Bessel Functions and Their Application to the Eigenvalues of the Laplace Operator

Matthew Jin

June 1, 2014

1 Introduction

Bessel functions of the first kind, J_n, are the solutions of Bessel's differential equation $x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - n^2)y = 0$ that do not have singularities at the origin. One can mathematically define $J_n(x)$ via its Taylor series expansion around $x = 0$: $J_n(x) = (\frac{1}{2}x)^n \sum_{k=0}^{\infty}(-1)^k \frac{(\frac{1}{4}x^2)^k}{k!\Gamma(n+k+1)}$. They appear as follows:

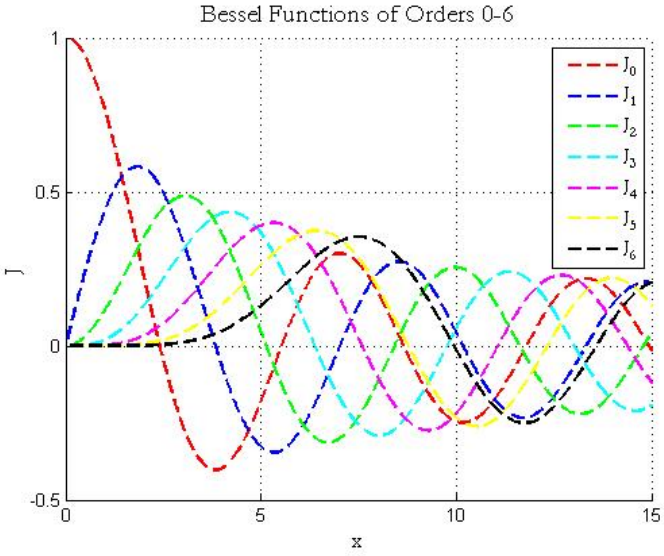

Figure 1: Plot of Bessel functions of the first kind for orders 0 through 6

This paper will discuss how to compute Bessel functions using Miller's algorithm, and how these functions can be applied to finding the vibration frequencies of a thin circular membrane.

2 Computing Bessel Functions

2.1 Recurrence Relation and Its Stability

The first task in this project was to compute Bessel functions, which was done using Miller's downwards recurrence algorithm. The algorithm applies the recurrence relation $J_{n-1}(x) = \frac{2n}{x} J_n(x) - J_{n+1}(x)$, where n is the order of Bessel function J—it computes Bessel functions in descending order from a higher seed order n_0.

To understand why this method is effective at computing Bessel functions, one must understand the stability of the recurrence relation. In general, a three-term recurrence relation has two linearly independent solutions, one of which corresponds to the sequence of terms that one wishes to compute. The other solution may be exponentially shrinking or growing in the direction that one is proceeding, or neither. In any case, it is impossible to compute the desired solution from the recurrence relation in the direction in which the unwanted solution is exponentially growing.

When computing Bessel functions, the unwanted solution exponentially grows for $n > x$ in the forwards direction of the recurrence relation, $J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x)$. One can heuristically see this by assuming that $\frac{2n}{x}$ is constant and applying the quadratic formula to find the solutions of $J_{n+1} - 2\frac{n}{x} J_n(x) + J_{n-1}(x) = 0$

$r^{n+1} - 2\frac{n}{x} r^n + r^{n-1} = 0$
$r^2 - 2\frac{n}{x} + 1 = 0$

One finds $r = \frac{n}{x} \pm \sqrt{\frac{n^2}{x^2} - 1}$. Note that the forward recurrence relation is stable when $|r| \leq 1$, which occurs when $n \leq x$, and unstable otherwise. In other words, for $n > x$ the unwanted solution catastrophically increases and renders it impossible to compute the desired minimal solution, which meanwhile decays. This can be seen from the following plots of J_n in the x, n plane:

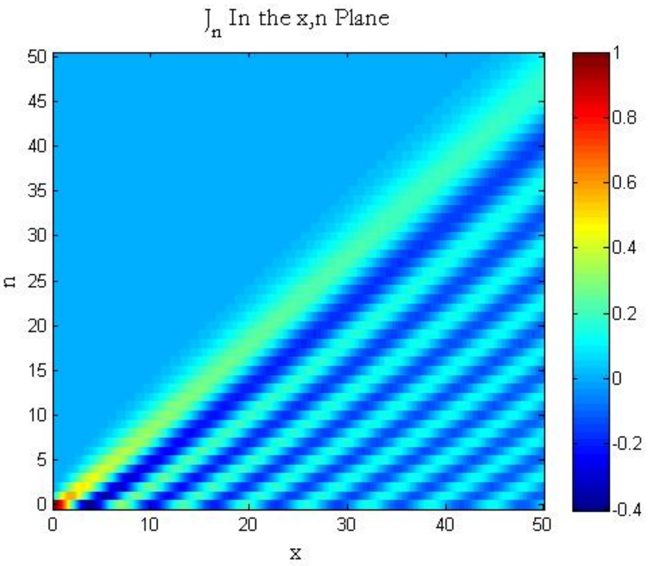

Figure 2: Imagesc plot of Bessel functions of the first kind in the x,n plane

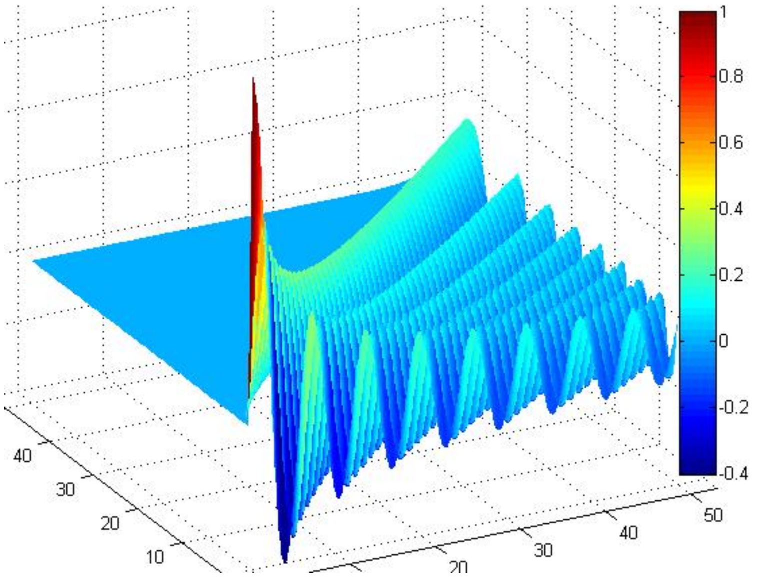

Figure 3: 3D plot of Bessel functions of the first kind in the x,n plane

One can see that for $n > x$, the desired minimal solution J_n decays rapidly, whereas for $n < x$, it oscillates with decaying amplitude.

2.2 Downwards Recurrence

If a solution catastrophically increases in one direction, then it will rapidly decrease in the opposite one. This means that when computing Bessel functions, for $n > x$, the undesired solution rapidly decreases in the backwards direction, allowing one to compute the desired Bessel function values, multiplied by a normalization factor $(J_0+2J_2(x)+2J_4(x)+...)$. The following diagram compares Bessel function values computed using a forwards recurrence method starting at J_0 and J_1 against the downwards recurrence method in which the initial n is always larger than x.

