

# Métodos numéricos para análisis estructural de bifurcaciones con seguimiento de las distintas ramas de post-pandeo

Santiago Delgado

## Resumen

El análisis lineal de las estructuras posee grandes limitaciones a la hora de considerar estructuras esbeltas, en particular en lo referente a la estabilidad de las mismas. Entre los distintos métodos para el análisis no lineal de estructuras, los métodos de longitud de arco han ganado gran renombre debido a su capacidad de estudiar el comportamiento estructural aún superando los puntos de “snap-through” y “snap-back” pero no logran analizar las distintas ramas de una bifurcación. Debido a la importancia de poder evaluar el comportamiento post-crítico en las mismas, en particular frente a un análisis de sensibilidad a las imperfecciones; se implementa en este trabajo dos metodologías de análisis de ramificaciones en reticulados, mediante el código ONSAS, las cuales son testeadas en dos estructuras de distinta complejidad.

## 1. Introducción

Al diseñar una estructura se debe evaluar una serie de fenómenos que pueden generar el colapso como por ejemplo la rotura o deformación excesiva de un elemento estructural, la capacidad de la estructura de mantener el equilibrio y no volcar o deslizar, entre otros. Uno de los fenómenos de mayor trascendencia en estructuras esbeltas es la inestabilidad.

La inestabilidad de las posiciones de equilibrio de sistemas estáticos, como las que se estudian en este trabajo, se asocian a la incapacidad de soportar pequeñas variaciones del mismo, obteniendo grandes desplazamientos a partir de dichas variaciones. [1]

En el presente trabajo se estudian estructuras reticuladas con materiales hiperelásticos lineales por lo cual los fenómenos de inestabilidad surgen a partir de fenómenos de no linealidad geométrica. De manera resumida la no linealidad geométrica se basa en aplicar el equilibrio de las fuerzas internas y externas en la configuración deformada de la estructura.

Como se presenta en [2] las bifurcaciones del comportamiento estructural surgen a partir de los denominados puntos críticos. En el método de los elementos finitos estos son los puntos de la curva carga-desplazamiento en los cuales la matriz de rigidez tangente es singular. No todos los puntos críticos se traducen en bifurcaciones del comportamiento estructural si no que pueden existir los denominados puntos límite.

El estudio de las bifurcaciones resulta de especial interés a la hora de predecir el comportamiento de una estructura con imperfecciones, debido a que permite no solo estimar la curva carga-desplazamiento sino predecir

qué estructuras poseen un alto grado de sensibilidad a las imperfecciones. Las bifurcaciones simétricas inestables y asimétricas, según se definen en [2], poseen un alto grado de sensibilidad a las imperfecciones obteniendo importantes reducciones de la capacidad portante en caso de presentar imperfecciones.

Para evaluar estructuras complejas, de varios grados de libertad, resulta de gran utilidad poseer herramientas informáticas capaces de identificar y seguir las distintas ramas de post-pandeo de una estructura. El objetivo principal del presente trabajo es generar, para el código ONSAS, métodos numéricos que permitan esto; aspirando a que en futuras versiones de éste puedan incorporarse los mismos.

En la sección 2 se presenta brevemente el código ONSAS el cual es el código que se usa como base para los métodos numéricos a implementar. En la sección 3 se presenta el caso de estudio con el que se prueban los métodos numéricos para validar los mismos.

Luego en la sección 4 se presenta el primer método numérico implementado, el cual surge intuitivamente a partir del método de longitud de arco tradicional. Posteriormente en la sección 5 se presenta el segundo método numérico implementado asociado al método de “branch switching” de Crisfield [3] el cual además de ser validado se aplica a un caso de mayor complejidad en la sección 6.

Finalmente, en la sección 7 se presentan las conclusiones asociadas al trabajo y algunos hilos de trabajos a futuro.

## 2. Código de base

El código ONSAS, desarrollado por el grupo MISEs del Instituto de Estructuras y Transporte de la Facultad de Ingeniería de la Universidad de la República, es un código para el análisis estático y dinámico, lineal y no lineal de estructuras escrito en el lenguaje de GNU-Octave.

El mismo, el cual puede obtenerse del repositorio ubicado en: <https://github.com/ONSAS/ONSAS.m>, posee, como métodos para el análisis estático no lineal de estructuras, Newton-Raphson incremental y longitud de arco cilíndrica, siendo ninguno de estos capaces de estudiar las distintas ramificaciones en una bifurcación.

La versión de ONSAS que se utilizó como base para el presente trabajo es la v0.2.1 la cual, ya a la fecha no es la versión más reciente. Debido a esto, de desearse utilizar los métodos que aquí se desarrollan, los cuales son brindados al final de este informe, se debe tener especial cuidado con la compatibilización de versiones del código.

## 3. Caso de estudio para validación

Como caso de estudio, para validar los métodos numéricos que se desarrollan, se hace uso de la cercha de Von Mises según se presenta en la figura 1 el cual tiene como parámetros  $L = 200$  cm (longitud de las barras),  $A = 2,5$  cm<sup>2</sup> (área de la sección) y  $E = 210$  GPa (módulo elástico). La medida de deformación que se utiliza es la deformación de Green.

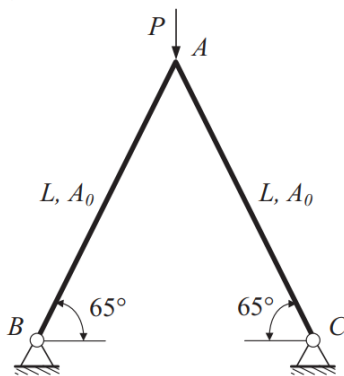


Figura 1: Estructura para el estudio de bifurcaciones [4]

La curva carga-desplazamiento vertical del nodo no apoyado es según se presenta en la figura 2. La bifurcación es simétrica inestable según los desplazamientos horizontales, los cuales son nulos hasta que se alcanza el punto crítico. [4] La carga a la cual se produce la bifurcación es  $P_c = 12\,578$  kN. [4]

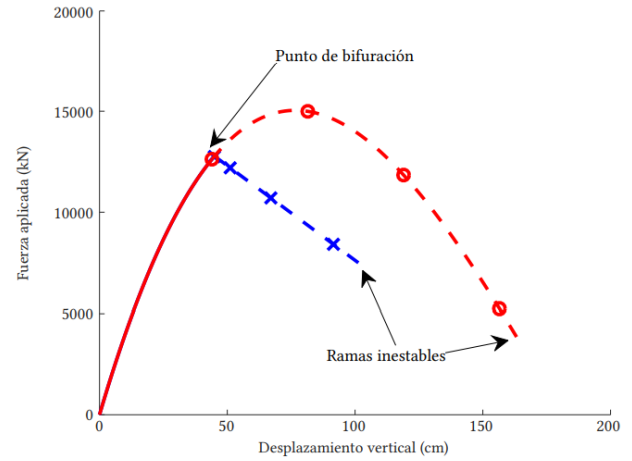


Figura 2: Curva carga- desplazamiento vertical [2]

En la figura 2 en rojo se presenta la rama fundamental y en azul las ramas (aparecen superpuestas) que surgen por la bifurcación y en línea continua aparecen los puntos de equilibrio estable y en línea discontinua los puntos de equilibrio inestable.

