

Módulo de Simulación de Trozado de Árboles del Software “*calcuLEMUs*”

Federico Jorge Letourneau ¹

¹ INTA – Centro Regional Patagonia Norte,
Campo Forestal Gral. San Martín,
Estación Experimental Agropecuaria Bariloche,
Ruta Nacional Nro 40 Km 1.911
Lago Puelo, Prov. Chubut.

letourneau.federico@inta.gob.ar

Resumen. Planificar el manejo forestal sustentable requiere estimar tasas de extracción, y caracterizar productos del trozado del tronco. El procesamiento de datos es intenso y tedioso de realizar manualmente, cuando se trabaja con grandes volúmenes de datos. *calcuLEMUs* calcula datos de inventarios forestales y simplifica al usuario esta tarea. Aquí se presenta el funcionamiento del módulo de trozado de troncos, basado en un algoritmo computacional que evalúa la obtención de hasta siete clases de productos definidos por el usuario. Para esto emplea un modelo matemático que describe la forma del tronco desde el tocón hasta el ápice. Estima el volumen y cantidad de los productos por árbol, priorizando por tamaño, en las tres especies de árboles más comunes en plantaciones de Patagonia Andina. Presenta una vista gráfica por árbol, y los resultados se reportan por árbol, y acumulado de la parcela en formato de planilla de cálculo.

Palabras clave: plantaciones forestales, programación modular, coníferas, Patagonia.

1 Introducción

El software *calcuLEMUs* es un calculador de datos de inventarios forestales, su nombre es un juego de palabras entre el término derivado de *calculus* o *calculi* que en latín era la piedra o tabla utilizada para sumar o realizar cuentas matemáticas, y por otra parte el vocablo de origen Mapuche *LEMU*, utilizado para denominar a un bosque o arboleda. Su principal ventaja respecto de otras alternativas es su simplicidad para procesar y administrar información, y está destinado a personal técnico que gestiona recursos forestales. Este se presentó en el XII Concurso Nacional de Innovaciones, y fue distinguido con la publicación en el catálogo INNOVAR 2016 en el rubro de Investigaciones Aplicadas (bajo el Nro 19556, Ministerio de Ciencia ,

Tecnología e Innovación Productiva, 2016), y su versión básica puede ser descargada desde <http://sipan.inta.gob.ar/#Herramientas>.

Este trabajo tiene por objeto presentar el desarrollo de un módulo que simula el trozado del tronco de árboles, y describir como se comporta cuando es invocado. En términos de diseño *calcuLEMUs* hace uso de programación modular, es decir que su código es un ensamble de distintos módulos, que facilita una implementación eficiente de las operaciones y mantenimiento del mismo [1].

Los inventarios forestales pueden ser de muy distinta magnitud, hay inventarios nacionales y provinciales / regionales [2], o locales [3]. En cualquiera de estas situaciones el cálculo de los productos con destino industrial u otros usos (rollizos, postes, varas o leña) requiere de realizar múltiples operaciones en planillas de cálculo, y ecuaciones de diversa complejidad. El manejo de esta información es una tarea tediosa, demandante de tiempo, y no exenta de errores.

Simular el trozado de un árbol requiere de una descripción matemática muy precisa de el perfil de su tronco. Esto se realiza a través de una ecuación que predice el diámetro de el tronco a una cierta altura. En coníferas con desarrollo monopódico, la forma de el tronco puede dividirse en tres partes características. Primero la porción inferior, desde donde el árbol desarrolla sus raíces hacia arriba (neiloide), segundo la sección intermedia y de mayor desarrollo (frustum de paraboloides), y tercero el extremo superior de el tronco próximo al ápice (cónico) [4]. La prevalencia de cada una de estas partes en el tronco depende de la especie y el tamaño de el árbol. Existen diversos modelos matemáticos para describir esto, y en todos los casos los parámetros son estimados mediante técnicas de ajuste estadístico. Es deseable que esta función matemática posea ciertas propiedades, que dan consistencia a las estimaciones. Por ejemplo, que las tres porciones características de el tronco puedan ser descriptas por una sola ecuación, sin necesidad de realizar supuestos en cuanto a la forma del árbol. También es importante que la estimación de los diámetros tenga sentido biológico y consistencia matemática.

