccionar para la correcta representación.

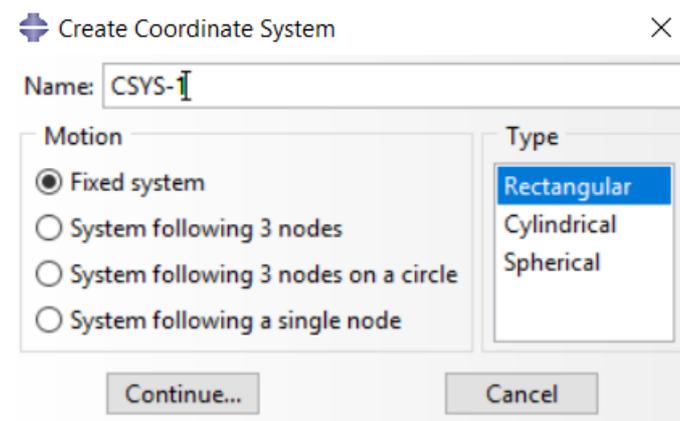


Ilustración 18. Creación de un sistema de coordenadas.

A partir de este punto el procedimiento seguido ha sido el explicado anteriormente: Variar los distintos parámetros que afectan a las condiciones de arrastre natural y forzado y estimar el valor del rozamiento para el que se produce el arrastre para cada caso.

Por lo que para cada material (Aluminio y Acero) y terna de valores (R, h_0, h_f) , se ha obtenido un modelo para el cual se han hecho varios análisis variando el valor del rozamiento comenzando como primer “disparo” con el valor teórico que debería tener.

De manera que una vez se realice el análisis, solo hay que comprobar si se produce el arrastre, y si se produce para el valor más próximo anterior y posterior (en este proyecto se ha trabajado con incrementos de 1% respecto del valor teórico).

Esto queda ilustrado con el siguiente ejemplo para un caso de arrastre natural ($h_0 = 0.04, h_f = 0.375 * h_0, R = 0.2$, Aluminio, μ teórico=0.35355).

- Se produce arrastre: Para $\mu = \mu_{teórico} = 0.3536$ se produce la laminación del material.

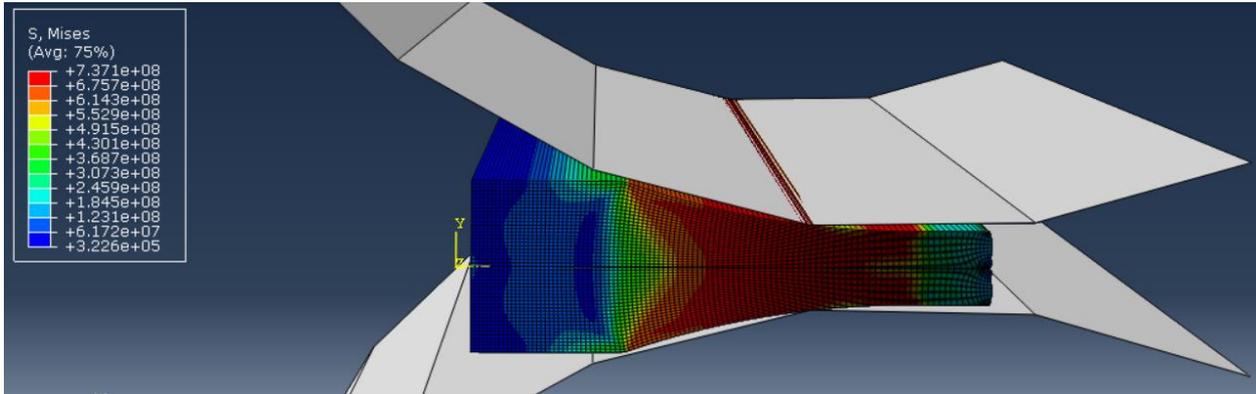


Ilustración 19. Arrastre.

- No se produce arrastre: Para $\mu = 0.99 * \mu_{teórico} = 0.350064$ el material no pasa entre los rodillos:

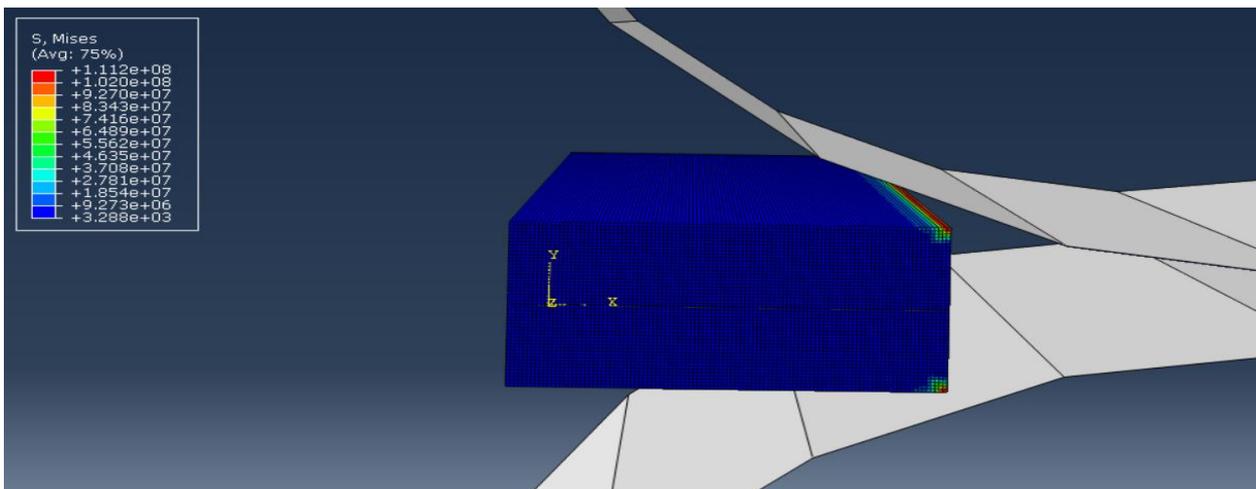


Ilustración 20. No se produce arrastre.

El objetivo de aquí en adelante será obtener aquellos valores para los que se produce el arrastre, tanto forzado como natural, así como el error respecto al valor que le correspondería según la teoría de laminación, con el fin de estudiar: si su evolución con la variación de los parámetros es la adecuada, como varía el error respecto al valor teórico, y si se puede obtener alguna conclusión que explique cualitativamente el por qué de ese error.

6.1 Arrastre Natural

En primera instancia se procede a estudiar la condición de arrastre natural para el aluminio, según diferentes variaciones de Radio, h_0 , h_f . Posteriormente se repetirá el proceso para el Acero, comprobándose que como era de esperar, los valores no varían significativamente para ambos materiales.

6.1.1 Aluminio.

- Aluminio. Radio=0.2 m; $h_0 = 0.01$ m.

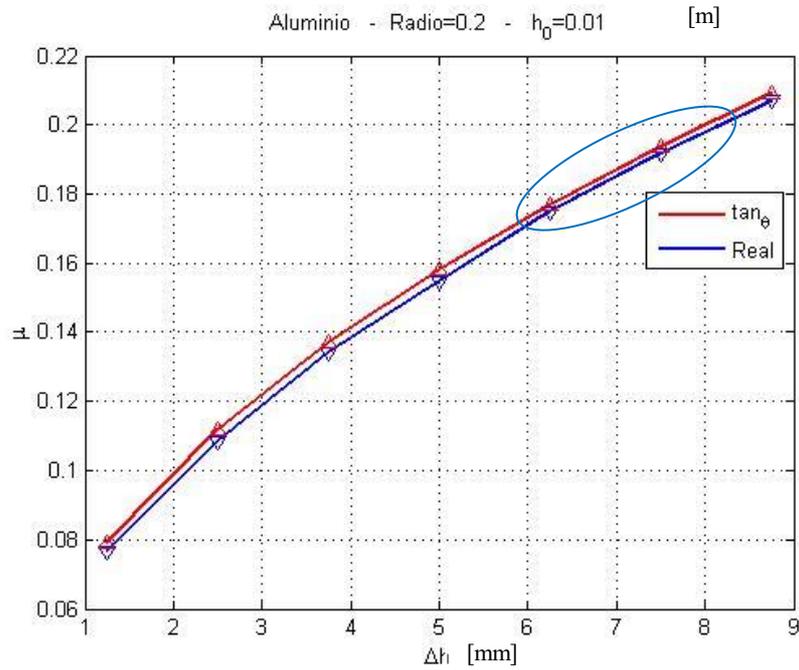


Ilustración 21. AN- Al -1-: μ frente a Δh

- Aluminio. Radio=0.2 m; $h_0 = 0.04$ m.

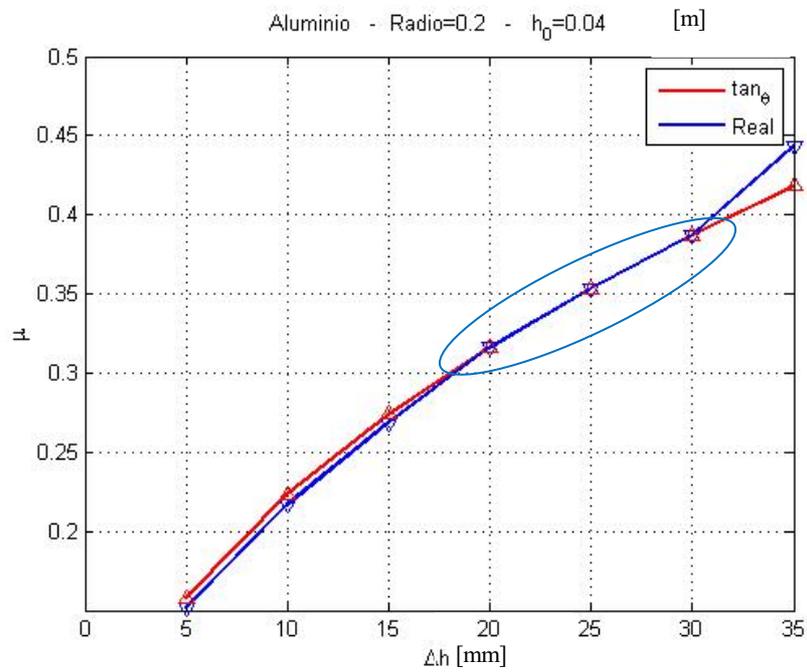


Ilustración 22. AN- Al -2-: μ frente a Δh

- Aluminio. Radio=0.2 m ; $h_0=0.08$ m.

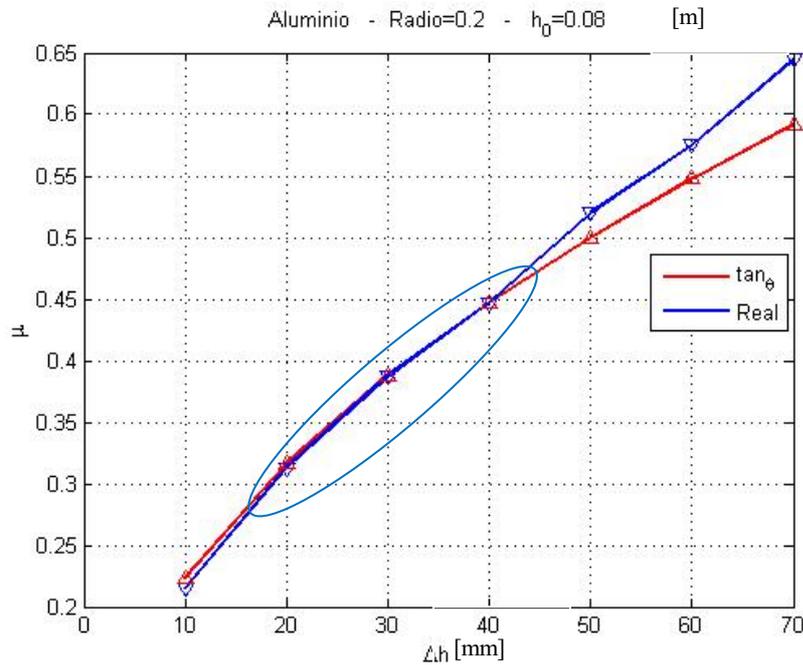


Ilustración 23. AN - Al -3-: μ frente a Δh

Se observa que en los tres casos, para Radio y h_0 fijos, a medida que aumenta Δh , y por lo tanto disminuye h_f , aumenta el valor del rozamiento real, tal y como lo predice la teoría. Además, Para radio fijo, se puede observar que a medida que aumenta h_0 , los valores de rozamientos obtenidos son mayores, ya que Δh será mayor y por lo tanto $\mu = \sqrt{\Delta h/R}$ crece.

Además los resultados parecen tener sentido puesto que a mayor valor de Δh , mayor será la longitud de contacto L_p , y por lo tanto mayor fuerza de fricción se necesitará para introducir la placa puesto que además la presión estará aplicada sobre una superficie mayor.

Los valores numéricos obtenidos son:

Radio	h_0 [m]	h_f [m]	Δh [mm]	L/\bar{h}	$\mu_{teórico}$	μ_{real}	L [mm]	Error %
0,2	0,01	0,125*h0	9	7	0,2091	0,2070	41.8	-1
		0,25*h0	8	6	0,1936	0,1917	38.7	-1
		0,375*h0	6	5	0,1767	0,1750	35.4	-1
		0,5*h0	5	4	0,1581	0,1549	31.6	-1
		0,625*h0	4	3	0,1369	0,1341	27.4	-2
		0,75*h0	3	3	0,1118	0,1084	22.4	-3
		0,875*h0	1	2	0,07905	0,0735	15.8	-3
		0,125*h0	35	4	0,4183	0,4434	83.7	5
		0,25*h0	30	3	0,3873	0,3872	77.5	0
		0,375*h0	25	3	0,3536	0,3535	70.7	0

0,2	0,04	0,5*h0	20	2	0,3162	0,3162	63.2	0
		0,625*h0	15	2	0,2739	0,2683	54.8	-2
		0,75*h0	10	1	0,2236	0,2168	44.7	-3
		0,875*h0	5	0,8	0,1581	0,1517	31.6	-4
0,2	0,08	0,125*h0	70	2,6	0,5912	0,6448	118.3	9
		0,25*h0	60	2,1	0,5477	0,5751	109.5	5
		0,375*h0	50	1,8	0,5	0,525	100	4
		0,5*h0	40	1,4	0,4472	0,4427	89.4	0
		0,625*h0	30	1,2	0,3873	0,3872	77.5	0
		0,75*h0	20	0,9	0,3162	0,313	63.2	-1
		0,875*h0	10	0,6	0,2236	0,2102	44.7	-4

Tabla 3. Resultados Arrastre Natural para R=0.2 m

Y en lo que respecta a los errores encontrados, en los tres casos anteriores se puede observar como para valores de rozamiento bajos, el arrastre natural se produce antes de lo que predice la teoría, y para valores elevados, se produce después de lo que se supone que debería hacerlo.

Además se puede observar que máximos errores positivos ($\mu_{real} > \mu_{teórico}$) se producen para h_0 más elevados y con un gran reducción de espesor. Ello parece ser debido a que la longitud de contacto en estos casos es más elevada, por lo que la fuerza real que debe aplicar el rodillo para desplazar tanto material es mayor.

Sin embargo los errores negativos ($\mu_{real} < \mu_{teórico}$) más altos, se producen para las menores reducción de espesor. Ello parece razonable puesto que la longitud de contacto será menor, por lo que el rodillo encuentra menos resistencia a la hora de hacer al material fluir.

Sin embargo cuando se aumenta el radio del rodillo se obtiene:

Radio	h0	hf	Ah[mm]	L/hmed	mu teórico	múreal	L[mm]	Error %
0.5	0,01	0,125*h0	9	11.75	0.1323	0.128	66.1	-3
		0,25*h0	8	9.79	0.1225	0.118	61.2	-3
		0,375*h0	6	8.13	0.1118	0.108	56	-3
		0,5*h0	5	6.67	0.1	0.097	50	-3
		0,625*h0	4	5.32	0.086	0.083	43.3	-4
		0,75*h0	3	4.04	0.07	0.067	35.4	-5
		0,875*h0	1	2.66	0.05	0.047	25	-6
0,5	0,04	0,125*h0	35	3.71	0.2646	0.254	132	-2
		0,25*h0	30	3.09	0.2249	0.240	122	-2
		0,375*h0	25	2.57	0.2236	0.216	112	-3
		0,5*h0	20	2.10	0.2	0.194	100	-3
		0,625*h0	15	1.68	0.1732	0.166	86.6	-4

		0,75*h ₀	10	1.22	0.1414	0.134	70	-5
		0,875*h ₀	5	0.84	0.1	0.094	50	-6
0,5		0,125*h ₀	70	4.15	0.3742	0.366	187.1	-2
		0,25*h ₀	60	3.46	0.3464	0.339	173.2	-2
		0,375*h ₀	50	2.87	0.3162	0.309	158.1	-2
		0,5*h ₀	40	2.35	0.2828	0.274	141.4	-3
		0,625*h ₀	30	1.88	0.2449	0.237	122.5	-3
		0,75*h ₀	20	1.42	0.2	0.192	100	-4
	0,08	0,875*h ₀	10	0.94	0.1414	0.132	70	-6

Tabla 4. Resultados Arrastre Natural para R=0.5 m.

Donde como se puede comprobar, a medida que la reducción de espesor es mayor, aumenta la longitud característica, y el valor del rozamiento se va incrementando. Sin embargo puesto que las longitudes de contacto son mayores que en el caso anterior, los errores positivos deberían ser mayores también, y a pesar de que va aumentando, en algunos casos no llegan ni a superar el valor teórico.

Esto nos hace pensar que la longitud de contacto entre rodillo y placa no es el parámetro que nos ayude a comprobar por que se producen dichos errores.

A continuación se muestran gráficamente los resultados obtenidos para R=0.5:

- Aluminio. R=0.5 m; h₀=0.01 m.

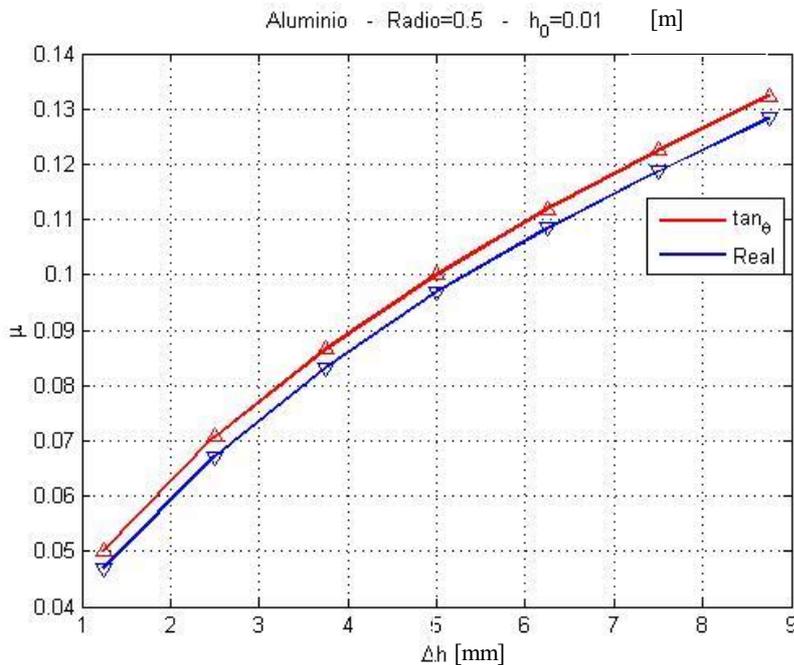


Ilustración 24. AN- Al -4-: μ frente a Δh

- Aluminio. $R=0.5$ m; $h_0=0.04$ m.

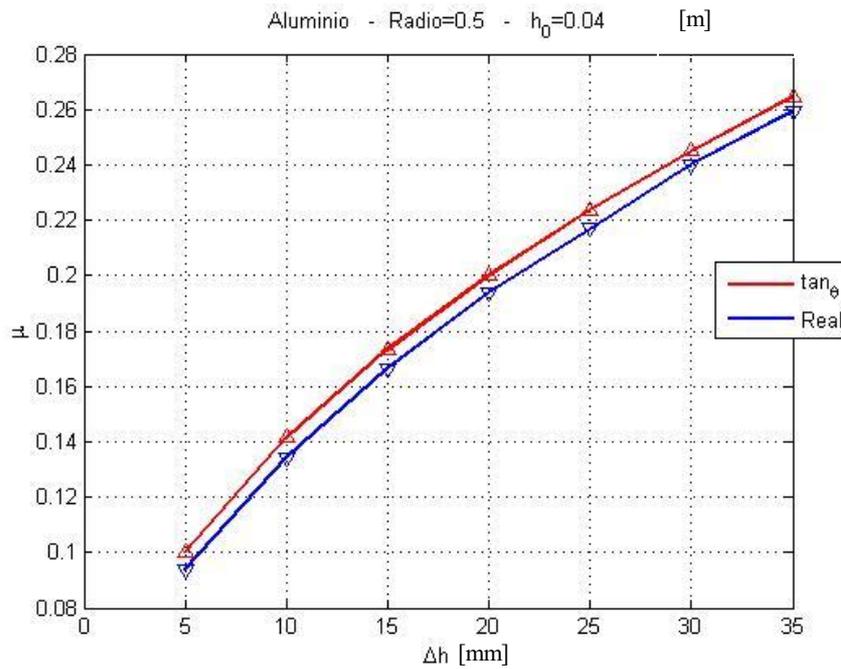


Ilustración 25. AN - Al -5-: μ frente a Δh

- Aluminio. $R=0.5$ m; $h_0=0.08$ m.

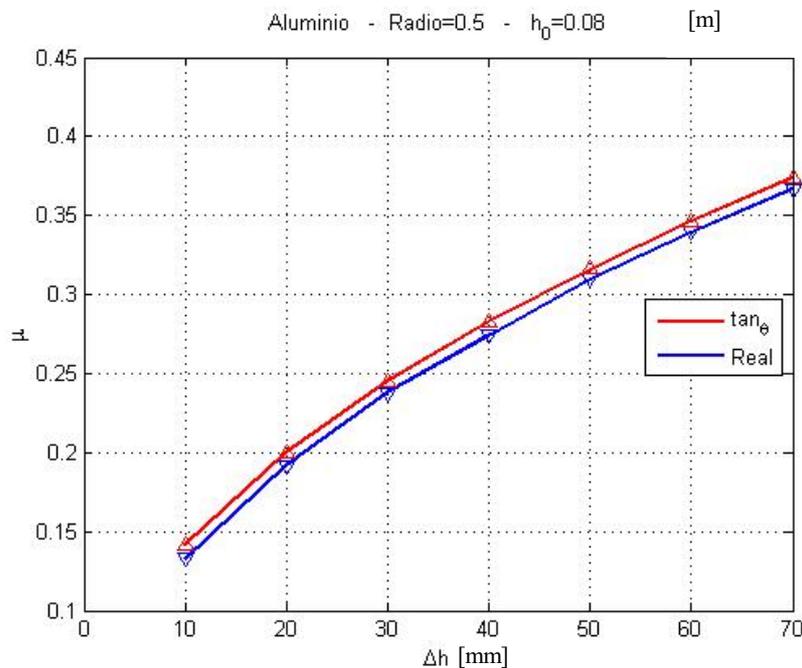


Ilustración 26. AN - Al -6-: μ frente a Δh

Donde como se puede observar que independientemente de los errores encontrados respecto a teoría, los rozamientos obtenidos respetan la evolución que sigue la misma ya que se obtienen las mismas tendencias que para $R=0.2$ m, aunque con valores inferiores a estos casos puesto que el radio es mayor.