## 4. Método de longitud de arco intuitivo

Para describir el primer método numérico que se desarrolla para el estudio de bifurcaciones, vale la pena en primer lugar describir brevemente de qué se tratan los métodos de longitud de arco para resolución de problemas no lineales.

Dichos métodos se basan en resolver los valores  $\{x_k; \lambda_k\}$  tales que:

$$\begin{cases} f_{int}(x_k) = \lambda_k f_{ext} \\ \Delta x_k^T \Delta x_k + \Delta \lambda_k^2 \psi^2 f_{ext}^T f_{ext} = \Delta l^2 \end{cases} \quad (1)$$

donde:

- $x_k$  es el vector de desplazamientos de los grados de libertad de la estructura para el paso  $k$ . Las entradas del mismo se definen como  $\{x_k^1; x_k^2; \dots; x_k^m\}$  para una estructura de  $m$  grados de libertad.
- $\lambda_k$  es el factor de carga externa para el paso  $k$ .
- $f_{int}(x_k)$  es el vector de fuerzas internas para los desplazamientos  $x_k$ .
- $f_{ext}$  es el vector de fuerzas externas para un factor de carga unitario.
- $\Delta x_k = x_k - x_{k-1}$  siendo  $x_{k-1}$  el vector de desplazamientos de los grados de libertad de la estructura solución del paso anterior.
- $\Delta \lambda_k = \lambda_k - \lambda_{k-1}$  siendo  $\lambda_{k-1}$  el factor de carga solución del paso anterior.

- $\psi$  es un factor de escala.
- $\Delta l$  es el parámetro de longitud de arco. A mayor valor más inexacto será el trazado de la curva carga-desplazamiento pero se requerirá menos tiempo computacional para trazar la misma.

De manera intuitiva, el método de longitud de arco se basa en partir de una solución conocida, trazar una bola de radio  $\Delta l$  centrada en esta solución y definir como “siguiente solución” al punto donde la bola corte a la solución del problema no lineal.

El primer método de estudio de bifurcaciones se basa en esta definición intuitiva del método de longitud de arco ya que si existe una bifurcación habrán varios puntos de la bola que serán solución.

El método se basa en el concepto de que una esfera n-dimensional de radio  $\Delta l$  se puede parametrizar a partir de n-1 ángulos según se muestra a continuación. Sean los puntos  $\{x_k^1; x_k^2; \dots; x_k^{n-1}; \lambda_k\}$  pertenecientes a la esfera n-dimensional de radio  $\Delta l$  y centro  $\{x_{k-1}; \lambda_{k-1}\}$  esto quiere decir:

$$\sum_{i=1}^{i=n-1} (\Delta x_k^i)^2 + \Delta \lambda_k^2 \psi^2 = \Delta l^2 \quad (2)$$

donde se está considerando que el módulo de  $f_{ext}$  se encuentra contenido en el parámetro de escala.

Se define un primer ángulo azimutal  $\theta_1$  tal que cumpla la ecuación (3).

$$\text{sen}(\theta_1) = \frac{\psi \Delta \lambda_k}{\Delta l} \quad (3)$$

Está claro ver que de esta manera los puntos  $\{x_k^1; x_k^2; \dots; x_k^{n-1}\}$  pertenecen a una esfera (n-1)-dimensional según la siguiente ecuación:

$$\sum_{i=1}^{i=n-1} (\Delta x_k^i)^2 = \Delta l^2 \cos^2(\theta_1) \quad (4)$$

Luego se define el ángulo  $\theta_2$  tal que cumpla la ecuación (5).

$$\text{sen}(\theta_2) = \frac{\Delta x_k^{n-1}}{\Delta l \cos(\theta_1)} \quad (5)$$

Nuevamente se puede apreciar que los puntos  $\{x_k^1; x_k^2; \dots; x_k^{n-2}\}$  pertenecen a una esfera (n-2)-dimensional según la siguiente ecuación:

$$\sum_{i=1}^{i=n-2} (\Delta x_k^i)^2 = \Delta l^2 \cos^2(\theta_1) \cos^2(\theta_2) \quad (6)$$

Se continua de manera análoga definiendo los ángulos  $\{\theta_3; \theta_4; \dots; \theta_{n-2}\}$  hasta llegar a la ecuación (7).

$$(\Delta x_k^1)^2 + (\Delta x_k^2)^2 = \Delta l^2 \prod_{i=1}^{i=n-2} \cos^2(\theta_i) \quad (7)$$

Con esta ecuación se define el ángulo  $\theta_{n-1}$  de manera análoga a la ecuación (5). Una interpretación geométrica de la definición de los ángulos, para  $n = 3$ , es presentada en la figura 3.

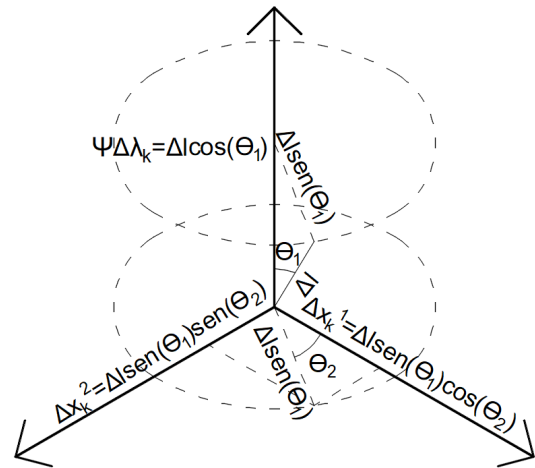


Figura 3: Esquema geométrico de la definición de ángulos

Lo anterior muestra que con variar los  $\{\theta_1; \theta_2; \dots; \theta_{n-2}\}$  en  $[-\pi/2; \pi/2]$  y  $\theta_{n-1}$  en  $[0; 2\pi]$  y despejando de las ecuaciones los coeficientes  $x_k^i$  y  $\lambda_k$  se obtienen todos los puntos de la esfera n-dimensional.

El método se basa en definir una cantidad discreta de ángulos que recorren los intervalos anteriormente mencionados y definen una cantidad discreta de puntos donde se evalúa el equilibrio de fuerzas. Aquellos puntos que cumplan con determinada tolerancia y estén suficientemente alejados entre sí y con las soluciones anteriores, son considerados soluciones del problema y trazan las distintas ramificaciones.

Se puede apreciar que los tiempos de cálculo de este método crecen exponencialmente con la cantidad de grados de libertad de la estructura ya que cada nuevo grado de libertad implica un nuevo ángulo que varía en su respectivo entorno por lo cual el método no es adecuado para estructuras de muchos grados de libertad.

En la versión actual del método, si no existen puntos que cumplan con la tolerancia buscada se elige como solución el punto con el menor error posible. Esto último no es correcto y debería ser mejorado, lo que se podría hacer eligiendo un entorno del conjunto de ángulos que dan lugar a la solución de menor error y buscar en el mismo soluciones que cumplan la tolerancia. Como se verá en la sección 4.3 este método tiene tiempos de

cómputo bastante largos, aún con este error, por lo que se decidió no continuar desarrollando el mismo y pasar al método de la sección 5 para cumplir con los objetivos planteados.