El algoritmo de trozado compara las dimensiones de el tronco respecto de el tamaño de los productos que se desean obtener. Para realizar estas estimaciones hay que desarrollar un código de computación que opere las distintas ecuaciones matemáticas y genere salidas de resultados para cada árbol y parcela de la base de datos. La integración de este complejo proceso con un software preexistente debe abordarse a través de un diseño de software apropiado. A continuación se describe en detalle el algoritmo computacional de simulación de trozado y sus componentes.

2 Materiales y Métodos

El software *calcuLEMUs* está programado en MATLAB® [5], y es distribuido por medio de una compilación para el sistema operativo Microsoft Windows® para computadoras de 64 Bit. MATLAB® al igual que otros lenguajes de programación permite crear funciones para realizar tareas específicas dentro de un programa. En este

caso se optó por desarrollar una función (ecuación 1) para simular el número y el volumen de productos (trozas, postes, varas, etc) que un árbol puede brindar de acuerdo a la tabla de especificación de productos deseados. La función que se propone acepta argumentos como el diámetro y altura del árbol, una caracterización de la forma del tronco que es propia de la especie (ecuación 2), una forma de calcular el volumen de las trozas (ecuación 3), parámetros para la salida gráfica, y otros datos para asegurar la consistencia de la información en relación a otros procedimientos de *calcuLEMUs* (ecuación 4).

Esta función está programada para que el proceso de simulación de trozado sea secuencial o por prioridad y recursivo (Tabla 1). Es secuencial por que procede de forma unidireccional desde la base del árbol hacia el ápice, y prioriza la obtención de los productos de mayor tamaño definidos por el usuario. Es recursivo por que en cada paso de la secuencia prueba una y otra vez, mediante un bucle, si es posible obtener un producto con una determinada especificación.

$$\text{resumenProductos} = \text{trozarArbol}(Dap, A_{total}, \text{especProductos}, \text{perfil}, \text{volBrtInd}, \text{activarGrafico}, \text{Parcela}, \text{Arbol}) \quad (1)$$

Donde:

- *resumenProductos*: vector de tamaño $1 \times 2 \times m$, donde se detalla el volumen y cantidad de productos obtenidos por simulación de trozado. Donde m es la cantidad de productos definidos por el usuario.
- *trozarArbol*: nombre de la función de trozado.
- *Dap*: es un escalar, que cuantifica el diámetro del tronco a 1,3 m de altura, expresado en milímetros.
- *A_{total}*: es un escalar, que cuantifica la altura total del árbol expresado en metros, desde el nivel del suelo.
- *especProductos*: es una matriz $n \times m$, donde $n = 3$ y $m \leq 7$. En la primera fila se describe j , el número de producto, la segunda fila el diámetro en punta fina (D_j) expresado en milímetros y la tercera fila es la longitud del producto ($longProducto_j$) expresado en metros, cada columna corresponde a un producto diferente.
- *perfil*: En una expresión en formato de texto que se pasa a la función para que esta la evalúe internamente. La expresión es la función de perfil de fuste descrita en la ecuación (2), con los parámetros propios de la especie de árbol (Tabla 2).
- *volBrtInd*: volumen bruto del tronco del árbol expresado en metros cúbicos, es el volumen total del tronco, incluye el tocón. El volumen, de acuerdo a la especie, es estimado mediante ecuaciones de volumen disponible para Pino ponderosa [6], Pino Oregón [7] y Pino Murrayana [8], y se emplea para conciliar los valores estimados mediante el uso del perfil de fuste de tal forma de brindar resultados consistentes entre el módulo de trozado y otros parámetros de *calcuLEMUs*.
- *activarGrafico*: es una variable de tipo boolean que indica a la función si es que tiene que generar una salida gráfica (TRUE o FALSE).

- *Parcela, arbol*: es el número de identificación del árbol y su parcela en la base de datos y se emplea en la salida gráfica.
- *j*: Es un índice utilizado para identificar las especificaciones de un producto en la matriz *especProductos*, toma valores de acuerdo a $1 \leq j \leq 7$.

