Special functions of mathematical physics

~

Spherical Bessel functions

M. Dapor - ECT* - dapor@ectstar.eu

Bessel equation of order ν

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - \nu^2)y = 0$$

The solution of this equation is a linear combination of the Bessel functions $J_{-\nu}$ and $J_{+\nu}$

Spherical Bessel functions

$$j_l(x) = \sqrt{\frac{\pi}{2x}} J_{l+\frac{1}{2}}(x)$$

Spherical Neumann functions

$$n_l(x) = (-1)^{l+1}\sqrt{\frac{\pi}{2x}}J_{-l-\frac{1}{2}}(x)$$

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

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

$$j_2 = \left(\frac{3}{x^3} - \frac{1}{x}\right)\sin x - \frac{3}{x^2}\cos x$$

$$n_0 = -\frac{\cos x}{x}$$

$$n_1 = -\frac{\cos x}{x^2} - \frac{\sin x}{x}$$

$$n_2 = \left(-\frac{3}{x^3} + \frac{1}{x}\right)\cos x - \frac{3}{x^2}\sin x$$

$$j_l(x) \underset{x\to 0}{\sim} \frac{x^l}{1\cdot 3\cdot ...\cdot(2l+1)}$$

$$n_l(x) \underset{x\to 0}{\sim} -\frac{1\cdot 3\cdot ...\cdot(2l-1)}{x^{l+1}}$$

$$j_l(x) \underset{x\to\infty}{\sim} \frac{\sin(x-\frac{l\pi}{2})}{x}$$

$$n_l(x) \underset{x\to\infty}{\sim} -\frac{\cos(x-\frac{l\pi}{2})}{x}$$

The following recursion relations are very useful when using any linear combination of Bessel and Neumann spherical functions

$$x f_{l-1} - (2l+1) f_l + x f_{l+1} = 0$$

$$x f_{l-1} - (l+1) f_l - x \frac{df_l}{dx} = 0$$

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define PI 3.141592654

double x,j,jj,n,nn,xmax,dx;
long int l;
FILE *bessj,*bessn;
char bj[20],bn[20];
```

```
void besselj()
{
    long il;
    double jp, jm, jz, norm;
```

```
if(l==0 && x==0)
{
    j=1;
    jj=0;
    return;
}
```

```
if(l>0 && x==0)
{
   j=0;
   jj=0;
   return;
}
```

```
jm=0.;
jz=9.9999e-300;
il=230;
```

```
while (il-- > 1)
{
   jp=(((2.*il+1)*jz)/x)-jm;
   if((il-1)==l) j=jp;
   if((il-1)==l+1) jj=jp;
   jm=jz;
   jz=jp;
}
```

```
  norm=sin(x)/x/jz;
  j=norm*j;
  jj=norm*jj;
  return;
}
```

```
void besseln()
{
   long il;
   double np, nm, nz;

   nm=(-cos(x))/x;
   nz=((-cos(x))/(x*x))-(sin(x))/x;
```

```
if (l==0)
{
    n=nm;
    nn=nz;
    return;
}
```

```
if (l==1)
{
   n=nz;
   nn=(cos(x))*(-(3/(x*x*x))+(1/x))-(sin(x))*(3/(x*x));
   return;
}
```

```
il=0;
while (il++ < l)
{
    np=(((2.*il+1)*nz)/x)-nm;
    nm=nz;
    nz=np;
}
```

```
    n=nm;
    nn=nz;
    return;
}
```

```
int main ()
{
    printf("j function file name = ");
    scanf("%s",bj);
    printf("n function file name = ");
    scanf("%s",bn);
    printf("l = ");
    scanf("%ld",&l);
```