Para el parámetro de escala  $\psi$  se utilizó la fórmula de la ecuación (8).

$$\psi = \frac{\text{mean}(|\text{eig}(K_T)|)}{\|f_{ext}\|} \quad (8)$$

donde  $K_T$  es la matriz de rigidez tangente de la estructura,  $\text{eig}$  es el conjunto de los valores propios y  $\text{mean}$  es la función promedio.

Para la búsqueda de la primer solución el ángulo  $\theta_1$  se varía en  $[0; \pi/2]$  para asegurar un aumento de la carga en la primer solución.

De esta manera, los parámetros numéricos que se deben ingresar al programa para utilizar este método son:

- “analysisSettings.methodName = 'slowIntuitiveBranching';”. Este es simplemente el nombre del método para que ONSAS sepa que debe trabajar con el mismo.
- “analysisSettings.deltaT = 1 ; analysisSettings.finalTime = NP ;”. Estos dos definen la cantidad de puntos que se calculan desde la condición inicial. Cada ramificación que se encuentre tiene NP puntos (si deltaT = 1, lo cual se recomienda).
- “analysisSettings.stopTolForces”. Este parámetro define la tolerancia en el error relativo de las fuerzas para definir si un punto es solución.
- “analysisSettings.incremArcLen”. Este parámetro define la longitud de arco.
- “analysisSettings.numberOfAngles”. Este parámetro define la cantidad de valores que puede tomar cada ángulo en su respectivo intervalo.
- “analysisSettings.distanceBetweenBranches”. Este parámetro define la distancia mínima que debe existir entre dos soluciones desde el mismo punto. Esto se basa en que si se encuentra una ramificación, dos puntos cercanos de pequeña tolerancia pero asociados a la misma rama no se consideren como ramas distintas.

#### 4.1. Pseudocódigo

Se presenta en el algoritmo 1 un pseudocódigo que describe la metodología que se sigue para calcular la solución k-ésima (o soluciones).

---

#### Algoritmo 1 Pseudocódigo para el método actual

---

- 1: Se definen todos los  $x_k$  y  $\lambda_k$  candidatos según los distintos ángulos
  - 2: **if** no es la primer solución **then**
  - 3:     Se descartan todos los puntos que se encuentran a menos de  $\Delta l$  de la solución anterior
  - 4: **end if**
  - 5: Se calcula la fuerza interna y la fuerza externa para cada candidato
  - 6: Se calcula el error relativo de cada candidato
  - 7: **if** Existe candidato que cumpla la tolerancia **then**
  - 8:     Se descartan todos los candidatos que no cumplan la tolerancia
  - 9:     Se elije el mejor candidato y se lo considera como primer solución:  $x_k^{sol1}$
  - 10:     Se descartan todos los candidatos que estén a menos de “distanceBetweenBranches” de  $x_k^{sol1}$
  - 11:     Se empieza a contar el número de ramificaciones: NumBranches=1
  - 12:     **while** Quedan candidatos por considerar **do**
  - 13:         Se elije el mejor candidato restante y se lo considera como nueva solución:  $x_k^{sol(NumBranches+1)}$
  - 14:         Se descartan todos los candidatos que estén a menos de “distanceBetweenBranches” de  $x_k^{sol(NumBranches+1)}$
  - 15:         Se actualiza el número de ramificaciones: NumBranches=NumBranches+1
  - 16:     **end while**
  - 17:     Para cada solución se repite el análisis para buscar la siguiente hasta tener todas las buscadas
  - 18: **else**
  - 19:     Se elije el mejor candidato y se lo considera como única solución:  $x_k^{sol1}$
  - 20:     Se repite el análisis para buscar la siguiente hasta tener todas las buscadas
  - 21: **end if**
- 

#### 4.2. Implementación en ONSAS

Para lograr implementar este método en ONSAS se modificaron las siguientes funciones:

- *ONSAS.m*: En esta función se indica que por utilizar este método, en vez de utilizar *ONSAS\_solve.m* se utilice *ONSAS\_branchingSolve.m* la cual es una nueva función que se crea para la implementación.
- *computeRHS.m*: En esta función se indica que para este método se calcule el vector de fuerzas externas y la diferencia entre las fuerzas internas y externas, para cada uno de los candidatos.
- Se modifican otras funciones simplemente para evitar conflictos con funciones *if* que requieren el nom-

bre del método.

También se crearon las siguientes funciones:

- *NumberOfTestsCalculator.m*: Esta función sirve para calcular el número de candidatos que se consideran en cada iteración. Esto es para poder generar una salida a la consola de comandos que indique el progreso de cada iteración.
- *ONSAS\_branchingSolve.m*: Esta es la función encargada de resolver el algoritmo 1. Trabajando de forma recursiva va resolviendo cada rama hasta encontrarse con una bifurcación donde se llama a sí misma para resolver cada una de las ramificaciones. Para lograr almacenar todas las soluciones crea *cells* de las variables *matUs*, *loadFactorsMat* y *stabilityBoolean*. La primera, la cual ya existe en el código de base, es una matriz que contiene para cada paso el vector de desplazamientos de los nodos de la estructura. La segunda, la cual también existe en el código de base, es una matriz que contiene para cada paso, los factores de carga. La tercera, es una variable lógica nueva que indica para cada paso si la estructura es estable o no.
- *slowBranchingTests.m*: Esta es una función recursiva que sirve para seleccionar todos los candidatos y hacer el cálculo del error relativo si el número de grados de libertad de la estructura es mayor o igual a dos. El motivo por el cual se genera esta función de manera recursiva es por el comportamiento intrínsecamente recursivo de la definición de los ángulos para la esfera n-dimensional.

Los detalles exactos de la implementación pueden verse en el repositorio con los códigos utilizados al cual puede accederse mediante el enlace ubicado en la sección 8.

### 4.3. Aplicación al caso de estudio

Se aplicó este método para el caso de estudio presentado en la sección 3 para validar el mismo. Se decidió hacer 30 pasos en cada ramificación, con una tolerancia del error de fuerzas de  $1 \times 10^{-3}$ , una longitud de arco de 4 cm, una distancia mínima entre ramificaciones de 1 cm y 152 valores para cada ángulo.

En la figura 4 se presenta el camino recorrido por el nodo no apoyado, partiendo de una posición en  $z \approx 1,81$  m y en  $x \approx 0,85$  m. Se puede ver como comienza descendiendo el nodo sin desplazamientos en la horizontal hasta que se acerca el punto crítico y comienzan las bifurcaciones. Se aprecia un primer problema del método, asociado a la falta de comunicación entre distintas ramas, en el cual desde una rama se puede fácilmente saltar a la otra

aumentando rápidamente el número a calcular. En el análisis se terminaron obteniendo 46.080 ramificaciones las cuales en realidad son solo 3.