Para describir el perfil de fuste del tronco se empleó un modelo trigonométrico (ecuación 2, [9]). El ajuste estadístico de los parámetros se realizó con la función “nls” del paquete estadístico nlme [10] del software R [11], empleando el IDE Rstudio [12].

$$D_i = Dap * (\log(\sin(\pi/2 * ((A_i / A_{total})))) / \log(\sin(\pi/2 * (1.3 / A_{total})))) ^ { (p1 + (p2 * \sin(\pi/2 * (A_i / A_{total}))) + (p3 * \cos(3 * \pi/2 * (A_i / A_{total}))) + (p4 * \sin(\pi/2 * (A_i / A_{total})) / (A_i / A_{total})) + (p5 * Dap) + (p6 * (A_i / A_{total}) * (Dap ^ 0.5)) + (p7 * (A_i / A_{total}) * (A_{total} ^ 0.5)))) } \quad (2)$$

Donde:

- D_i : diámetro del tronco a la altura A_i expresado en milímetros.
- Dap : como se definió con anterioridad.
- A_{total} : como se definió con anterioridad.
- A_i : Altura del tronco, toma valores entre A_{tocon} y A_{total}
- A_{tocon} : es la altura del tocón definido al simular el volteo del árbol, por defecto el simulador usa un valor constante de 30 cm.
- $p1$ a $p7$: parámetros de la ecuación de forma del tronco ajustados por métodos estadísticos.

Para el ajuste de las funciones de perfil de el tronco se empleó la base de datos de árboles muestra del proyecto PNFOR 1104074 [13] para la zona Andino Patagonica. En el caso de *Pseudotsuga mensiezii* se contó con 76 árboles, con Dap entre 195 mm y 490 mm y A_{total} entre 11,7 m y 36,9 m. Para *P. ponderosa* se emplearon 117 árboles con Dap de 90 mm a 560 mm, y A_{total} de 6,8 m a 35,8 m. Mientras que para *P. con-torta* se emplearon 55 árboles de Dap 195 mm a 409 mm y A_{total} 11,7 m a 36,9 m

En la Tabla 1, se presenta el código del algoritmo recursivo y secuencial de simulación del trozado. El uso de distintos condicionantes evita la inconsistencia de los resultados, como por ejemplo simular el trozado de árboles pequeños que no tiene el tamaño suficiente para generar un producto. El resultado de este algoritmo, es una matriz (*trozado*) que contiene las alturas sobre el nivel del suelo en donde se trozó el árbol, el valor del diámetro en ese punto del tronco y el código que identifica cuál es el producto obtenido. Con esta información se calcula el volumen de cada troza de acuerdo a la ecuación 3 [4], y estos valores son acumulados en una matriz denominada $\sum volTroza$.

Tabla 1. Código de programación del bucle de simulación de trozado.

```

Ai = 0.3;% altura del tocón en metros
Di = round(eval(perfil));% diámetro del tronco en el tocón
trozado = [Ai Di especProductos{1,j}];% almacena dimensiones del producto
trozado
for j = 1:length(especProductos{2,:})%para cada producto definido hacer lo
sgte:
    if j == 1% Si se evalúa el primer producto entonces
        if Ai + especProductos{3,j} <= Atotal % si la altura del tocón más la
            longitud del primer producto es menor o igual a la altura del árbol
            hacer:
                Ai = Ai + especProductos{3,j};%actualizar la altura de trozado
                Di = eval(perfil);% estimar el diámetro del tronco en Ai
                if Di < especProductos{2,j}% si el diámetro del tronco es me-
                    nor que la especificación del producto no hacer nada.
                else %caso contrario
                    trozado = [trozado; [Ai Di especProductos{1,j}]]; almacena
                        altura y diámetro en punta fina y el número de identifica-
                        ción del producto trozado.
                end
            end
        else % en caso de que se evalúe el segundo producto o posterior entonces
            if (Ai - especProductos{3,j-1}) < 0 % si la altura de trozado es ne-
                gativa entonces no hacer nada.
            else %si la altura de trozado es positiva
                Ai = Ai - especProductos{3,j-1}; %recalcular la altura de corte des-
                    haciendo la longitud de corte del producto y utilizar la altura del
                    producto anterior.
                Di = eval(perfil); % estima el diámetro en punta fina
            end
        end
        while Di >= especProductos{2,j}% Mientras que el diámetro del tronco en
            la altura de trozado sea mayor que el diámetro del producto hacer:
            Ai = Ai + especProductos{3,j};% Actualizar la altura del trozado
            Di = eval(perfil); % estimar el diámetro del tronco en Ai
            if Di < especProductos{2,j}% si el diámetro del tronco es menor que
                el producto entonces, quiere decir que no es posible trozar el pro-
                ducto j
                j = j + 1;% pasa a la especificación del producto siguiente
                break; ;% sale del bucle del while
            end
            trozado = [trozado; [Ai Di especProductos{1,j}]];% Almacena las di-
                mensiones de la trozas para su posterior cálculo de volumen.
            end
        end %Fin del bucle
    end