```
bessj=fopen(bj,"w");
bessn=fopen(bn,"w");
x=1.0;
xmax=15.0;
dx=0.01;
```

```
while (x<=xmax)
{
   besselj();
   besseln();
   fprintf(bessj,"%lg %lg \n",x,j);
   fprintf(bessn,"%lg %lg \n",x,n);
   x=x+dx;
}
```

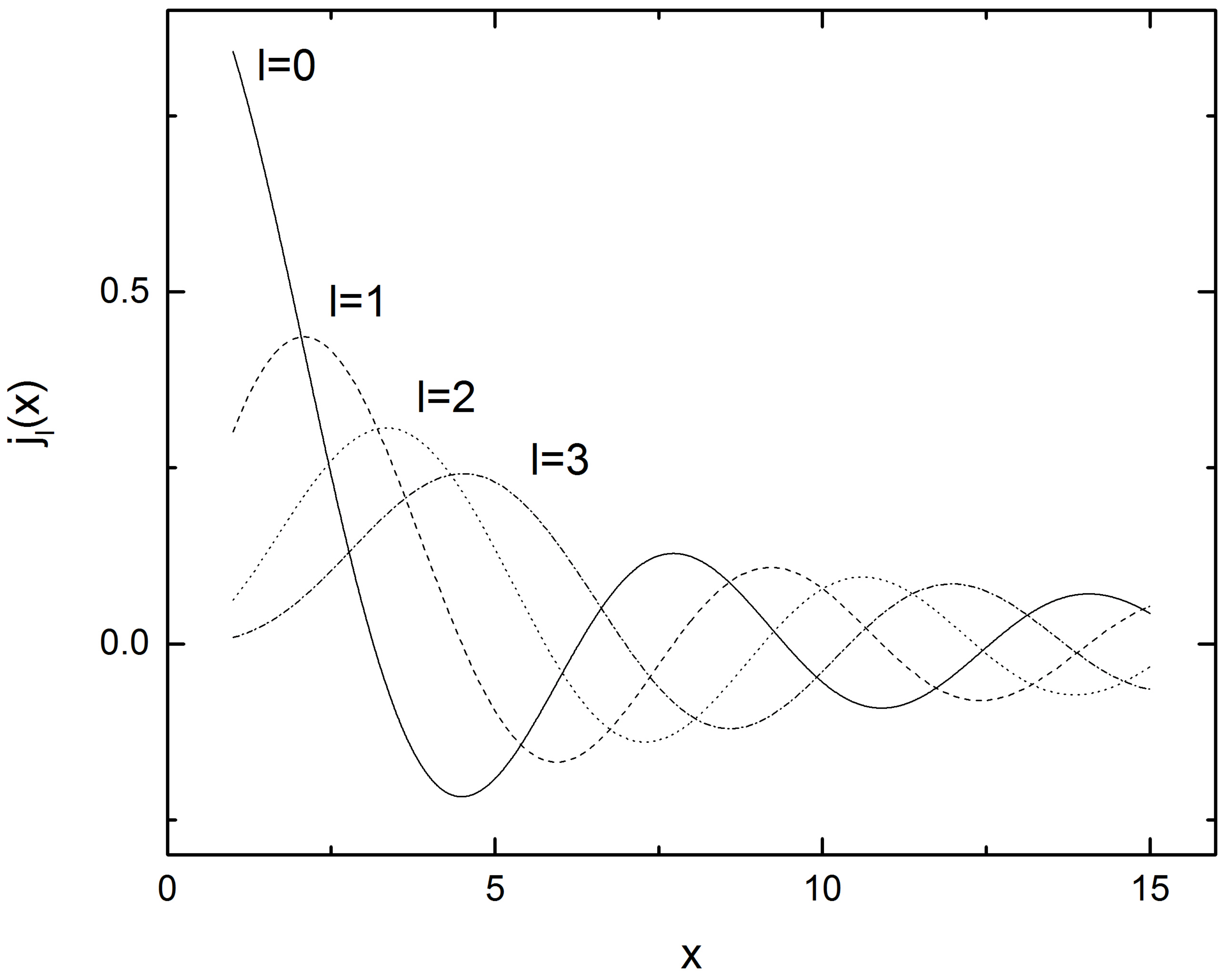
```
    fclose(bessn);
    fclose(bessj);