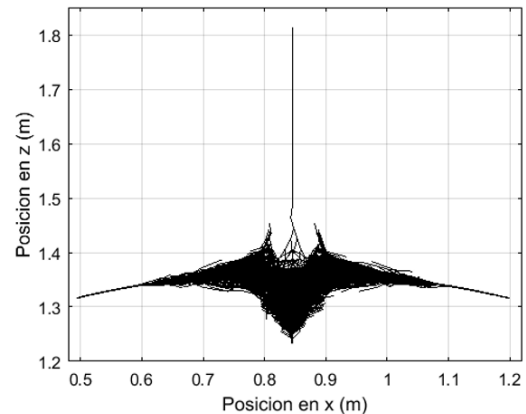


Figura 4: Trayectoria del nodo

Otro detalle que se logra apreciar es que existen algunas ramas que divergen, las cuales forman los “cuernos” de la figura. Se desconoce el motivo por el cual ocurren las mismas pero se sabe que de no haber admitido valores que no cumplieran la tolerancia, las mismas no hubieran ocurrido.

Vale la pena apreciar que, a pesar de las dificultades del método, se logra claramente apreciar las formas que adopta la estructura en las respectivas ramas de post pandeo, con posibles desplazamientos importantes en la horizontal o pudiendo mantener la rama principal y continuar descendiendo.

En la figura 5 se presenta el gráfico de la carga en función del desplazamiento horizontal. En azul con forma de  $\times$  se presentan los puntos estables y en rojo con forma de  $+$  se presentan los puntos inestables. Se logra apreciar claramente la forma simétrica inestable de la bifurcación según predecía la teoría.

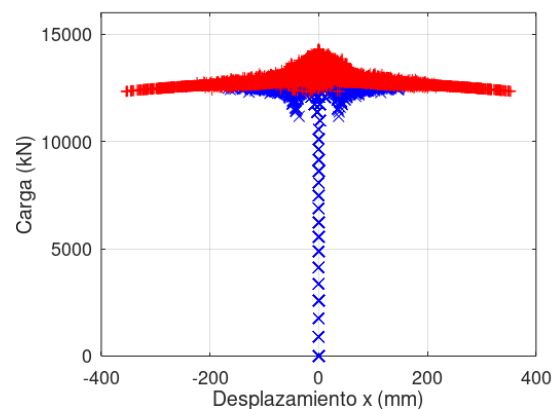


Figura 5: Curva carga - desplazamiento horizontal

Otro punto débil del método que logra apreciarse con este gráfico es la dificultad del mismo para determinar

claramente el punto crítico, existiendo una nube importante de puntos en su entorno.

Finalmente en las figuras 6 y 7 se presenta el gráfico de la carga en función del desplazamiento vertical, siendo la segunda figura una ampliación de la primera. Nuevamente se tiene el criterio de en azul y con forma de  $\times$  los puntos estables y en rojo con forma de  $+$  los puntos inestables. En este caso se logra apreciar, salvo las ramificaciones que divergen, una forma muy similar a la presentada en la figura 2 lo cual es muy satisfactorio.

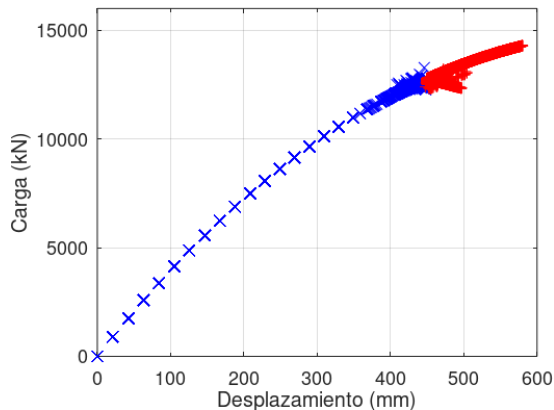


Figura 6: Curva carga - desplazamiento vertical

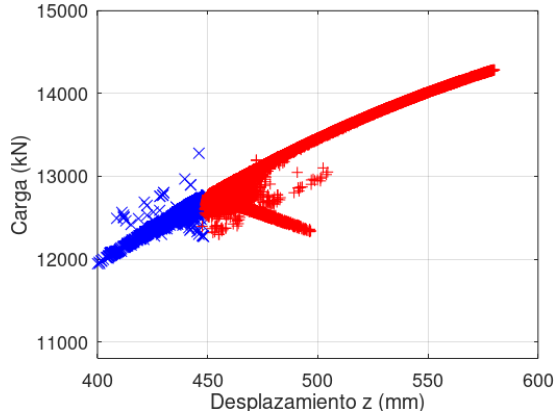


Figura 7: Ampliación a la bifurcación

Una última gran desventaja del método implementado son los tiempos de cómputo. En una computadora de escritorio común, de 16 GB de memoria RAM el mismo tardó 20 días en finalizar, e incluso en una computadora de alto poder de cómputo de 125 GB de memoria RAM apenas se logró bajar de 38 horas. Esto resulta ser un gran problema, ya que la estructura es sencilla y de aumentar levemente el número de grados de libertad los tiempos serían completamente inviables.

Se concluye que los resultados del método, para el caso de estudio, son bastante satisfactorios, pudiendo encontrar las distintas ramas de post-pandeo y permitiendo obtener una aproximación del punto crítico. De

todas formas, como existen desventajas muy importantes, se requiere intentar implementar un método con mayor optimización y que mejore las desventajas del actual.

En cuanto a la formación del autor de este informe, el haber desarrollado este método favoreció en cuanto al entendimiento de los métodos numéricos de longitud de arco y en la lógica de programación del código ONSAS lo que facilita la implementación del próximo método.

## 5. Método de branch switching de Crisfield

El segundo método que se presenta, está basado en los capítulos 20.2, 21.1 y 21.2 de [3] el cual se asocia a ir resolviendo la curva carga-desplazamiento por algún método estándar, pero al cruzarse un punto crítico se itera hasta hallar el mismo con cierta precisión y una vez ahí se desarrollen por el mismo método original las posibles ramificaciones.

El método que se utiliza para ir resolviendo la curva carga-desplazamiento hasta el punto crítico es el método de longitud de arco ya implementado en el código ONSAS. Una vez que se cruza un punto crítico se itera modificando la longitud de arco hasta obtener la precisión buscada.

Para estudiar cuándo se cruza un punto crítico se utilizan  $n$  funciones de prueba ( $\tau$ ), siendo  $n$  el número de valores propios negativos de la matriz de rigidez tangente (al inicio del paso)  $+1$ , acotado superiormente por el número de grados de libertad de la estructura. Las funciones  $\tau$  son los  $n$  menores valores propios de la matriz de rigidez tangente.

Se considera que se cruzó un punto crítico cuando alguna de las funciones  $\tau$  cambia de signo entre dos pasos consecutivos. En este momento comienza un proceso de “bracketing” ajustando la longitud de arco hasta lograr obtener con suficiente precisión el punto crítico según se explica a continuación.

