- **Contexto Pregunta**: Hace aproximadamente 2000 años Herón propuso la siguiente fórmula para el cálculo del área de un triángulo con longitud de lados igual a a, b y c, respectivamente, ver figura 1,

$$\mathcal{A} = \sqrt{s\,(s-a)\,(s-b)\,(s-c)}, \quad s = \frac{a+b+c}{2}.$$

La fórmula resulta muy útil para triángulos que tiene tamaños de lados similares, sin embargo produce resultados incorrectos cuando un lado es mucho más pequeño que los otros en aritmética de punto flotante, por ejemplo en *double precision*. Para el contexto de esta pregunta, considere el triángulo con vértices A, B y C definido en la figura 1,

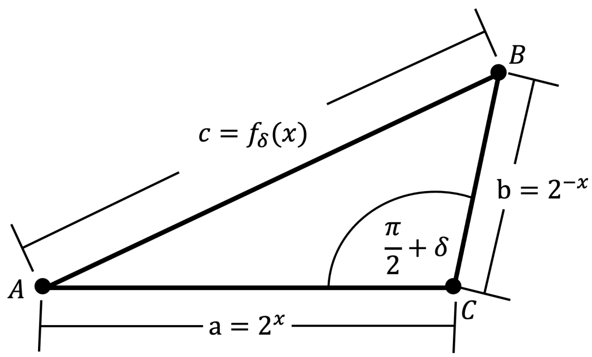

Figura 1: Diagrama de triangulo obtuso definido por los vertices A, B, y C.

Notar que la longitud de los lados del triángulo se definen de la siguiente forma,

$$\overline{AC} = a = 2^x, \tag{1}$$
$$\overline{CB} = b = 2^{-x}, \tag{2}$$
$$\overline{AB} = c = f_\delta(x) = \sqrt{2^{2\,x} + 2^{-2\,x} - 2\,\cos\left(\frac{\pi}{2}+\delta\right)} = 2^x\,\sqrt{1+2^{-4\,x}+\sin(\delta)\;2^{-2\,x+1}} = 2^x\,\sqrt{1+g_\delta(x)}, \tag{3}$$

donde $x \geq 0$ y $g_\delta(x) = 2^{-4\,x} + \sin(\delta)\;2^{-2\,x+1}$. Entonces, para obtener el área $\mathcal{A}$ se deben computar los siguientes términos:

$$s = \frac{a+b+c}{2} = \frac{2^x + 2^{-x} + 2^x\,\sqrt{1+g_\delta(x)}}{2} = \frac{2^x\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}, \tag{4}$$
$$s-a = s-2^x = \left(\frac{2^x\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right) - 2^x, \tag{5}$$
$$s-b = s-2^{-x} = \left(\frac{2^x\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right) - 2^{-x}, \tag{6}$$
$$s-c = s-2^x\,\sqrt{1+g_\delta(x)} = \left(\frac{2^x\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right) - 2^x\,\sqrt{1+g_\delta(x)}. \tag{7}$$

Particularmente en este caso se puede obtener que $\mathcal{A} = \frac{|\cos(\delta)|}{2}$, sin embargo lo interesante ocurre en los pasos intermedios de la computación. Más aún, dado que se conoce el resultado final, que es independiente del valor de la variable x, se puede utilizar este problema como un *test* para estudiar si los pasos intermedios se han obtenido adecuadamente.

Tabla de Derivadas

La siguiente lista de derivadas considera la siguiente notación: $x \in \mathbb{R}$, $u := u(x)$, $v := v(x)$, $c \in \mathbb{R}$ y es una constante, $a \in \mathbb{R}^+$ y es una constante, $u' = \frac{\mathrm{d}}{\mathrm{d}x} u(x)$, y $v' = \frac{\mathrm{d}}{\mathrm{d}x} v(x)$.

