

# Física Numérica

## Tarea #2

D. A. Vázquez Gutiérrez

*Escuela Superior de Física y Matemáticas, Instituto Politécnico Nacional, Unidad Profesional "Adolfo López Mateos", Zacatenco, Edificio 9, Col. San Pedro Zacatenco, C.P. 07730 del. Gustavo A. Madero, Ciudad de México, México*

*email: dvazquezg1600@alumno.ipn.mx*

26 de marzo de 2024

### 1. Cancelación Sustractiva

Las distintas formas en que podemos representar matemáticamente una expresión pueden tener un gran impacto en la estabilidad numérica y la precisión de los cálculos, especialmente cuando trabajamos con números cercanos entre sí o cuando nos acercamos a ciertos límites, como lo es el caso del cero.

Una de las razones por las cuales ciertas expresiones pueden funcionar mejor que otras numéricamente es la cancelación sustractiva.

Recordando que en general el número más pequeño que la computadora puede almacenar es mucho menor que la precisión de máquina ya que como recordemos de los resultados nuestra tarea anterior, nuestro límite de *underflow* calculado es:

$$l_{ufw} = 4,940656 \times 10^{-324} \quad (1)$$

Mientras que nuestra precisión de máquina es:

$$\epsilon_m = 1,11023 \times 10^{-16} \quad (2)$$

Por lo tanto la cancelación numérica puede ocurrir cuando restamos cantidades muy cercanas entre sí, siendo estas irreconocibles para la máquina. Al reescribir una expresión de una manera que evite esta resta directa, como usar identidades trigonométricas o factorización, podemos evi-

tar la cancelación numérica y obtener resultados más precisos.

Ahora mostraremos como reescribir algunas expresiones para evitar la cancelación para los argumentos indicados.

$$(a) \sqrt{x+1} - 1, x \approx 0$$

Entonces comenzamos multiplicando ambas partes por un uno en forma de  $\sqrt{x+1} + 1$  sobre si mismo.

$$\begin{aligned} (\sqrt{x+1} - 1) \cdot 1 &= (\sqrt{x+1} - 1) \cdot \frac{\sqrt{x+1} + 1}{\sqrt{x+1} + 1} \\ &= \frac{(x+1) - 1}{\sqrt{x+1} + 1} \\ &= \frac{x}{\sqrt{x+1} + 1} \end{aligned}$$

De tal forma que :

$$\sqrt{x+1} - 1 = \frac{x}{\sqrt{x+1} + 1} \quad (3)$$

Donde por mas que  $x \approx 0$ , mientras (3) no llegue a *underflow* no habrá ceros inesperados al evitar sustracciones.

$$(b) \operatorname{sen} x - \operatorname{sen} y, x \approx y$$

Partimos recordando lo siguiente :

$$\sin(a \pm b) = \sin(a)\cos(b) \pm \cos(a)\sin(b)$$

$$\Rightarrow \sin(a+b) - \sin(a-b) = 2\cos(a)\sin(b)$$

Entonces resolviendo  $a+b = x$  y  $a-b = y$  obtenemos:

$$\sin x - \sin y = 2\cos\left(\frac{x+y}{2}\right)\sin\left(\frac{x-y}{2}\right) \quad (4)$$

Donde aun teniendo una sustracción en el termino del seno , y suponiendo que  $x$  y  $y$  sean menores que la precision de maquina, este tomaría el valor de  $x$  que aun que muy pequeño , es distinto de cero , evitándonos la cancelación.

$$(c) \quad x^2 - y^2, x \approx y$$

Notemos que podemos factorizar esta expresion , obteniendo :

$$x^2 - y^2 = (x+y)(x-y) \quad (5)$$

Una pregunta que nos podríamos hacer es , ¿Por que esta forma es efectiva si aun tenemos una raíz con sustracción  $x-y$  ? Esto es debido a que aun se conserva la diferencia entre ellos , y aun que puede haber perdida de precision en la diferencia , esta es menor que si se representara directamente como  $x^2 - y^2$ .