Con el fin de encontrar una tendencia para la evolución de los errores encontrados, se decide comprobar la

evolución respecto la homogeneidad en la deformación, y los resultados obtenidos son los siguientes:

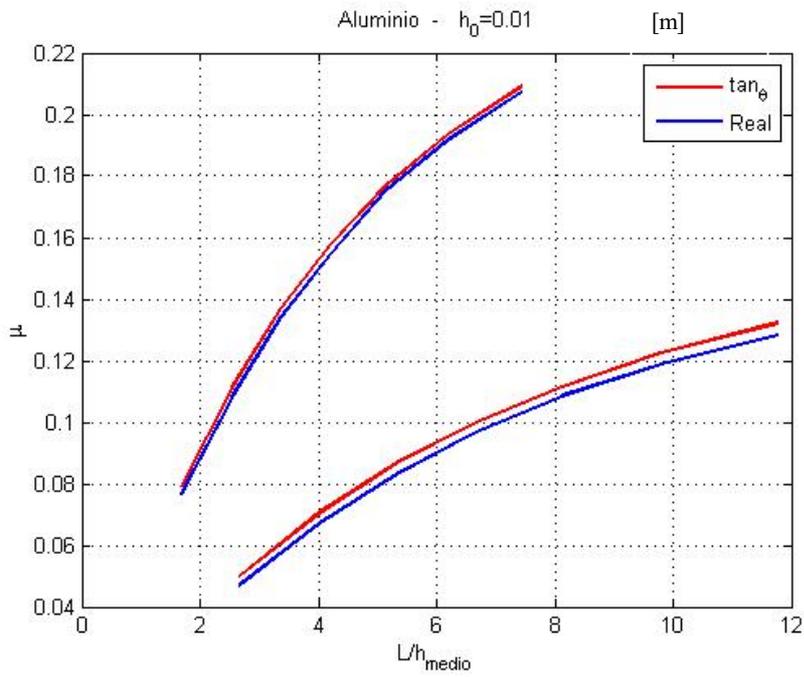


Ilustración 27. AN - Al -7-: μ frente a L/h_{med}

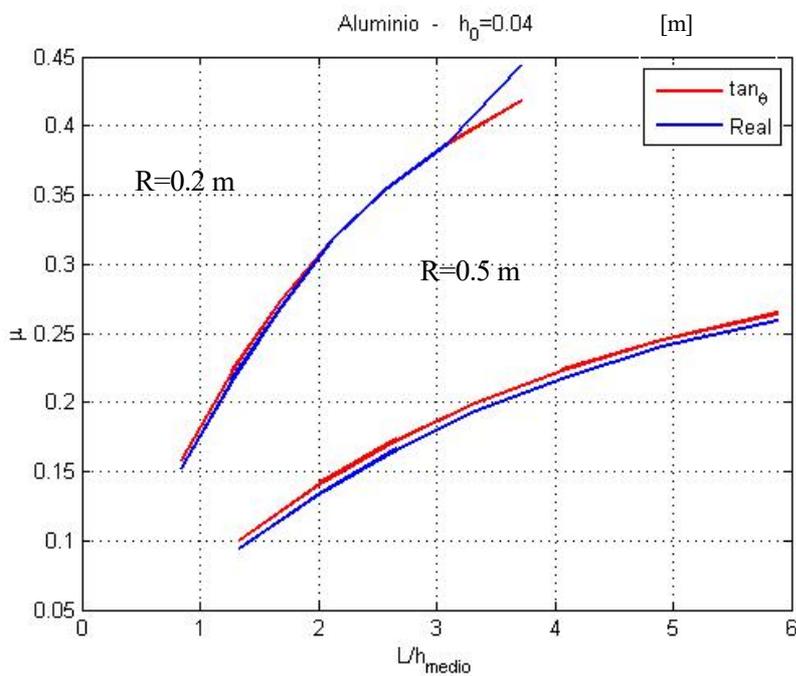


Ilustración 28. AN - Al -8-: μ frente a L/h_{med}

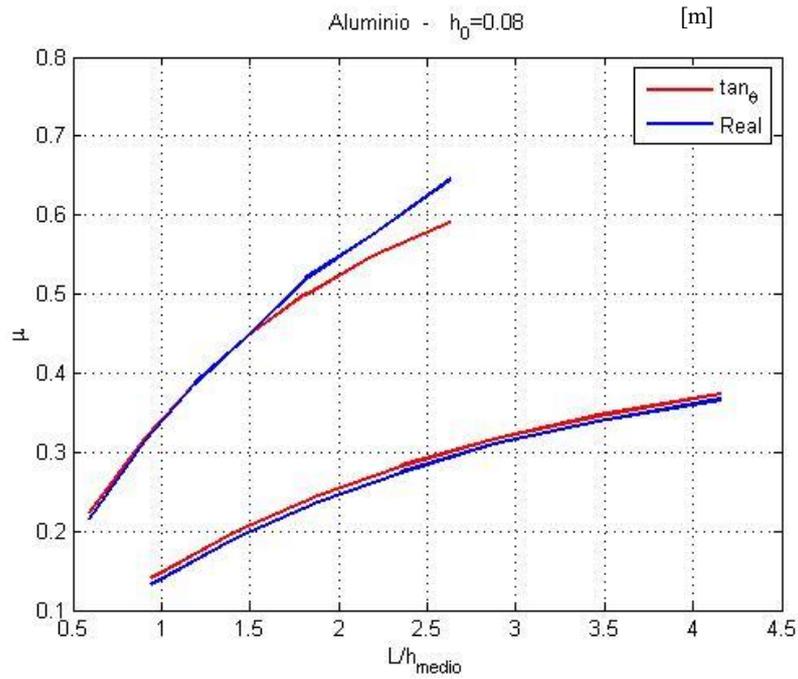
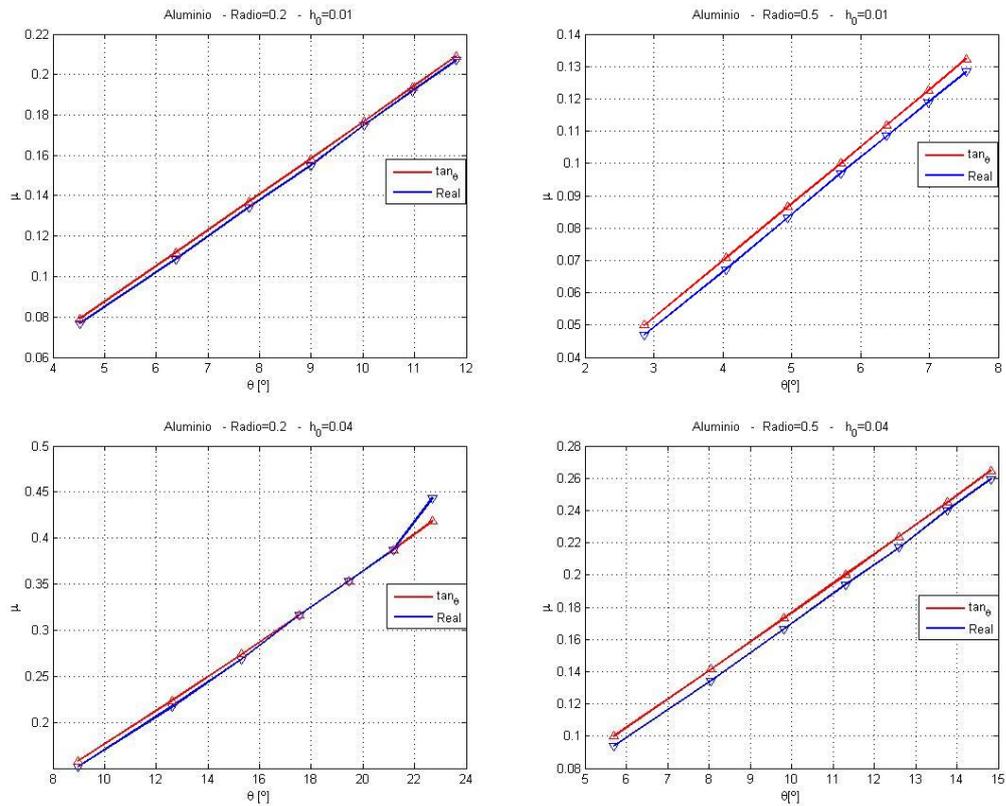


Ilustración 29. AN - Al -9-: μ frente a L/h_{med}

Se puede comprobar que no existe un rango perfectamente definido en el que podamos decir que el error encontrado sea el mínimo. Si que se puede afirmar que a medida que aumenta L/\bar{h} , va aumentando el error respecto a la teoría de manera progresiva.

Finalmente se opta por definir entre que ángulos el error es el mínimo e intentar contrastarlo con resultados experimentales para ver si el modelo computacional desarrollado los cumple:



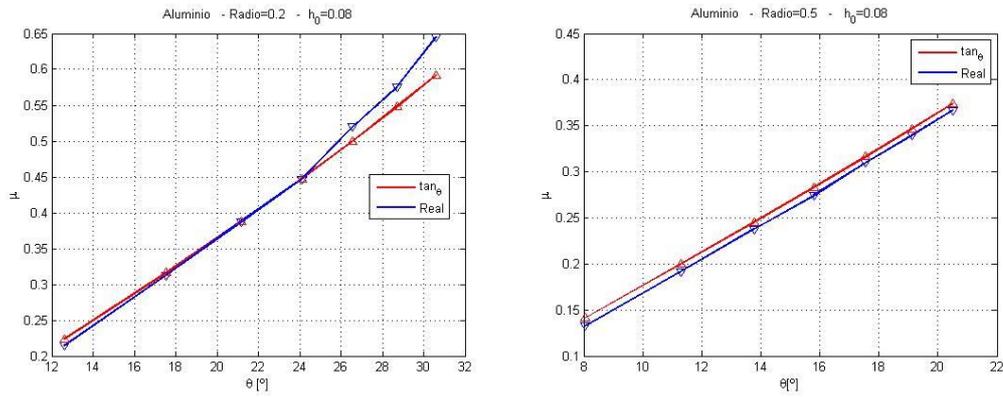


Ilustración 30. AN - Al -10-: μ frente a θ

Donde se puede aproximar que los mejores resultados se obtienen para ángulos entre los 10-22 grados, que se corresponde con coeficientes de rozamiento de 0.1763-0.3639. Esto no parece muy descabellado puesto que los ángulos típicos para los cuales se realiza la laminación en frío no superan dichos valores (Aunque debido a la dificultad en la medición experimental de rozamientos la bibliografía para consultar dichos valores es excasa).

Finalmente se va a representar el efecto que tiene el mallado adaptativo a la hora de simular las condiciones de arrastre natural:

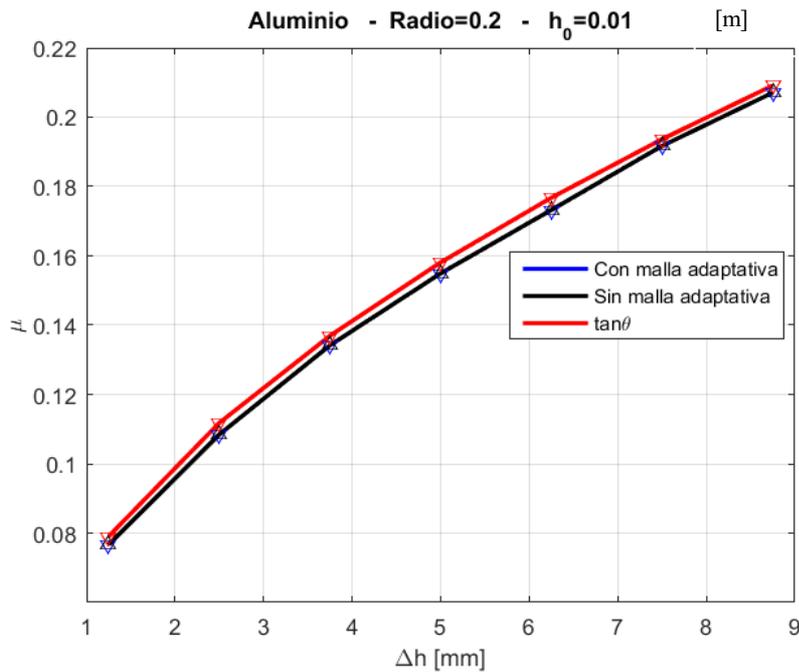


Ilustración 31. AN - Al -11-: Efecto del Mallado Adaptativo.

Donde se observa claramente que el valor del rozamiento obtenido para el modelo de estimación del coeficiente de arrastre natural es independiente del uso o no de dicho tipo de malla, ya que los resultados para la malla adaptativa y la no adaptativa aparecen superpuestos.

Esto quiere decir que los resultados obtenidos tienen un alto grado de fiabilidad y exactitud, puesto que son independientes del tipo de mallado escogido.

6.1.2 Acero

A continuación se van a mostrar los valores obtenidos para el Acero, donde a priori se deben obtener los mismos valores obtenidos anteriormente, o al menos, con un error muy pequeño respecto a estos.

- Acero. $R=0.2$ m; $h_0=0.01$ m.

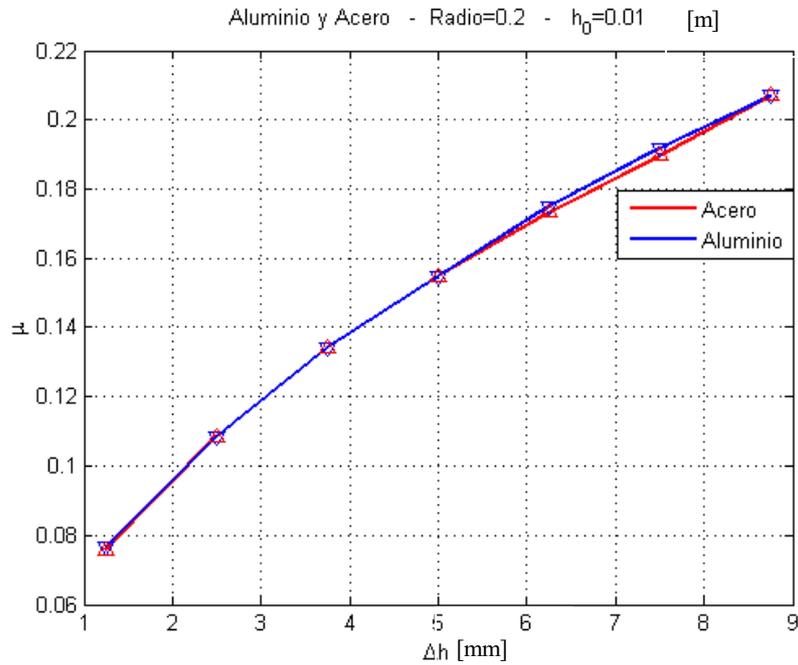


Ilustración 32. AN - AC -1-: μ frente a Δh

Los máximos errores encontrados entre los coeficientes de rozamiento entre el caso del aluminio y del acero son de un 1%, por lo que se demuestra que es independiente del tipo de material. Lo mismo sucede para el resto de casos, que no necesitan de mayor explicación.

- Acero. $R=0.2$ m; $h_0=0.04$ m.

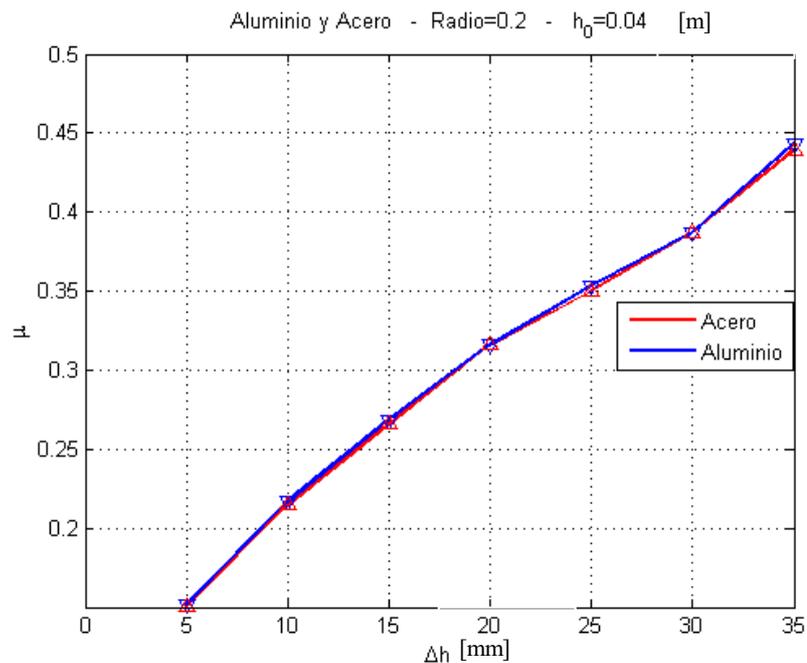


Ilustración 33. AN - AC -2-: μ frente a Δh

- Acero. $R=0.2$ m; $h_0=0.08$ m.

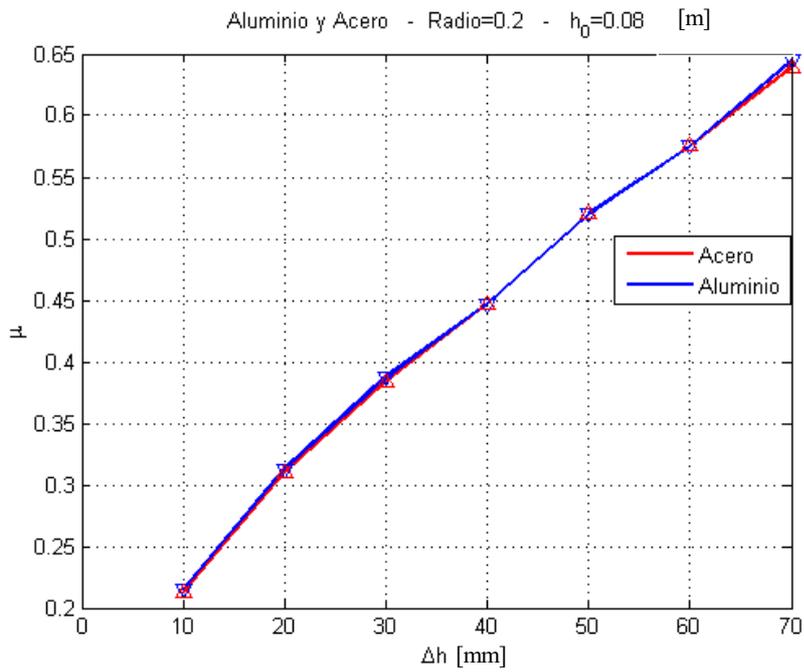


Ilustración 34. AN - AC -3-: μ frente a Δh

- Acero. $R=0.5$ m; $h_0=0.01$ m.

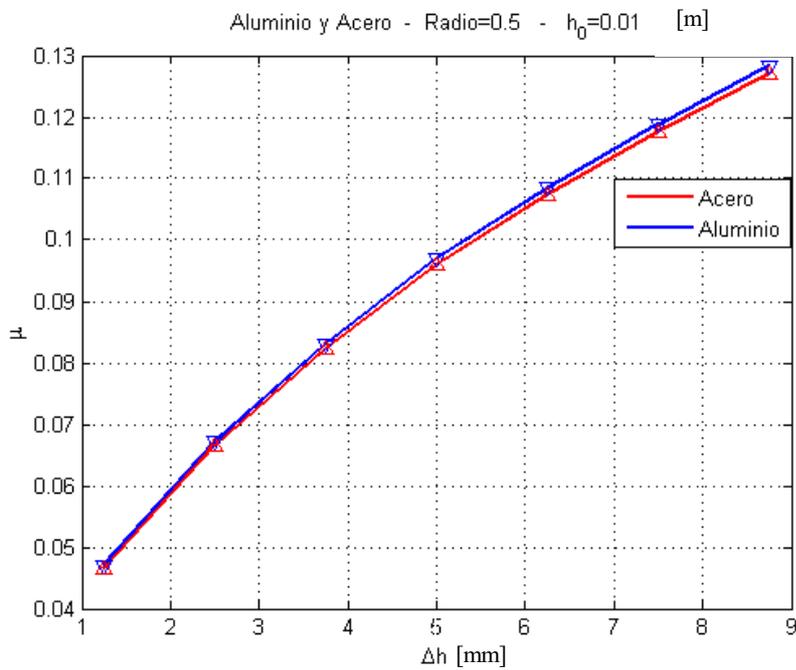


Ilustración 35. AN - AC -4-: μ frente a Δh

- Acero. $R=0.5$ m; $h_0=0.04$ m.

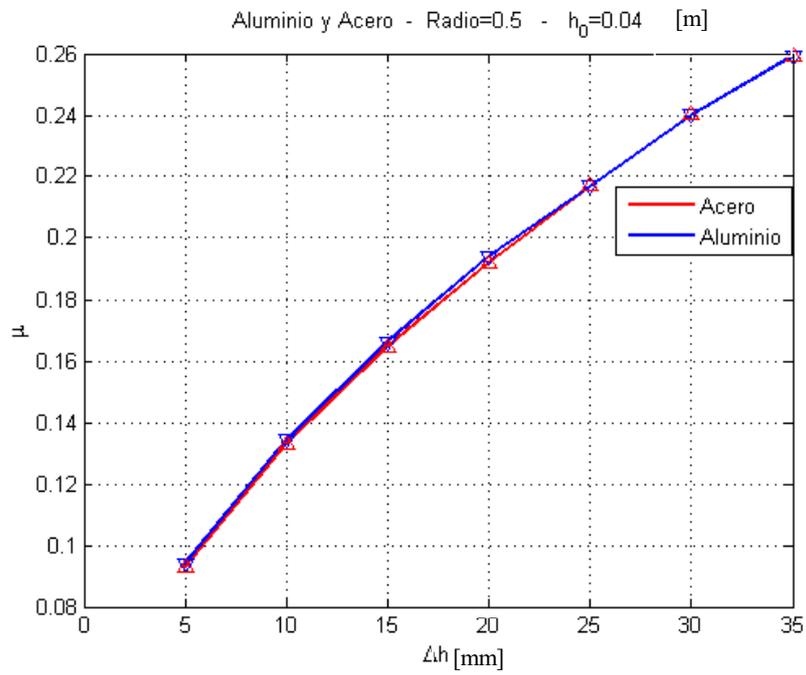


Ilustración 36. AN - AC -5-: μ frente a Δh

- Acero. $R=0.5$ m; $h_0=0.08$ m.

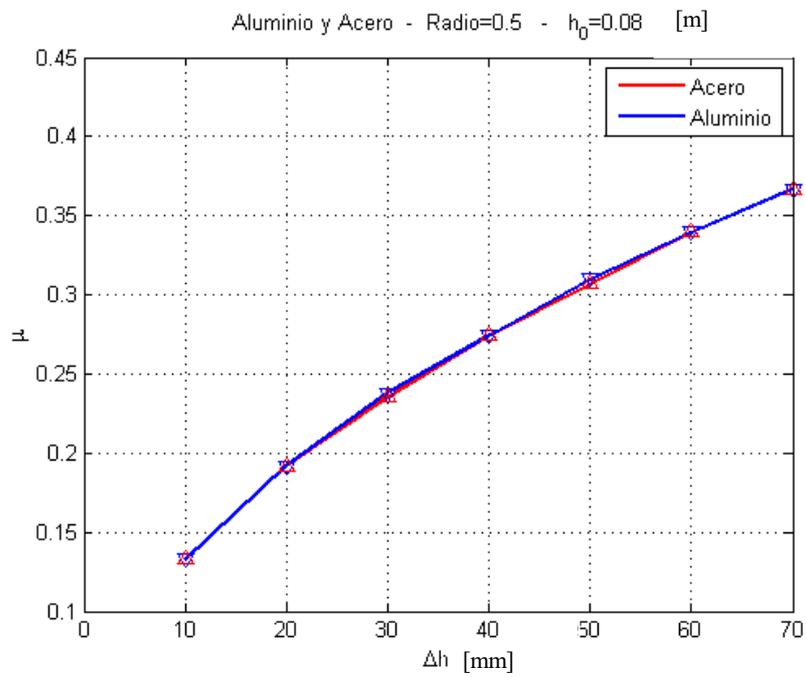


Ilustración 37. AN - AC -6-: μ frente a Δh

6.2 ARRASTRE FORZADO

A continuación se procede a realizar el mismo estudio anterior pero para la estimación del coeficiente de arrastre forzado, según el cual el arrastre de la placa se producía para valores del rozamiento:

$$\mu \geq \tan\left(\frac{\theta}{2}\right) \quad (6 - 1)$$

Al igual que antes primero se estudiará para el Aluminio 7075 y posteriormente para el Acero al carbono dulce, destacando que puesto que los errores obtenidos son mayores y existe una mayor dispersión en los resultados que para el caso de arrastre natural, se va a partir como primer “disparo” del valor teórico, y variando a incrementos de un 3% a partir de este.

6.2.1 Aluminio.

Se va a realizar el mismo estudio paramétrico que se realizó para el modelo de arrastre natural:

- Aluminio. $R=0.2$ m. $h_0=0.01$ m.

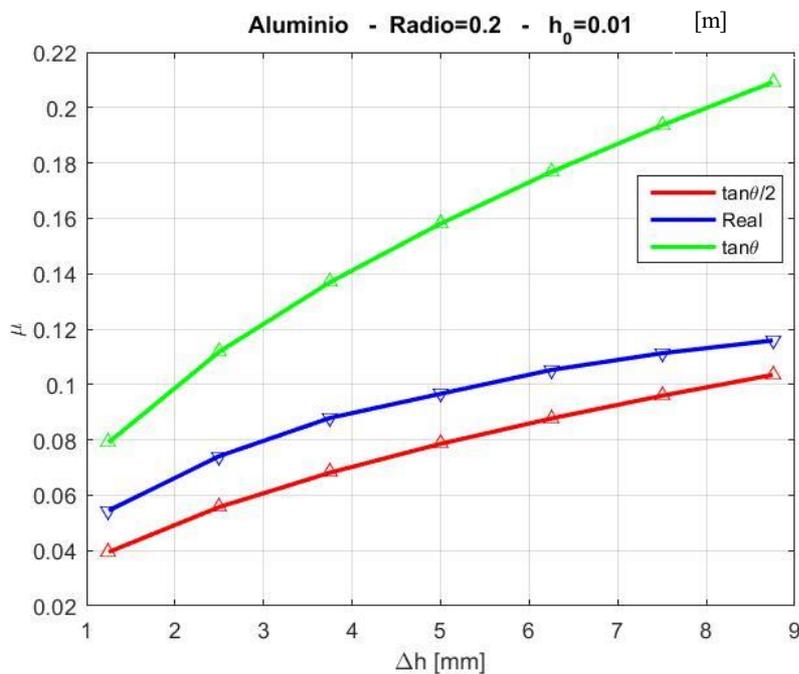


Ilustración 38. AF - AL -1-: μ frente a Δh

Donde se observa que la tendencia es igual que la observada para el caso de arrastre natural: A medida que la reducción de espesor aumenta, se incrementa el valor del coeficiente de rozamiento. Dicha tendencia se va a dar de igual manera con la longitud característica de contacto, el ángulo, y L/h_{med} , con lo cual se obtienen gráficas con la misma tendencia que para el caso del arrastre natural pero con valores inferiores. Esto es razonable ya que:

$$\mu = \tan\left(0.5 * \operatorname{atan}\left(\sqrt{\frac{\Delta h}{R}}\right)\right) \quad (6 - 2)$$

Por lo que se observa claramente que la evolución a seguir con los parámetros va a ser la misma que para el caso de arrastre natural.

- Aluminio. R=0.2 m. $h_0=0.04$ m.

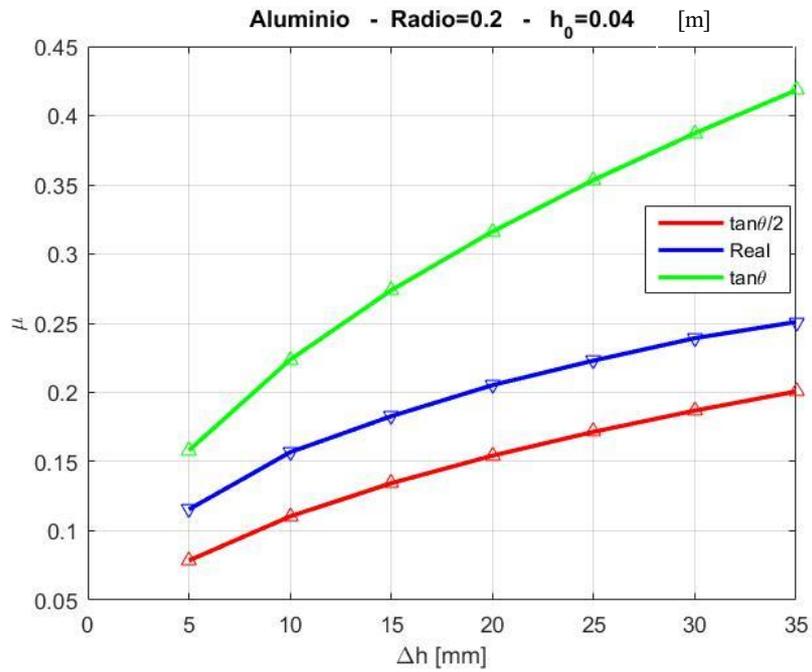


Ilustración 39. AF - AL -2-: μ frente a Δh

- Aluminio. R=0.2 m. $h_0=0.08$ m.