Se definen los siguientes cuatro parámetros:

- $x_1$  y  $x_2$  las longitudes de arco que encierran, por izquierda y por derecha respectivamente, la longitud de arco que daría exactamente el punto crítico. En otras palabras, son las longitudes de arco en las cuales, tras determinar el “próximo paso”, se obtienen los valores de funciones de prueba  $\tau_1$  y  $\tau_2$ .
- $\tau_1$  y  $\tau_2$  son las funciones de prueba, asociadas al punto crítico, tal que  $\tau_1$  mantiene el mismo signo que la función de prueba en el paso anterior y  $\tau_2$  es del

signo contrario. Ambas funciones tienen el menor valor absoluto obtenido hasta el momento en el proceso de “bracketing”.

En la primera iteración  $x_1 = 0$  y  $x_2 = \Delta l$ . El siguiente intento de longitud de arco ( $x$ ) es según la ecuación (9).

$$x = x_2 - \tau_2 \frac{x_1 - x_2}{\tau_1 - \tau_2} \quad (9)$$

Una vez definido  $x$  se calcula el siguiente paso de la curva carga-desplazamiento utilizando como longitud de arco este valor y se evalúa la función de prueba en esta solución. Según el signo de la misma se actualizan los valores de  $\tau_1$  y  $x_1$  o  $\tau_2$  y  $x_2$  y se repite el proceso hasta que la función definida en la ecuación (10) tenga un valor menor a cierta tolerancia definida por el usuario o se supere el máximo de iteraciones permitidas.

$$\eta = \sqrt{\frac{|\bar{\tau}| \Delta x}{\Delta l}} \quad (10)$$

con:

$$\Delta x = x_2 - x_1 \quad (11)$$

$$\bar{\tau} = \frac{\tau}{\sqrt{|\tau_1 \tau_2|}} \quad (12)$$

siendo  $\tau$  el valor de la función de prueba de la solución dada por  $x$ .

Una vez encontrado el punto crítico se calcula una base ortonormal de los vectores propios de la matriz de rigidez tangente en este punto, asociados a valores propios cuyo valor absoluto sea menor a la tolerancia que debía cumplir la función  $\eta$ . Estos vectores, denominados  $z$  en lo que resta del documento, son las direcciones de las que parten las bifurcaciones asociadas al punto crítico.

Se estudia para cada una de las direcciones  $z$  si las mismas están asociadas a una bifurcación propiamente dicha o a un punto límite; mediante evaluar el producto interno de este vector con el vector de fuerzas externas. Si el producto interno resulta nulo, según la tolerancia dada para el error en el balance de fuerzas, la dirección se asocia a una bifurcación, en caso contrario se asocia a un punto límite.

Una vez definidas las direcciones de la bifurcación, en primer lugar se repite el proceso de ir estudiando la rama fundamental hasta el próximo punto crítico partiendo de la solución en la que cambió el signo de la función de prueba.

Luego, para cada una de las direcciones  $z$  asociadas a bifurcaciones se repite el proceso, estudiando puntos críticos y futuras bifurcaciones a menos que el usuario

decida no querer estudiarlas en ramificaciones de la rama fundamental. Para esto se parte, para el método de longitud de arco, suponiendo  $\Delta x = \pm z \Delta l$ , obteniendo dos ramas por dirección.

De esta manera, los parámetros numéricos que se deben ingresar al programa para utilizar este método, sin considerar los ya requeridos para el método de longitud de arco de ONSAS, son:

- “analysisSettings.methodName = 'branchArcLength';”. Este es simplemente el nombre del método para que ONSAS sepa que debe trabajar con el mismo.
- “analysisSettings.stopTestFunction”. Este parámetro define la tolerancia para el valor de  $\eta$  de la ecuación (10).
- “analysisSettings.branchFromBranches”. Este parámetro es una variable lógica que define si se estudian puntos críticos y ramificaciones en las ramificaciones de la rama fundamental.

## 5.1. Pseudocódigo

Se presenta en el algoritmo 2 un pseudocódigo que describe la metodología que se sigue en el cálculo de cada paso.

---

### Algoritmo 2 Pseudocódigo para el método actual

---

- 1: Se calcula la siguiente solución ( $x_k$  y  $\lambda_k$ ) mediante el método de longitud de arco.
- 2: CritFoundBoolean=FALSE
- 3: **if** Cambió el signo de alguna función de prueba **then**
- 4:     Se corre el algoritmo 3
- 5:     Se actualiza  $x_k$  y  $\lambda_k$  como la solución final del anterior algoritmo
- 6:     CritFoundBoolean=TRUE
- 7: **end if**
- 8:  $K_t$ : Matriz de rigidez tangente dado  $x_k$ .
- 9: **if** CritFoundBoolean **then**
- 10:     Se calculan  $V$  los vectores propios de  $K_t$
- 11:     Se descartan todos cuyos valores propios asociados tengan valor absoluto  $\geq$  “stopTestFunction”.
- 12:      $z$ : Vectores  $V$  restantes.
- 13:     Se ortonormaliza  $z$ .
- 14:     Se corre este algoritmo sin modificar nada
- 15:     Se guardan todas las soluciones del paso anterior encabezadas por las soluciones hasta este punto.
- 16:     **for** Index=1:(Cantidad de  $z$ ) **do**
- 17:          $q$ : Vector de fuerzas externas
- 18:         **if**  $\langle q; z(Index) \rangle \leq ||q||$  “stopTolForces” **then**
- 19:             Se corre este algoritmo considerando para primera iteración del próximo paso  $\Delta x = z(Index) \Delta l$

```

20:         Se guardan todas las soluciones del pa-
           so anterior encabezadas por las soluciones hasta este
           punto
21:         Se corre este algoritmo considerando pa-
           ra primera iteración del próximo paso  $\Delta x =$ 
            $-z(Index)\Delta l$ 
22:         Se guardan todas las soluciones del pa-
           so anterior encabezadas por las soluciones hasta este
           punto
23:         end if
24:     end for
25:     Se termina el algoritmo
26: else
27:     Se repite el análisis para buscar la siguiente solu-
           ción hasta tener todas las buscadas
28: end if

```

También se presenta en el algoritmo 3 un pseudocódigo que describe la iteración para hallar el punto crítico.

---

### Algoritmo 3 Pseudocódigo para la búsqueda del punto crítico

---

```

1:  $\Delta l_0 =$  Longitud de arco original
2:  $\tau_1 =$  Función de prueba en paso anterior
3:  $\tau_2 =$  Función de prueba que determinó la cruza de un
   punto crítico
4:  $x_1 = 0$ 
5:  $x_2 = \Delta l_0$ 
6:  $\eta = +\infty$ 
7:  $Tol_\tau$ : Tolerancia para hallar puntos críticos
8:  $Max_{it}$ : Máximo número de iteraciones
9:  $it = 1$ 
10: while  $\eta \geq Tol_\tau$  &  $it \leq Max_{it}$  do
11:      $x = x_2 - \tau_2(x_1 - x_2)/(\tau_1 - \tau_2)$ 
12:     Se actualiza la longitud de arco:  $\Delta l = x$ 
13:     Se recalculan  $x_k$  y  $\lambda_k$  con la nueva longitud de
       arco
14:     Se calcula  $\tau$ : Función de prueba dada  $x_k$ 
15:     if Signo de  $\tau =$  Signo de  $\tau_1$  then
16:          $x_1 = x$ 
17:          $\tau_1 = \tau$ 
18:     else
19:          $x_2 = x$ 
20:          $\tau_2 = \tau$ 
21:     end if
22:     Se calcula  $\eta$  según ecuación (10)
23:      $it = it + 1$ 
24: end while
25: Se restaura la longitud de arco:  $\Delta l = \Delta l_0$ 