$$(d) \quad \frac{1-\cos x}{\sin x}, x \approx 0$$

Recordando lo siguiente :

$$1 - \cos x = 2\sin^2\left(\frac{x}{2}\right)$$

$$\begin{aligned} \Rightarrow \frac{1 - \cos x}{\sin x} &= \frac{2\sin^2\left(\frac{x}{2}\right)}{\sin x} \\ &= 2\sin\left(\frac{x}{2}\right) \cdot \frac{\sin\left(\frac{x}{2}\right)}{\sin x} \end{aligned}$$

Sin embargo también tenemos la siguiente propiedad trigonométrica:

$$\sin(a) = 2\sin\left(\frac{a}{2}\right)\cos\left(\frac{a}{2}\right)$$

Por lo tanto

$$\frac{1 - \cos x}{\sin x} = 2 \frac{\sin\left(\frac{x}{2}\right)}{\cos\left(\frac{x}{2}\right)} \quad (6)$$

En este caso nos deshacemos de todo tipo de sustracción, y mientras  $x/2$  no provoque un *underflow* , aun para cantidades muy pequeñas , la forma de la ecuación debe funcionar perfectamente.

$$(e) \quad c = (a^2 + b^2 - 2a \cdot b \cdot \cos\theta)^{1/2}, a \approx b, |\theta| \ll 1$$

Intentemos hacer una expansión de Taylor de la función coseno alrededor de  $\theta = 0$  :

$$\cos\theta \approx \cos(0) - \sin(0)\theta - \frac{\cos''(0)}{2!}\theta^2 + \frac{\cos'''(0)}{3!}\theta^3 - \dots$$

Dado que el  $\theta$  es muy pequeño , entonces, para potencias mayores que 2 , las consideramos cero, esto nos da como resultado .

$$\cos\theta \approx 1 - \frac{\theta^2}{2!}$$

Sustituyamos esta ecuación en la ecuación principal y nos da

$$\begin{aligned} c &= (a^2 + b^2 - 2a \cdot b \cdot \cos\theta)^{1/2} \\ &\approx (a^2 + b^2 - 2a \cdot b \cdot (1 - \frac{\theta^2}{2!}))^{1/2} \\ &= (a^2 + b^2 - 2ab + ab\theta^2)^{1/2} \\ &= ((a-b)^2 + ab\theta^2)^{1/2} \end{aligned}$$

Por lo tanto

$$(a^2 + b^2 - 2a \cdot b \cdot \cos\theta)^{1/2} = ((a-b)^2 + ab\theta^2)^{1/2} \quad (7)$$

Aquí podemos notar que aunque  $\theta$  sea muy pequeña, no implica que se cancele toda la expresión, por otra parte  $a - b$  da un valor distinto de cero, o por lo menos con una precisión mayor que la expresión anterior, perdiendo así la posibilidad de cancelación por sustracción.

## 2. Funciones de Bessel esféricas

### 2.1. Up & Down

Nos piden escribir un programa de tal forma que utilizando las formulas de recursión hacia arriba (*up*) y hacia abajo (*down*) para calcular  $j_l(x)$  para los primeros 25 valores de  $l$  para  $x = 0, 1, 1, 10$ , donde sabemos que estas relaciones de recurrencia son :

$$j_{l+1}(x) = \frac{2l+1}{x} j_l(x) - j_{l-1}(x), up$$

$$j_{l-1}(x) = \frac{2l+1}{x} j_l(x) - j_{l+1}(x), down$$

Sin embargo haremos una modificación a estas, en primera instancia, dado que nos interesan sobre todo los resultados, pondremos la función *up* y *down* en función de un  $r$  final, esto es:

$$j_r(x) = \frac{2r-1}{x} j_{r-1}(x) - j_{r-2}(x), up$$

$$j_r(x) = \frac{2l+3}{x} j_{r+1}(x) - j_{r+2}(x), down$$

En el caso de la función recursiva *up*, basta con conocer el valor de  $j_0$  y de  $j_1$ , estos, por nuestras clases de métodos matemáticos sabemos que son:

$$j_0(x) = \frac{\sin x}{x}$$

$$j_1(x) = \frac{\sin x}{x^2} - \frac{\cos x}{x}$$

A partir de estas deberíamos de ser capaces de calcular hacia arriba cualquier otra función esférica de Bessel

Por otro lado, hay una modificación importante que hacerle a la función recursiva *down*, ya que partimos de un valor que desconocemos  $j_L^N$ , el cual al suponerlo, nos llevara a resultados aceptables, pero con un cierto error.