```

$$volTroza_{ij} = (((\pi * D_i^2)/4 + (\pi * D_{i-1}^2)/4)/2) * longProducto_j \quad (3)$$

Donde:

- $volTroza_{ij}$: volúmen de la troza i expresado en metros cúbicos.
- D_i : diámetro de el tronco a la altura A_i , donde $D_i < D_{i-1}$, estimado por (2)
- D_{i-1} : diámetro de el tronco a la altura A_{i-1} , donde $D_{i-1} > D_i$, estimado por (2)
- $A_i > A_{i-1}$
- π : (pi) es la relación entre la longitud de una circunferencia y su diámetro
- $longProducto_j$: es la longitud del producto j de acuerdo a los definido en *especProductos*.

Dado que *calcuLEMUs*, por fuera de la función de trozado emplea otras ecuaciones matemáticas para estimar el volumen bruto de el tronco entero, pueden entonces observarse diferencias entre estas y el procedimiento que aquí se describe. Por ejemplo podría suceder que la sumatoria de los volúmenes de los productos simulados más los desperdicios superen el valor *volBrInd*. Por ello se incorporó un paso de cálculo para conciliar estos valores donde la estimación final de el volumen de trozas se obtiene mediante la ecuación 4. De esta forma el volumen estimado mediante el trozado nunca supera el *volBrInd*. Para evaluar esta diferencia entre *volBrPerfil* y *volBrInd* se realizó una comparación gráfica empleando veintin árboles de *P. ponderosa* con Dap entre 25 mm a 630 mm y alturas totales entre 2 m y 34 m.

$$resumenVolProductos = (\sum volTroza_{ij} / volBrPerfil) * volBrInd \quad (4)$$

Donde:

- *resumenVolProductos*: es una matriz que almacena el resultado en volumen obtenido mediante la simulación.
- $\sum volTroza_{ij}$: es una matriz que almacena el volúmen de cada producto j simulado.
- *volBrInd*: como ya se definió.
- *volBrPerfil*: Es un escalar equivalente al volumen total del tronco incluyendo el tocón calculado como la sumatoria de 40 trozas de igual longitud, con diámetros estimados mediante *perfil* y cubicados empleando la ecuación 3.

	A	B	C	D	E	F	G	H	I
1	Número de Producto	1	2	3	4	5	6	7	
2	Diámetro en Punta Fina [mm]	400	270	150	140	NA	NA	NA	
3	Longitud de Troza [m]	3,15	3,15	2,55	2,2	NA	NA	NA	
4									
5									
6									
7									
8									
9									
10									

Fig 1. Definición de productos a obtener por simulación del trozado (matriz *especProductos*). Los datos se ingresan a través de planilla de cálculo, y se pueden definir hasta 7 productos (B1: H7).

3 Resultados y Discusión

La lógica de la simulación se describe de forma completa a través de el diagrama de la Figura 3. Este diagrama enumera los argumentos de entrada y salida y todas las decisiones y procesos internos que realiza la función de trozado (ecuación 1). El usuario no tiene posibilidad de observar su funcionamiento en tiempo de ejecución, solo puede ver la salida gráfica (Fig. 6) o tabulada (Tabla 3). Esta función de trozado es convocada por procedimiento de jerarquía superior, que calcula distintos valores para cada árbol de la base de datos. La función tiene como argumentos de entrada la matriz de especificación de productos (Figura 1), la ecuación del perfil del tronco propio de la especie (ecuación 2 y tabla 2), las dimensiones del árbol y los parámetros de salida gráfica.

En el diagrama de la Figura 3, la pregunta: ¿Se puede trozar un producto? se aborda a través de el código de programación de la Tabla 1. Este bucle controla el proceso de simulación por medio de una sentencia de el tipo “for - end”, probando de forma secuencial cada una de las definiciones de productos almacenadas en *especProductos*. Dentro del bucle hay otras dos sentencias bien diferenciadas, una primera de el tipo condicional “if - end” y una segunda del tipo bucle “while - end”. La primer sentencia tiene por objetivo establecer una serie de controles que evitan inconsistencias en la simulación. Como querer cortar el tronco a una altura superior a la altura de el árbol. Mientras que la segunda sentencia es la que estima las dimensiones de el tronco remanente y prueba si es posible trozar un producto, lo hace tantas veces como sea posible

hasta que el tronco remanente sea más pequeño que las especificaciones del producto simulado, momento en el cuál termina un ciclo del bucle “for – end”.