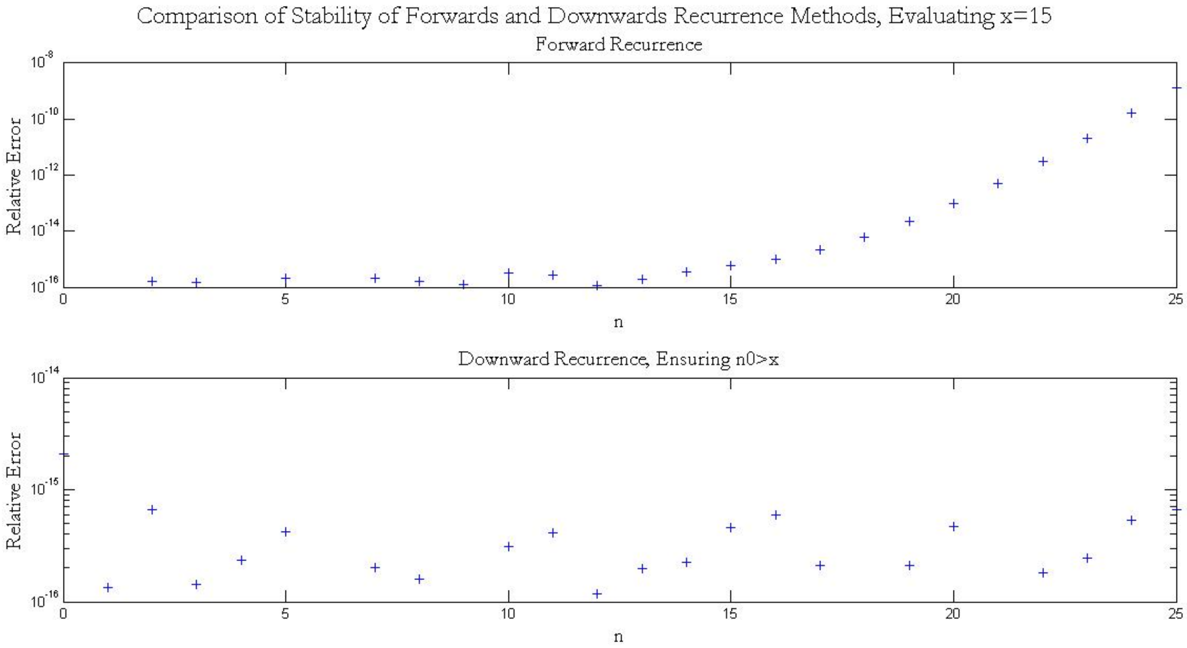

Figure 4: A comparison of the relative error when computing J_n using forward recurrence vs. downward recurrence

For forwards recurrence, although the error is near machine precision for $n < x$, the error catastrophically increases for $n > x$. However, for downwards recurrence, for an initial $n_0 > x$, one can achieve near machine precision for $J_n(x)$ at all n. Thus one can see that Miller's algorithm is able to reliably evaluate Bessel functions of all orders for all x, as long as the initial order at which one begins the downwards recurrence is large enough.

2.3 Initial Seed Order Size

One must immediately wonder, how large must n_0 be before the sequence has converged upon $J_n(x)$ with high accuracy? According to William Press's Numerical Recipes in C, one must start higher than n by approximately $\sqrt{Cn}$, where $\sqrt{C}$ is approximately the number of digits of accuracy. For this project, for input $n \geq x$, the starting seed order was $5 + n + \lceil\sqrt{100n}\rceil$, and for $x \geq n$, the starting seed order was $5 + x + \lceil\sqrt{100x}\rceil$. This guaranteed that for each computation, the initial starting order n_0 was always larger than both x and the desired n by a sufficient amount. Theoretically, one would expect this to give $\sqrt{100} = 10$ digits of accuracy. In practice, the function written for this project typically grants 14 to 16 significant digits of accuracy, which can be seen from Figure 4.

2.4 Algorithm Speed

The approximate algorithm speed relative to the approximate order of x and n is detailed in the following table:

Order of n and x (digits)	Seconds (Order)
1 and 2	10^{-4}
3	10^{-3}
4	10^{-2}
5	10^{-1}
6	1

Figure 5: This table compares the size of n and x against the amount of time it takes the downwards recurrence algorithm written for this project to compute $J_n(x)$. Note that order of n and x is given in number of digits, so order 1 and 2 in this case refers to numbers ranging from 1 to 99, and 5 refers to numbers ranging from 10000 to 99999. Furthermore, the number of seconds is given in terms of order of magnitude, so 10^{-4} indicates times ranging from 10^{-4} to 10^{-3}, and 1 indicates numbers ranging from 1 to 10. Note that the observed times are approximate and subject to variation.

One can see that for n and x of order around 6 digits and less, the time scales approximately linearly with slope 1 against the magnitude of n and x. For n and x of higher magnitude, the time blows up significantly and was not measured. Figure 6 illustrates the time the algorithm takes to compute $J_n(x)$ in the x,n plane.