Esto lo podemos corregir apartir del algoritmo de Miller, sea  $j_l^C(x)$  el valor numérico calculado como aproximación de  $j_l(x)$ ; sabemos que  $j_0^C(x) \neq j_0(x)$ , apartir de esta premisa, utilizamos el mecanismo de Miller

$$j_l^N(x) = j_l^C(x) \times \frac{j_0(x)}{j_0^C(x)}$$

De tal forma, conseguimos el siguiente programa :

```

1 import numpy as np
2
3 def bessell_up(l, x):
4     if l == 0:
5         return np.sin(x) / x
6     elif l == 1:
7         return (np.sin(x) / x**2) - (np.
8             cos(x) / x)
9     else:
10         return ((2*l - 1) / x) * bessell_up(
11             l-1, x) - bessell_up(l-2, x)
12
13 """
14 Se anade el ajuste para el up
15 """
16 def bessell_down_a(l, x):
17     if l == 24:
18         return 0
19     elif l == 23:
20         return 1
21     else:
22         return ((2*l + 3) / x) *
23             bessell_down_a(l+1, x) - bessell_down_a(
24                 l+2, x)
25
26 """
27 Primeraparte de algoritmo de miller
28 conseguimos la version con errores
29 para despues conseguir una normalizacion
30 respecto a la base jC0
31 """
32 def bessell_down(l, x):
33     if l == 24:
34         return 0*((np.sin(x) / x)/
35             bessell_down_a(0,x))
36     elif l == 23:
37         return 1*((np.sin(x) / x)/
38             bessell_down_a(0,x))
39     else:

```

```

30         return (((2*l +3) / x) *
        bessell_down_a(l+1, x) - bessell_down_a(
        l+2, x))*((np.sin(x) / x)/
        bessell_down_a(0,x))

31 """
32 Se anade el ajuste para el down
33 """
34
35 x_values = [0.1, 1, 10]
36 for x in x_values:
37     print("Para x = {x}:")
38     for l in range(25):
39         j_l_up = bessell_up(l, x)
40         j_l_down = bessell_down(l, x)
41         Error_l= abs(j_l_up-j_l_down)/(abs
        (j_l_up)+abs(j_l_down))
42         print(f"J_{l}({x}) (Up): {j_l_up},
        J_{l}({x}) (Down): {j_l_down} ,
        Error_{l} : {Error_l}")
43 #Escribe y compara los mismos balores de
        la ecuacion esferica de bessell up y
        down

```

**Código 1:** Programa que obtiene la funcion mediante la funcion de recurrencia up y down . NOMBRE : "Bessell.py"

Entonces , como vemos en las figuras 1 - 3, notamos que la función Up y Down, tomamos el primer y segundo valor como 0 y 1 respectivamente, al compararlasy,tiene un buen comportamiento para  $l$  pequeño, así como para valores de  $x$  altos , mientras que al crecer , las cancelaciones sustractivas perjudican cada vez mas los resultados .

Para x = 0.1:

j_0(0.1) (Up): 0.9883341664682815, j_0(0.1) (Down): 0.9883341664682815, Error_0 :0.0
j_1(0.1) (Up): 0.0330801190255594, j_1(0.1) (Down): 0.0330801190255594, Error_1 :2.4484076344924058e-14
j_2(0.1) (Up): 0.000664396081396563, j_2(0.1) (Down): 0.000664396081396563, Error_2 :1.0727811440384734e-11
j_3(0.1) (Up): 9.51851722737736e-06, j_3(0.1) (Down): 9.51851722808557e-06, Error_3 :1.286170654949181e-07
j_4(0.1) (Up): 1.056066987518434e-07, j_4(0.1) (Down): 1.057201582098734e-07, Error_4 :1.000810630582598539e-08
j_5(0.1) (Up): -1.45509318244035e-08, j_5(0.1) (Down): 9.63135222516448e-09, Error_5 :1.1e-09
j_6(0.1) (Up): 1.69586887082126e-06, j_6(0.1) (Down): 7.3973418935771e-12, Error_6 :1.1e-09
j_7(0.1) (Up): -0.0002044649572666014, j_7(0.1) (Down): 4.931887475731746e-14, Error_7 :1.1e-09
j_8(0.1) (Up): -0.03386557549813292, j_8(0.1) (Down): 2.9812081925381987e-16, Error_8 :1.1e-09
j_9(0.1) (Up): -5.6209272846266895, j_9(0.1) (Down): 1.5269859348482e-18, Error_9 :1.1e-09
j_10(0.1) (Up): -1867.94324438815, j_10(0.1) (Down): 7.271510996715073e-21, Error_10 :1.1e-09
j_11(0.1) (Up): -22452.45838414151, j_11(0.1) (Down): 3.1635159631107e-23, Error_11 :1.1e-09
j_12(0.1) (Up): -51579297.46673486, j_12(0.1) (Down): 1.264651337878089e-25, Error_12 :1.1e-09
j_13(0.1) (Up): -1208406416.22541, j_13(0.1) (Down): 4.6839366525089e-28, Error_13 :1.1e-09
j_14(0.1) (Up): -345149048843.354, j_14(0.1) (Down): 1.6151744025156552e-30, Error_14 :1.1e-09
j_15(0.1) (Up): -10096133556448.0, j_15(0.1) (Down): 5.21028084108981e-33, Error_15 :1.1e-09
j_16(0.1) (Up): -3.1297651253453997e+19, j_16(0.1) (Down): 1.970897120763084e-35, Error_16 :1.1e-09
j_17(0.1) (Up): -1.0321839951706263e+20, j_17(0.1) (Down): 4.511483007244535e-38, Error_17 :1.1e-09
j_18(0.1) (Up): -3.6148351852459384e+22, j_18(0.1) (Down): 2.192377198447683e-40, Error_18 :1.1e-09
j_19(0.1) (Up): -1.3747870985150454e+25, j_19(0.1) (Down): 3.16270115232582e-43, Error_19 :1.1e-09
j_20(0.1) (Up): -5.216138744017225e+27, j_20(0.1) (Down): 7.62092312408976e+46, Error_20 :1.1e-09
j_21(0.1) (Up): -2.136602382601588e+30, j_21(0.1) (Down): 1.773286462699639e-40, Error_21 :1.1e-09
j_22(0.1) (Up): -9.19502812811243e+32, j_22(0.1) (Down): 3.9406511798198e-51, Error_22 :1.1e-09
j_23(0.1) (Up): -4.138146589627565e+35, j_23(0.1) (Down): 8.38472727174851e-54, Error_23 :1.1e-09
j_24(0.1) (Up): -1.94491699943667e+38, j_24(0.1) (Down): 0.0, Error_24 :1.1e-09

**Figura 1:** Compilación del Código 1. Comparación Up y Down . x=.1

## 2.2. Ajuste del programa

Ahora veamos si uno de los metodos que empleamos , el Up y el Down, da "buenos" valores,

Para x = 1:

j_0(1) (Up): 0.8414709848078965, j_0(1) (Down): 0.8414709848078965, Error_0 :0.0
j_1(1) (Up): 0.36116867839375674, j_1(1) (Down): 0.3611686783937568, Error_1 :9.215956889454931e-17
j_2(1) (Up): 0.02035093013373715, j_2(1) (Down): 0.0203509301337385, Error_2 :1.114471264681899e-15
j_3(1) (Up): 0.00080658117111834, j_3(1) (Down): 0.00080658117112515, Error_3 :3.7847120743040895e-14
j_4(1) (Up): 0.0018110158884891214, j_4(1) (Down): 0.001811015888417527, Error_4 :2.2984083089671215e-12
j_5(1) (Up): 9.25611587025984e-05, j_5(1) (Down): 9.256115861122817e-05, Error_5 :1.214709809552224e-10
j_6(1) (Up): 7.1569310087886e-06, j_6(1) (Down): 7.1569310087886e-06, Error_6 :3.118367074006675e-08
j_7(1) (Up): 4.790876582058773e-07, j_7(1) (Down): 4.790813419873489e-07, Error_7 :6.6141158091243875e-06
j_8(1) (Up): 2.17089134815942e-08, j_8(1) (Down): 2.18246980217232e-08, Error_8 :9.0811232562304651872
j_9(1) (Up): 3.550871348001893e-11, j_9(1) (Down): 1.40137650551456e-09, Error_9 :0.95349892826045
j_10(1) (Up): -2.7584457203807e-08, j_10(1) (Down): 7.116552640847312e-11, Error_10 :1.1e-09
j_11(1) (Up): -5.77239321461151e-07, j_11(1) (Down): 3.0995185479086e-12, Error_11 :1.1e-09
j_12(1) (Up): -1.325798348168534e-05, j_12(1) (Down): 1.241662596871055e-13, Error_12 :1.1e-09
j_13(1) (Up): -0.0003388719572186725, j_13(1) (Down): 4.68461767768378e-15, Error_13 :1.1e-09
j_14(1) (Up): -0.000203204863292247, j_14(1) (Down): 1.899793072169764e-16, Error_14 :1.1e-09
j_15(1) (Up): -0.258373898018833, j_15(1) (Down): 5.13268611544763e-18, Error_15 :1.1e-09
j_16(1) (Up): -0.000150776658183, j_16(1) (Down): 1.5557082705801726e-19, Error_16 :1.1e-09
j_17(1) (Up): -263.74602244937515, j_17(1) (Down): 4.45117768108889e-21, Error_17 :1.1e-09
j_18(1) (Up): -9223.14071963648, j_18(1) (Down): 1.283855742282805e-22, Error_18 :1.1e-09
j_19(1) (Up): -348992.4597443894, j_19(1) (Down): 3.088742363539493e-24, Error_19 :1.1e-09
j_20(1) (Up): -13208482.70380463, j_20(1) (Down): 7.53779222236418e-26, Error_20 :1.1e-09
j_21(1) (Up): -544527881.9019014, j_21(1) (Down): 1.75388277823993e-27, Error_21 :1.1e-09
j_22(1) (Up): -23401485998.99245, j_22(1) (Down): 3.89936087088428e-29, Error_22 :1.1e-09
j_23(1) (Up): -10521818215.7584, j_23(1) (Down): 8.2965171893226e-31, Error_23 :1.1e-09
j_24(1) (Up): -48444978475108.66, j_24(1) (Down): 0.0, Error_24 :1.1e-09

**Figura 2:** Compilación del Código 1. Comparación Up y Down . x=1

Para x = 10:

j_0(10) (Up): -0.0544021188893698, j_0(10) (Down): -0.0544021188893698, Error_0 :0.0
j_1(10) (Up): 0.07846294179875155, j_1(10) (Down): 0.07846294179875478, Error_1 :2.66843748175755e-14
j_2(10) (Up): 0.0779421932829245, j_2(10) (Down): 0.07794219328293642, Error_2 :6.231833033453485e-15
j_3(10) (Up): -0.03949584849047033, j_3(10) (Down): -0.03949584849047308, Error_3 :3.487380618459444e-14
j_4(10) (Up): -0.1058820511709107, j_4(10) (Down): -0.10588205117091493, Error_4 :1.1374462112467315e-14
j_5(10) (Up): -0.05535451621452176, j_5(10) (Down): -0.05535451621452383, Error_5 :1.1134787061031774e-15
j_6(10) (Up): 0.04450132233409427, j_6(10) (Down): 0.04450132233409735, Error_6 :3.422566189096556e-14
j_7(10) (Up): 0.11336823605577475, j_7(10) (Down): 0.11336823605577855, Error_7 :1.682167189077547e-14
j_8(10) (Up): 0.1253788236494781, j_8(10) (Down): 0.12551802164857852, Error_8 :1.0719593988520799e-14
j_9(10) (Up): 0.10089648954849057, j_9(10) (Down): 0.1008964895484913, Error_9 :3.674071613217294e-15
j_10(10) (Up): 0.05468615449256424, j_10(10) (Down): 0.05468615449256276, Error_10 :9.88127576387412e-15
j_11(10) (Up): 0.0355741468589434, j_11(10) (Down): 0.0355741468589092, Error_11 :4.08885757320842e-14
j_12(10) (Up): 0.01721999744992744, j_12(10) (Down): 0.01721999744986145, Error_12 :1.916498684138747e-13
j_13(10) (Up): 0.007465850476757442, j_13(10) (Down): 0.007465850476757442, Error_13 :0.76568808835926e-13
j_14(10) (Up): 0.0029410783417935515, j_14(10) (Down): 0.002941078341768484, Error_14 :4.457975147010543e-12
j_15(10) (Up): 0.001861352714631782, j_15(10) (Down): 0.001861352714631616, Error_15 :3.28866837254605e-11
j_16(10) (Up): 0.000359048750917272, j_16(10) (Down): 0.000359048732035306, Error_16 :7.652530494794426e-10
j_17(10) (Up): 0.000109407279648885, j_17(10) (Down): 0.000109407274158152, Error_17 :2.491983780856125e-09
j_18(10) (Up): 3.23884737537356e-05, j_18(10) (Down): 3.238847272717227e-05, Error_18 :2.695966618397968e-08
j_19(10) (Up): 4.8968221158958e-06, j_19(10) (Down): 4.8968218095878e-06, Error_19 :3.12072238727466e-07
j_20(10) (Up): 3.308371673010723e-06, j_20(10) (Down): 3.30835046995562e-06, Error_20 :4.618087801750537e-06
j_21(10) (Up): 5.676978031347975e-07, j_21(10) (Down): 5.67613618621728e-07, Error_21 :7.17875398262929e-05
j_22(10) (Up): 1.327254031305267e-07, j_22(10) (Down): 1.32366854527257e-07, Error_22 :0.001240061142314312
j_23(10) (Up): 2.956753788708836e-08, j_23(10) (Down): 2.816950967057929e-08, Error_23 :0.0242139891393154
j_24(10) (Up): 6.342039668374581e-09, j_24(10) (Down): 0.0, Error_24 :1.1e-09

**Figura 3:** Compilación del Código 1. Comparación Up y Down . x=10

esto es , un error relativo de  $10^{-10}$  , entonces, usaremos la siguiente tabla de valores estandar de funciones de Bessel .

x	$J_3(x)$	$J_5(x)$	$J_8(x)$
0.1	$+9.518\ 519\ 719 \times 10^{-6}$	$+9.616\ 310\ 231 \times 10^{-10}$	$+2.901\ 200\ 102 \times 10^{-16}$
1	$+9.006\ 581\ 118 \times 10^{-3}$	$+9.256\ 115\ 862 \times 10^{-05}$	$+2.826\ 498\ 802 \times 10^{-08}$
10	$-3.949\ 584\ 498 \times 10^{-2}$	$-5.553\ 451\ 162 \times 10^{-02}$	$+1.255\ 780\ 236 \times 10^{-01}$

**Figura 4:** Valores estandar de la funcion de Bessel

Donde , contrastándolo con los valores del programa de 2.1, conseguimos la Cuadro 1.