Fuente parcial: https://tutormemath.net/assets/derivative_integrals.pdf.

$$\frac{\mathrm{d}}{\mathrm{d}x}[c\,u] = c\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[u \pm v] = u' \pm v',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[u\,v] = u\,v' + v\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{u}{v}\right] = \frac{v\,u' - u\,v'}{v^2},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[c] = 0,$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[u^n] = n\,u^{n-1}\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[x] = 1,$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[|u|] = \frac{u}{|u|}(u'), \quad u \neq 0,$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\log(u)] = \frac{u'}{u},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\exp(u)] = \exp(u)\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\log_a(u)] = \frac{u'}{\log(a)\,u},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[a^u] = \log(a)\,a^u\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\sin(u)] = (\cos(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\cos(u)] = -(\sin(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\tan(u)] = (\sec^2(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\cot(u)] = -(\csc^2(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\sec(u)] = (\sec(u)\;\tan(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\csc(u)] = -(\csc(u)\;\cot(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\arcsin(u)] = \frac{u'}{\sqrt{1-u^2}},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\arccos(u)] = -\frac{u'}{\sqrt{1-u^2}},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\arctan(u)] = \frac{u'}{1+u^2},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\sinh(u)] = (\cosh(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\cosh(u)] = (\sinh(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\tanh(u)] = (\mathrm{sech}^2(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\coth(u)] = -(\mathrm{csch}^2(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\mathrm{sech}(u)] = -(\mathrm{sech}(u)\;\tanh(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[\mathrm{csch}(u)] = -(\mathrm{csch}(u)\;\coth(u))\,u',$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\sinh^{-1}(u)\right] = \frac{u'}{\sqrt{u^2+1}},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\cosh^{-1}(u)\right] = \frac{u'}{\sqrt{u^2-1}},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\tanh^{-1}(u)\right] = \frac{u'}{1-u^2},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\coth^{-1}(u)\right] = \frac{u'}{1-u^2},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\mathrm{sech}^{-1}(u)\right] = \frac{-u'}{u\sqrt{1-u^2}},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\mathrm{csch}^{-1}(u)\right] = \frac{-u'}{|u|\sqrt{1+u^2}},$$

$$\frac{\mathrm{d}}{\mathrm{d}x}u(v(x)) = u'(v)\,v'.$$

- **Desarrollo Pregunta 1**:

(a) En esta pregunta se estudiará la evaluación de la ecuación (7). Por ejemplo, considere la siguiente implementación para la obtención del área $\mathcal{A}$, donde se considera que `a`, `b` y `c` se obtuvieron de la evaluación de las ecuaciones (1) a (3), respectivamente, para algún valor de `x` definido como input,

```
1: s  = (a+b+c)/2.
2: s1 = s-a
3: s2 = s-b
4: s3 = s-c
5: A  = np.sqrt(s*s1*s2*s3)
```

Notar que los números al lado izquierdo de cada línea de código representa el número de línea respectivo, por ejemplo "`4: s3 = s-c`", corresponde a la línea 4 del código donde se ejecuta la computación de "`s-c`" y se asigna a la variable "`s3`".

(I) Por simplicidad de computación considere que $\delta = 0$ y $x \in \mathbb{N}$ solamente para la pregunta a continuación: ¿Cuál es el menor valor de $x > 0$ tal que la expresión "`s3`" (línea 4 del código anterior) es exactamente igual a 0.0 en *double precision*? Por completitud, se incluye nuevamente acá la ecuación (7) considerando en este caso $\delta = 0$,

$$s - c = \left(\frac{2^x \left(1 + \sqrt{1 + g_0(x)}\right) + 2^{-x}}{2} \right) - 2^x \sqrt{1 + g_0(x)} \quad \text{donde} \quad g_0(x) = 2^{-4\,x}.$$

- Notar que $g_0(x) = 2^{-4\,x}$, por lo que sabemos que $g_0(x)$ tiende a 0 a medida que x aumenta.
- En este caso sabemos que $x \in \mathbb{N}$, lo que implica que debemos estudiar solo los números naturales.
- La primera observación que uno puede destacar es que a medida que x aumenta, el factor 2^x aumenta y el factor 2^{-x} disminuye, lo que hace que el valor de `s` se acerque al valor de `c`.
- El primer valor crítico que se puede observar es cuando el factor $1 + g_0(x)$ se hace 0 en *double precision*, lo que significa que $g_0(x) = 2^{-4\,x} = 2^{-53}$. Lo cual se reduce a $x = \left\lceil \frac{53}{4} \right\rceil = 14$. Lo que significa que para $x \geq 14$, el valor de $1 + g_0(x) = 1$ en *double precision*. Esto implica la siguiente simplificación de la expresión anterior,