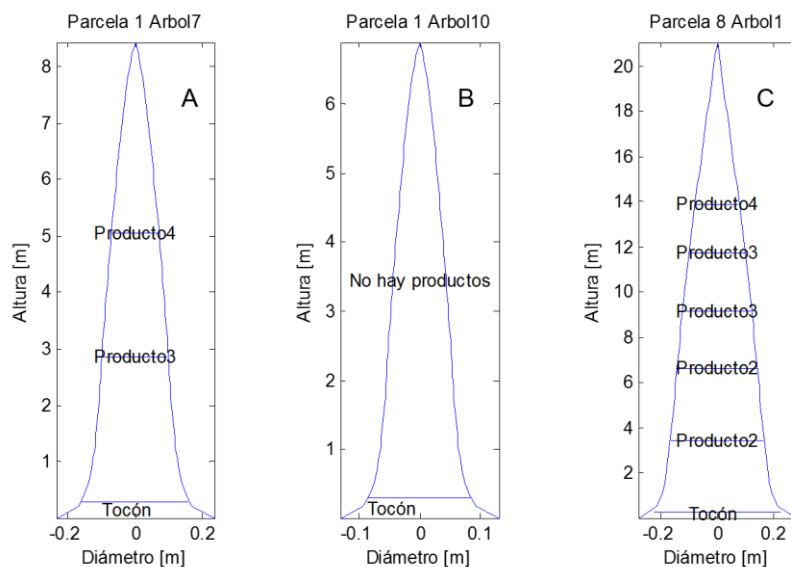


Fig. 2. Salida gráfica de la función de simulación de trozado para tres árboles de dos parcelas diferentes.

En el caso de que el usuario active la opción de visualizar el resultado, entonces el programa muestra en tiempo real de ejecución el gráfico a medida que avanza procesando cada árbol de la base de datos. Si la opción se desactiva, al finalizar el procesamiento de todos los árboles *calculEMUs* presenta la salida tabulada individual para cada árbol (Tabla 3) o un resumen de la parcela de inventario, de similar estructura a la Tabla 3. En la Figura 2 se muestra un ejemplo de salida gráfica que genera la función para tres árboles. El primer árbol (Fig 2A) produce una pieza o unidad de el producto 3 y otra de el producto 4. El segundo árbol es pequeño y no llega a brindar producto alguno (Fig. 2B). El tercer árbol de mayor tamaño produce dos piezas de el producto 2, dos del producto 3, y una del producto 4 (Fig. 2C). En los tres casos además hay un desperdicio de la punta que no se cuantifica.

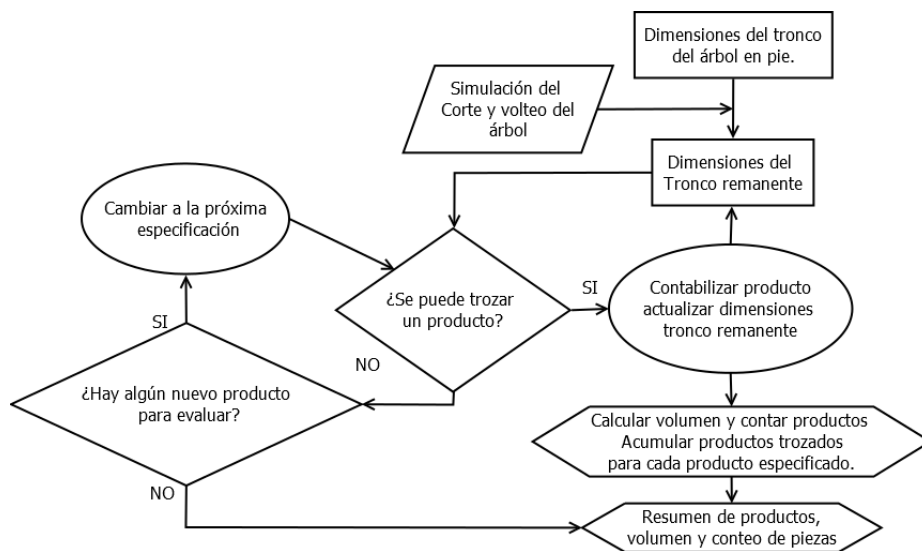
Tabla 3a. Matriz *resumenProductos* del ejemplo de la Figura 6.