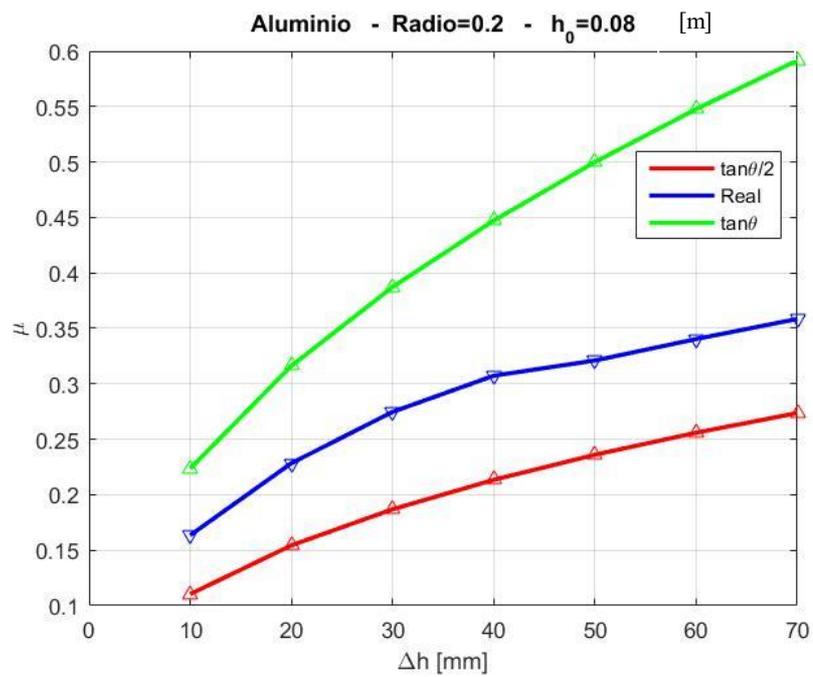


Ilustración 40. AF - AL -3-: μ frente a Δh

Como se puede observar en los tres casos anteriores los errores obtenidos son considerablemente mayores a los obtenidos para el caso del arrastre natural, encontrándose de media errores del 30%.

Además cabe destacar que a diferencia del modelo de arrastre natural, para el caso de arrastre forzado lo valores de rozamientos obtenidos usando o no malla adaptativa difieren en más un 50% para algunos casos:

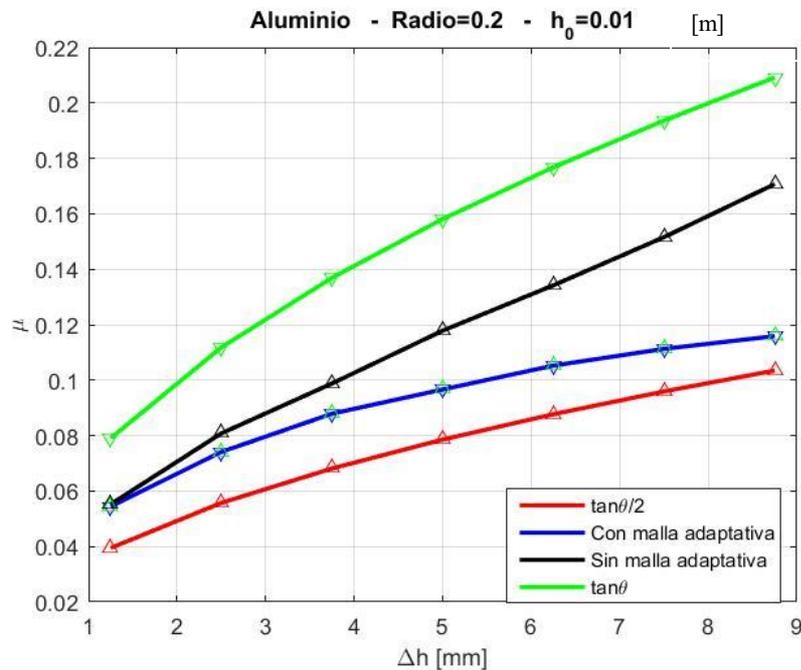


Ilustración 41. AF - AL -4-: Comparación con y sin Malla Adaptativa.

Esto nos da una idea de la complejidad del caso de estudio que está realizando, en el que el contacto inicial es más complejo que en el modelo realizado para el arrastre natural. Además la teoría usa varias hipótesis (ángulos pequeños, presión constante, etc.) que pueden hacer que los resultados se vean alterados.

Por lo que en resumen, la obtención de los errores para el caso del arrastre forzado pueden deberse a diversas razones:

- Mallado insuficiente.
- Teoría mal postulada.
- Imposibilidad del programa por simular el problema en cuestión.
- Errores matemáticos que dan lugar a que el contacto no sea el adecuado.

Para intentar subsanar el posible problema en el contacto inicial (ya que puede que debido a los errores matemáticos asociados a los modelos numéricos sea el culpable de los elevados errores existentes), se va a proponer un modelo alternativo que presenta las siguientes novedades:

- Consta de dos *Steps*, uno de ellos de un tiempo de 0.003 segundos, y otro de 0.07 s.
- El *Step-1 de 0.003* será exactamente igual al del modelo de arrastre forzado anterior, salvo por que el rozamiento asociado a dicho *step* será el correspondiente al modelo de arrastre natural (Mismo h_0 , Δh , R , y material).
- El nuevo *step de 0.07* s llevará asociado el rozamiento correspondiente al arrastre forzado. Habrá que crear por lo tanto en dicho *step* una nueva interacción y desactivar la del *step* anterior.

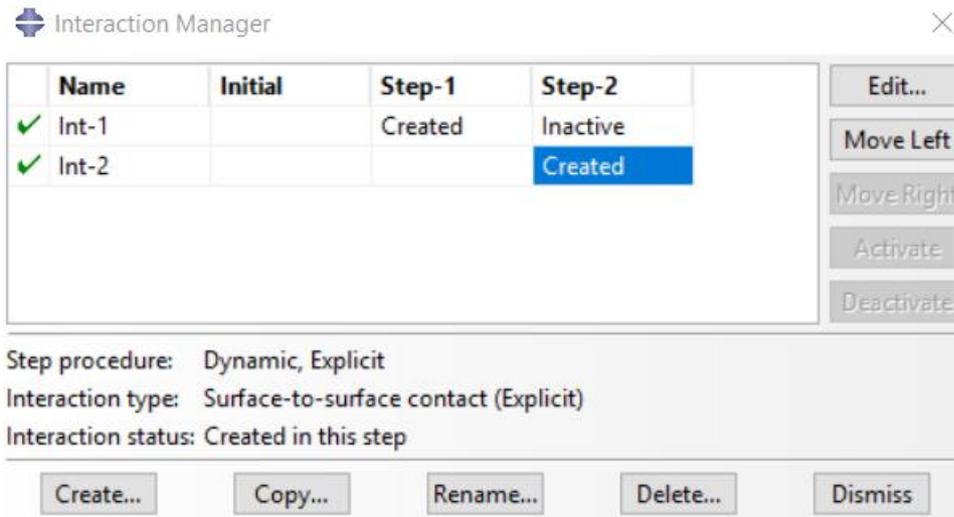


Ilustración 42. Modelo “Real 2” con dos Steps y dos rozamientos.

Con esto se consigue que el material fluya un poco entre los rodillos, sin modificarse los parámetros geométricos que determinan el coeficiente de arrastre forzado en el instante en que se inicia el *step 2*. Por lo que de esta forma nos aseguramos que el contacto entre rodillo y placa es perfecto.

Los resultados obtenidos con el nuevo modelo (se va a llamar **Real 2** en las gráficas) son mucho mejores y se muestran a continuación:

- Aluminio. $R=0.2$ m, $h_0=0.01$ m, con 2 Steps.

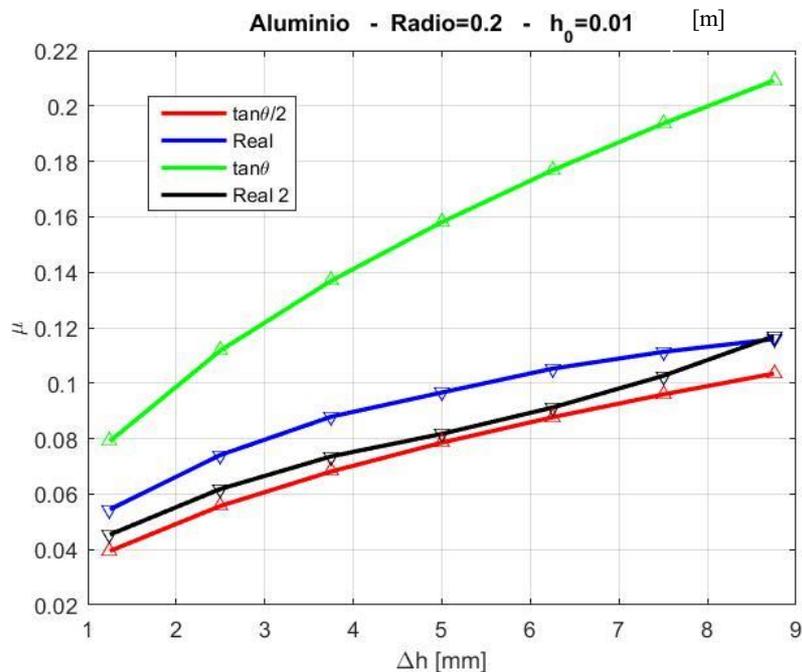


Ilustración 43. AF - AL -5-: μ frente a Δh

- Aluminio. $R=0.2$ m, $h_0=0.04$ m, con 2 Steps.

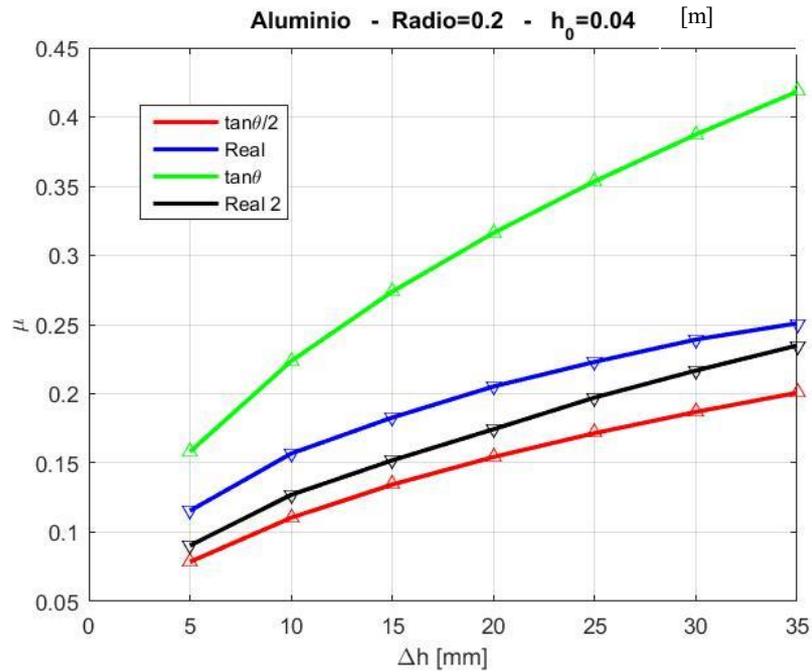


Ilustración 44. AF - AL -6-: μ frente a Δh

- Aluminio. $R=0.2$ m, $h_0=0.08$ m, con 2 Steps.

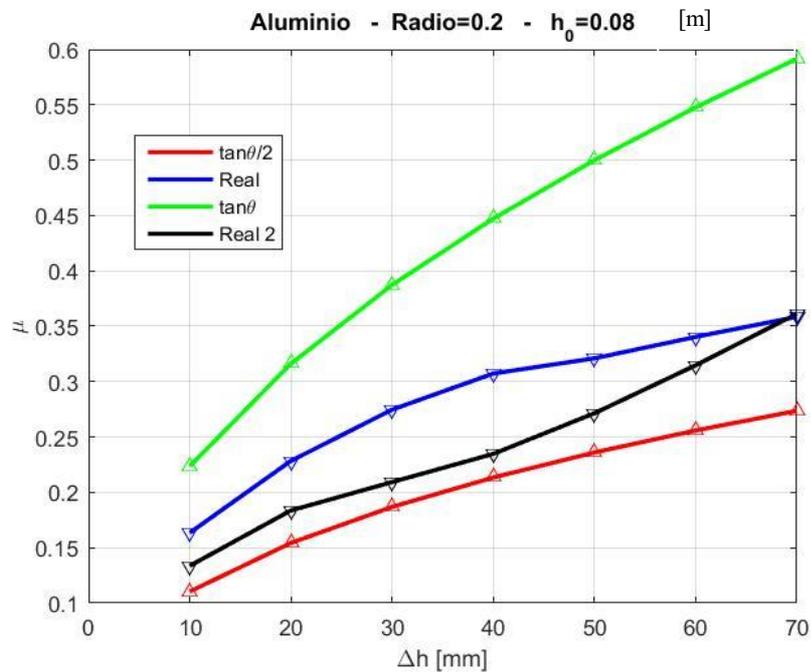


Ilustración 45. AF - AL -7-: μ frente a Δh

Finalmente representamos los valores obtenidos para radios de 0.5 m según los dos casos anteriores (El caso normal y el caso en el que se fuerza a pasar el material para mejorar el contacto):

- Aluminio. $R=0.5$ m. $h_0=0.01$ m.

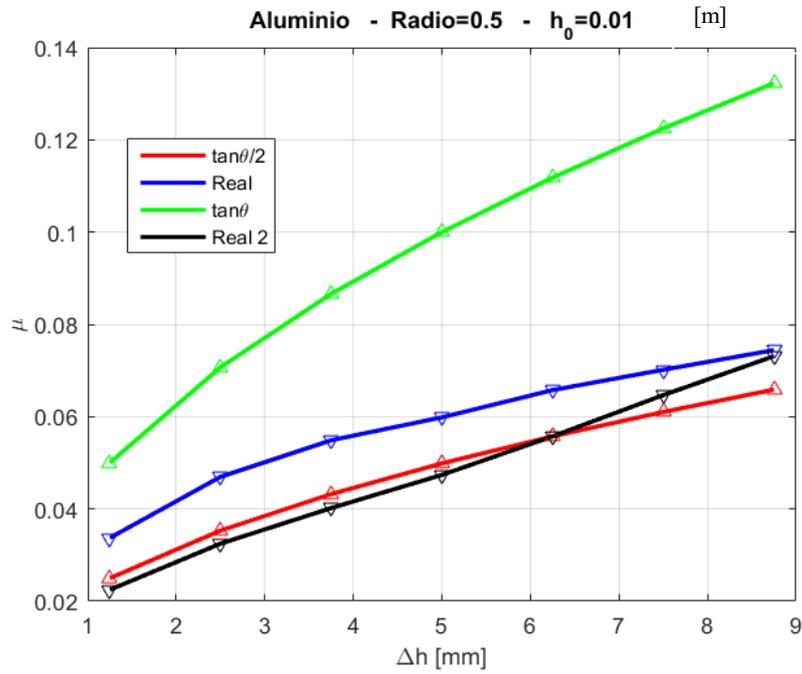


Ilustración 46. AF - AL -8-: μ frente a Δh

- Aluminio. $R=0.5$ m. $h_0=0.04$ m.

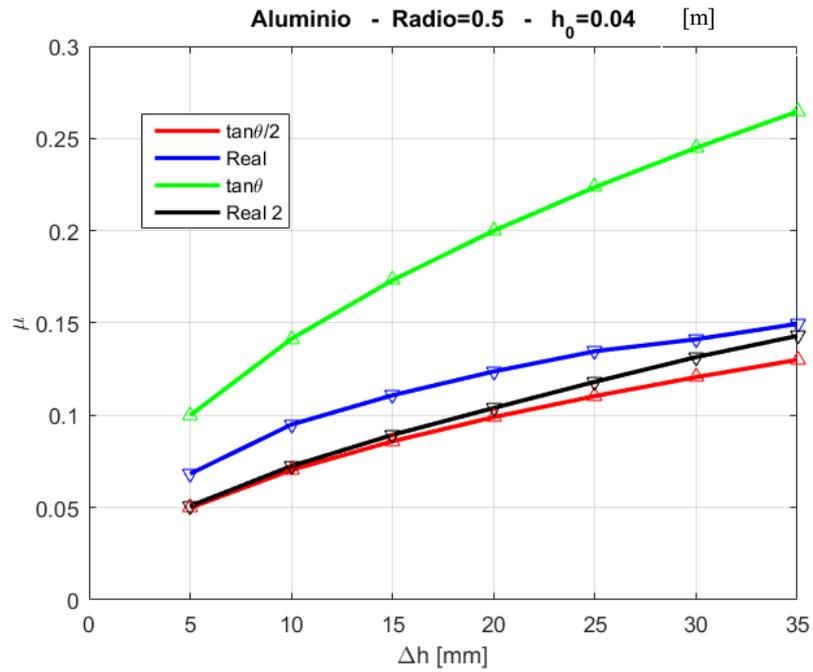


Ilustración 47. AF - AL -9-: μ frente a Δh

- Aluminio. $R=0.5$ m. $h_0=0.08$ m.

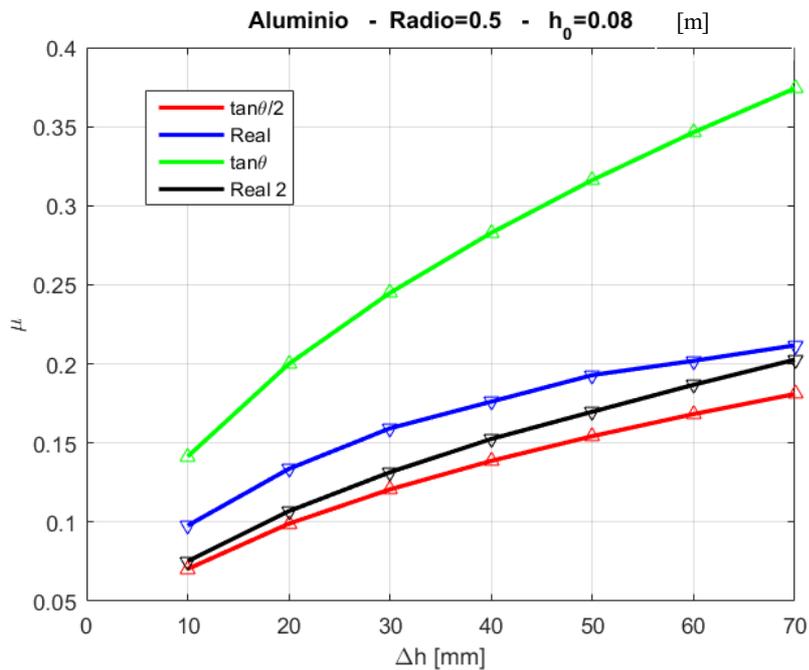


Ilustración 48. AF - AL -10-: μ frente a Δh

Se ha observado que el nuevo modelo proporciona resultados mucho mejores que el caso con un solo *Step*, lo que parece indicar que los grandes errores encontrados anteriormente se deben principalmente a que el contacto inicial entre el rodillo y la superficie de la placa no es perfecto.

Además se comprueba que a medida que aumenta la reducción de espesor, los errores encontrados son más elevados, alejándose de los valores teóricos y acercándose a los valores obtenidos para el caso “**Real**”, con un solo *Step*. Sin embargo para pequeñas reducciones de espesor, los errores encontrados son menores al 10 % para todos los casos.

Al final del capítulo, en el apartado de conclusiones, se dará una explicación al por qué los resultados obtenidos con este segundo modelo “**Real 2**” se aproximan tan bien para pequeñas reducciones de espesor, y por que se aleja considerablemente de los valores teóricos cuando Δh se hace mayor.

6.2.2 Acero.

Con el acero al carbono sucede exactamente lo mismo que con el caso de arrastre natural: Los resultados obtenidos son prácticamente análogos a los del Aluminio, salvo por pequeños errores de índole computacional.

Puesto que el análisis y la evolución de las gráficas son idénticas a la del apartado anterior, representamos todas las gráficas sin entrar en detalle en cada una de ellas, representando además el caso correspondiente con el contacto inicial sin penetración previa del material:

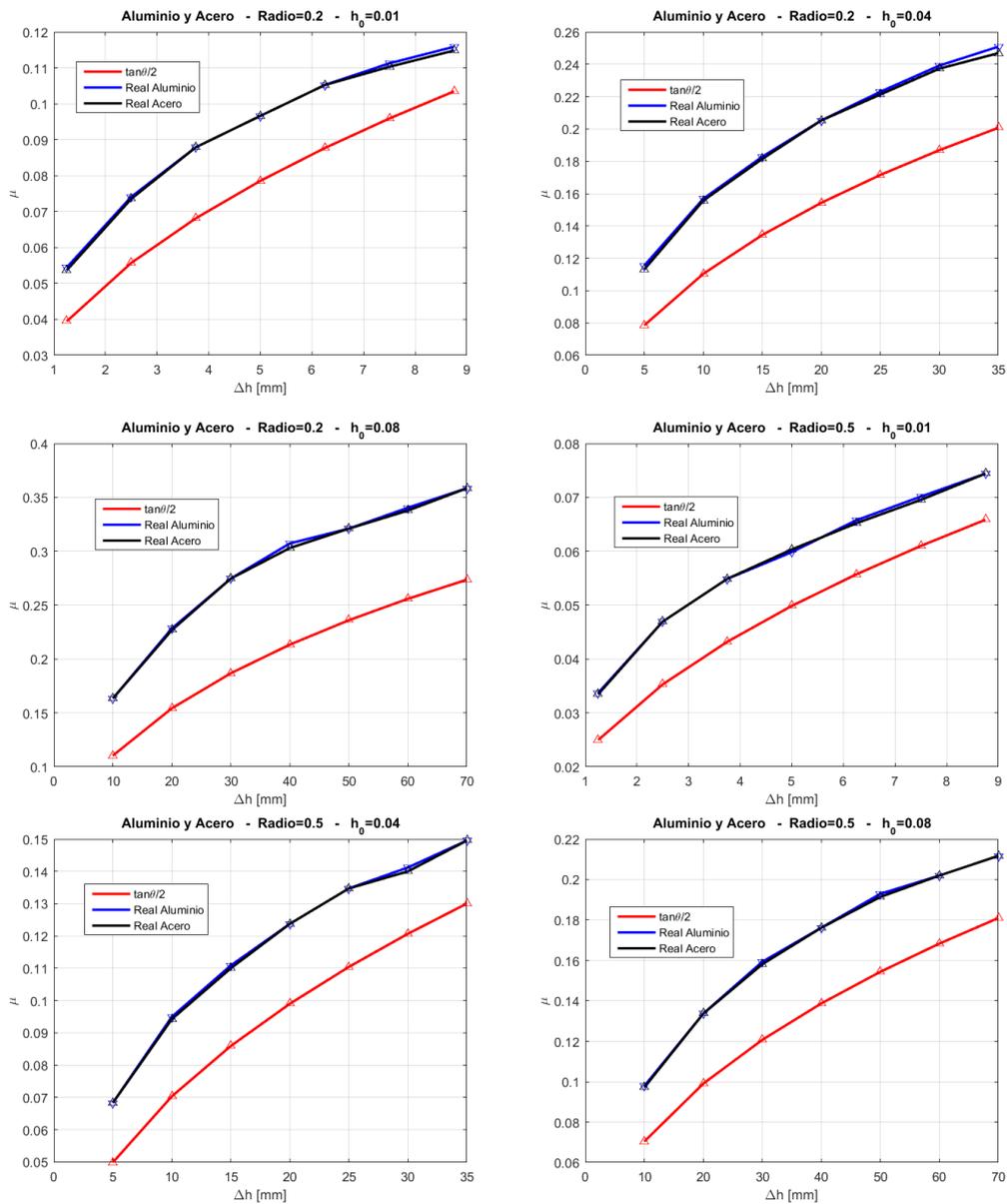


Ilustración 49. AF - AC -1-: μ frente a Δh

6.3. Conclusiones.

Se han estimado los valores del coeficiente de rozamiento mínimo de arrastre natural y forzado para varias configuraciones de chapa y rodillos mediante el método de los elementos finitos y comparados con los valores que arrojan las expresiones para los mismos recogidas en la bibliografía. Del estudio sacamos las siguientes conclusiones:

- Para Arrastre Natural:
 - El mallado adaptativo no influye en los resultados.
 - Con el modelo de elementos finitos realizado para la estimación del coeficiente de arrastre natural se obtienen valores similares a los obtenidos según la teoría, con discrepancias a un 7%.
 - Los errores más elevados se producen para valores de $\mu_{real} < \mu_{teórico}$.
 - También se ha observado que para pequeñas reducciones de espesor, $\mu_{real} < \mu_{teórico}$, y para grandes reducciones, $\mu_{real} > \mu_{teórico}$, encontrándose un rango entre $\mu = [0.17, 0.37]$ donde $\mu_{real} = \mu_{teórico}$.
 - Independencia del material a la hora de estimación del coeficiente de arrastre natural, ya que se obtienen prácticamente los mismos resultados para Aluminio y Acero.
- Para el modelo de Arrastre Forzado:
 - La malla adaptativa afecta significativamente a los resultados, reduciendo discrepancias en casi un 50%.
 - Las discrepancias con los valores teóricos son del orden del 25%, pudiéndose deber a problemas de contacto en el inicio de la laminación.
 - Debido a ello, se propuso un segundo modelo con el que eliminar dicho posible error de contacto, y los errores se redujeron considerablemente, obteniéndose errores muy bajos (menores al 10%) excepto para grandes reducciones de espesor, donde los resultados son similares a los del punto anterior (Errores en torno al 20-30 %).
 - Dichos errores pueden deberse a las grandes deformaciones y cambios de geometría que sufre el modelo, los cuales podrían ser tratados y reducidos con el uso más intensivo y específico de la malla adaptativa.
 - Independencia del material a la hora de la estimación del coeficiente de arrastre forzado.