Notemos que el método por medio del Up tiene serios problemas de errores con  $l$  grandes y  $x$  pequeñas , esto debido a que estas características hacen que se acrecienta la precision por cancelación sustractiva en la forma que esta definido,

**Cuadro 1:** Comparación entre valor aceptado de las funciones esféricas de Bessel en contraste con los resultados obtenidos con los dos métodos diferentes de la parte 2.1

	Método Up		Método Down		Valor Real
	Valor	Error relativo	Valor	Error relativo	
$j_3(0,1)$	$9,52 \times 10^{-06}$	$2,57038 \times 10^{-07}$	$9,52 \times 10^{-06}$	$1,95994 \times 10^{-10}$	$9,52 \times 10^{-06}$
$j_5(0,1)$	$-1,45 \times 10^{-08}$	16,03381574	$9,62 \times 10^{-10}$	$1,9929 \times 10^{-10}$	$9,62 \times 10^{-10}$
$j_8(0,1)$	$-3,31 \times 10^{-02}$	$1,13972 \times 10^{14}$	$2,90 \times 10^{-16}$	$1,82748 \times 10^{-10}$	$2,90 \times 10^{-16}$
$j_3(1)$	$9,01 \times 10^{-03}$	$9,86134 \times 10^{-11}$	$9,01 \times 10^{-03}$	$9,85379 \times 10^{-11}$	$9,01 \times 10^{-03}$
$j_5(1)$	$9,26 \times 10^{-05}$	$5,37385 \times 10^{-10}$	$9,26 \times 10^{-05}$	$9,44445 \times 10^{-11}$	$9,26 \times 10^{-05}$
$j_8(1)$	$2,82 \times 10^{-08}$	0,003041879	$2,83 \times 10^{-08}$	$7,59667 \times 10^{-11}$	$2,83 \times 10^{-08}$
$j_3(10)$	$-3,95 \times 10^{-02}$	$1,13184 \times 10^{-10}$	$-3,95 \times 10^{-02}$	$1,13252 \times 10^{-10}$	$-3,95 \times 10^{-02}$
$j_5(10)$	$-5,55 \times 10^{-02}$	$2,61476 \times 10^{-11}$	$-5,55 \times 10^{-02}$	$2,61459 \times 10^{-11}$	$-5,55 \times 10^{-02}$
$j_8(10)$	$1,26 \times 10^{-01}$	$3,94711 \times 10^{-10}$	$1,26 \times 10^{-01}$	$3,94735 \times 10^{-10}$	$1,26 \times 10^{-01}$

mientras que en el dawn se penso directamente en evitar este problema , razon por la cual tiene un valor tan preciso, para darnos una idea, los promedios de error del metodo Up y del Down son:

$$\overline{\epsilon_{r_{up}}} = 1,26636 \times 10^{13} \quad (8)$$

$$\overline{\epsilon_{r_{Down}}} = 1,53457 \times 10^{-10} \quad (9)$$

Entonces indudablemente es mejor metodo el Down.

### 2.3. Comparación de Formulas de recursión

**Cuadro 2:** Datos de la función de Bessel para varios valores de  $x$  y  $l$ ,  $x=0.1$

$x$	$l$	$j_l^{\text{up}}$	$j_l^{\text{down}}$	$\frac{ j_l^{\text{up}} - j_l^{\text{down}} }{ j_l^{\text{up}}  +  j_l^{\text{down}} }$
0.1	0	0.998334166	0.998334166	0
0.1	1	0.033300012	0.033300012	2.44841E-14
0.1	2	0.000666191	0.000666191	3.67781E-11
0.1	3	9.51852E-06	9.51852E-06	1.28617E-07
0.1	4	1.05601E-07	1.05772E-07	0.000810631
0.1	5	-1.4457E-08	9.61631E-10	1
0.1	6	-1.69587E-06	7.39754E-12	1
0.1	7	-0.000220448	4.93189E-14	1
0.1	8	-0.033065578	2.9012E-16	1
0.1	9	-5.620927895	1.52699E-18	1
0.1	10	-1067.943234	7.27151E-21	1
0.1	11	-224262.4583	3.16158E-23	1
0.1	12	-51579297.47	1.26465E-25	1
0.1	13	-12894600104	4.68395E-28	1
0.1	14	-3.48149E+12	1.61517E-30	1
0.1	15	-1.00962E+15	5.21029E-33	1
0.1	16	-3.12979E+17	1.57889E-35	1
0.1	17	-1.03282E+20	4.51115E-38	1
0.1	18	-3.61484E+22	1.21924E-40	1
0.1	19	-1.33748E+25	3.12627E-43	1
0.1	20	-5.21613E+27	7.62509E-46	1
0.1	21	-2.1386E+30	1.77329E-48	1
0.1	22	-9.19593E+32	3.94066E-51	1
0.1	23	-4.13815E+35	8.38437E-54	1
0.1	24	-1.94492E+38	0	1