```

---

## 5.2. Implementación en ONSAS

Para lograr implementar este método en ONSAS se modificaron las siguientes funciones:

- *computeMatrix.m*: En esta función simplemente se agregó el nuevo método a los cálculos estáticos de la matriz de rigidez tangente.
- *computeRHS.m*: En esta función se agrega el nuevo método a la sección originalmente destinada a *arcLength*.
- *computeDeltaU.m*: Además de un cambio totalmente análogo a de *computeRHS.m*, se implementó un “kill switch” tal que si en algún paso el discriminante del método de longitud de arco es negativo, se termine el cálculo de la ramificación en curso sin dar error en todo el programa.
- *timeStepIteration.m*: Además de agregar el nuevo método a la sección originalmente destinada a *arcLength*, se cambia el punto de partida de las iteraciones, en el caso de una bifurcación, por el indicado en el pseudocódigo. También se generan unos cambios para implementar el “kill switch” de la función anterior.
- *ONSAS.m*: Similar a la sección 4.2, se indica que por utilizar este método se utilice *ONSAS\_branchingSolve.m*.
- *ONSAS\_branchingSolve.m*: Se agrega este nuevo método entre las opciones que puede resolverse, teniendo en cuenta los detalles presentados en el pseudocódigo.

Los detalles exactos de la implementación pueden verse en el repositorio con los códigos utilizados al cual puede accederse mediante el enlace ubicado en la sección 8.

## 5.3. Aplicación al caso de estudio

Se aplicó este método para el caso de estudio presentado en la sección 3 para validar el mismo. En este caso se decidió hacer 50 pasos en cada ramificación, con tolerancias en el error de fuerzas y desplazamientos de  $1 \times 10^{-6}$ , una longitud de arco de 3 cm, una tolerancia para hallar puntos críticos de  $1 \times 10^{-4}$  y buscando ramificaciones en las ramificaciones.

En la figura 8 se presenta nuevamente el camino recorrido por el nodo no apoyado, mediante el nuevo método. En este caso se aprecia claramente como el nodo comienza descendiendo sin desplazamientos horizontales hasta el punto crítico en el cual se bifurcan 3 caminos, uno que mantiene el descenso vertical sin deformaciones laterales y dos simétricos que parten de un desplazamiento puramente horizontal. Esta es una primera verificación del modelo, en particular porque, a diferencia del método anterior, el número de bifurcaciones encontradas coincide con el real.



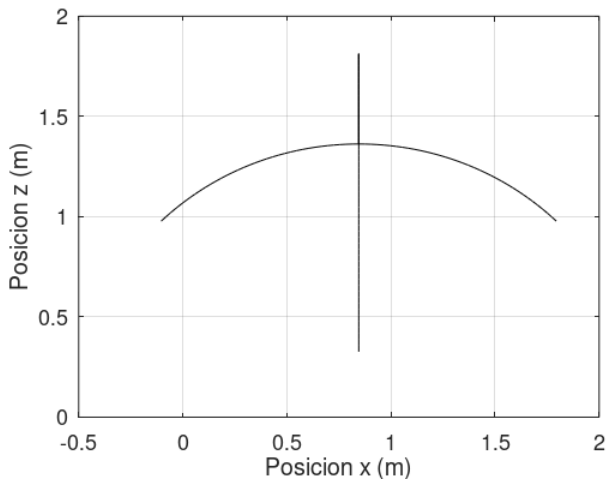


Figura 8: Trayectoria del nodo

Por otra parte, en la figura 9, se presenta el gráfico de la carga en función del desplazamiento horizontal. En azul y con forma de  $\times$  se presentan los puntos estables, en rojo con forma de  $+$  se presentan los puntos inestables y en verde con forma circular se presenta el punto crítico (según la tolerancia anteriormente definida). Nuevamente se logra apreciar claramente la forma simétrica inestable predicha por la teoría.

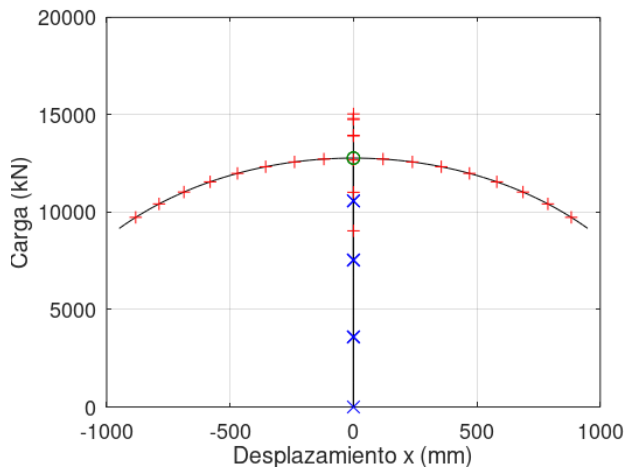


Figura 9: Curva carga - desplazamiento horizontal

Vale la pena notar la existencia de puntos inestables superpuestos con puntos estables en la figura 9. Los mismos se deben a la fracción inestable y estable de la rama fundamental las cuales tienen desplazamientos horizontales nulos (por lo que aparentan superpuestas en la figura) pero presentan desplazamientos verticales distintos según se presentará a continuación.

Finalmente, en las figuras 10 y 11 se presenta el gráfico de la carga en función del desplazamiento vertical con el mismo criterio de colores y formas que la figura anterior. En esta figura se logra apreciar una forma virtualmente idéntica a la presentada en la figura 2 lo que implica una nueva verificación del método.

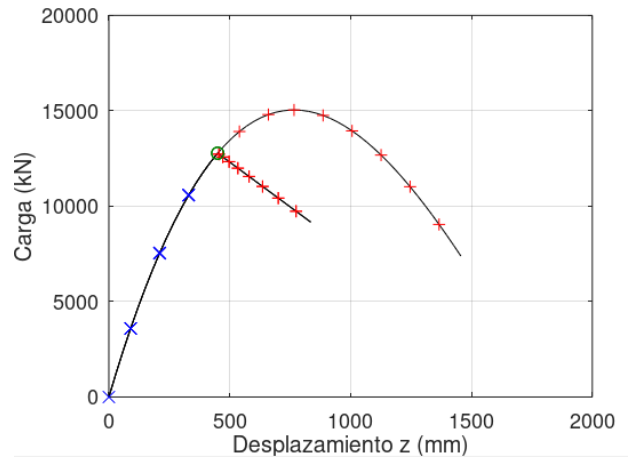


Figura 10: Curva carga - desplazamiento vertical

En la figura 11 se aprecia que el valor de la carga crítica es  $P_{crit} = 12\,777\text{ kN}$  el cual es bastante similar al indicado en la referencia por lo que el método resulta satisfactorio.