6.3.1. Posibles soluciones de mejora.

Como se ha observado a lo largo del estudio del modelo de estimación del coeficiente de arrastre forzado, se han encontrado errores elevados cuya naturaleza se podía deber a varias razones:

- Teoría mal postulada.
- Errores computacionales: principalmente en el contacto de las superficies implicadas.
- Limitación computacional: En lo que respecta al número de elementos de la malla.

Por ello se plantean dos líneas de investigación futuras:

6.3.1.1. Mejora del mallado adaptativo.

Se comprobó que forzando la fluencia del material entre los rodillos (en un porcentaje muy pequeño) de manera previa al estudio de las condiciones de arrastre forzado se mejoraron los resultados considerablemente (salvo para Δh elevados), aunque habría que estudiar en mayor profundidad los efectos que tiene sobre el modelo y si introduce desviaciones importantes respecto a lo impuesto por el desarrollo teórico.

Es por ello que se plantean como líneas de investigación futuras observar el efecto que tiene el mallado adaptativo en el modelo de arrastre forzado en cuestión, puesto que variando los parámetros que definen dicho tipo de mallado se cree que se pueden mejorar considerablemente los resultados obtenidos.

De hecho los errores encontrados para reducciones de espesor elevadas se deben a que en dichos casos predomina la deformación no homogénea, que lleva asociada un cambio muy grande en la geometría de la preforma. Por lo tanto el mallado adaptativo es muy necesario en estos casos ya que se necesita de un remallado continuo de la geometría para que este se adapte a la geometría deformada para cada intervalo de tiempo que se crea necesario.

Por lo que se plantea que en futuros estudios se realice un remallado mas exhaustivo de la geometría, a intervalos de tiempo mas pequeños:

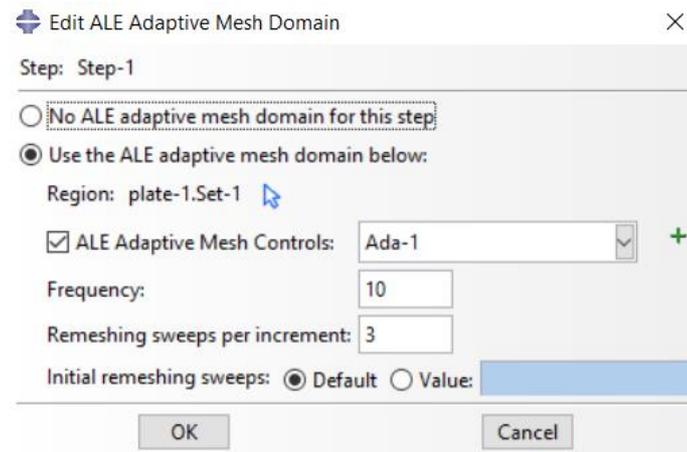


Ilustración 50. Parámetros de remallado.

Para lo cual habría que ir jugando con la **frecuencia y los remallados por incremento**.

Por otro lado se podría estudiar para cada caso los parámetros propios de la técnica ALE proporcionada por Abaqus:

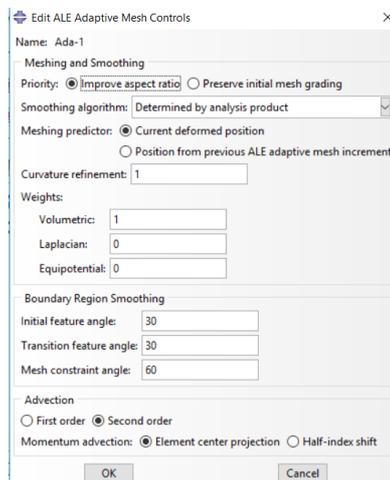


Ilustración 51. Parámetros de la técnica ALE de Abaqus.

Puesto que según el caso de estudio, pueden interesar más unos valores u otros.

Finalmente podría combinarse dicho estudio sobre los parámetros influyentes en la malla adaptativa con el estudio en profundidad de los parámetros influyentes en la condición de contacto, *overclosure controls*, puesto que dichos parámetros permiten restringir los errores iniciales que pueden darse cuando se impone el contacto entre geometrías de cierta complejidad, evitando la penetración de unos *Parts* en otros así como “falsos contactos iniciales”.

6.3.1.2. Estudio de casos extremos.

Por otro lado, con el fin de demostrar la veracidad de la teoría, se plantea el cálculo del coeficiente de arrastre forzado para casos extremos de poco interés práctico, pero que pueden dislumbrar si la teoría postulada es correcta o si no lo es, posibilitar la búsqueda de una teoría alternativa.

Dicho caso extremo consiste en el cumplimiento de las dos hipótesis planteadas para la obtención del coeficiente de arrastre forzado:

- Considerar el material perfectamente elástico, lo cual encajaría cómo se vio en el **capítulo 1** con la hipótesis de presión constante sobre la placa a lo largo del proceso de laminación.
- Ángulos de contacto pequeños.

ANEXO A. MODELO PASO A PASO

En el presente Anexo se van a modelar con ABAQUS los dos problemas que se quieren resolver con el proyecto: Se van a simular las condiciones de arrastre natural y arrastre forzado para la comprobación del cumplimiento o no de los valores dados por la teoría.

Antes de entrar en profundidad en cada modelo, cabe destacar algunos aspectos comunes que presentarán.

Como es sabido el proceso de laminación es una sencilla técnica de fabricación que se utiliza para transformar placas prefabricadas por otros procesos en otras con formas o dimensiones más adecuadas para posteriores procesos de fabricación.

Dicho proceso es principalmente quasi-estático, ya que la laminación se produce normalmente a velocidades bajas, del orden de 1 m/s, por lo que los efectos de inercia no son muy importantes. Esto a priori nos indicaría que realizar y resolver el modelo como un problema estático no debería suponer un problema, y los resultados no deberían diferir mucho de la realidad.

Sin embargo, debido al comportamiento del material y los fenómenos no lineales que tienen lugar en el proceso (contacto y fricción), se va a utilizar un modelo dinámico, y por lo tanto se utiliza el solver explícito del que dispone abaqus para dicho tipo de problemas.

A continuación se proceden a explicar los modelos realizados para una placa genérica de x mm de altura inicial y con reducción de altura del $y\%$ m donde la hipótesis principal a tener en cuenta es que se va a considerar *plane strain*, debido a que como ya se comentó en el capítulo 1, se puede formular el problema asumiendo que existen un estado de deformación plana.

A.1 Arrastre natural

Se va a explicar que modelo se ha llevado a cabo para simular las condiciones de arrastre natural. Cabe destacar que en primer lugar se realizó el modelo con el interfaz gráfico de abaqus, hasta que se consiguió un modelo con el que se obtenían resultados aceptables y que serviría de base para realizar el posterior estudio paramétrico en PYTHON.

Se van a explicar a continuación todos los pasos seguidos para la obtención del modelo así como los detalles que se consideren de interés.

A.1.1. Modelo y Geometría.

Se va a modelar solo la mitad del problema debido a la simetría de geometría y de cargas que este presenta. En cuanto a la geometría, esta consistirá en un rectángulo con las dimensiones adecuadas, que es el material que se quiere laminar, y un rodillo, que es el encargado de realizar dicho proceso de laminación.

Antes de comenzar a explicar cómo se ha desarrollado el modelo cabe destacar que lo más importante del problema es determinar el punto de contacto exacto entre el cilindro y la placa de material, ya que dicho punto de contacto viene dado por la teoría de la laminación y debe ser el adecuado para que se de la condición de arrastre natural.

Se ha procedido de dos formas para asegurar que el contacto entre cilindro y material se produzca en dicho punto, y aquí es donde interviene la geometría del rodillo:

- Dividir el arco de 90° en dos arcos: El primero de ellos empieza en el punto final de contacto entre rodillo y placa, y acaba en el punto de contacto a ángulo θ respecto al primero (punto de contacto para que se de arrastre natural). El segundo arco comenzaría en el segundo punto mencionado anteriormente, y acabaría a 90 grados.
- Arco de 90° y *Datum Point* en el punto de contacto deseado.

Se va a explicar en primer lugar el segundo de los casos mencionados, dejando el otro para el modelo de simulación de condición de arrastre forzado. Los pasos llevados a cabo son los que siguen:

1. Iniciar sesión en ABAQUS/CAE. Se creará un modelo automáticamente (model-1). Podemos cambiar el nombre de dicho modelo aunque en este caso se dejará el que viene por defecto.
2. Crear un part. Como se explicó anteriormente ABAQUS permite trabajar desde el árbol, la barra de herramientas, y con los iconos que aparecen en la ventana de trabajo. De aquí en adelante se va a trabajar indiferentemente de estas tres formas.

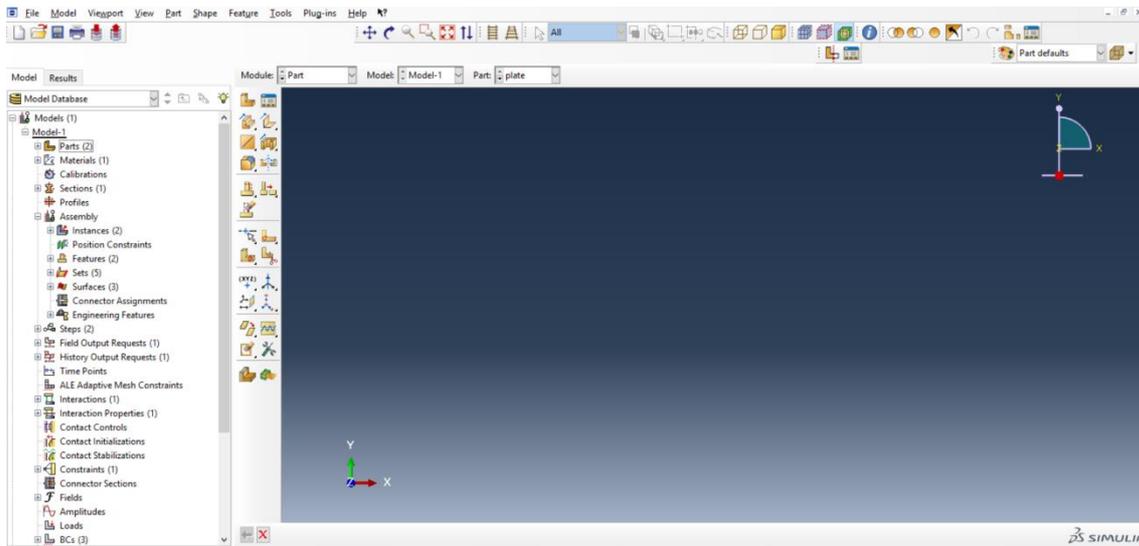


Ilustración 52. Interfaz de Abaqus.

Por lo tanto o bien se hace doble clic en *Model* → *Part*, se usa el icono correspondiente *Part Manager*, o se abre el manager desde la barra de herramientas *Part* → *Part Manager*. Se crea un nuevo Part llamado Plate, y se escoge la opción de *2D planar* puesto que por ahora solo se va a trabajar en dos dimensiones. El tipo será *Deformable* y basado en *Shell (placa)*. En cuanto a las dimensiones del *sketch* se escoge en función del tamaño de la geometría que se va a crear, y en nuestro caso se escoge un *Sketch Area* de 0.1.

3. Se selecciona *Create lines: Rectangle (4 Lines)*, y se introducen las coordenadas del material de partida, teniendo en cuenta que se está usando la simetría del problema para simplificar el modelo, por lo que las dimensiones serán la mitad de las de el material de partida. Por lo tanto se introducen las coordenadas (0,0) y (-1.2L, -h/2) = (-0.05, -0.005), donde se supone que la longitud de la placa es un 20% superior a la longitud característica de contacto, lo cual no afectará al problema y simplificará su resolución.

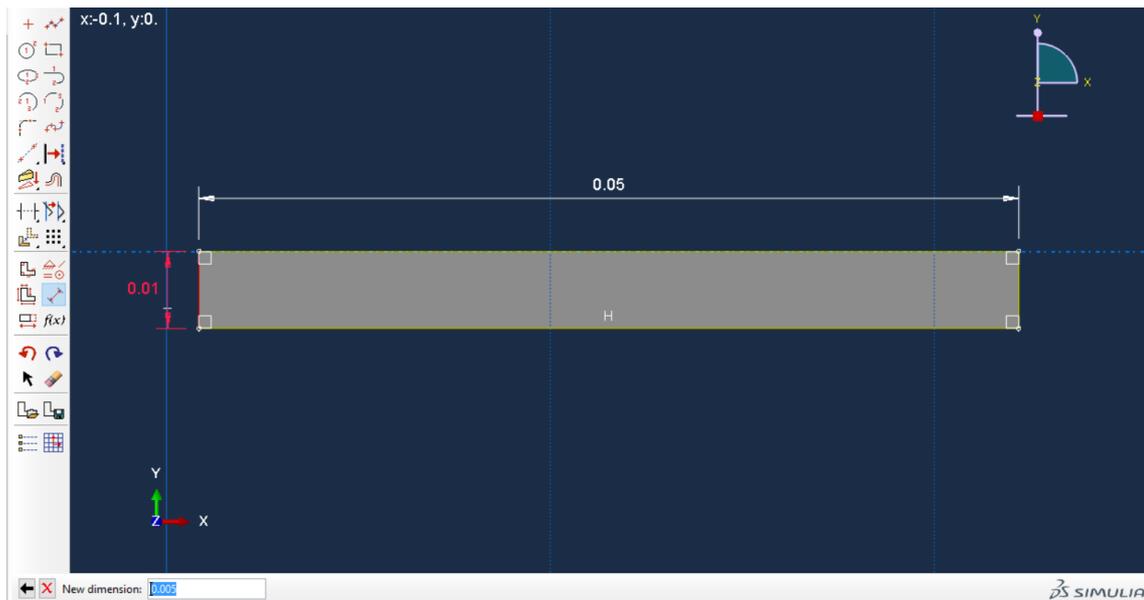


Ilustración 53. Creación de la geometría de partida.

4. En el árbol del modelo, se hace doble clic en Part, y se escoge la opción de modelado espacial *2D Planar*, tipo *Analytical rigid*. En cuanto a la dimensión del sketch, se escoge *Sketch Area* de 0.5.
5. Finalmente se crea un arco con *Create Arc*, centrado en el punto (0,0), y que va desde (0,-R), a (-R, 0).
6. A continuación es necesario crear un punto de referencia que servirá a posteriori para definir algunos aspectos del *step*, y será el punto en torno al que se centran las condiciones de contorno y el análisis en general. Para ello, desde la barra de herramientas, se hace clic en *Tools* → *Reference Point*, y se escoge como punto de referencia el centro del arco.

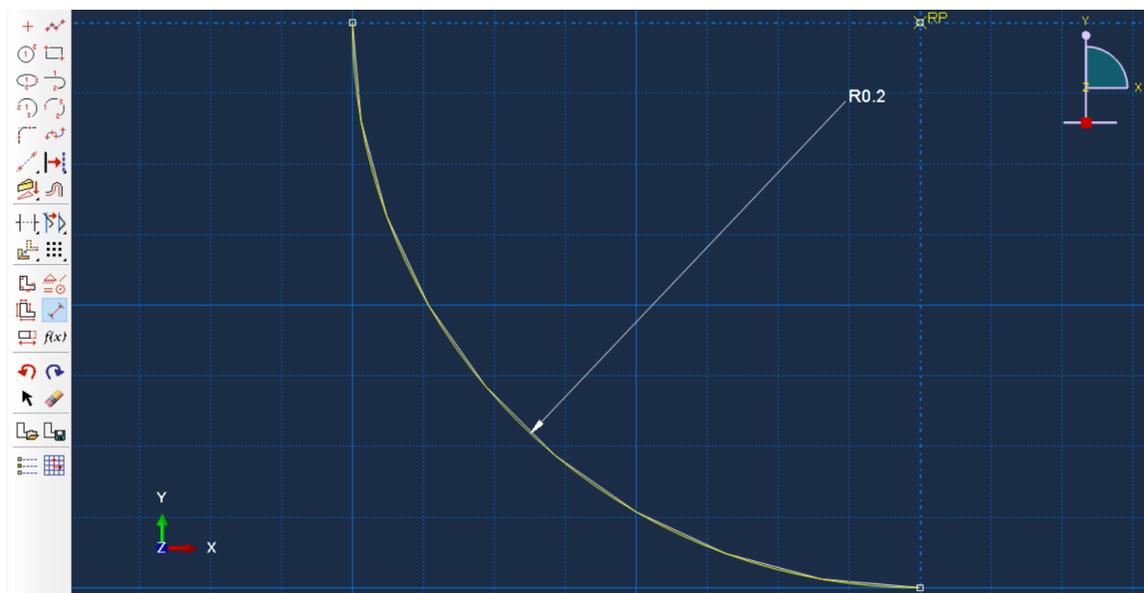


Ilustración 54. Creación del arco que conforma el rodillo.

A.1.2. Material y Sección.

Como se describió en el **capítulo 4**, se va a realizar un estudio para distintos materiales. En este primer caso se va a realizar para el aluminio utilizado en la industria aeronáutica, AL-7075.

Cabe destacar que se va a considerar un comportamiento elástico-plástico, por lo que habrá que introducir diferentes propiedades tanto elásticas como plásticas (**Vease ANEXO D**).

Para definir el material y la sección se procede de la siguiente manera:

1. Doble clic en el árbol de modelo en *Materials*. Se crea un nuevo material denominado Aluminio-7075, y se introducen las siguientes propiedades:

- Densidad: 2810 kg/m^3
- Modulo de elasticidad: $7.2 \text{ E}10 \text{ N/m}^2$
- Coeficiente de Poisson: 0.33
- Deformaciones plásticas y tensiones asociadas: En este caso se han tenido en cuenta 20 puntos de deformaciones, lo cual se considera suficiente para reflejar con claridad el comportamiento plástico del material. Además se ha variado el número de puntos comprobándose que no se producen alteraciones en los resultados (con 10, 20, y 40 puntos).

Yield Stress (Pa)	Plastic Strain
418110000	0
579695510	0.05
607229461	0.1
625462338	0.15
639455160	0.2
650955905	0.25
660794895	0.3
669437301	0.35
677172075	0.4
684191979	0.45
690632530	0.5
696592913	0.55
702148072	0.6
707356128	0.65
712263132	0.7
716906240	0.75
721315901	0.8
725517400	0.85
729531987	0.9
733377701	0.95
737070000	1

Tabla 5. Propiedades Plásticas del Aluminio 7075

2. Para crear una sección, se hace doble clic en *Sections* y se crea una sección sólida homogénea con espesor *Default thickness* de 1.0.
3. Finalmente, se asocia la sección a la geometría del material de partida mediante *Section Assignments*, ya sea como se mencionó antes usando el árbol o los iconos de ayuda del programa.

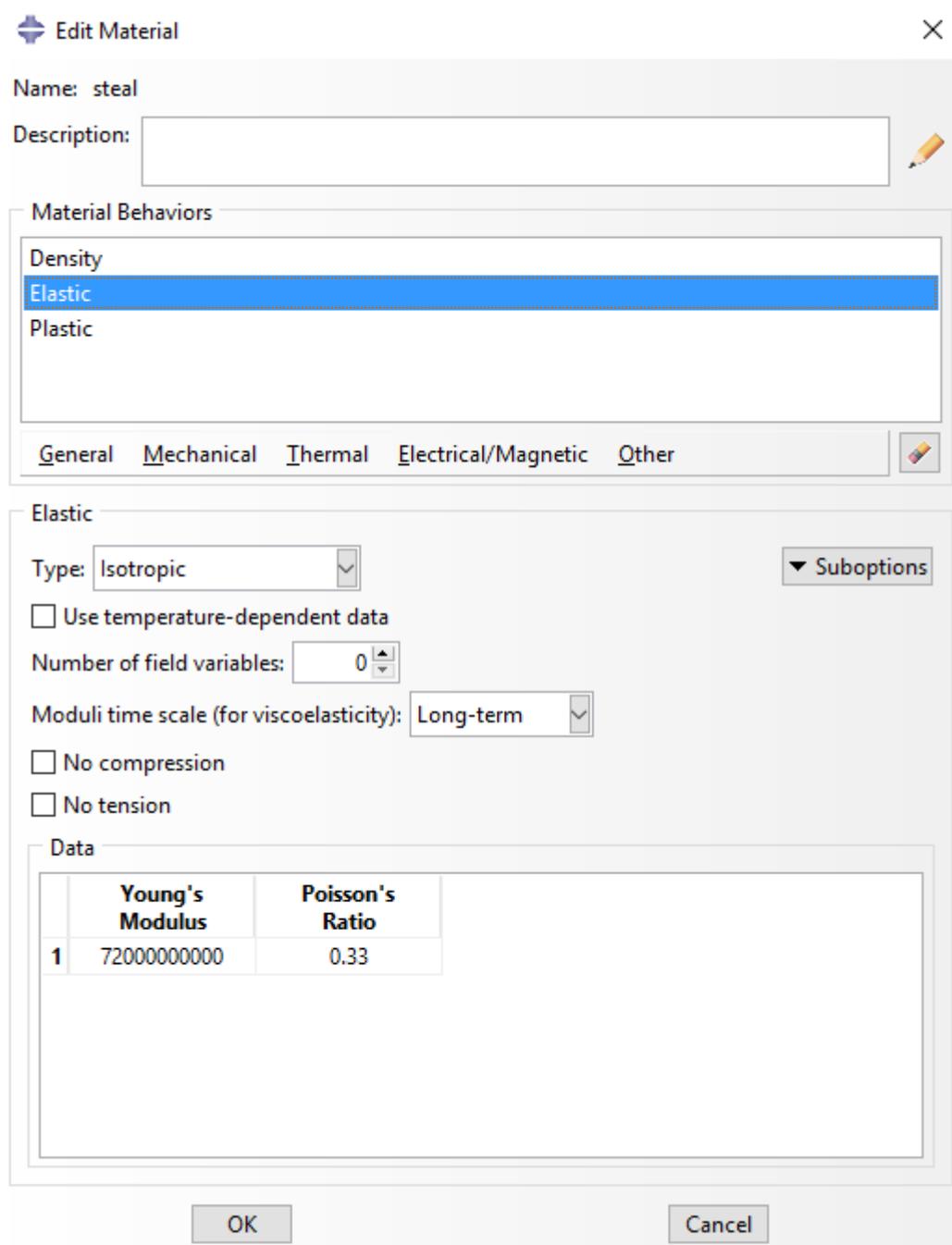


Ilustración 55. Definición de las propiedades del material en Abaqus.

A.1.3. Assembly.

Una vez definida la geometría y el modelo con los que se va a trabajar, se procede a montar el modelo de manera que refleje de la manera más fiable posible lo que sucede en la realidad en el proceso de laminación.

Puesto que el modelo que se está explicando es el que consta de un solo rodillo, es necesario definir el punto del

rodillo en el que debe producirse el primer contacto entre rodillo y material para que se den las condiciones de arrastre natural. Para ello se va a usar una herramienta de abaqus llamada *Datum point*, que permite introducir un punto que sirva como referencia, pero el cual no forma parte de la geometría del modelo, es decir, no es un punto físico del rodillo o material de partida, por lo cual no requiere que se apliquen restricciones ni condiciones de contorno.

Para ello, se hace clic en *Tools* → *Datum* en la barra de herramientas y aparece la ventana que se muestra a continuación:

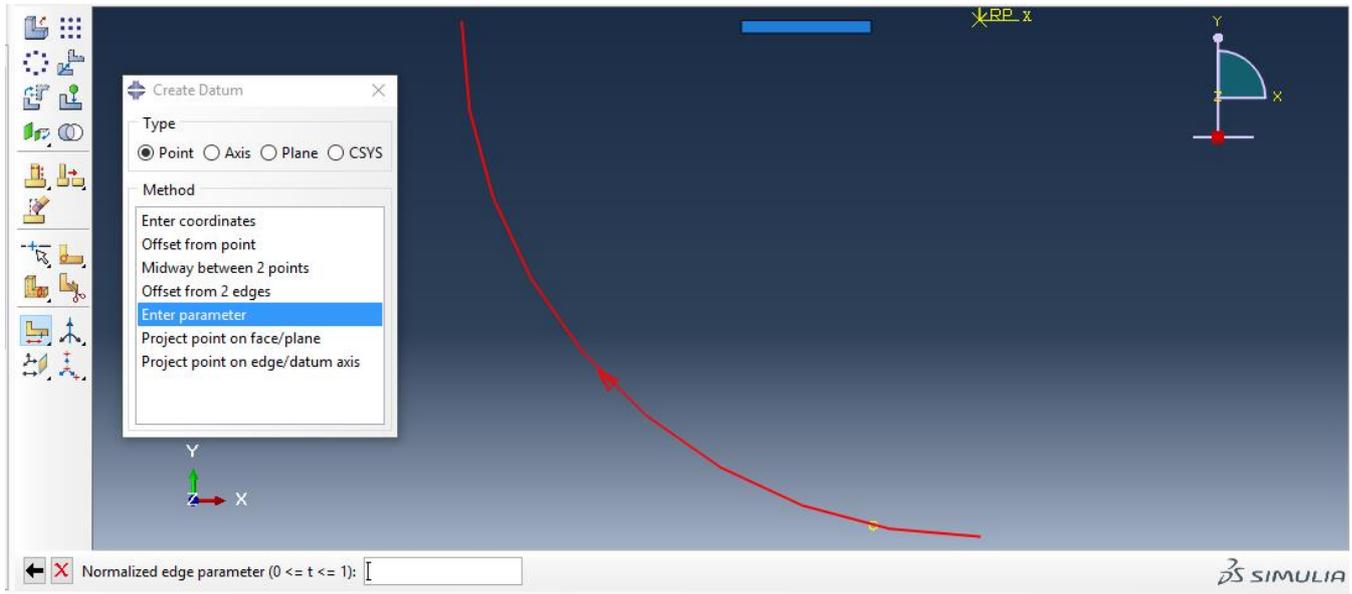


Ilustración 56. Datum Point.