    return(0);
}
```

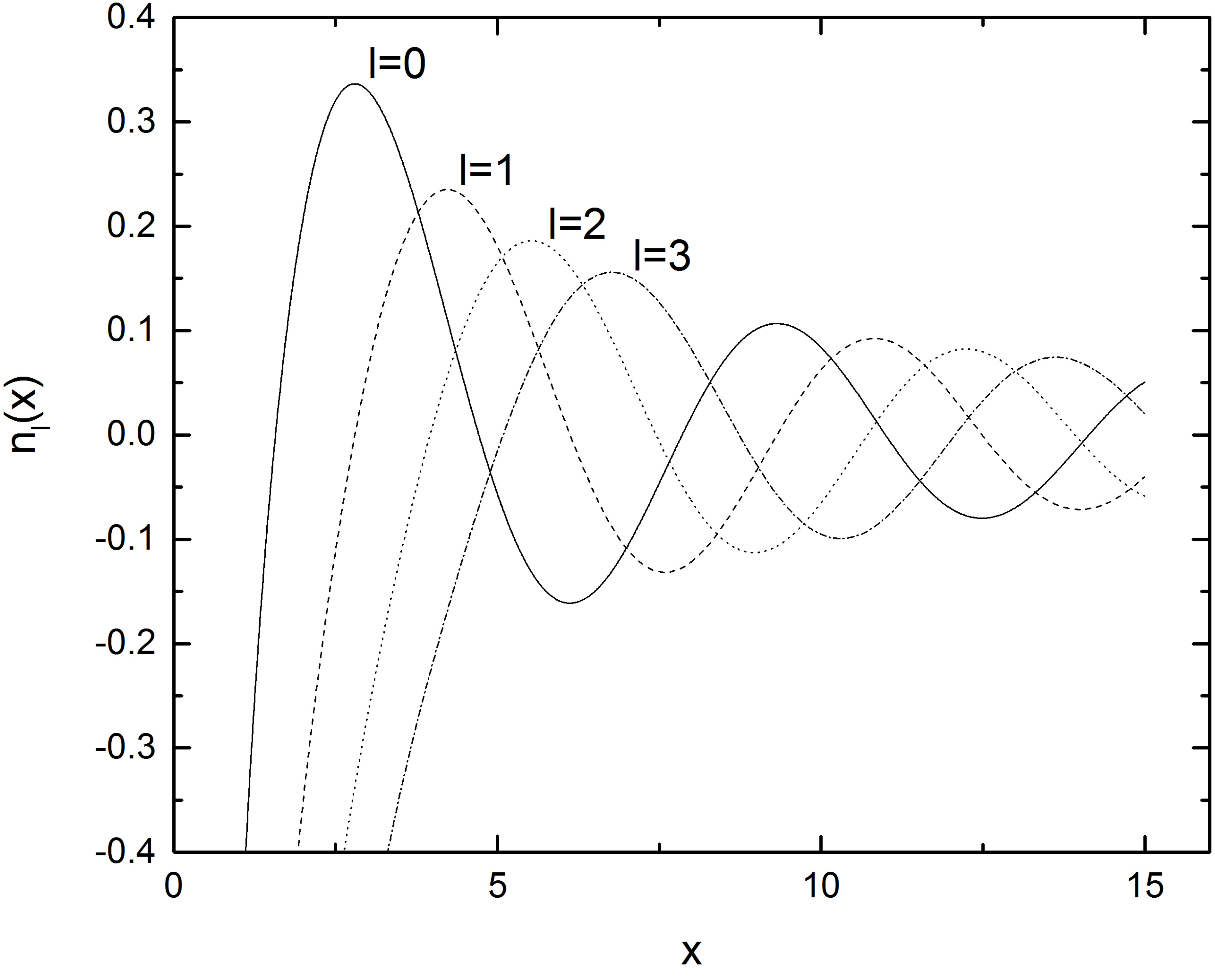
Compile a c file

```
gcc filename.c -o filename -lm
```

Run a compiled program

```
./filename
```


l=0
l=1
l=2
l=3
$j_l(x)$
0.5
0.0
0
5
10
15
x


l=0
l=1
l=2
l=3
$n_l(x)$
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
0
5
10
15
x