<i>parcela</i>	<i>especie</i>	<i>arbol</i>	<i>Dap</i>	<i>Atotal</i>	<i>volBrInd</i> [m ³]
1	2	7	237	8,4	0,18
1	2	10	123	6,9	0,06
8	4	1	384	21,1	1,03

Tabla 3b. Matriz resumenProductos del ejemplo de la Figura 6.

parcela	arbol	Vol Prod. 1 [m ³]	Vol. Prod. 2 [m ³]	Vol. Prod. 3 [m ³]	Vol. Prod. 4 [m ³]	C. Prod 1	C. Prod 2	C. Prod 3	C. Prod 4
1	7	0	0	0,12	0,04	0	0	1	1
1	10	0	0	0	0	0	0	0	0
8	1	0	0,64	0,25	0,06	0	2	2	1

Donde: parcela: número de parcela a la que pertenece el árbol, especie: código de identificación de especie (2 *P. ponderosa*, 4 *P. contorta*), árbol: número de identificación del árbol dentro de la parcela, *Dap*, *A_{total}* y *volBrtInd*: como se definió antes, *Vol. Prod. 1* y *C. Prod 1* es la suma de los volúmenes y cantidades de productos tipo 1 y sucesivos simulados.

**Fig 3.** Diagrama de flujo de operaciones y decisiones del módulo de trozado.

La Figura 4 muestra la distribución de residuales producto de los ajustes estadísticos de la ecuación 2, para *P. ponderosa* (Fig. 4A), *P. contorta* (Fig. 4B) y *P. mensiezi* (Fig. 4C) los parámetros resultantes se presentan en la tabla 2. En dichas figuras se observa una adecuada distribución de residuales y otras características (como a valores predichos 1 y 0 le corresponden valores estandarizados de residuales igual a cero), que denotan que el modelo propuesto para describir la forma del tronco es preciso e insesgado.

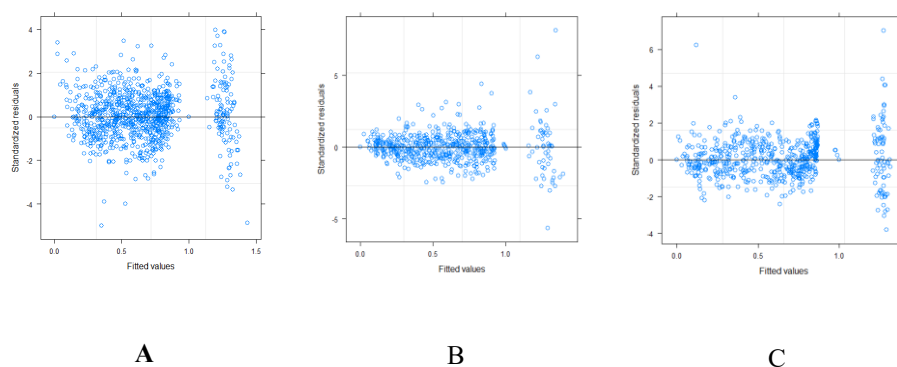


Fig. 4. Distribución de residuales estandarizados para los valores predichos por el modelo de la ecuación 2 de perfil de fuste de (A) *Pinus ponderosa*, (B) *Pinus contorta* y (C) *Pseudotsuga mensiezii*.

Tabla 2. Parámetros del modelo de la ecuación 2 por especie.

Parámetro	<i>P. ponderosa</i>	<i>P. mensiezii</i>	<i>P. contorta</i>
p1	2.2429434	3.4410959	1.7103949
p2	-0.8820692	-1.2375913	-0.2880448
p3	-0.1412407	-0.2240591	-0.0564792
p4	-1.0232666	-1.7221383	-0.8862977
p5	-0.0001415	-0.0000651	0.0002081
p6	0.0074955	0.0264240	0.0155042
p7	-0.0251282	-0.1117471	-0.1046544