Como se puede observar ABAQUS permite introducir un punto, eje, plano, o sistema de referencia, existiendo diferentes métodos para introducir el objeto de interés.

Para el caso del punto, se puede añadir directamente por coordenadas, mediante la distancia a un punto, como punto intermedio entre dos puntos, mediante proyección en una cara o plano, o introduciendo un parámetro de interés.

Dicha opción es la que más nos interesa en nuestro caso ya que a pesar de que se conocen las coordenadas exactas del punto de contacto, debido a pequeños errores numéricos puede que el punto que se defina no forme parte del rodillo, y aunque el error que presente sea del orden de la micra, no se produciría la laminación puesto que las partes de interés de nuestro problema (rodillo y plato) no estarían inicialmente en contacto.

Por ello se va a usar la opción *Enter parameter*, ya que permite seleccionar una eje de la geometría existente, y mediante la definición de un parámetro “t”, definir en qué posición dentro de dicho eje se encuentra el punto que se quiere definir.

En el caso del problema que se está tratando de resolver, el razonamiento a seguir es el siguiente:

- Es conocido el ángulo que forma la vertical que pasa por el centro del arco, y el punto de contacto deseado para que se produzca el arrastre natural:

$$\tan \theta = \sqrt{\frac{\Delta h}{R}} \rightarrow \theta = \tan^{-1} \left(\sqrt{\frac{\Delta h}{R}} \right) \quad (A - 1)$$

- Por lo que si el parámetro t está definido por defecto de 0 a 1 en sentido de las agujas del reloj, el punto inicial se corresponderá con un valor de t=0 y el punto final correspondiente con un ángulo recorrido total de $\pi/2$ se corresponde con un valor de t=1.
- De manera que para obtener el punto de contacto exacto y por ende el valor del parámetro t a introducir, haciendo una sencilla regla de tres se obtiene:

$$t = \frac{\theta}{\pi} (A - 2)$$

Una vez conocido el punto de contacto, se hace clic en el icono de *Translate instance*, se selecciona el *part* a trasladar, que en este caso es el plato, se selecciona como punto inicial el vértice superior derecho puesto que es el primer punto que entrará en contacto con el rodillo, y como punto final el *Datum Point* creado previamente:

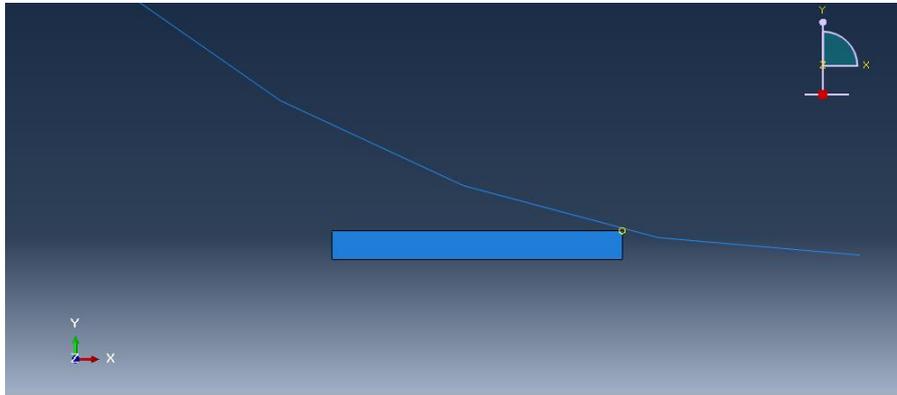


Ilustración 57. Ensamblado del modelo.

A.1.4. Superficies y Sets.

Antes de proceder con la definición del caso de estudio y de las interacciones que tienen lugar entre los componentes del modelo es conveniente definir diferentes *Sets* o Subpartes del modelo que facilitarán en gran medida la aplicación de condiciones de contorno y de cargas o interacciones en determinadas zonas.

1. Se despliega el contenido de *Assembly* en el árbol del modelo, y se hace doble clic en *Sets* para crear las ayudas geométricas correspondientes:

- Bottom: El eje inferior del plato.
- Plate: El plato entero.
- RefPnt= Punto de referencia del rodillo creado anteriormente.

2. A continuación se van a crear algunas superficies de interés que nos serán de gran ayuda a la hora de definir las interacciones de contacto entre ambas partes que conforman el modelo:

- Top: El eje superior y el eje derecho del plato. Es muy importante escoger ambos ejes, ya que si solo se elige el eje superior, no se tendrá en cuenta que cuando la placa se deforme a causa de la presión ejercida por el rodillo, el material situado en el eje derecho pasará a entrar en contacto con el rodillo, por lo que si no se define de esta manera, el resultado no será real, y además pueden aparecer diversos errores en el análisis.
- Roller: Todo el eje del rodillo.

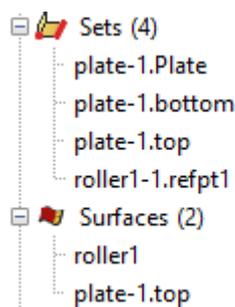


Ilustración 58. Creación de los Sets.

A.1.5. Step.

A continuación se va a definir uno de los puntos más críticos a la hora de conseguir que el análisis converja de la manera más eficiente computacionalmente. Se ha tratado de simplificar y hacer que el análisis sea lo más rápido posible, sin embargo, algunas de las muchas utilidades que presenta ABAQUS para reducir el tiempo de simulación, pueden alterar los resultados obtenidos, así como dificultar la automatización del modelo, que es uno de los objetivos planteados en el presente proyecto con vistas a su posterior programación en PYTHON.

Por otro lado se van a aceptar en gran medida los parámetros que Abaqus sugiere por defecto, de manera que el único parámetro que debemos tener en cuenta a la hora de realizar el análisis dinámico explícito es el tiempo de análisis que se va a llevar a cabo. Para ello hay que estudiar algunos parámetros propios del proceso de laminación que indican cuanto tiempo tarda en la realidad en producirse la laminación. La manera de proceder es la siguiente:

1. Se crea un *Dynamic Explicit Step* haciendo doble clic en steps en el árbol de modelo con un periodo de 0.1 segundos.

Se ha escogido dicho periodo teniendo en cuenta que:

- Velocidad de giro del rodillo es de 2π rad/s.
- Radio= 0.2 m.
- Longitud de contacto L=0.095 m.

Por lo tanto, teniendo en cuenta que el rozamiento entre rodillo y la placa es el adecuado para que se produzca el arrastre natural, y que este sucede instantáneamente en el momento en que entran en contacto ambos, el tiempo característico del fenómeno viene dado por la velocidad tangencial en el punto de contacto:

$$\omega = 2\pi \frac{\text{rad}}{\text{s}} \rightarrow V = 2\pi * 0.2 = 1.25 \frac{\text{m}}{\text{s}} \rightarrow \text{Tiempo} = \frac{\text{Longitud de contacto}}{V} = \frac{0.095}{1.25} = 0.076 \quad (\text{A} - 3)$$

Puesto que se van a realizar diferentes modelos, con algunos de ellos de mayor tamaño y con Longitud de contacto mayor, se va a escoger un tiempo de 0.1 segundos, el cual observando el resultado obtenido se estima adecuado para el estudio que se pretende llevar a cabo.

2. A continuación se va a crear un historial de resultados en el que se obtendrán todos aquellos resultados que consideremos necesarios y de interés. En este caso, puesto que el proyecto consiste en un estudio paramétrico del proceso de laminación para observar como varía el coeficiente de rozamiento, no se requiere gran cantidad de datos de salida. Por ello solo se va a requerir que ABAQUS calcule las reacciones que sufre el rodillo, con vistas a poder representar algunas gráficas y obtener así conclusiones.

Para ello se hace doble clic en *History Output Request*, y lo editamos de la siguiente manera:

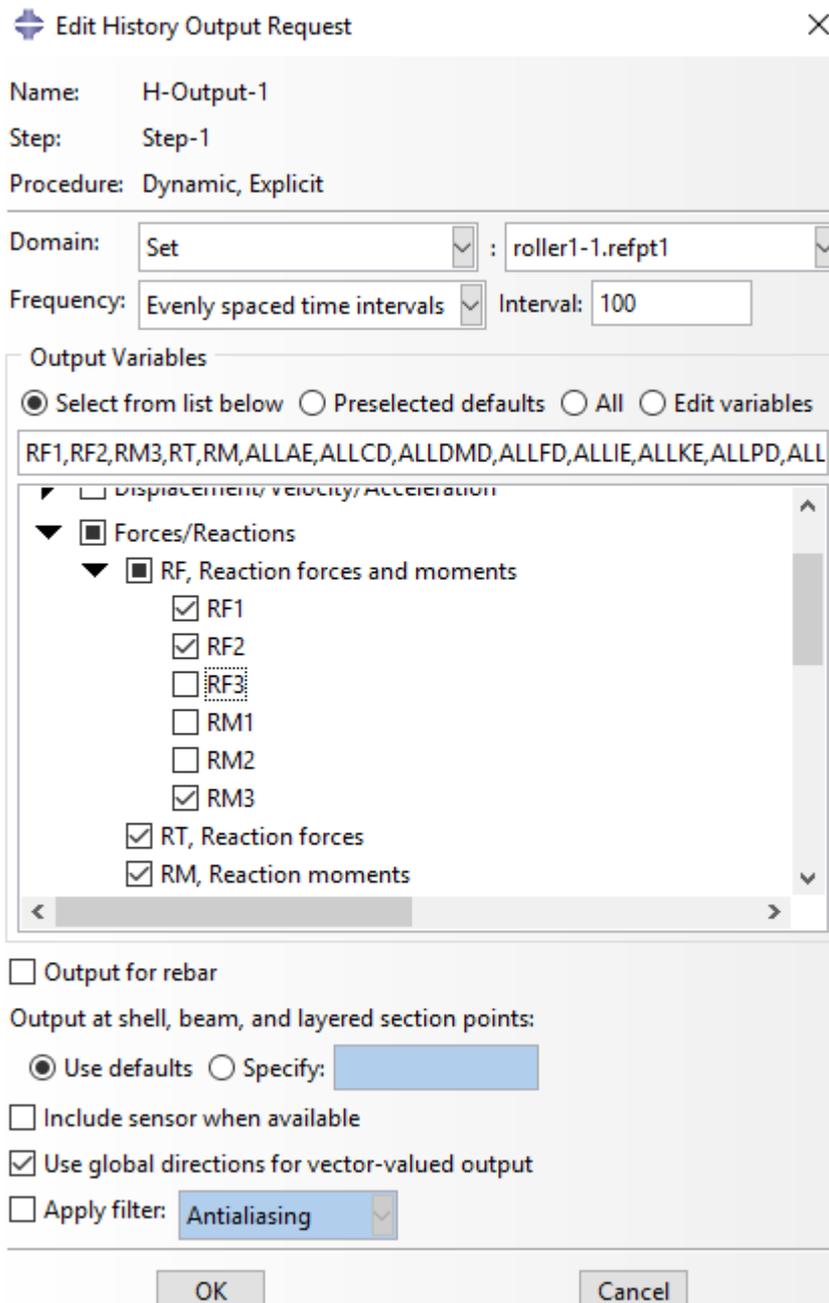


Ilustración 59. History Output.

- Se cambia el dominio a *Set* y se selecciona el punto de referencia del rodillo *Refpnt*.
- Se divide el intervalo de tiempo de manera equiespaciada en 100 intervalos.
- Se selecciona entre las variables de salida las reacciones que sufrirá el rodillo, que se encuentran en *Forces and Reactions* → *Rf*, *Forces/Reactions* → *RF1, RF2, RM3*.

Finalmente, antes de proceder a definir las condiciones de contacto que rigen el problema, se va a introducir en dicho apartado algunos aspectos relevantes sobre el tipo de mallado que se va a utilizar.

Para el análisis, como ya se explicó en el capítulo 4 y se ampliará en el anexo C se va a utilizar una malla adaptativa. Para configurar dicho mallado, se hace clic en *Others* en la barra de herramientas y seleccionamos *Adaptative Mesh Domain* → *Manager* → *Edit*:

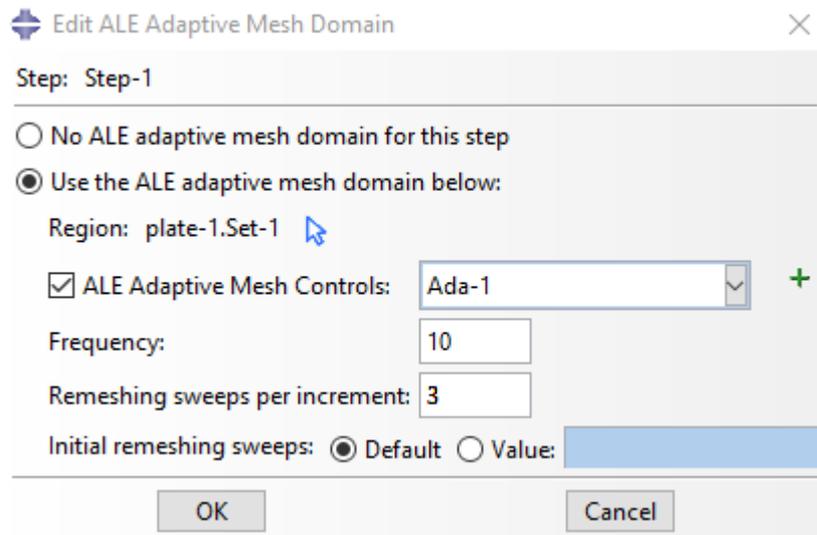


Ilustración 60. Definición de la malla adaptativa.

- Se selecciona *Use the ALE adaptive mesh domain for this step*. Es decir, permitimos a ABAQUS que use mallado de tipo adaptativo en el step-1 del modelo.
- Se selecciona como región de aplicación de dicho mallado la plancha de material a laminar.
- Se activa los controles de malla adaptativa *ALE adaptive Mesh Controls*, creándolos con nombre *Ada-1* y aceptando los parámetros que abaqus propone por defecto.
- Finalmente se selecciona una frecuencia de cada diez intervalos y 3 remallados por incremento en *Remeshing sweeps per increment*.

A.1.6. Contact.

Este apartado es el más importante de todo el modelo puesto que es el que se usará para determinar el coeficiente de rozamiento necesario para que tenga lugar la condición de arraste natural. A continuación se explica paso a paso como proceder:

1. Lo primero que se va a hacer es definir el rozamiento que va a existir entre rodillo y material a laminar. Para ello se hace clic en *Interaction Property Manager*, y se crea una nueva propiedad. Como lo que nos interesa es el comportamiento tangencial existente, se escoge como opción de propiedad de contacto *Tangential behavior*. Como método de contacto, por encontrarnos en un análisis explícito, el más adecuado es el método *Penalty*, por lo cual se selecciona dicho método.

Finalmente se dejan todos los parámetros restantes por defecto salvo el coeficiente de fricción, que es lo que se quiere hallar con la realización del proyecto. Como valor inicial del mismo, se tomará el de la teoría de laminación para el caso del arraste natural, ya que lo que se quiere demostrar es la fiabilidad de dicha teoría. Por ello se toma un valor de:

$$\mu = \tan\theta = \sqrt{\frac{\Delta h}{R}} = 0.201 \quad (A - 4)$$

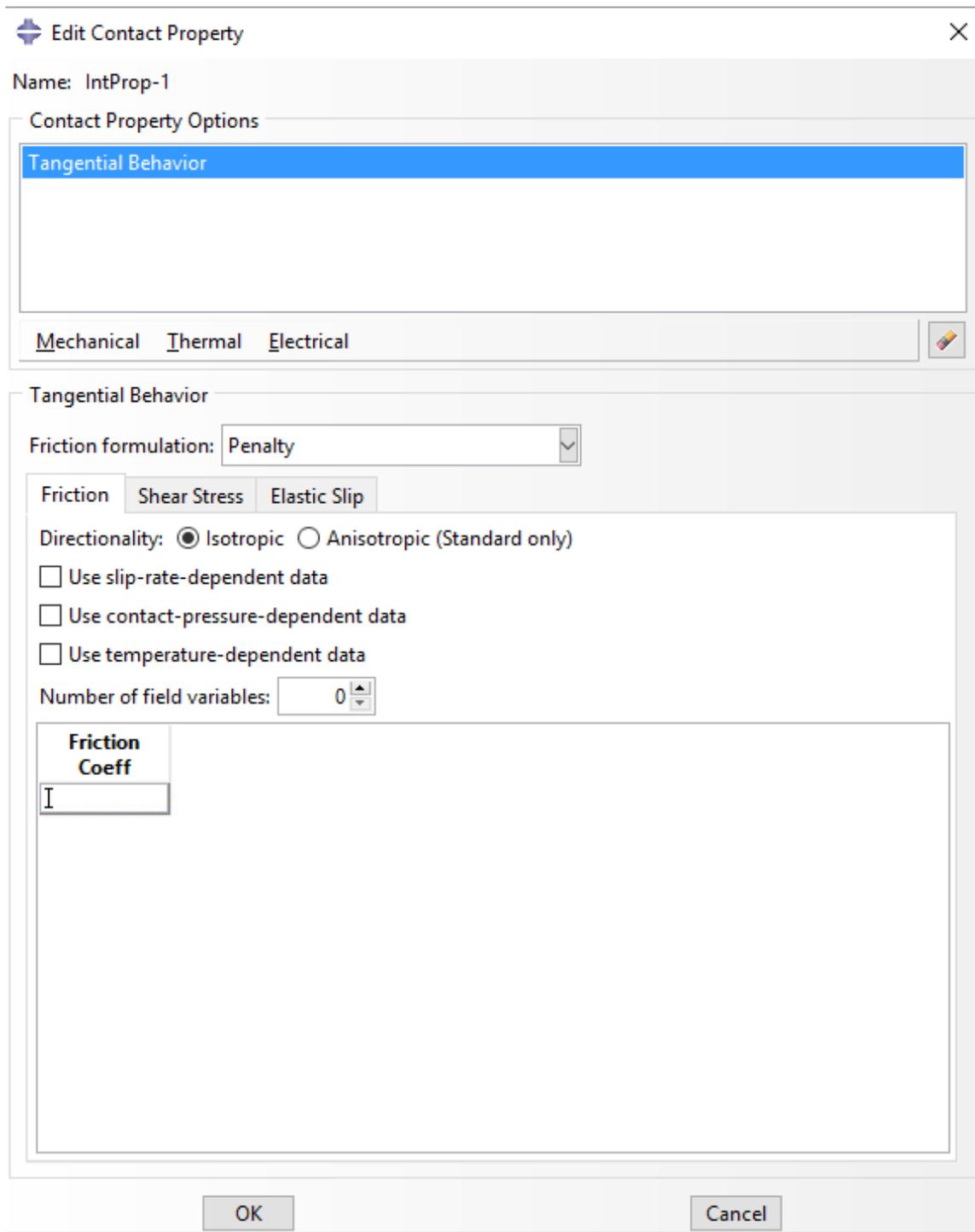


Ilustración 61. Introducción del valor del rozamiento.

2. A continuación se va a crear una interacción entre rodillo y material, y dicha interacción no va a ser otra que el contacto entre ambos cuerpos. Para ello se hace clic en *Create Interaction*, y se selecciona *Surface-to-surface contact (Explicit)*.

Se escoge como primera superficie o *Master surface* el rodillo, y como segunda superficie o *Slave surface* el material. En cuanto a la formulación escogida, se puede elegir *kinematic contact method* o *penalti method* indiferentemente, aunque es cierto que el segundo de los mencionados es el que se aconseja para este tipo de análisis. Finalmente se aceptan el resto de parámetros por defecto y crea dicha interacción.

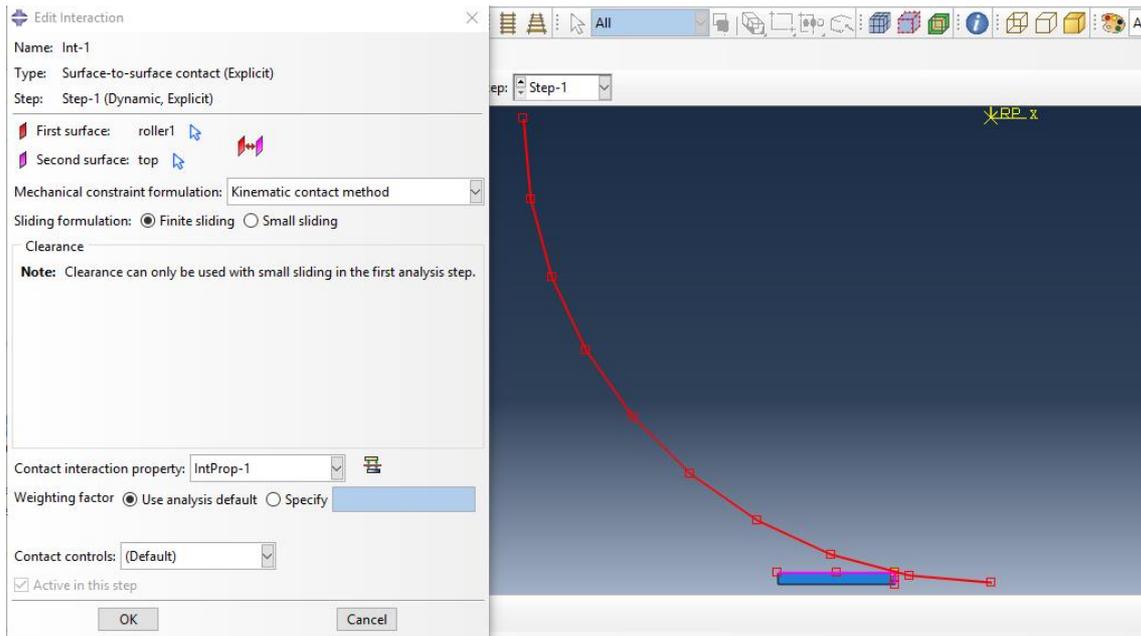


Ilustración 62. Contacto entre rodillo y placa.

A.1.7. Condiciones de contorno.

Este es el último paso que hay que realizar para definir completamente la física del modelo y pasar a procesar el mallado del mismo. A continuación se muestran las condiciones de contorno que se van a aplicar y el modo de hacerlas con ABAQUS:

Nombre	Step	Region	Condición
Bottom	Inicial	Bottom (Base)	Y-Simetría
FixRoll	Inicial	RefPnt	U1=U2=0
Vroll	Step 1	RefPnt	VR3= 2pi rad/s

Tabla 6. Condiciones de Contorno.

1. Hacer doble clic en *Manager* en el apartado *BC's* del árbol del modelo. Se desplegará un menú con todas las condiciones de contorno que se pueden aplicar según el tipo de step que haya sido definido.

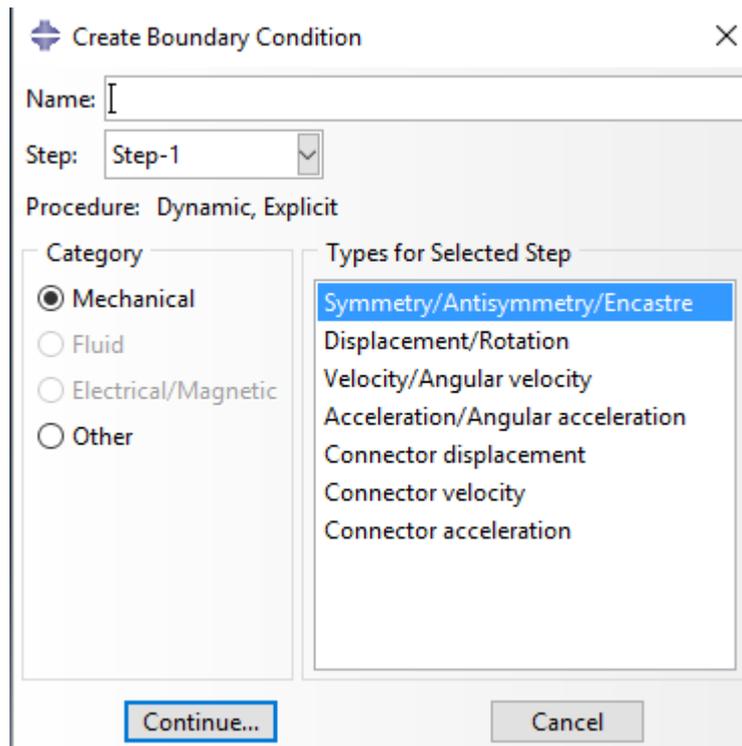


Ilustración 63. Condiciones de contorno.

2. Para la condición de simetría, se selecciona para el step inicial el tipo *Symmetry/Antisymmetry/Encastre* → *Y-Symmetry*, y se selecciona la región de aplicación, que no es otra que el Set *Bottom* creado anteriormente.
 3. Se crea una condición de contorno que limite el desplazamiento de los rodillos y solo permita su rotación en torno al eje Z. Para ello, en el step inicial se selecciona el tipo *Displacement/Rotation*, se escoge como región e aplicación el punto de referencia creado para el rodillo en pasos anteriores, y se hace clic en U1, U2, lo cual pondrá dichos valores a cero y los proparará para el *Step 1*, permitiendo solo la rotación del rodillo.
 4. Finalmente, se crea la condición de contorno correspondiente a la rotación del rodillo. Esta se va a crear en el *Step-1*, puesto si prestamos atención, las dos condiciones anteriores eran condiciones iniciales del problema referente a desplazamientos. Sin embargo, esta se crea en un *Step* separado por que la rotación del cilindro es propia del problema en sí.
- Para ello se selecciona el *Step 1*, *Velocity/Angular velocity*, y se introduce en VR3 el valor de la velocidad deseado (V1, V2 no se tocan puesto que están restringidos los desplazamientos en dichas direcciones).