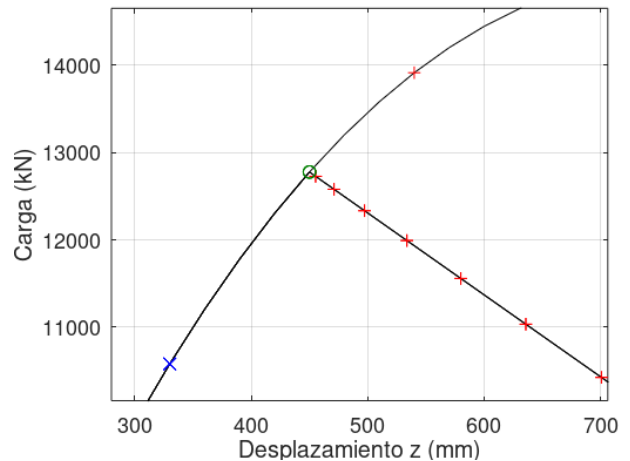


Figura 11: Ampliación a la bifurcación

Se concluye que los resultados del método, para el caso de estudio, son completamente satisfactorios, pudiendo concluir que el mismo se encuentra validado para esta estructura en particular y puede probarse en estructuras de mayor complejidad.

A su vez, el nuevo método posee tiempos de cómputo considerablemente mejores que el anterior, pudiendo resolver esta estructura con las tolerancias mencionadas en tan solo 30 segundos. Debido a estos dos motivos, este es el método utilizado para resolver la estructura de la siguiente sección.

## 6. Aplicación a un caso tridimensional

Tras haber validado el método de la sección 5 con el caso de estudio de la sección 3, se procede en esta sección a utilizar este método para resolver una estructura tridimensional de mayor complejidad. Se hace uso del domo

tridimensional de la sección 22.3 de [3] el cual es presentado en la figura 12 cuyas unidades están en metros. La rigidez a directa de las barras que conforman la estructura es  $EA = 1 \times 10^4$  N y la medida de deformación no lineal que se utiliza es la deformación de Green.

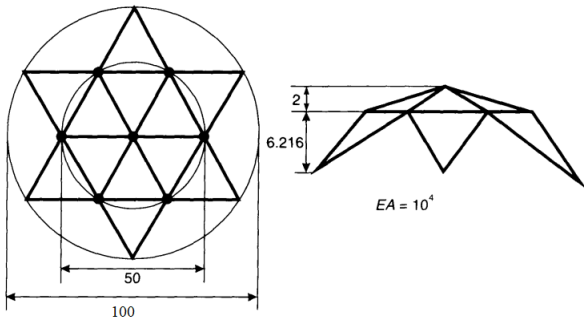


Figura 12: Estructura a analizar [3]

Los nodos del anillo exterior se encuentran apoyados en todas las direcciones y todos estos apoyos son articulaciones. Los nodos del anillo interior se encuentran cargados con cargas verticales descendentes de valor  $P$  mientras que el nodo de la cúspide se encuentra cargado con una carga vertical descendente de valor  $P/2$ . En los gráficos que se presentarán a continuación el eje de “cargas” indica el valor de  $P$ . En la figura 13 se presenta el modelo tridimensional que es ingresado al programa.

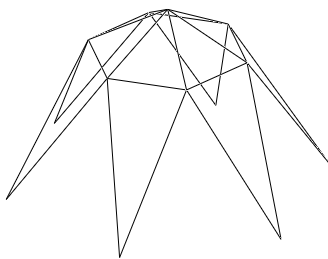


Figura 13: Esquema 3D de la estructura

Las posibles curvas carga-desplazamiento vertical del nodo de la cúspide es según se presenta en la figura 14 donde vale la pena apreciar la existencia de bifurcaciones dobles en el segundo y tercer punto crítico de la rama fundamental.

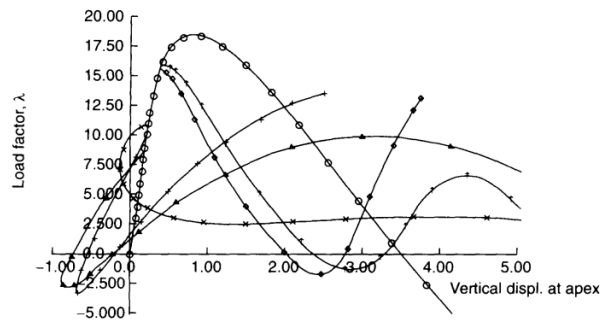


Figura 14: Curva de la referencia [3]

Se aplicó el método desarrollado utilizando 8000 pasos en cada ramificación, con tolerancias en el error de fuerzas y desplazamientos de  $1 \times 10^{-6}$ , una longitud de arco de 2 mm, una tolerancia para hallar puntos críticos de  $1 \times 10^{-4}$  y únicamente buscando ramificaciones de la rama fundamental. La versión análoga a la figura 14 obtenida con el método es la figura 15.

La gran cantidad de pasos corridos y la pequeña longitud de arco utilizada, los cuales llevaron a que el método tardara 2 horas en una computadora común, se asocian fundamentalmente a la búsqueda de poder trazar la curva asociada a la tercer bifurcación la cual, como se presentará más adelante, es compleja de obtener.

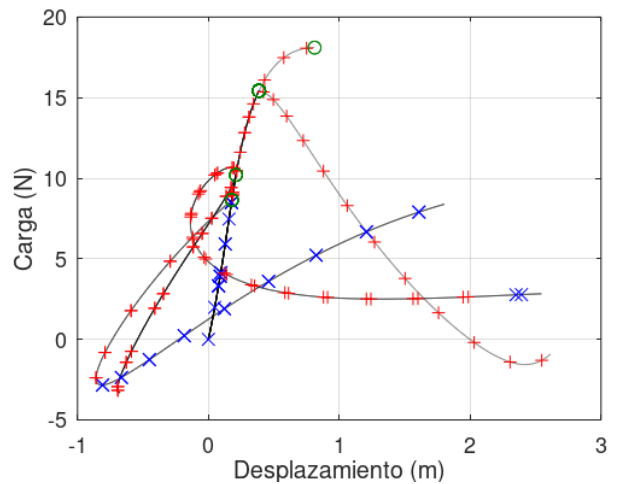


Figura 15: Curva obtenida por el método implementado

En primer lugar, vale la pena apreciar que la forma de las curvas es satisfactoriamente similar a la presentada en la referencia. La única diferencia notable que se obtiene es la falta de una ramificación en el tercer punto crítico la cual no pudo obtenerse en un amplio espectro de longitudes de arco consideradas. Las otras diferencias son algunas ramificaciones que para los parámetros utilizados no pudieron ser recorridas completamente.