La comparación del volumen del tronco estimado (Figura 5) mediante *volBrtrPerfil* (ecuación 2) y *volBrtrInd*, muestra grandes diferencias en árboles pequeños mientras que para árboles de mediano a gran tamaño ($> 0,5 \text{ m}^3$), las diferencias se estabilizan, con una sobreestimación promedio del 6% por parte de *volBrtrPerfil*. Los datos empleados para los ajustes de los parámetros de *volBrtrInd* y los de la ecuación 2 se realizaron con la misma base de datos para las tres especies. Por ello la diferencia es estrictamente metodológica en relación a la forma de calcular el volumen de los tocones. En el caso de *volBrtrInd* la cubicación de el tocón se realizó considerando que tiene forma de cilindro [6, 7, 8]. Como se explicó con anterioridad para *volBrtrPerfil* el tocón se asume con forma neiloide. Por ello la estabilización de la diferencia entre estas dos formas de estimar el volumen ocurre por la proporción que representa el volumen de el tocón respecto de todo el tronco. Es decir que la diferencias observadas se producen a nivel de el tocón, y por ello no afecta las estimaciones de los productos. Así la ecuación 4 permite que el módulo de simulación de trozado mantenga consistencia de resultados con los otros procedimientos que operan en *calcuLEMUs*.

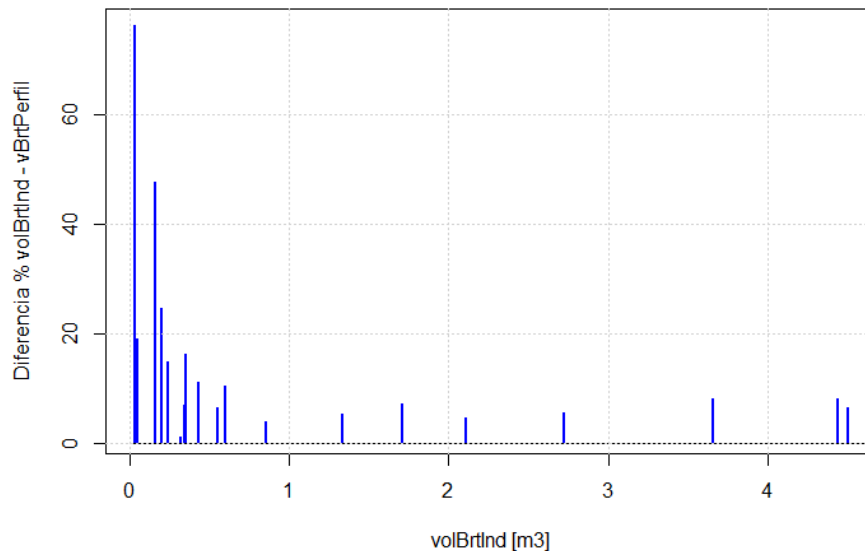


Fig. 5. Diferencia porcentual absoluta entre estimaciones de *volPerfilBrt* y *volBrtInd*.

El módulo de trozado de *calcuLEMUs* prioriza la obtención de los productos de mayor tamaño. Es decir que no realiza un proceso de optimización combinatoria del tipo “Knapsack” [14 y 15]. La optimización se realiza con el objeto de maximizar el valor de trozas producido por un árbol, y suele abordarse mediante programación dinámica [16 y 17]. Sin embargo, dado que aquí la descripción de los productos considera trozas de longitudes fijas e igual valor comercial, entonces los resultados que provee el módulo se aproxima a los obtenidos mediante un algoritmo de optimización [18].

4 Conclusiones

A futuro es necesario ampliar la descripción de los productos, donde se incluya distintas calidades de el tronco de el árbol (por podas o defectos que requieran un saneamiento o potencialidad de uso) o rangos de dimensiones de productos, con distintos valores comerciales. En ese caso será imprescindible incorporar un algoritmo de optimización. El diseño de programación modular permite sumar un cálculo muy complejo al código pre-existente de forma sencilla. En el procedimiento de jerarquía superior solo requirió incorporar pocas líneas de programación, para convocar la función y almacenar su resultado. Programar en módulos permitió trabajar en el dise-

ño, prueba e implementación de forma independiente al resto de el programa, resultando que el proceso de depuración de errores fuese ágil, dinámico e interactivo.

Dado estos resultados se considera que el diseño modular de el software es la forma de continuar incorporando funcionalidades a *calcuLEMUs*. Este diseño de programación redundará en que la inclusión de nuevas funcionalidades, como por ejemplo el algoritmo de optimización, se pueda realizar a bajo costo y con una alta confiabilidad en los resultados.