$$\begin{aligned} s - c &= \left(\frac{2^x \left(1 + \sqrt{1}\right) + 2^{-x}}{2} \right) - 2^x \sqrt{1} \\ &= \left(\frac{2^x\, 2 + 2^{-x}}{2} \right) - 2^x \\ &= \left(\frac{2^{x+1} + 2^{-x}}{2} \right) - 2^x \end{aligned}$$

- Ahora debemos evaluar la expresión $2^{x+1} + 2^{-x}$, considerando que $x \geq 14$. En este caso observamos que ocurrirá pérdida de importancia cuando exista una diferencia de magnitud entre los término mayor a 2^{52}. En particular uno podría factorizar por 2^{x+1} y se obtiene,

$$2^{x+1} + 2^{-x} = 2^{x+1} \left(1 + 2^{-2\,x-1}\right).$$

  Entonces, si $2^{-2\,x-1} = 2^{-53}$ se exhibe perdida de importancia. Despejando x se obtiene que $x = 26$.
- Por lo tanto, para $x = 26$ se obtiene la siguiente evaluación de *double precision*,

$$\begin{aligned} s - c &= \left(\frac{2^{26+1} + 2^{-26}}{2} \right) - 2^{26} \\ &= \left(\frac{2^{26+1}}{2} \right) - 2^{26} \\ &= \left(2^{26}\right) - 2^{26} \\ &= 0. \end{aligned}$$

- Finalmente la respuesta es $x = 26$.

(II) Proponga una forma alternativa de evaluar la ecuación (7) tal que permita utilizar valores de x mayores al valor que obtuvo en la pregunta anterior. Por completitud, se incluye nuevamente acá la ecuación (7), ahora en su forma original,

$$s-c=\left(\frac{2^{x}\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right)-2^{x}\,\sqrt{1+g_\delta(x)}\quad\text{donde}\quad g_\delta(x)=2^{-4\,x}+\sin(\delta)\;2^{-2\,x+1}.$$

- En este caso se considera la expresión completa. Notar que se incluyen todos los pasos para claridad del desarrollo.

$$\begin{aligned}
s-c&=\left(\frac{2^{x}\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right)-2^{x}\,\sqrt{1+g_\delta(x)}\\
&=\left(\frac{2^{x}\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right)-\frac{2\,2^{x}\,\sqrt{1+g_\delta(x)}}{2}\\
&=\left(\frac{2^{x}\left(1+\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\right)-\frac{2\,2^{x}\,\sqrt{1+g_\delta(x)}}{2}\\
&=\frac{2^{x}\left(1-\sqrt{1+g_\delta(x)}\right)+2^{-x}}{2}\\
&=2^{x-1}\left(1-\sqrt{1+g_\delta(x)}\right)+2^{-x-1}\\
&=2^{x-1}\left(1-\sqrt{1+g_\delta(x)}\right)\left(\frac{1+\sqrt{1+g_\delta(x)}}{1+\sqrt{1+g_\delta(x)}}\right)+2^{-x-1}\\
&=2^{x-1}\left(\frac{\left(1-\sqrt{1+g_\delta(x)}\right)\left(1+\sqrt{1+g_\delta(x)}\right)}{1+\sqrt{1+g_\delta(x)}}\right)+2^{-x-1}\\
&=2^{x-1}\left(\frac{-g_\delta(x)}{1+\sqrt{1+g_\delta(x)}}\right)+2^{-x-1}\\
&=-2^{x-1}\left(\frac{2^{-4\,x}+\sin(\delta)\;2^{-2\,x+1}}{1+\sqrt{1+g_\delta(x)}}\right)+2^{-x-1}\\
&=-\frac{2^{-3\,x-1}+\sin(\delta)\;2^{-x}}{1+\sqrt{1+g_\delta(x)}}+2^{-x-1}
\end{aligned}$$

- Por lo tanto la expresión propuesta corresponde a,

$$\begin{aligned}
s-c&=-\frac{2^{-3\,x-1}+\sin(\delta)\;2^{-x}}{1+\sqrt{1+g_\delta(x)}}+2^{-x-1}\\
&=-\left(\frac{\left(2^{-3\,x-1}+\sin(\delta)\;2^{-x}\right)}{1+\sqrt{1+g_\delta(x)}}\right)+2^{-x-1}.
\end{aligned}$$

  Se incluyen paréntesis para notar el orden de evaluación.