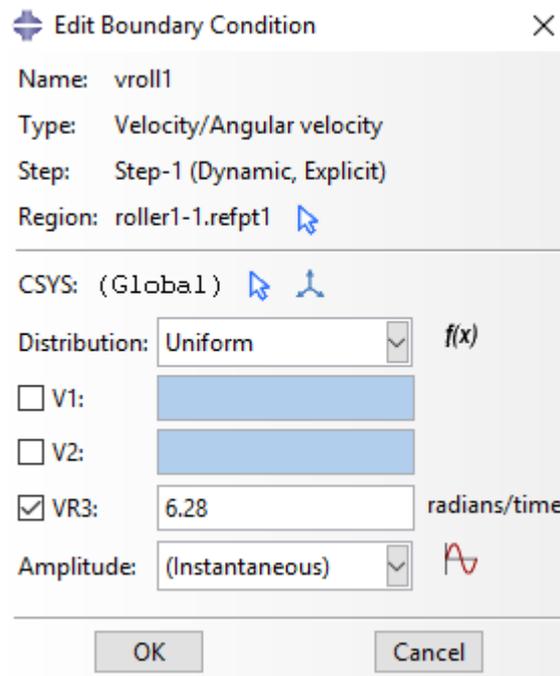


Ilustración 64. Velocidad angular de los rodillos.

3. Finalmente se va explicar cómo introducir una condición inicial en velocidad a la placa de material que va a ser laminado. Cabe destacar que dicha condición de contorno es opcional y que no influirá en el resultado del problema, aunque si hará que el análisis vaya más rápido puesto que elimina la discontinuidad en el campo de velocidades que sufre la placa, ya que parte de una velocidad nula a adquirir una velocidad de salida en cuanto entra en contacto con el rodillo.

Para ello, en el árbol del modelo se hace doble clic en *Fields*, y en la ventana de diálogo que aparece, se selecciona el Step inicial, *Mechanical category* → *Velocity type*. Se escoge la región a la que se quiere dar dicha velocidad inicial, que es la placa de material, y se introduce la velocidad deseada.

Cabe destacar que la discontinuidad en el campo de velocidades se produce en el momento en que se inicia el proceso de laminación en el punto de contacto entre rodillo y placa, donde la velocidad es igual a la velocidad tangencial asociada a la velocidad angular del rodillo. Por lo que un valor adecuado para dicha velocidad inicial, es por lo tanto, el valor de la velocidad tangencial en el punto de contacto:

$$V_{inicial} = \omega * R = 2\pi * 0.2 \sim 1.25 \quad (A - 5)$$

A.1.8. Mallado.

Este es el último apartado en lo que se refiere a definición del modelo, ya que posteriormente se procede a definir el análisis y la visualización de resultados.

Como se introdujo en capítulos y apartados anteriores se va a utilizar un mallado adaptativo, para un caso de deformación plana.

1. Para introducir dicha característica a nuestro mallado, es necesario definir qué tipo de elemento se va a utilizar. Para ello, en la barra de herramientas principal del programa, se selecciona *Mesh* → *Element type*, se indica que se está realizando un análisis dinámico de tipo explícito, y se selecciona la hipótesis de *Plane Strain*. El elemento asociado a las características impuestas es el elemento CPE4R o CPE3 (según se usen elementos cuadrados o triangulares).

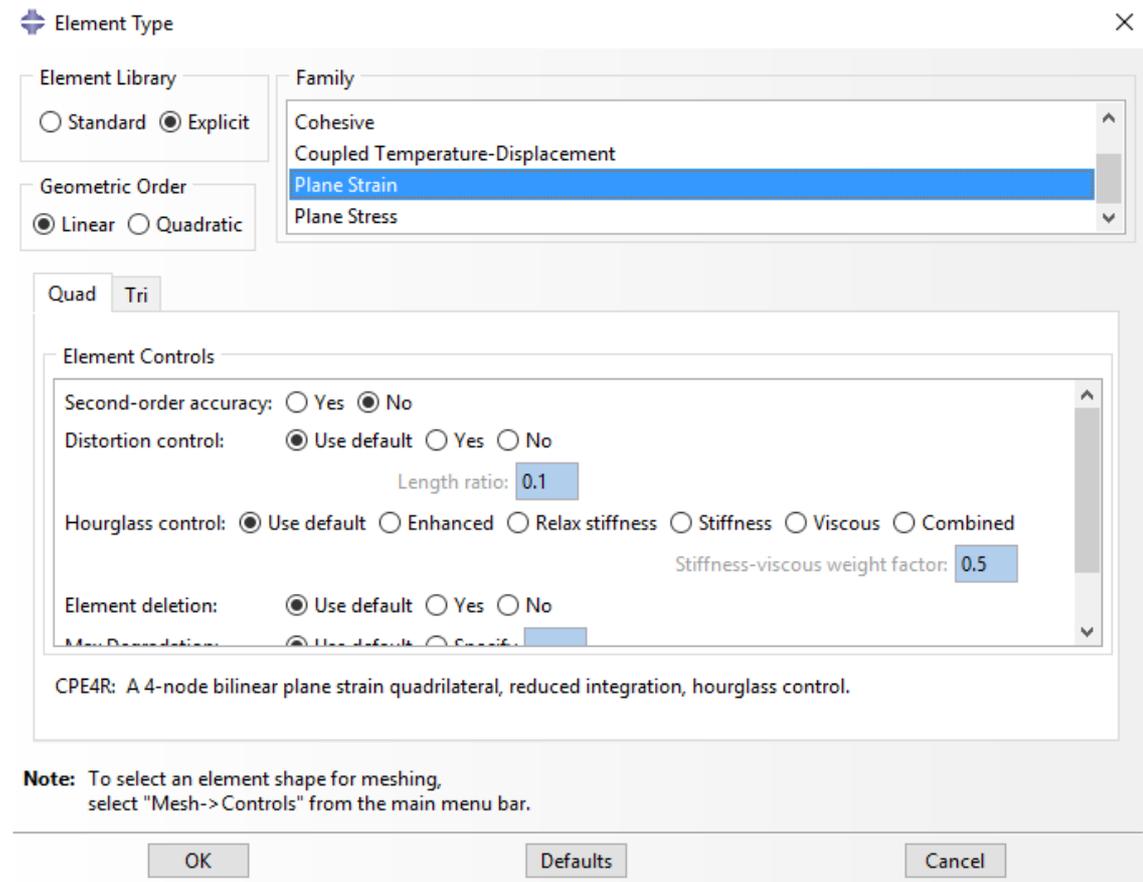


Ilustración 65. Tipo de elemento escogido.

2. A continuación también se puede imponer que el mallado sea estructurado o no, lo cual es opcional ya que para la geometría de partida un mallado libre o menos estructurado también dar buenos resultados y no induce a error. Lo que si hay que tener en cuenta es que para que el mallado adaptativo sea efectivo es necesario trabajar con mallas libres de elementos triangulares o predominantemente cuadrados. Estas opciones se escogen en la barra de herramientas haciendo clic en *Mesh* → *Mesh controls*.

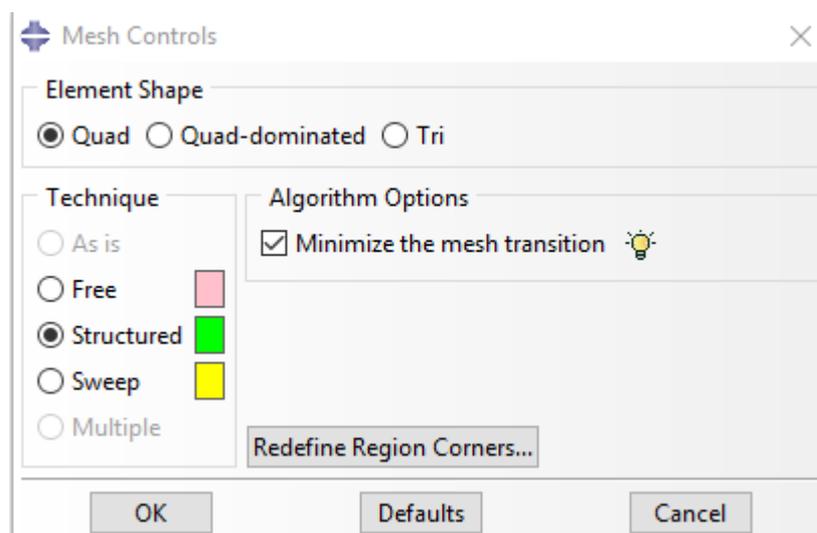


Ilustración 66. Estructura de la malla.

3. Finalmente se hace una semilla de mallado para observar cómo va a ser el mallado que se va a realizar y se malla. Para ello se selecciona en la barra de herramientas *Mesh* → *Part* y se selecciona el tamaño del elemento a malla. Como se explicó en capítulos anteriores el mallado va a tener entre 2500- 5000 elementos por lo que hay que calcular el tamaño del elemento. Finalmente, se selecciona *Mesh* → *Part* y se malla la placa.

A.1.9. Análisis.

Una vez creado y modelado el proceso de laminación que se va a llevar a cabo, hay que crear un caso de trabajo para posteriormente observar los resultados obtenidos en el postprocesador de ABAQUS:

1. Se crea un nuevo *job* de estudio haciendo doble clic en el árbol del modelo en *Job*. Puesto que el objetivo del trabajo es realizar un estudio paramétrico y se realizarán numerosos modelos, conviene identificar cada modelo con un nombre que resulte fácil de identificar. Por lo que le damos al caso de estudio el nombre de 02_004_05-AN, el cual se corresponde con Radio rodillo_ h_0 _%reducción_ Arrastre natural o forzado.

Finalmente se hace clic en *submit* para cargar el trabajo e iniciar el análisis, y una vez obtenido los resultados, se procederá a la visualización de los mismos.

A.1.10. Visualización.

Una vez se complete en análisis de manera correcta nos disponemos a observar los resultados obtenidos. Para ello se hace clic en el módulo *Results*.

Cabe destacar que no se va a explicar en este anexo como acceder a ciertos resultados, ya que ya se hizo en el **capítulo 6**. Lo que si se va a hacer es mostrar algunas indicaciones para que la visualización de los mismos sea más dinámica y realista:

1. En primer lugar lo que se va a hacer es dar profundidad a la visualización, es decir, visualizar el problema como un problema en tres dimensiones. Para ello se hace clic en *View* → *ODB Display Options* → *Sweep and Extrude*, y seleccionamos la dimensión de la extrusión que se quiere dar tanto a la placa (*Deformable body*), como al rodillo (*Analytical rigid*). Se asigna una extrusión igual a un metro a cada uno de ellos.

2. También se va a representar el problema completo, es decir, se va a hacer gráficamente la simetría que ya se hizo computacionalmente en el modelo mediante las condiciones de contorno.

Para ello se debe crear un sistema coordenado respecto al que realizar la simetría, el cual se va a hacer en la base del modelo (que es donde se aplicó la misma). Para ello se hace clic en el icono de *Create Coordinate System* → *Fixed System* → *Rectangular*. Se disponen los ejes de manera análoga a los dados por abaqus como referencia, solo teniendo en cuenta que el eje x debe coincidir con la base del modelo.

Para hacer visible la simetría finalmente se hace clic en *View* → *ODB Display Options* → *Mirror and Pattern*, se escoge el sistema de referencia creado, y como plano de simetría el plano XZ.

A.2. Arrastre forzado.

En el proyecto de investigación que se está realizando se tenía como objetivo demostrar el alcance de la teoría de laminación en frío en lo que se refiere a la estimación de los coeficientes de rozamiento necesarios para que se diera el arrastre natural y el arrastre forzado de las placas de material.

Una vez explicado el modelo realizado en ABAQUS para simular las condiciones de arrastre natural se procede por lo tanto a hacer lo mismo con las condiciones de arrastre forzado.

Cabe destacar que los modelos realizados para simular ambas condiciones son prácticamente idénticos, salvo por la geometría de partida de la placa de material, que se encuentra introducida entre los rodillos como se explico en el primer capítulo del presente texto.

Otra diferencia plausible es que como se explicó al inicio de este Anexo, se ha trabajado con dos modelos tanto para arrastre natural como para arrastre forzado: Uno de ellos usando un *datum point* para determinar el punto de contacto, y otro usando dos arcos para modelar el cilindro, donde el paso de un arco al otro se produce en dicho punto de contacto.

Por lo que a continuación se van a explicar dichas modificaciones, prestando principal atención en la geometría, y explicando a modo resumen como se realizaría el modelo usando la configuración B explicada en el párrafo superior.

A.2.1 Geometría.

En este caso se van a crear tres parts, uno de ellos deformable (placa de material), y otros dos que se corresponden con superficies rígidas (*analytical Surfaces*).

- Plate.

En este caso la geometría de partida es más compleja que en el caso de arrastre natural ya que el material se ha introducido previamente entre los rodillos, de manera que nuestra geometría de inicio ya está realmente deformada. Por lo tanto habrá que realizar una placa de material con un arco en su esquina superior derecha, simulando dicha entrada entre los rodillos.

Para ello hay que tener en cuenta los parámetros característicos del proceso de laminación, que son básicamente la longitud de contacto L , y el incremento de altura Δh .

$$L \sim \sqrt{R\Delta h} \quad (A - 6)$$

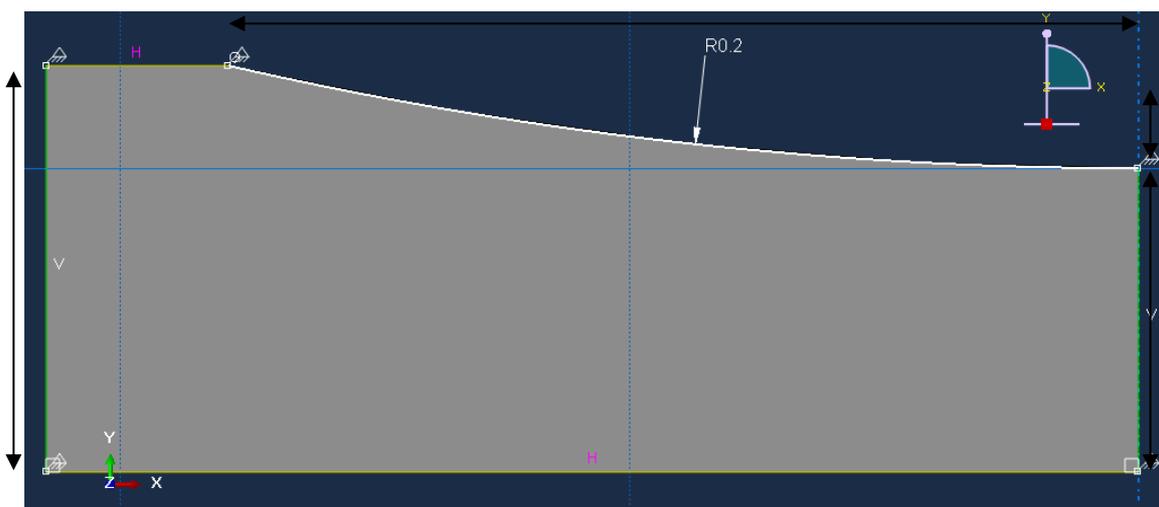
$$\Delta h = h_o - h_f \quad (A - 7)$$

Para realizar el arco, se usa la herramienta *Create arc: Center and 2 Endpoints*, es decir, un arco por tres puntos. Por lo tanto se introducirán las coordenadas de los tres puntos de la siguiente forma:

$$(x_0, y_0) = (0,0); \quad (x_1, y_1) = (0, -R) = (0, -0.2); \quad (x_2, y_2) = \left(-L, -R + \frac{\Delta h}{2}\right) \quad (A - 8)$$

Cabe destacar que existen pequeños errores matemáticos, y que ABAQUS ajusta el arco que se ha introducido a la circunferencia posible más cercana. De todas formas los errores detectados son del orden de $1E-5$, por lo que no son representativos.

Finalmente, a partir de dichos puntos es sencillo completar la geometría de la siguiente forma:



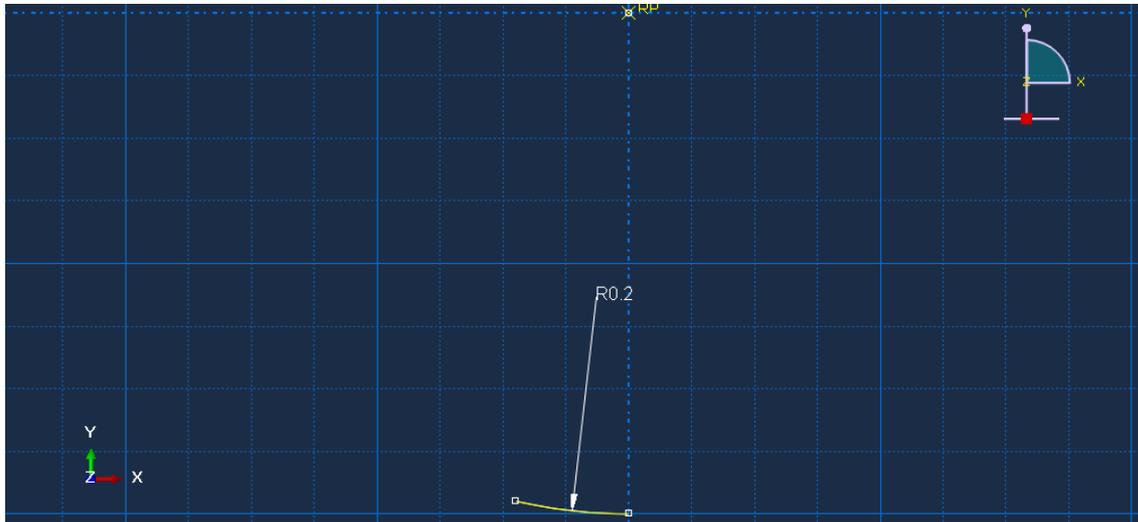
- Rodillos.

Como se explicó anteriormente, para algunos casos del modelo de arrastre forzado se ha utilizado la segunda de las formas explicadas para determinar el punto de contacto: usar dos rodillos.

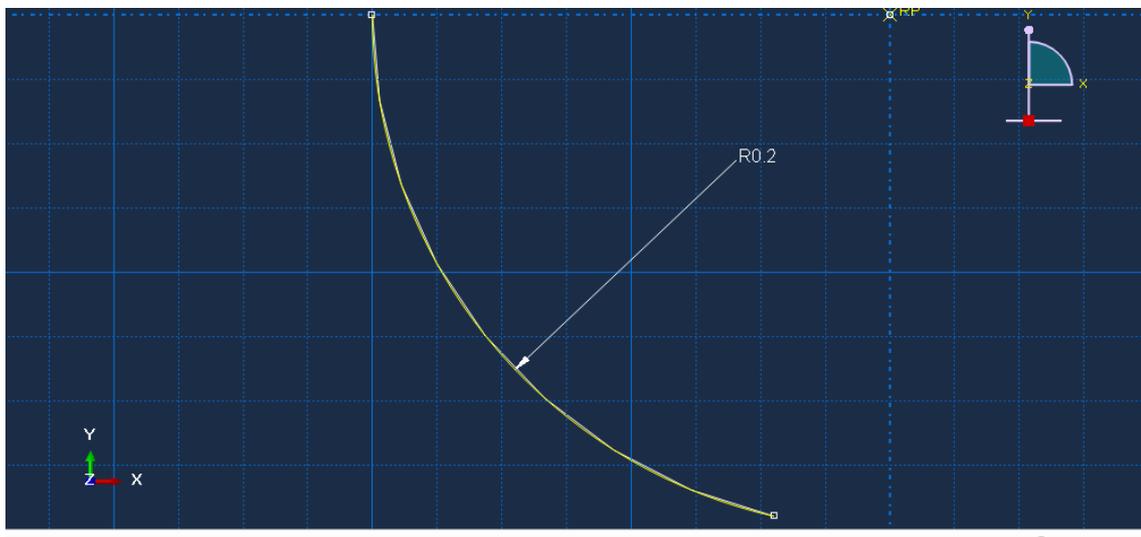
El procedimiento seguido para crear las dos superficies rígidas analíticas es el siguiente:

1. En un part de nombre roller1, usar la herramienta *Create arc: Center and 2 Endpoints*, es decir, un arco por tres puntos. Se introducirán las coordenadas de los tres puntos (son los mismos que se definió para la placa) de la siguiente forma:

$$(x_0, y_0) = (0,0); \quad (x_1, y_1) = (0, -R) = (0, -0.2); \quad (x_2, y_2) = \left(-L, -R + \frac{\Delta h}{2}\right) \quad (A - 9)$$



2. En un part de nombre roller 2, completar la circunferencia hasta alcanzar 90 grados, es decir, crear un arco con mismo centro, punto inicial el punto final del roller 1, y punto final $(-R,0)$:



3. Finalmente se crean dos Referent Point, uno para cada arco.

A.2.2. Sets and Surfaces:

En lo que respecta a los *Sets* y las superficies, solo existe una diferencia palpable entre el modelo realizado para simular la condición de arrastre natural, y su análogo para el arrastre forzado, que es la definición de la superficie *Top* del modelo como se muestran en las siguientes fotografías:

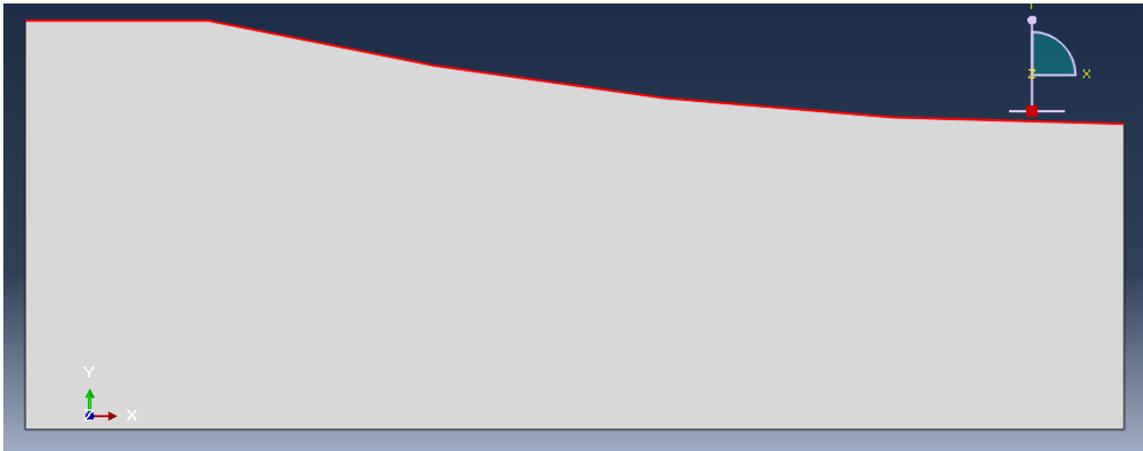


Ilustración 67. Superficie *Top* en Arrastre forzado.

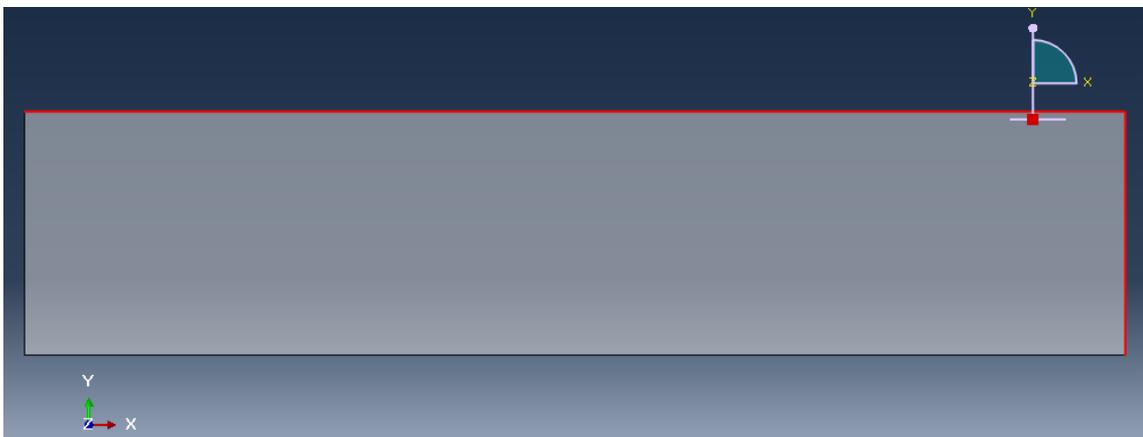


Ilustración 68. Superficie *Top* en Arrastre Natural.

A.2.3. Assembly

En lo que respecta al ensamblaje del conjunto, la forma de proceder es análoga al de AN obteniéndose el siguiente resultado:

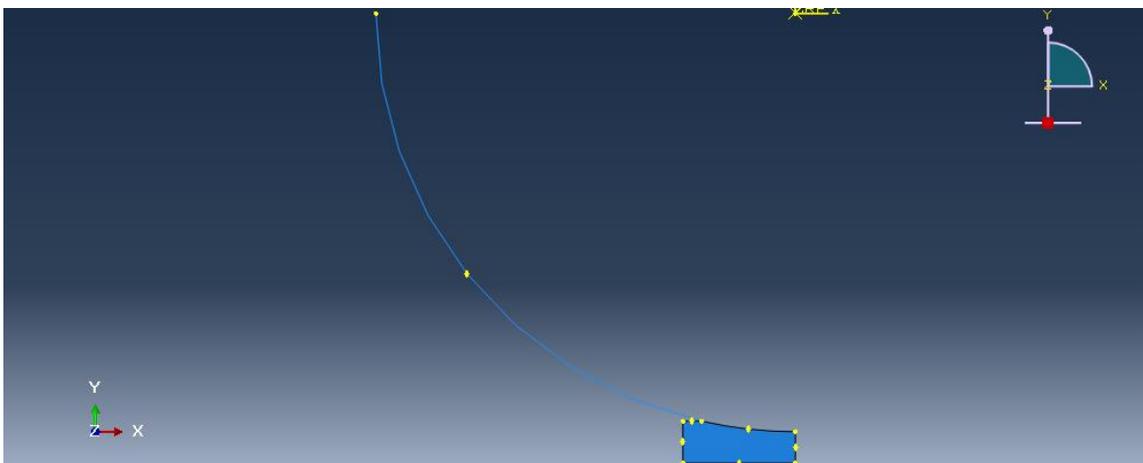


Ilustración 69. Ensamblado del model de Arrastre Forzado.

A.2.4. Resto del modelo.

Finalmente el resto del modelo no sufrirá ninguna modificación si se define el punto de contacto entre placa y rodillo mediante un Datum Point, y en el caso de utilizar la opción con dos arcos para obtener dicho punto de contacto, la principal diferencia respecto al modelo de arrastre natural radica en que hay que crear el doble de interacciones de contacto, y el doble de condiciones de contorno, pero todas igual a las definidas en el modelo anterior.