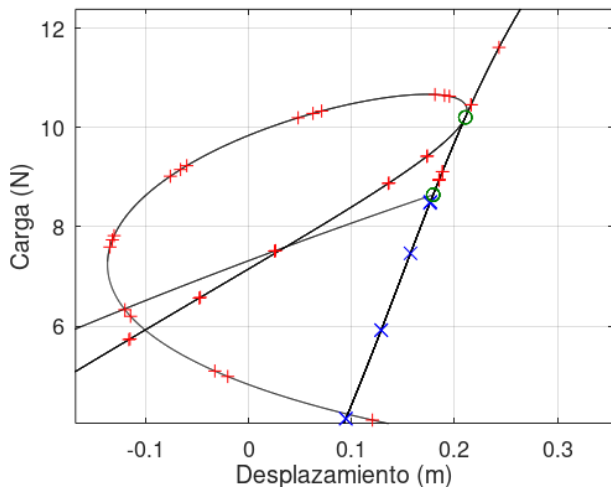


Figura 16: Ampliación a las dos primeras bifurcaciones

En la figura 16 se logra apreciar el comportamiento estructural en la segunda bifurcación, la cual es doble. Esto implica que son dos los vectores propios, ortogonales entre sí, los que forman las 4 posibles ramificaciones a seguir. Se aprecian solamente dos ya que están superpuestas en este gráfico.

Por otra parte, en la figura 17 se presenta una ampliación a la bifurcación obtenida del tercer punto crítico, donde se logra apreciar la dificultad asociada a trazar las ramificaciones desde el mismo mediante el método actual. Los puntos verdes representan configuraciones en las que el mínimo valor absoluto de los valores propios de la matriz de rigidez tangente es menor a la tolerancia indicada. Esta ramificación, junto con las que no pudieron ser trazadas, poseen en sus inicios matrices de rigidez tangente casi singulares las cuales dificultan el cálculo de las soluciones.

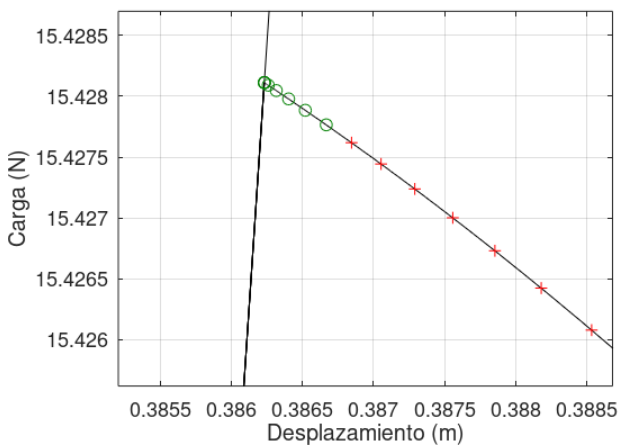


Figura 17: Ampliación a la tercer bifurcación

En las figuras 18 a 22 se presenta a modo ilustrativo las configuraciones deformadas del domo para las distintas ramificaciones, indicadas por el esquema de arriba a la derecha de cada figura, donde se presenta con un círculo

verde la configuración de carga y desplazamientos considerada.



Figura 18: Deformada de la primer ramificación

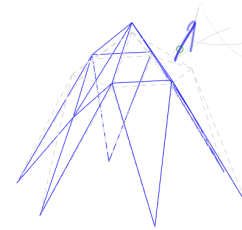


Figura 19: Deformada de la segunda ramificación tipo 1

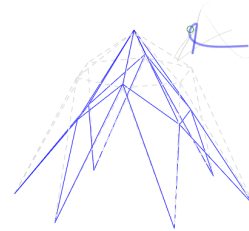


Figura 20: Deformada de la segunda ramificación tipo 2

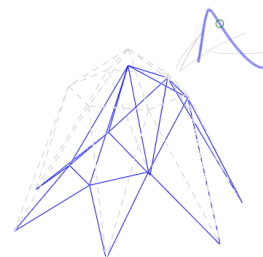


Figura 21: Deformada de la tercer ramificación

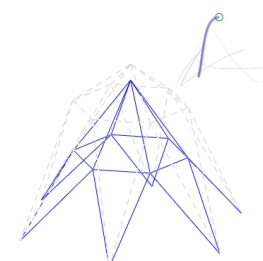


Figura 22: Deformada sobre el punto límite

Se concluye que, aún en una estructura compleja y tri-dimensional, el método desarrollado es capaz de resolver satisfactoriamente el comportamiento pre y post crítico de las estructuras, si bien requiere algunas mejoras para lograr captar con precisión ciertos comportamientos complejos, en particular cuando la matriz de rigidez tangente posee valores propios de reducido valor absoluto.

## 7. Conclusiones y trabajos a futuro

Finalizado el trabajo se puede concluir que se logró implementar en ONSAS dos métodos basados en el método de longitud de arco, capaces de estudiar las distintas ramas de post-pandeo que pueden ocurrir en estructuras de reticulados.

El primero de los métodos posee una clara desventaja en lo referente a los tiempos de cómputo y a la precisión alcanzable por el mismo. Entre los posibles trabajos a futuro se podría buscar mejorar este método, buscando mecanismos para lograr una mayor eficiencia computacional y evitando redundancia de ramificaciones.

El segundo método resulta completamente satisfactorio para lograr estudiar el comportamiento post crítico de una estructura sencilla; pero a la hora de estudiar una estructura compleja el mismo puede sufrir complicaciones al evaluar el comportamiento estructural en situaciones en que la matriz de rigidez tangente es aproximadamente singular. Como trabajo a futuro se podría indagar en el uso de formulaciones del método de longitud de arco, de “branch switching” y de “bracketing” que consideren derivadas de mayor orden.

Además se podría considerar la posibilidad de extender estos métodos a formulaciones de elementos finitos más complejas (vigas, cáscaras, elementos 3d, etc.) y buscar implementar mecanismos de variación de la longitud de arco para evitar el cálculo de una gran cantidad de puntos en los tramos cuasi-lineales de las estructuras.

Finalmente, se presenta la posibilidad de profundizar en el comportamiento de post-pandeo en las bifurcaciones de mayor orden ya que, si bien en el segundo método implementado se evalúa el comportamiento en distintas direcciones ortogonales entre sí, existen infinitas direcciones para poder seleccionar.

## 8. Códigos utilizados e información de contacto.

El código adaptado de ONSAS junto con los ejemplos utilizados pueden encontrarse en el siguiente vínculo:

<https://gitlab.fing.edu.uy/santiago.delgado/trabajo-anle-2021>. En el mismo también puede encontrarse una versión pdf del presente informe.

Ante cualquier consulta referente al informe, es posible contactar al autor mediante el correo electrónico: *sdelgado97@gmail.com*.

## Referencias

- [1] Zdeněk Bažant and Luigi Cedolin. *Stability of Structures. Elastic, Inelastic, Fracture and Damage Theories*. World Scientific, 1st edition, 2010. Capítulo 3.5.
- [2] Juan Bruno Bazzano and Jorge Pérez Zerpa. *Introducción al análisis no lineal de estructuras*. Universidad de la República, Montevideo, 1st edition, 2017.
- [3] M.A. Crisfield. *Non-linear finite element analysis of solids and structures. Vol 2: Advanced Topics*. John Wiley & Sons, 3rd edition, 2000.
- [4] Lei Li and Kapil Khandelwal. *Topology optimization of geometrically nonlinear trusses with spurious eigenmodes control*. Engineering Structures 131, pages 324-344, 2017.