Referencias

1. Parnas D.L. 1972. On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the Association for Computing Machinery. Vol 15 (12). Pp 1053 – 1058.
2. Basil G., Letourneau F.J., Antoci V., Ancalao M., Cobelo C., Rusch V., Ayesa J., Umaña F., Sarasola M., Cuevas J., De Agostini N., Claps L., Andenmatten E. 2013. Reserva Forestal “Loma del Medio – Río Azul”. Plan de Uso Múltiple. Pp. 189. Instituto Nacional de Tecnología Agropecuaria.
3. Bava J. Benitez R. 2016. Inventario Nacional de Plantaciones en Secano de Patagonia. Eds. CIEFAP, UCAR, Ministerio de Agroindustria, Presidencia de la Nación. Pp. 176.
4. Avery T.E., Burkhart H.E. 1994. Forest Measurements. Eds McGraw – Hill. Pp 408.
5. MATLAB version 7.10.0, 8.3.0.532 (R2014a). Natick, Massachusetts: The Math-Works Inc., 2014.
6. Andenmatten, E.; F. Letourneau. 1995. “Pino Ponderosa (*Pinus ponderosa* (Dougl.)Laws. Tabla de volumen estándar de aplicación en la región Andina de Río Negro y Chubut”. IV Jornadas Forestales Andino Patagónicas. M. I. Zingoni editora. San Martín de los Andes, Neuquén, Argentina. Pág. 266-271.
7. Rey, M.; Andenmatten, E.; F. Letourneau. 1995: “Tarifa de volumen para Pino Oregon (*Pseudotsuga menziesii* (Mirb) Franco) en la región Andina de las Provincias de Río Negro y Chubut”. IV Jornadas Forestales Andino Patagónicas. M. I. Zingoni editora. San Martín de los Andes, Neuquén, Argentina. Pág. 306-311.
8. Letourneau, F.J. Andenmatten E., De Agostini N., 2011. Estimación del volumen de fuste para Pino Murrayana (*Pinus contorta*), en la región Andino Patagónica de Argentina. Comunicación técnica Nro 38 Silvicultura. Pp 1 – 5. Instituto Nacional de Tecnología Agropecuaria. EEA Bariloche “Dr. Grenville Morris”, San Carlos de Bariloche, Argentina.
9. Bi H. (2000). Trigonometric Variable – Form Taper Equations for Australian Eucalypts. Forest Science, Vol. 46, No. 3. Pp 397 – 409.
10. Pinheiro J, Bates D, DebRoy S, Sarkar D and R Core Team (2017). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-131, <https://CRAN.R-project.org/package=nlme>.
11. R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
12. RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL <http://www.rstudio.com/>. Version 1.0.136.

13. Letourneau F.J. 2017. Proyecto Nacional Forestal 1104074. Base de Datos del Proyecto. Desarrollo de Modelos de Crecimiento y Producción Forestal, Caracterización de la Calidad de los Productos Forestales y Economía Forestal. Informe Interno.
14. Martello S., Toth P. Knapsack Problems. Algorithms and Computer Implementations, 1990. John Wiley & Sons, England, Pp 306.
15. Pisinger D., 1995. Algorithm for Knapsack Problems. Ph.D. Thesis. Dept of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Denmark. Pp 200.
16. Laclau P. 1995. Optimización del trozado de rollizos en el monte. Tomo 2. IV Jornadas Forestales Andino Patagónicas. M. I. Zingoni editora. San Martín de los Andes, Neuquén, Argentina. Pp 355 – 372.
17. Yapura P., Sañudo G., Fassola H.E., Crechi E., Keller A.E. Optimización Del Trozado De Fustes De Pinus Taeda, "Pino Híbrido" (Pinus Elliottii X Pinus Caribaea F2) Y Eucalyptus Grandis, En Las Provincias De Misiones Y Noreste De Corrientes, PIA 10107 En: Investigación forestal 2011-2015: los proyectos de investigación aplicada. 1a ed. Ministerio de Agroindustria. Unidad para el cambio rural -UCAR. Libro digital. ISBN 978-987-1873-41-8. Pp. 371 - 374.
18. Herrera, J.F., Barrios, A., López, A.M. & Nieto, V. (2015). Generación de patrones de corte óptimos para árboles individuales a partir de productos demandados en plantaciones comerciales. Colombia Forestal, 18(2), Pp 193-206.