Homework 4: Bessel Function

Riccardo Zangarini

December 23, 2021

1 Code

1.1 Description

The aim of this homework is to solve the Bessel's differential equation:

$$x^2 \frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - \nu^2)y = 0 \tag{1}$$

using the *tridiagonal* method. The boundary conditions are:

$$y(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } x = 10 \end{cases} \tag{2}$$

In order to solve the matricial equation, we need to find the arrays a, b and c, which are the central diagonals of the tridiagonal matrix.
Using the 2nd order finite difference approximation, we find a system of $N - 2$ equations:

$$y_{i+1}c_i + y_i b_i + y_{i-1}a_i = 0 \tag{3}$$

where:

$$a_i = 2x_i^2 - hx_i, \tag{4}$$

$$b_i = -4x_i^2 + 2h^2(x_i^2 - 1), \tag{5}$$

$$c_i = 2x_i^2 + hx_i. \tag{6}$$

The first and last element of the array y are defined by (2).
Setting $N = 100$ and $\nu = 1$, we can evaluate the solution for each interval Δx.

1.2 Code

```
#include "my_header.h"

int main(){
  int i, n = 100;
  double x_r = 10., x_l = 0., x_i;
  double alpha = 0, beta = 1., h = fabs(x_r - x_l) / (double)(n - 1);
  ofstream fdata;

  double *y;
  y = new double[n];
  y[0] = alpha; y[n - 1] = beta;             // initial conditions

  double *r;
  r = new double[n];

  double *a;
  a = new double[n];

  double *b;
  b = new double[n];

  double *c;
  c = new double[n];

  // defining arrays
  for(i = 0; i < n; i++){
    x_i = x_l + (double)i * h;

    a[i] = (2. * x_i * x_i) - (h * x_i);
    b[i] = -(4. * x_i * x_i) + (2. * h * h * (x_i * x_i - 1.));
    c[i] = (2. * x_i * x_i) + (h * x_i);

    r[i] = 0.;
  }
  r[1] = -(alpha * a[0]);                    // the rhs function is not included because
  r[n - 2] = -(beta * c[n - 1]);             // its value is always 0.

  Tridiag(r + 1, a + 1, c + 1, b + 1, y + 1, n - 2);

  fdata << setiosflags(ios::scientific);
  fdata.open("bessel.txt", std::ofstream::out);

  // plot of the y[i] with respect to the x
  for(i = 0; i < n; i++){
    x_i = x_l + (double)i * h;
    fdata << x_i << " \t " << y[i] << "\n";
  }

  fdata.close();

  delete r;
  delete a;
  delete c;
  delete b;
  delete y;

  return 0;
}
```

2 Output

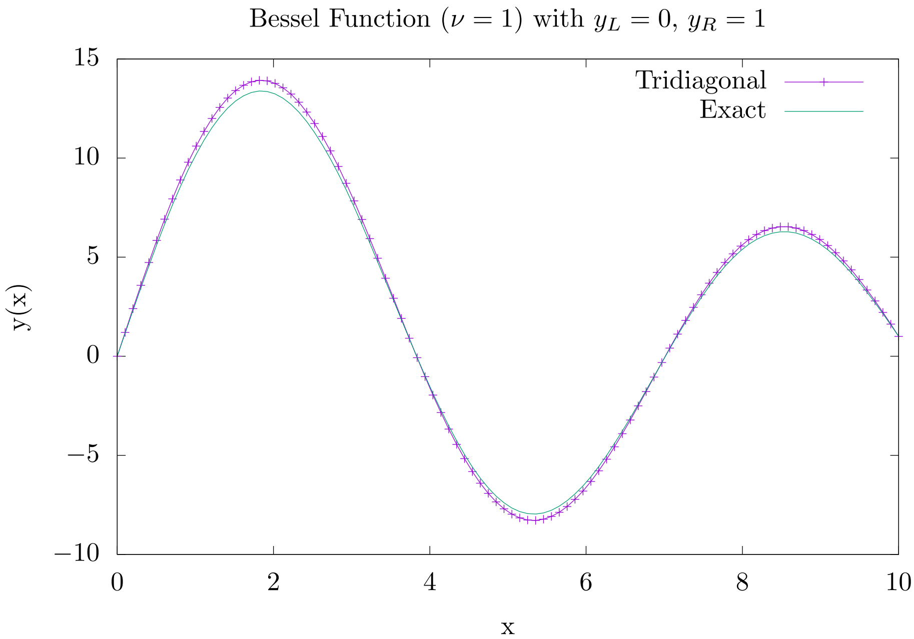

Figure 1: Comparison between the Bessel Function solved with the tridiagonal method and the analytical solution. The second one is normalized with the rightmost point $J_1(x = 10.0)$.

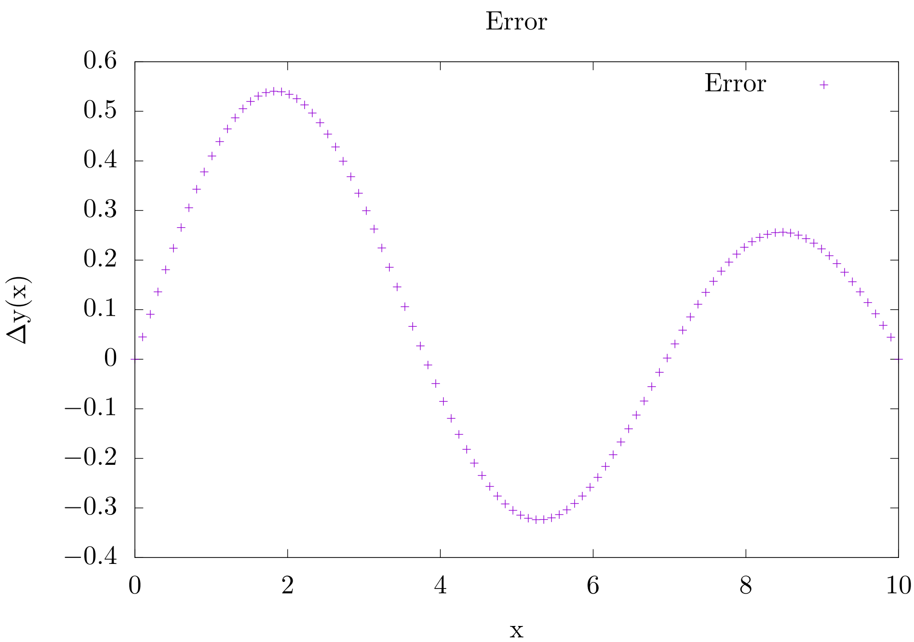

Figure 2: Error $J_1^{tridiag}(x) - J_1^{exact}(x)$ using the same results from Figure 1.