También destacar que puesto que el caso de arrastre forzado es más complejo como se verá en capítulos posteriores, para dicho caso se ha usado un mallado con tipo de elementos triangulares, cuyo código de identificación en Abaqus es CPE3.

ANEXO B. ANÁLISIS EXPLÍCITO Vs IMPLÍCITO.

El software utilizado para la realización del proyecto, Abaqus, permite resolver problemas de naturaleza estática o dinámica. Puesto que el problema que se pretende resolver con este proyecto es de carácter dinámico, se optó por crear un caso de estudio dinámico.

Dentro de dicho problema dinámico, Abaqus ofrece dos métodos de resolución donde cada cual será más o menos conveniente en función del problema:

- Integración directa implícita.
- Integración directa explícita.

A la hora de seleccionar uno de dichos métodos hay que tener en cuenta que tipo de problema se quiere resolver, las condiciones del mismo, y evaluar cual de los dos será capaz de obtener solución al problema, y en caso de ser válidas ambas opciones, escoger la más eficiente desde el punto de vista computacional.

A continuación se explican brevemente ambos métodos, y cuál de ellos ha sido el elegido para la realización del proyecto.

B.1. Metodo de Resolución.

B.1.1. Integración directa implícita

El método de integración directa implícita que posee Abaqus se basa en el operador implícito de Hilber-Hughes-Taylor.

La principal característica de dicho método es que requiere el cálculo de la matriz de rigidez para cada uno de los incrementos de tiempo, y su posterior inversión para la resolución de las ecuaciones dinámicas. Esto da lugar a un gran costo computacional, sobre todo para modelos de cierta envergadura.

Además hay que unir a dicha desventaja el hecho de que este método plantea más problemas que el método explícito en problemas fuertemente no lineales.

Sin embargo, algunas de las ventajas que presenta el método de integración directa implícita es que el valor del incremento de tiempo no tiene un límite prefijado, ya que este operador es incondicionalmente estable, lo cual lo hace apropiado para modelos en el que el tiempo de ejecución sea elevado. Dicho tiempo de ejecución, para nuestro caso, es equiparable al tiempo que tarda toda la placa en ser laminada.

Además dicho método presenta una potente ventaja: ofrece una mayor variedad de elementos para el mismo, lo cual permite adaptarse mejor a una mayor variedad de problemas.

B.1.2. Integración directa explícita.

El método de integración directa explícita emplea el operador de la diferencia-central, el cual calcula una solución del sistema en un instante a partir de su situación en el instante $t(n-1)=t(n)-h$, donde h es el incremento de tiempo definido para el problema dinámico.

La principal ventaja de dicho método es que desde el punto de vista de la convergencia, este método garantiza la convergencia con la única condición de que el tiempo de cada incremento sea el adecuado. Esto supone una gran ventaja para el estudio de problemas fuertemente no lineales, como es el caso de la laminación.

Además otra gran ventaja es que no integra la matriz de rigidez en cada uno de los incrementos establecidos para el problema, por lo que computacionalmente es más eficiente que el método implícito.

El principal problema que presenta el método es que es condicionalmente estable, guardando dicha estabilidad relación con el tiempo que tarda una onda en atravesar el elemento más pequeño del modelo y con el incremento de tiempo prefijado. Dicha onda es conocida como *dilatation wave*. Por lo tanto, el modelo será más estable si el incremento de tiempo prefijado es menor que el tiempo que tarda la onda en atravesar el elemento más pequeño (tiempo crítico), y menos estable si es mayor. Por lo tanto la eficiencia del problema depende de la relación existente entre el tiempo de ejecución (tiempo de laminación) y el incremento de tiempo máximo que podemos imponer.

Por lo tanto el principal problema de dicho método es que si no escogemos adecuadamente el tiempo de ejecución del problema así como los incrementos de tiempo asociados, el coste computacional del problema (tiempo de análisis) se puede incrementar mucho.

B.1.3. Aplicación al modelo realizado.

Teniendo en cuenta que el problema que se va a estudiar está caracterizado por grandes deformaciones y fuerte efectos no lineales (fricción), parece que en lo que respecta a la convergencia, es más conveniente escoger el método de integración directa implícito.

Además puesto que el tiempo de ejecución no es elevado, y se puede aproximar fácilmente, dicho método encaja mucho mejor con el tipo de problema a analizar.

Finalmente cabe destacar que el método de integración directa explícito de Abaqus ofrece una herramienta para solventar los problemas relacionados con la eficiencia asociada a la *dilatation wave* mencionada anteriormente. Dicha herramienta se conoce como *Mass Scaling* y se explicará en el Anexo número 3.

ANEXO C. MALLA ADAPTATIVA, MASS SCALING

Se van a explicar dos herramientas muy importantes en lo que respecta a la convergencia del modelo y la obtención de unos resultados reales, y a la eficiencia computacional del modelo. Dichas herramientas son el uso del mallado adaptativo o de la herramienta *Mass Scaling*.

C.1. Malla adaptativa.

Cuando se realizan análisis con el método de los elementos finitos se discretiza la geometría en elementos unidos entre si mediante nodos. En los análisis de conformado de metales e impactos de alta velocidad, un cuerpo puede experimentar una deformación plástica elevada, por lo que un único punto de integración de los elementos, que generalmente son robustos para largas deformaciones, puede dar imprecisión en los resultados incluso inestabilidad en el análisis.

Es por eso que en los programas de elementos finitos se comenzó a desarrollar herramientas para aumentar la fiabilidad de los resultados para aquellos fenómenos en los que tengan lugar deformaciones elevadas y es necesarios que el material fluya. El mallado que tiene en cuenta todos estos aspectos recibe el nombre de “mallado adaptativo”, y es el que se va a usar para llevar a cabo los análisis en este proyecto.

El mallado adaptativo permite obtener una malla de alta calidad a lo largo del cálculo, incluso cuando se producen grandes deformaciones. Ello se debe a que permite que el material fluya independientemente de la malla en aquellos puntos en los que sea necesario.

Cabe destacar que esta herramienta no crea una malla nueva ni modifica la tipología o disposición de la malla inicial, sino que reposiciona los nodos cuando los elementos se han visto muy deformados y permite por lo tanto que el material fluya a través de la malla, es decir, permite el remallado automático de la geometría durante el análisis para mejorar su integridad.

Además dicha herramienta viene de la mano de otra que permite aumentar el control de la malla para evitar una gran distorsión de los elementos de la misma. Dicha herramienta es la técnica ALE proporcionada por Abaqus, la cual aúna los dos enfoques existentes en una malla de elementos finitos en relación con el comportamiento cinemático de la misma y del material:

- Enfoque lagrangiano.
Este primer enfoque consiste en que la malla siempre está ligada al material de forma que se deforma tal y como lo hace este último. Dicho enfoque es utilizado para problemas estructurales en los que predominan las pequeñas deformaciones. Sin embargo en aquellos problemas en los que se producen grandes deformaciones, pueden surgir problemas de convergencia o incluso inestabilidad como ya se comunicó anteriormente, imposibilitando la obtención de una solución.
- Enfoque euleriano.
Un segundo enfoque, que es el utilizado por el mallado adaptativo, es aquel que permite que el material fluya a través de la malla, de modo que permanezca invariable en el espacio mientras el material fluye a través de ella.

Debido a esto, se va a utilizar el análisis tipo ALE el cual permite combinar ambos enfoques en el mismo problema.

C.2. Mass Scaling.

La herramienta *Mass Scaling* que ofrece Abaqus para aumentar la eficiencia de los análisis que usan como método de integración el método directo explícito se basa en modificar la masa de algunos elementos para reducir el coste computacional.

Esta herramienta permite aumentar la masa de aquellos elementos del modelo que debido a su pequeño tamaño reducen considerablemente el incremento de tiempo crítico, haciendo que sea posible resolver el problema para incrementos de tiempos mayores.

La expresión del tiempo crítico o límite de estabilidad para un análisis explícito se puede aproximar como:

$$\Delta t = \left(\frac{L_e}{c_d} \right) (C - 1)$$

Donde:

- L_e es la longitud característica del elemento más pequeño del modelo.
- $c_d = \sqrt{\frac{E}{\rho}}$ Es la velocidad de la *dilatational wave* para materiales con propiedades lineales elásticas, con E el modulo de Young y ρ la densidad del material.

Lo que hace el Mass Scaling es aumentar la masa del material multiplicando su densidad por un factor f^2 , (f es el factor que se introduciría en Abaqus), de manera que la velocidad de la onda disminuye, y por consiguiente aumenta el tiempo crítico posibilitando resolver el análisis con un menor número de incrementos e iteraciones.

Además Abaqus ofrece varias formas de aplicar este “Escalado de masa” según el tipo de problema que estemos resolviendo:

- Fixed. Se usa en problemas quasi-estáticos y se aplica una vez a todo el modelo en el inicio de cada step.
- Variable. Se usa en análisis quasi-estáticos y se aplica periódicamente a lo largo de cada step.

Por lo tanto según lo explicado, parece buena idea usar dicha herramienta. Sin embargo cuando se entró en profundidad en la aplicación de dicha herramienta se encontraron varios impedimentos:

- El proceso de deformación se caracteriza por un comportamiento altamente plástico predominado por grandes deformaciones, por lo que la aplicabilidad de la ecuación de la velocidad de onda no es exacta.
- El Mass scaling tiene gran influencia en las fuerzas de inercia, por lo que si no se aplica a la perfección, puede alterar mucho los resultados debido a que estas fuerzas se pueden hacer tan elevadas que el arrastre natural y forzado se produzca mucho antes de lo que debería.

Tras varios análisis y comprobaciones, se decidió por lo tanto no utilizar dicha herramienta para ganar fiabilidad en los resultados, a pesar de que el coste computacional sea mucho más elevado que con el Mass Scaling.

ANEXO D. LEYES DE COMPORTAMIENTO DE LOS MATERIALES.

Si hay algo que define el comportamiento de los metales en el proceso de laminación, es el endurecimiento por deformación que sufren cuando comienzan a fluir a través de los rodillos.

A grosso modo, endurecimiento por deformación o por trabajo quiere decir que el metal se está haciendo más fuerte conforme la deformación aumenta. Dicha propiedad es un factor muy importante en ciertos procesos de manufactura y especialmente en el proceso de laminación como ya se introdujo antes.

A continuación se va a estudiar el comportamiento de un metal en función de cómo le afecta dicha propiedad. Para ello hay que recordar la curva esfuerzo-deformación de los metales, la cual se muestra a continuación:

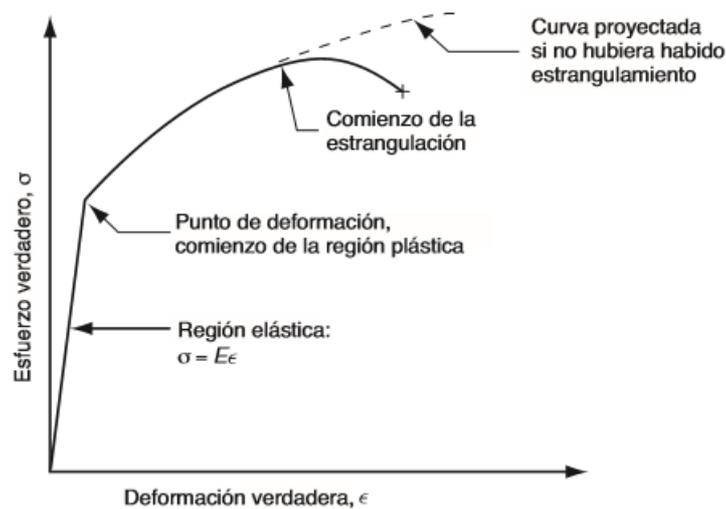


Ilustración 70. Curva esfuerzo-deformación.

Donde se puede observar la zona elástico-lineal, y la zona de deformación plástica. Si la parte de la curva correspondiente con la región plástica se graficara en una escala logarítmica, el resultado obtenido sería una relación lineal, por lo que la relación entre el esfuerzo verdadero y la deformación verdadera en la región plástica vendría dada por una línea recta de ecuación:

$$\sigma = K\epsilon^n \quad (D - 1)$$

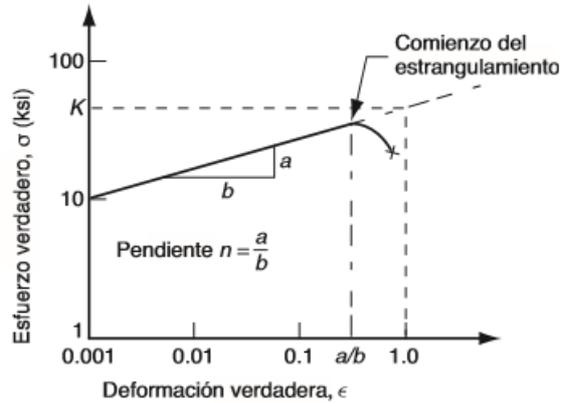


Ilustración 71. Zona plástica de la curva esfuerzo-deformación

Esta ecuación es conocida con el nombre de curva de flujo, y proporciona una aproximación buena del comportamiento de los metales en la región plástica, reflejando incluso la capacidad de endurecerse al producirse la deformación. Los parámetros implicados son:

- K = coeficiente de resistencia [Mpa]: Es igual al valor del esfuerzo verdadero para un valor igual a 1 de la deformación verdadera.
- n = Exponente de endurecimiento por deformación: es la pendiente de la recta que se observa en la gráfica anterior y su valor está relacionado con la tendencia que presenta un metal para endurecerse cuando está sometido a un trabajo.

Una vez introducido el concepto de endurecimiento por deformación se van a estudiar los tipos de relaciones esfuerzo-deformación existente.

D.1. Tipos de relaciones esfuerzo-deformación.

La curva esfuerzo-deformación introducida anteriormente puede proporcionar mucha información sobre el comportamiento elástico-plástico de los materiales. En dicha curva predominan dos comportamientos:

- La ley de Hooke ($\sigma = E\epsilon$), que gobierna el comportamiento del metal en la región plástica.
- La curva de flujo, que lo determina en la región plástica.

Por lo que según predominen un comportamiento u otro, o la combinación de ambos, los materiales se comportarán de una u otra forma, describiéndose su comportamiento según una de las tres formas básicas según las cuales se comportan casi todos los materiales sólidos:

- Perfectamente elástico. El comportamiento de dicho material se caracteriza por completo por su rigidez, indica según el módulo de Young E . Dicho tipo de materiales no presenta flujo plástico, sino que llegado el límite elástico, se fractura. Dichos materiales no son buenos candidatos para las operaciones de laminado.

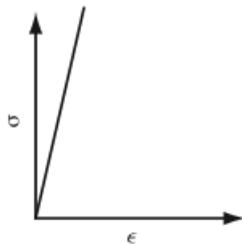


Ilustración 72. Material perfectamente plástico.

- Elástico y perfectamente plástico. Dicho material presenta una rigidez definida por E . Una vez que se

alcanza la resistencia de deformación, Y , el material se deforma plásticamente con el mismo nivel de esfuerzo, es decir, no, hay que aplicar una carga mayor para que siga deformándose. Dicho comportamiento se caracteriza por un valor de $K=Y$ e $n=0$. Los materiales se comportan de dicha forma cuando se calientan a temperaturas suficientemente altas que los cristalizan en lugar de endurecerlos por deformación. Un ejemplo de dicho comportamiento es el Plomo.

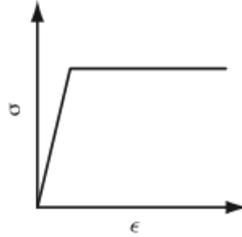


Ilustración 73. Material elástico y perfectamente plástico.

- Elástico y endurecimiento por deformación. Este material obedece la ley de Hooke en la región elástica y comienza a fluir a su resistencia de deformación Y . Una deformación continua en dicho tipo de materiales va a requerir un esfuerzo siempre incremental, dado por una curva de flujo cuyo coeficiente de resistencia K es mayor que Y , y cuyo exponente de endurecimiento por deformación n , es mayor que cero. En escala logarítmica, dicho comportamiento plástico se representa con una recta como se introdujo anteriormente. Cabe destacar que la mayor parte de los materiales dúctiles presentan este comportamiento cuando se trabajan en frío, y es por eso que se va a utilizar este comportamiento de los materiales para la realización del proyecto.

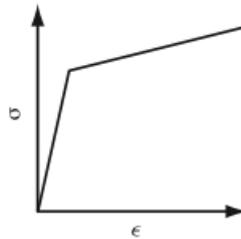


Ilustración 74. Material elástico y con endurecimiento por deformación

Por lo que los materiales que se van a laminar (Aluminio y Acero) seguirán dicho comportamiento.

D.2. Esfuerzo de fluencia.

Finalmente se va a introducir el concepto de Esfuerzo de fluencia, puesto que será de gran importancia para la definición del modelo.

El esfuerzo de fluencia es de especial interés en la curva de fluencia, que no es más que la curva esfuerzo-deformación en la región donde tiene lugar el conformado plástico del metal. Dicha curva indica el esfuerzo de fluencia del metal, además de la propiedad de resistencia que determinan las fuerzas y la potencia requerida para realizar una operación de conformado.

Por lo tanto el esfuerzo de fluencia se puede definir como el valor instantáneo del esfuerzo requerido para continuar la deformación del material o mantener fluyendo al metal. Esta es la resistencia a la fluencia del metal en función de la deformación, que puede expresarse como: $Y_f = K \epsilon^n$, con $Y_f >$ que el límite elástico del metal.

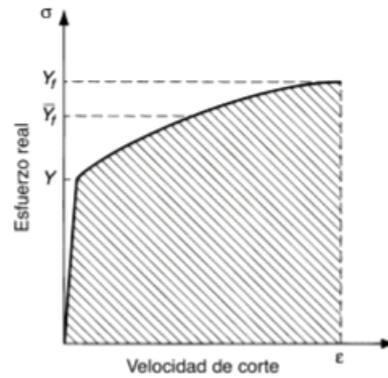


Ilustración 75. Curva de fluencia.

Finalmente es de especial interés definir el concepto de esfuerzo de fluencia promedio, puesto que es de vital importancia a la hora de determinar la fuerza necesaria para que se produzca la laminación.

El esfuerzo de fluencia promedio es el valor promedio de los esfuerzos sobre la curva esfuerzo-deformación desde el comienzo de la deformación hasta el valor final (máximo) que ocurre durante este proceso. El valor se ilustra en la gráfica anterior, y se determina integrando la ecuación de la curva de fluencia entre cero y el valor final de deformación que define el rango de interés, dando lugar a:

$$\bar{Y}_f = \frac{K\epsilon^n}{1+n} \quad (\mathbf{D-2})$$

Donde \bar{Y}_f es el esfuerzo de fluencia promedio y ϵ el máximo valor de la deformación durante el proceso de deformación.

ANEXO E. VISUALIZACIÓN DE ALGUNOS DE LOS CASOS ESTUDIADOS.

E.1 Arrastre Natural. Aluminio, $R = 0.2 \text{ m}$, $h_0 = 0.04 \text{ m}$, $h_f = 0.625 * h_0 \text{ m}$.

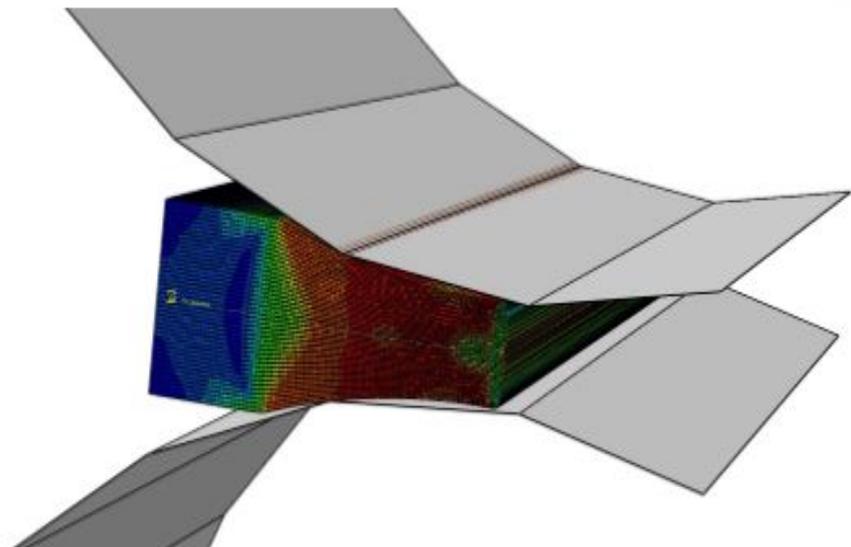
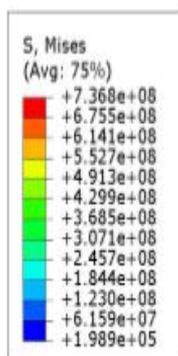
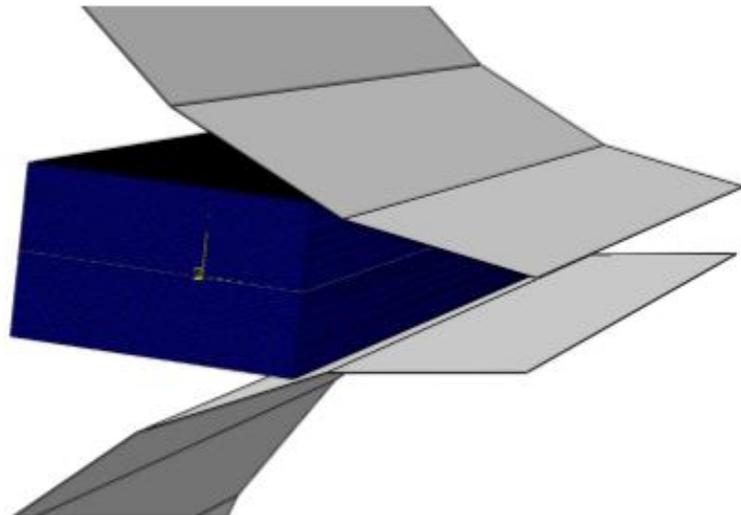
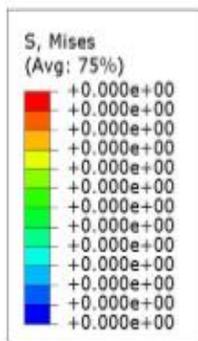


Ilustración 76. Representación gráfica proceso laminación-1

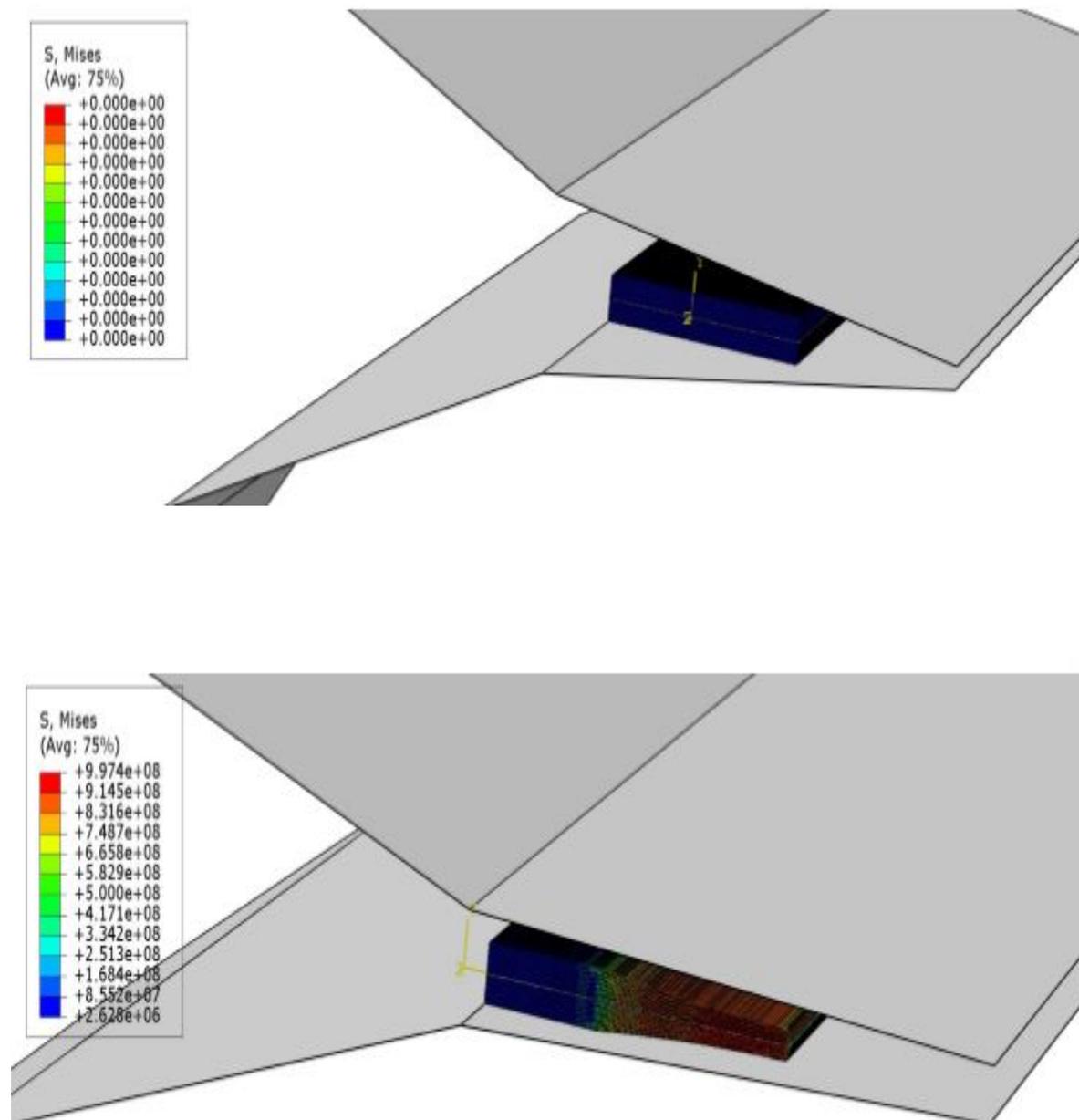
E.2. Arrastre Natural. Acero, $R = 0.5 \text{ m}$, $h_0 = 0.01 \text{ m}$. $h_f = 0.5 * h_0 \text{ m}$.

