

```

Jan 02, 11 20:33                bessel.cpp                Page 1/2
// file: bessel.cpp
//
// Spherical Bessel functions via up and down recursion
//
// Programmer: Dick Furnstahl furnstahl.1@osu.edu
//
// Revision history:
// 02-Jan-2011 new version, for 780.20 Computational Physics
//
// Notes:
// * compile with: "g++ -o bessel bessel.cpp"
// * adapted from: "Projects in Computational Physics" by Landau and Paez
//               copyrighted by John Wiley and Sons, New York
//               code copyrighted by RH Landau
// * data saved as: x y1 y2 --- should print column headings!!
//
//*****
// include files
#include <iostream>           // note that .h is omitted
#include <iomanip>            // note that .h is omitted
#include <fstream>           // note that .h is omitted
#include <cmath>
using namespace std;        // we need this when .h is omitted

// function prototypes
double down_recursion (double x, int n, int m); // downward algorithm
double up_recursion (double x, int n);         // upward algorithm

// global constants
const double xmax = 100.0; // max of x
const double xmin = 0.1;  // min of x > 0
const double step = 0.1;  // delta x
const int order = 10;     // order of Bessel function
const int start = 50;     // used for downward algorithm

//*****
int
main ()
{
    double ans_down, ans_up;

    // open an output file stream
    ofstream my_out ("bessel.dat");

    my_out << "# Spherical Bessel functions via up and down recursion"
             << endl;

    // step through different x values
    for (double x = xmin; x <= xmax; x += step)
    {
        ans_down = down_recursion (x, order, start);
        ans_up = up_recursion (x, order);

        my_out << fixed << setprecision (6) << setw (8) << x << " "
                << scientific << setprecision (6)
                << setw (13) << ans_down << " "
                << setw (13) << ans_up
                << endl;
    }
    cout << "data stored in bessel.dat." << endl;

    // close the output file
    my_out.close ();
    return (0);
}

```

```

Jan 02, 11 20:33                bessel.cpp                Page 2/2
//-----end of main program-----
// function using downward recursion
double
down_recursion (double x, int n, int m)
{
    double j[start + 2]; // array to store Bessel functions
    j[m + 1] = j[m] = 1.; // start with "something" (choose 1 here)
    for (int k = m; k > 0; k--)
    {
        j[k - 1] = ((2.* double(k) + 1.) / x) * j[k] - j[k + 1]; // recur. rel.
    }
    double scale = (sin (x) / x) / j[0]; // scale the result
    return (j[n] * scale);
}

//-----
// function using upward recursion
double
up_recursion (double x, int n)
{
    double term_three = 0.;
    double term_one = (sin (x)) / x; // start with lowest order
    double term_two = (sin (x) - x * cos (x)) / (x * x); // next order
    for (int k = 1; k < n; k += 1) // loop for order of function
    { // recurrence relation
        term_three = ((2.*double(k) + 1.) / x) * term_two - term_one;
        term_one = term_two;
        term_two = term_three;
    }
    return (term_three);
}

```