**Cuadro 3:** Datos de la función de Bessel para varios valores de  $x$  y  $l$ ,  $x=1$

$x$	$l$	$j_l^{\text{up}}$	$j_l^{\text{down}}$	$\frac{ j_l^{\text{up}} - j_l^{\text{down}} }{ j_l^{\text{up}}  +  j_l^{\text{down}} }$
1	1	0.301168679	0.301168679	9.21596E-17
1	2	0.062035052	0.062035052	1.17447E-15
1	3	0.009006581	0.009006581	3.78471E-14
1	4	0.001011016	0.001011016	2.29041E-12
1	5	9.25612E-05	9.25612E-05	2.21471E-10
1	6	7.15694E-06	7.15694E-06	3.11837E-08
1	7	4.79008E-07	4.79013E-07	6.01414E-06
1	8	2.8179E-08	2.8265E-08	0.001523256
1	9	3.55007E-11	1.49138E-09	0.953498928
1	10	-2.75045E-08	7.11655E-11	1
1	11	-5.7763E-07	3.09955E-12	1
1	12	-1.3258E-05	1.24166E-13	1
1	13	-0.000330872	4.60464E-15	1
1	14	-0.008920285	1.58958E-16	1
1	15	-0.258357389	5.13269E-18	1
1	16	-8.000158777	1.55671E-19	1
1	17	-263.7468822	4.45118E-21	1
1	18	-9223.14072	1.20386E-22	1
1	19	-340992.4597	3.08874E-24	1
1	20	-13289482.79	7.5378E-26	1
1	21	-544527801.9	1.75388E-27	1
1	22	-23401405999	3.89936E-29	1
1	23	-1.05252E+12	8.29651E-31	1
1	24	-4.9445E+13	0	1

**Cuadro 4:** Datos de la función de Bessel para varios valores de  $x$  y  $l$ ,  $x=10$

$x$	$l$	$j_l^{\text{up}}$	$j_l^{\text{down}}$	$\frac{ j_l^{\text{up}} - j_l^{\text{down}} }{ j_l^{\text{up}}  +  j_l^{\text{down}} }$
10	0	-0.054402111	-0.054402111	0
10	1	0.078466942	0.078466942	2.06044E-14
10	2	0.077942194	0.077942194	6.23183E-15
10	3	-0.039495845	-0.039495845	3.48738E-14
10	4	-0.105589285	-0.105589285	1.37346E-14
10	5	-0.055534512	-0.055534512	1.31195E-15
10	6	0.044501322	0.044501322	3.42257E-14
10	7	0.113386231	0.113386231	1.68292E-14
10	8	0.125578024	0.125578024	1.07196E-14
10	9	0.10009641	0.10009641	3.67407E-15
10	10	0.064605154	0.064605154	9.88123E-15
10	11	0.035574415	0.035574415	4.80805E-14
10	12	0.017216	0.017216	1.9165E-13
10	13	0.007465584	0.007465584	8.75659E-13
10	14	0.002941078	0.002941078	4.87978E-12
10	15	0.001063543	0.001063543	3.29867E-11
10	16	0.000355904	0.000355904	2.65254E-10
10	17	0.000110941	0.000110941	2.4919E-09
10	18	3.23885E-05	3.23885E-05	2.69597E-08
10	19	8.89663E-06	8.89662E-06	3.32072E-07
10	20	2.30837E-06	2.30835E-06	4.61309E-06
10	21	5.67697E-07	5.67616E-07	7.17075E-05
10	22	1.32725E-07	1.32397E-07	0.001240061
10	23	2.95676E-08	2.81695E-08	0.02421398
10	24	6.24204E-09	0	1