

Código para las funciones esféricas de Bessel

```
1 def esfericasBesselj(n,x):
2     if (x == 0e0): return 1e0
3     if (n == 0): return 0e0
4
5     j0 = sin(x) / x
6     if (n == 0): return j0
7     j1 = (j0 - cos(x)) / x
8     if (n == 1): return j1
9
10    nmax = 0
11
12    if(n >= fabs(x)):
13        jn = 1.
14        for i in range(n, n + 51):
15            jn *= (2*i - 1) / x
16            if (jn >= 1e8): nmax = i + 10; break
17
18    if (nmax == 0):
19        global j
20        for i in range(2, n+1):
21            j = (2*i - 1) / x*j1 - j0
22            j0 = j1 ; j1 = j
23        return j
24    else:
25        j2 = 0.; j1 = 1e-20
26        for i in range (nmax, -1, -1):
27            j = (2*i + 3) / x*j1 - j2
28            j2 = j1
29            j1 = j
30            if (i == n): jn = j
31
32        return (j0/j)*jn
33
34
35 for i in range(5):
36     x = []
37     y = []
```

```

38
39     for m in arange(0.1, 15., 0.1):
40         x.append(m)
41         y.append(esfericasBesselj(i, m))
42         plt.plot(x, y)
43
44
45 plt.ylim(-0.35,0.6)
46 plt.axhline(y=0, color='k', ls='dashed', lw=.75)
47 plt.title('Funciones esfericas de Bessel')
48 plt.show()

```

Puntos importantes:

1. La función `esfericasBessely` ocupa el algoritmo de Miller para calcular la función esférica de Bessel de orden n .
2. Para órdenes pequeños, la función utiliza las expresiones analíticas.
3. La variable `nmax` se usa como bandera de dirección para la iteración estable en la relación de recurrencia.