Ilustración 77. Representación gráfica proceso laminación-2

E.3. Arrastre Forzado. Aluminio, $R = 0.5 \text{ m}$, $h_0 = 0.08 \text{ m}$. $h_f = 0.375 * h_0 \text{ m}$.

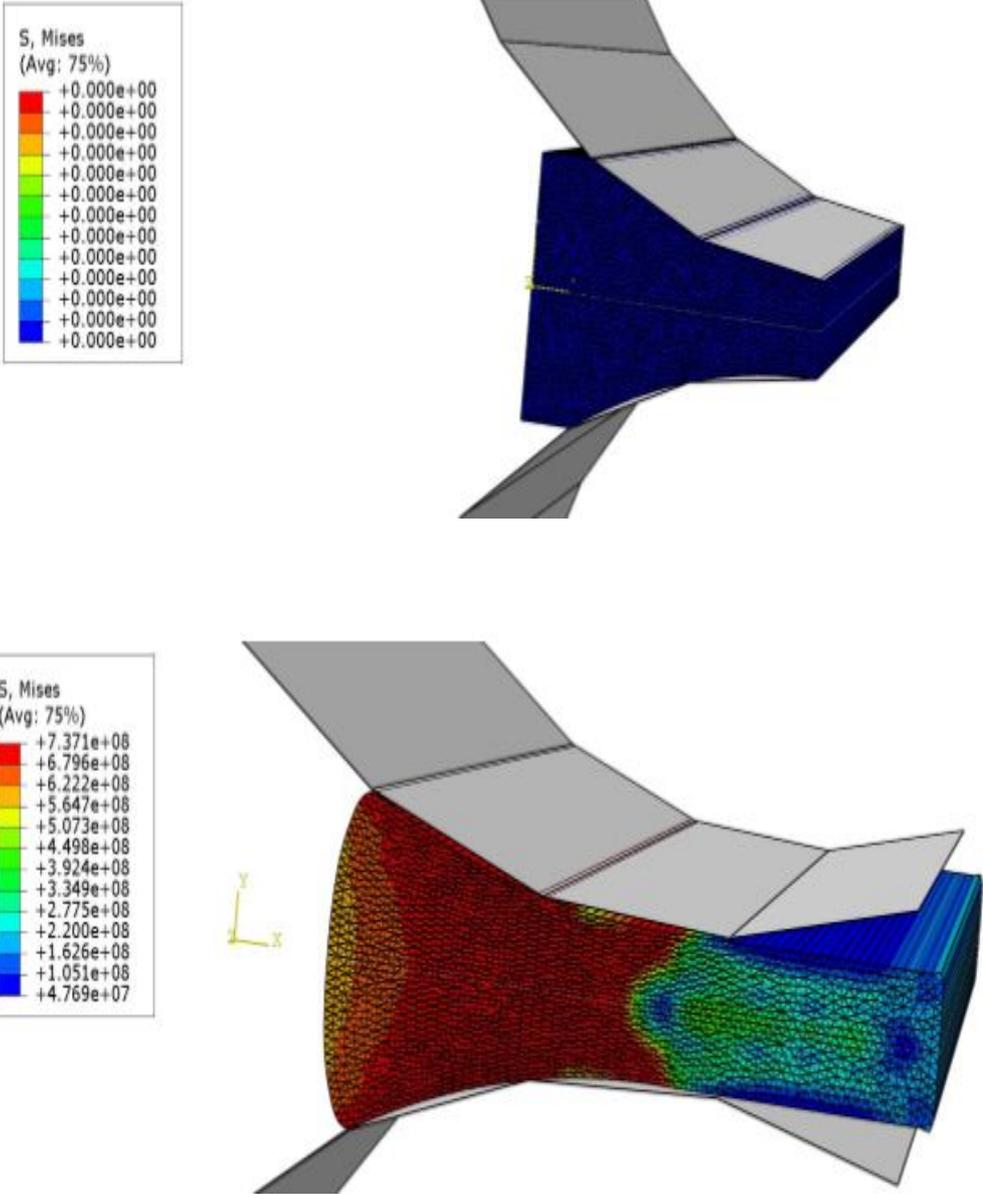


Ilustración 78. Representación gráfica proceso laminación-3

E.4 Arrastre Forzado. Acero, $R = 0.2 \text{ m}$, $h_0 = 0.01 \text{ m}$, $h_f = 0.125 * h_0 \text{ m}$.

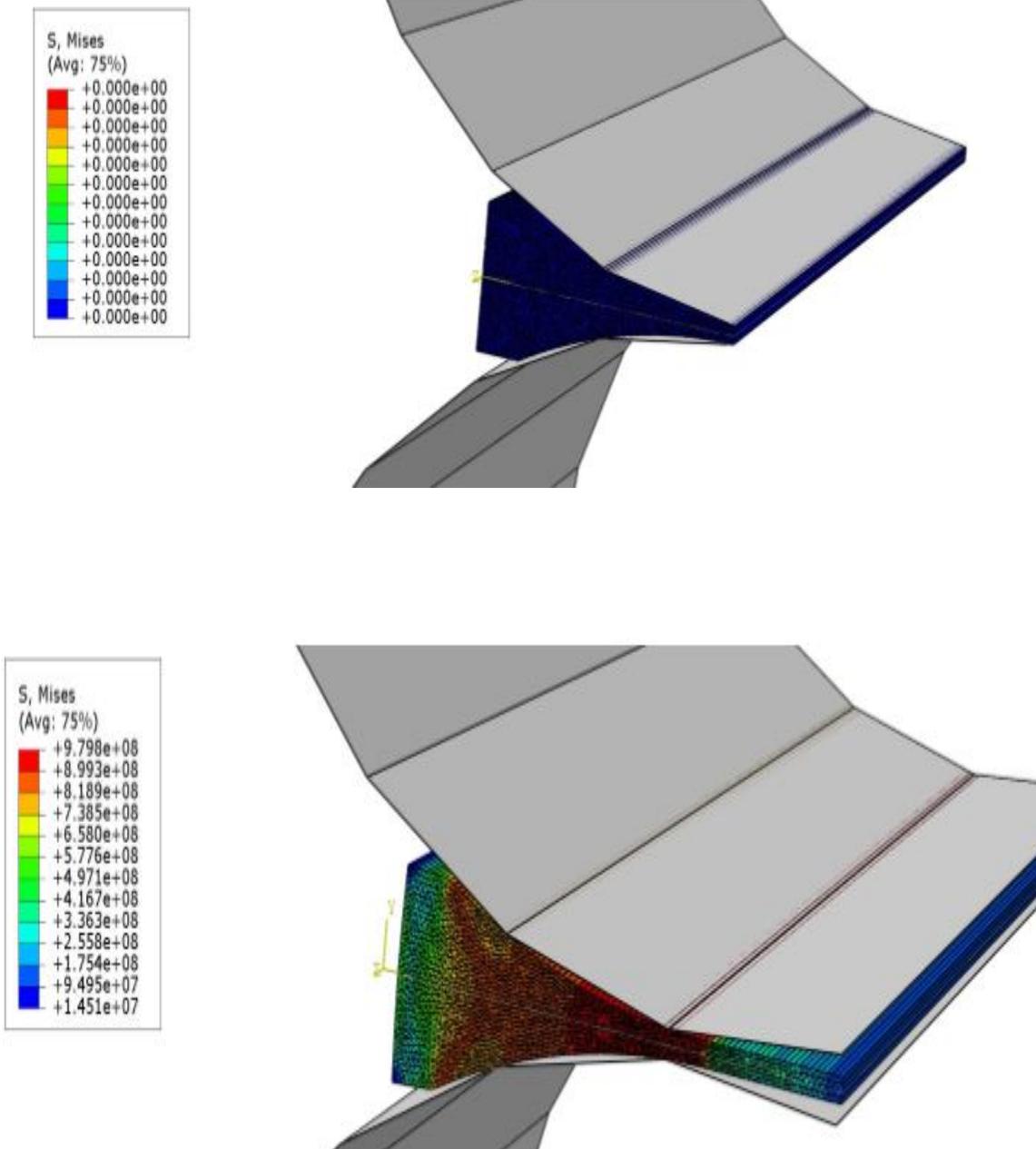


Ilustración 79. Representación gráfica proceso laminación-4

ANEXO F. CÁLCULO DE FUERZAS Y MOMENTOS DE LAMINACIÓN. TEORÍA.

En la introducción a la teoría de laminación quedo claro que ésta se producía principalmente por la fuerza de compresión ejercida por los rodillos sobre la placa de material, y que dicho proceso se podía agilizar incluyendo una fuerza de tracción en el sentido de laminación para facilitar la fluencia del material a través de los rodillos.

Esto se ilustra fácilmente en el criterio de Tresca. El material fluye cuando la tensión tangencial máxima alcanza el límite de fluencia tangencial, que viene dado por:

$$\tau_{ys} = \frac{\sigma_{ys}}{2} (F - 1)$$

Por lo tanto cualquier par de tensiones principales que provoquen la fluencia deberán estar en los extremos de un círculo de Mohr cuyo diámetro es σ_{ys} . Si en la dirección x aparece una tensión de tracción, la tensión según la dirección y tendrá que disminuir en una magnitud igual al valor absoluto de la anterior, siendo la tensión necesaria para hacer fluir el material $\sigma_{ys} - \sigma_x$, suponiéndose ambas tensiones positivas.

Planteando el equilibrio entre la fuerza F sobre los rodillos con las tensiones verticales en el material en la zona de deformación se obtiene:

$$\frac{F}{W} = \int_0^L \sigma_y(x) dx \cong \int_0^L (\sigma_{ys}(x) - \sigma_x(x)) dx \cong \int_0^L \sigma_{ys}(x) dx - \int_0^L \sigma_x(x) dx \quad (F - 2)$$

Y suponiendo una evolución lineal de tensiones σ_x en la zona de contacto se tiene:

$$\int_0^L \sigma_x(x) dx = \int_0^L \left(\sigma_0 + \frac{(\sigma_f - \sigma_0)}{L} x \right) dx = \frac{(\sigma_0 + \sigma_f)}{2} L \quad (F - 3)$$

Además el primer sumando de la ecuación (1) se puede aproximar usando el concepto de tensión de fluencia estudiado en el Anexo anterior, por lo que:

$$\int_0^L \sigma_{ys}(x) dx \cong \frac{L}{\Delta \epsilon} \int_{\epsilon_0}^{\epsilon_f} \sigma_{ys}(\epsilon) d\epsilon = \bar{\sigma}_{ys} * L \quad (F - 4)$$

Por lo que la ecuación de la fuerza en los rodillos queda de la forma:

$$F = \left(\bar{\sigma}_{ys} - \frac{\sigma_f + \sigma_0}{2} \right) * L * w \cong \left(\bar{\sigma}_{ys} - \frac{\sigma_f + \sigma_0}{2} \right) * \sqrt{R\Delta h} * w \quad (F - 5)$$

Con

- $\bar{\sigma}_{ys}$ =Tensión media de fluencia en el proceso.
- σ_0 =Tensión de tracción a la entrada del proceso.
- σ_f =Tensión de tracción a la salida del proceso.
- L =Longitud de la zona de contacto.
- w =Anchura de la chapa.
- Δh =Decremento de espesor de la chapa.
- R =Radio de los rodillos.

Y donde la tensión media de fluencia se calcula a partir de la curva de comportamiento $\sigma - \epsilon$ donde las deformaciones son según el espesor de la chapa:

$$\overline{\sigma}_{ys} = \frac{\int_{\epsilon_0}^{\epsilon_f} \sigma d\epsilon}{\epsilon_f - \epsilon_0} = \begin{cases} \frac{\int_{\epsilon_0}^{\epsilon_f} \sigma_{ys} d\epsilon}{\epsilon_f - \epsilon_0} = \sigma_{ys}(\dot{\epsilon}) = C \dot{\epsilon}^m \rightarrow & \text{Conformado en caliente} \\ \frac{\int_{\epsilon_0}^{\epsilon_f} K \epsilon^n d\epsilon}{\epsilon_f - \epsilon_0} = \frac{K}{n+1} \frac{\epsilon_f^{n+1} - \epsilon_0^{n+1}}{\epsilon_f - \epsilon_0} \rightarrow & \text{Conformado en frío} \end{cases} \quad (\mathbf{F} - 6)$$

Donde la deformación a la entrada vendrá dada de un proceso anterior y la deformación a la salida se puede estimar como:

$$\epsilon_f = \epsilon_0 + \ln\left(\frac{h_0}{h_f}\right) \quad (\mathbf{F} - 7)$$

ANEXO G. CÁLCULO DE FUERZAS Y MOMENTOS DE LAMINACIÓN. UN EJEMPLO CON ABAQUS.

En líneas futuras de investigación sería una buena idea calcular la fuerza necesaria para que se de la laminación mediante los resultados obtenidos con Abaqus.

Para ello, según lo visto en el Anexo G, sería necesario calcular las tensiones de fluencia, tensión a la entrada y a la salida del proceso para posteriormente aplicar:

$$F = \left(\overline{\sigma}_{ys} - \frac{\sigma_f + \sigma_0}{2} \right) * L * w \cong \left(\overline{\sigma}_{ys} - \frac{\sigma_f + \sigma_0}{2} \right) * \sqrt{R\Delta h} * w \quad (E - 1)$$

Para obtener dichos valores con Abaqus es necesario activar las salidas correspondientes en *Field Output*, y *History Output*, activando las pestañas correspondiente con las deformaciones plásticas y las tensiones en el primero, y la pestaña *Forces and Reaction* en el segundo, habiendo definido como lugar o *Set* de estudio el punto de referencia del rodillo previamente.

Una vez definidos los resultados de salida que se quieren calcular, es hora de crear las gráficas correspondientes.

Para el cálculo de la fuerza de laminación se puede observar que es necesario conocer la tensión de fluencia media, la de entrada en el proceso, y la de salida. Todas estas tensiones se pueden obtener representando la tensión frente al tiempo para los elementos de nuestro modelo. También sería conveniente representar las deformaciones frente al tiempo con el objetivo de comprobar que error existe entre la tensión de fluencia media calculada según las gráficas, y la obtenida según la ecuación (F-6). Ambas gráficas se muestran a continuación para el modelo de arrastre natural con $radio=0.2 \text{ m}$, $h_0 = 0.04 \text{ m}$, $h_f = 0.375 * h_0 \text{ m}$:

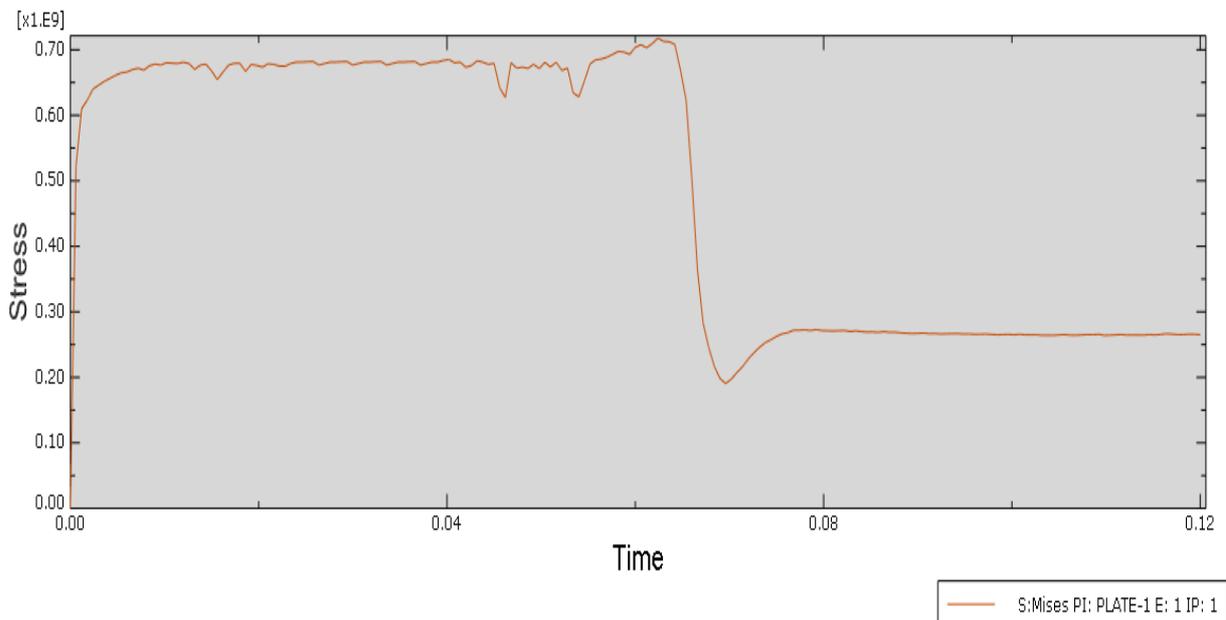


Ilustración 80. Tensión de un elemento.

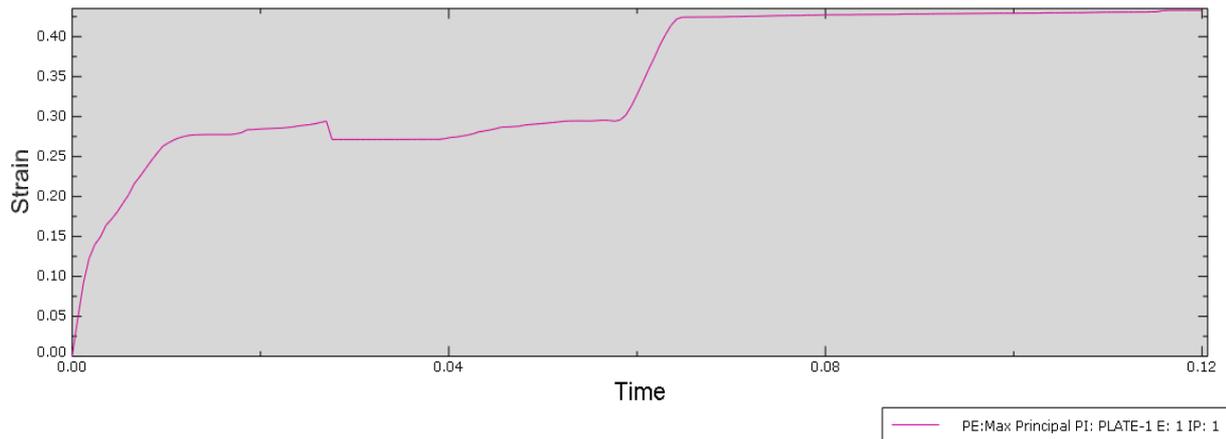


Ilustración 81. Deformación de un elemento.

Cabe destacar que las deformaciones y tensiones mostradas han sido calculadas para un solo elemento, por lo que habría que calcularlas para la placa en su conjunto, por lo que habría que integrar a lo largo de todos los elementos que componen la placa. Ello supone un proceso tedioso ya que los valores que determinan las tensiones de fluencia, entrada y salida se observan fácilmente gráficamente, pero debido al gran número de elementos existente resulta muy complejo y laborioso integrar para todos los elementos de la placa dichos valores. Habría que recorrer los vectores (X,Y) que exporta Abaqus tras el análisis y mediante un algoritmo seleccionar los valores de interés.

Por ello, a nivel ilustrativo y muy simplificado, se propone una forma de estimar el orden de magnitud de la fuerza de laminación:

1. Representar la reacción RM3, es decir, el momento en el eje del cilindro frente al tiempo.

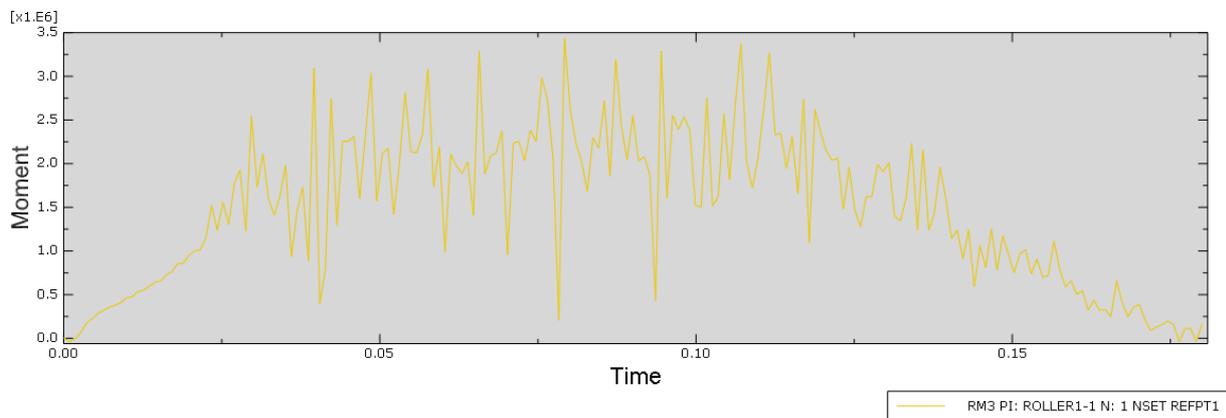


Ilustración 82. Reacción en el eje del rodillo.

2. Aproximar su valor a un valor medio debido a la gran dispersión y suponer que es el Par debido exclusivamente a la componente horizontal de la fuerza de fricción en el instante en que la placa comienza a entrar entre los rodillos (ya que la gráfica muestra la evolución del momento a lo largo del tiempo, por lo que dicha fuerza también varía).
3. Dividiendo entre el radio se obtendría: $\frac{T}{R} = F * \cos\theta$
4. Finalmente la fuerza de fricción es $F = \mu N = \frac{T}{R * \cos\theta}$.

Por lo que para el ejemplo que se está estudiando se obtiene $\frac{2}{(0.2 * 0.94279)} = 165 \text{ MN}$.

ANEXO H. CRITERIOS DE CONVERGENCIA.

Los criterios de convergencia que hay que tener en cuenta a la hora de realizar un análisis explícito en Abaqus atienden a criterios energéticos, usando una ecuación de balance para evaluar si la respuesta cuasi-estática del modelo es la adecuada.

En Abaqus dicha ecuación es la siguiente:

$$E_I + E_V + E_{FD} + E_{KE} - E_W = E_{TOT} = \text{constante} \quad (\mathbf{H} - \mathbf{1})$$

Donde:

- E_I = Es la energía interna (Energía de deformación plástica y elástica así como la energía asociada con *hourglass control*).
- E_V = Es la energía disipada por mecanismos viscosos.
- E_{FD} = Es la energía de disipación por fricción.
- E_{KE} = Energía cinética.
- E_W = Trabajo externo de las fuerzas.
- E_{TOT} = Energía total del sistema.

Además la energía cinética del material de deformación no debe exceder de una pequeña fracción de su energía interna a lo largo de la mayor parte del proceso de formación, siendo dicha pequeña fracción del orden de 1-5%, lo cual es un requisito extra difícil de conseguir.

A lo largo del proyecto se han realizado algunas comprobaciones de convergencia, cumpliéndose de manera aproximada dicha ecuación. Lo que sí que se ha comprobado para todos los casos es que la energía total permanezca constante cuando el análisis ha finalizado.

REFERENCIAS

- [1] Groover, M.P., “Fundamentals of Modern Manufacturing: Materials, Processes, and Systems”. ISBN:9780470467008, *Mc-Graw-Hill*, 2010.
- [2] Schey, J.A. and Cárdenas, J.L., “Procesos de manufactura”. ISBN:978970103733, *Mc-Graw-Hill*, 2002.
- [3] Carpofofo, Luis Valentín Bohorquez Jiménez, “Tema 6, Laminación”, *Apuntes de la asignatura de Procesos de Fabricación*, Sevilla, 2010.
- [4] Adarsh Dhingra, Geeta Agnihotri and K.K.Pathak, “A three dimensional finite element simulation of cold Rolling of a Steel plate and prediction of influence parameters”, *Revista, International journal of manufacturing technology and industrial engineering (IJMTIE)*, Vol. 1, No. 1, January-June 2011, pp. 13-17.
- [5] Abaqus 6.12, “Abaqus/CAE User`s Manual”.
- [6] Abaqus 6.12, “Example Problems Manual”, Volume I: Statics and Dynamics Analyses.
- [7] Abaqus 6.12, “Example Problems Manual”, Volume II: Other Applications and Analyses,
- [8] Guido van Rossum, “Guía de aprendizaje de Python”, Rev 2.0, Editor: *Fred L.Drake, Jr*, 16 de Octubre de 2000.
- [9] Shailendra Dwivedi, Dr. Geeta Agnihotri, Dr. K.K. Pathak, “Parametric Study of the Hot Rolling Process using FEM”, *International Journal of Engineering Research and Applications (IJERA)*, Vol. 2, Issue 5, Septembrer-October 2012, pp. 1517-1522.
- [10] Alfonso Ynigo Rivera, Rachid Cheriguene, “Modelos de Fricción para mecanizado de alta velocidad”, *Proyecto Fin de Carrera, Universidad Carlos III*, Madrid, 2015.
- [11] Ricardo Durnes Delgado, Alejandro Quesada González, “Empleo de Mallado Adaptativo en Simulaciones de Procesos de Conformado de Chapa por el Método de los Elementos Finitos”, *Proyecto Fin de Carrera, Universidad Carlos III*, Madrid, Diciembre de 2010.
- [12] Francisco Javier Foncubierta Boj, Luis Valentín Bohórquez Jiménez, “Análisis de la fuerza y momento de laminación en frío mediante el uso de elementos finitos”, *Proyecto Fin de Carrera, Escuela Técnica Superior de Ingeniería, Univeridad de Sevilla*, Sevilla, Mayo de 2015.
- [13] Abaqus, “Quasi-Estatic Analyses”, *Abaqus/Explicit Advanced Topics*, Lecture 5.

[14] Abaqus, “Single Pass Rolling of a Thick Plate”, *Abaqus Workshop 5*.

[15] Jose Luis Enríquez Berciano, Enrique Tremps Guerra, Susana de Elío de Bengy, Daniel Fernández Segovia, “Laminación”, *Monografías sobre tecnología del Acero*, Parte IV.