(b) Implemente en Python utilizando adecuadamente la librería NumPy (en especial su capacidad de vectorización) la forma adecuada de obtener en *double precision* las expresiones `s`,`s1=s-a`, `s2=s-b`, y `s3=s-c`. Para su implementación, considere que **solo** tiene a su disposición las siguientes funciones de la librería NumPy, además de las operaciones elementales, ciclos y condicionales propios de Python:

- `np.arange(n)`: Para `n` un número entero positivo entrega un vector de largo `n` con números enteros desde `0` a `n-1`.
- `np.abs(x)`: Entrega el valor absoluto de `x`.
- `np.sqrt(x)`: Entrega la evaluación de la raíz cuadrada no negativa de un vector o escalar `x`.
- `np.cos(x)`: Entrega la evaluación de la función coseno de los valores en radianes en el vector o escalar `x`.
- `np.sin(x)`: Entrega la evaluación de la función seno de los valores en radianes en el vector o escalar `x`.
- `np.log(x)`: Entrega la evaluación de la función logaritmo natural de los valores en el vector o escalar `x`.
- `np.power(x,y)`: Evalúa la expresión x^y si `x` e `y` son escalares. En caso de que `x` e `y` sean vectores, deben tener la misma dimensión y entrega la evaluación elemento a elemento. Si solo uno de los términos es un vector, entrega el vector donde el término constante se consideró para cada término de vector.
- `Newton1D(f,fp,x0)`: Implementa el método de Newton en 1D que entrega la aproximación de la raíz de `f`. Esta función recibe como parámetros la función `f`, la derivada `fp` de `f` y el *initial guess* `x0`. Por simplicidad se omite criterio de detención.

Notar que al momento de implementar usted debe decidir qué componentes se deben vectorizar y qué componentes no, considerando las funciones de NumPy antes mencionadas. ***Adicionalmente se recomienda evaluar si alguna otra expresión requiere algún tipo de corrección antes de implementarla***.

Considere la siguiente firma:

```
'''
input:
x       : (float) Input value 'x'.
delta   : (float) Input value 'delta'.

output:
s   : (float) Semi-perimeter s
s1  : (float) s-a
s2  : (float) s-b
s3  : (float) s-c
A   : (float) Area
'''
def compute_area_improved(x,delta):
```

- Se define $g_\delta(x)$ como `gd` y $\sqrt{1+g_\delta(x)}$ como `w` para utilizar en la computación posterior.
```
gd = lambda x: np.power(2.,-4*x)+np.sin(delta)*np.power(2.,-2*x+1)
w = lambda x: np.sqrt(1+gd(x))
```
- Se obtiene `s`:
```
s = ((np.power(2.,-x)+np.power(2.,x))+np.power(2.,x)*w(x))/2.
```
- Se obtiene `s1` pero corregido, ya que también sufre de errores de cancelación.
```
s1 = (np.power(2.,x-1)*gd(x))/(1+w(x))+np.power(2.,x-1)
```
- Se obtiene `s2`.
```
s2 = (np.power(2.,x)*(1+w(x))+np.power(2.,-x))/2.-np.power(2.,-x)
```
- Se obtiene `s3` desarrollado en la pregunta anterior.
```
s3 = -((np.power(2.,-3*x-1)+np.sin(delta)*np.power(2.,-x))/(1+w(x)))+np.power(2.,-x-1)
```
- Se obtiene `A`.
```
A = np.sqrt(s*s1*s2*s2)
```
- Se retorna lo obtenido.
```
return s, s1, s2, s3, A
```