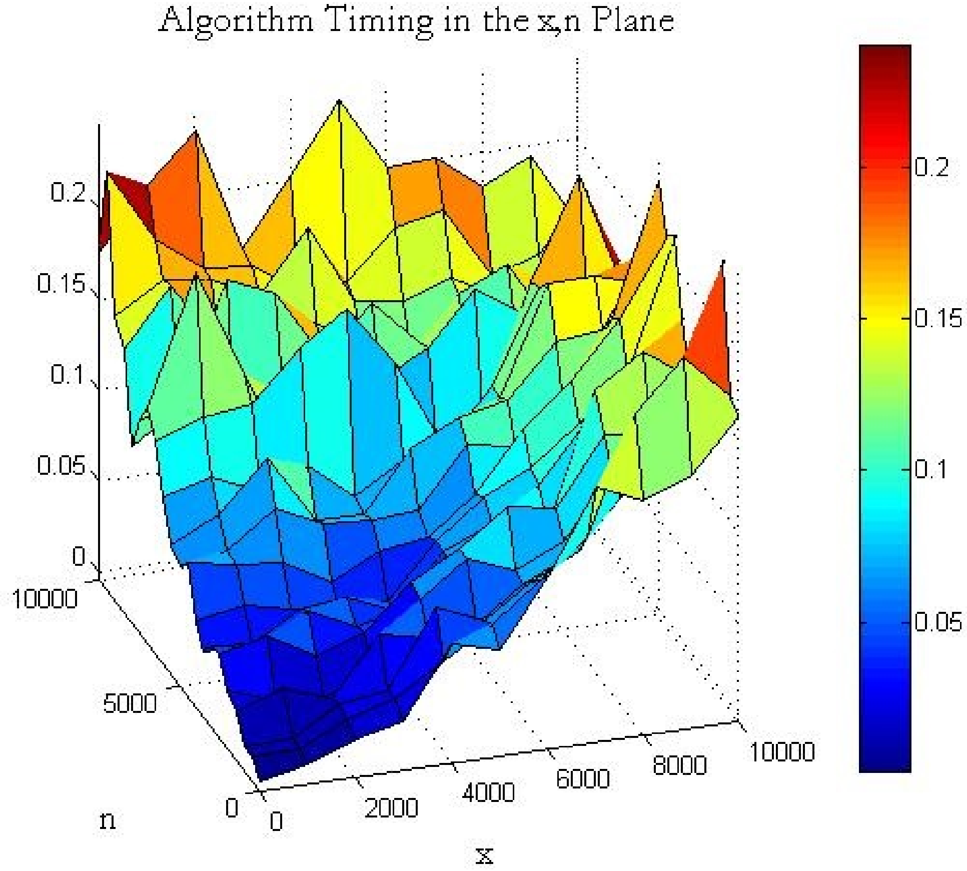

Figure 6: The algorithm timing scales with both n and x because both n and x determine the initial seed order size. If $n > x$, n determines the starting initial starting order n_0, whereas if $x > n$, x determines n_0.

3 Bessel Function Zeroes As the Eigenvalues of the Laplace Operator In a Unit Disc

Consider a unit disc D with a membrane stretched across it. Vibrations across it satisfy the two-dimensional wave equation with Dirichlet boundary conditions. We wish to find the solution to the following problem:

$$\begin{cases} -\Delta u = \lambda u & (1) \\ u(1,\theta) = 0 \ \forall \theta & (2) \end{cases}$$

Note that here we solve a version of the problem without time-dependence and will operate in polar coordinates. We begin by rewriting Δu:

$\Delta u = \frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial u}{\partial r}) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$

$\Delta u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$

Now, we apply separation of variables to u: $u(r,\theta) = R(r)\Theta(\theta)$.

$\Delta u = R''(r)\Theta(\theta) + \frac{1}{r}R'(r)\Theta(\theta) + \frac{1}{r^2}R(r)\Theta''(\theta)$

Applying (1) and simplifying, we find

$\frac{R''}{R} + \frac{1}{r}\frac{R'}{R} + \frac{1}{r^2}\frac{\Theta''(\theta)}{\theta} = -\lambda.$

$r^2\frac{R''}{R} + r\frac{R'}{R} + \frac{\Theta''(\theta)}{\theta} = -\lambda r^2$

We observe that $\frac{\Theta''(\theta)}{\theta}$ is a constant. Let this constant be $-n^2$:

$r^2\frac{R''}{R} + r\frac{R'}{R} + \lambda r^2 - n^2 = 0$

$R'' + \frac{1}{r}R' + (\lambda - \frac{n^2}{r^2})R = 0$

Now we scale this equation with $\rho = \sqrt{\lambda}r$. We observe that

$\frac{\partial R}{\partial r} = \frac{\partial R}{\partial \rho}\frac{\partial \rho}{\partial r} = \sqrt{\lambda}\frac{\partial R}{\partial \rho}$ and $\frac{\partial^2 R}{\partial r^2} = \lambda\frac{\partial^2 R}{\partial \rho^2}$, which implies that

$\lambda R_{\rho\rho} + \lambda\frac{1}{\rho}R_\rho + \lambda(1 - \frac{n^2}{\rho^2})R = 0$

$R_{\rho\rho} + \frac{1}{\rho}R_\rho + (1 - \frac{n^2}{\rho^2})R = 0$

This is Bessel's differential equation of order n. This implies that the solution is $R(\rho) = J_n(\rho) = J_n(\sqrt{\lambda}r)$. Applying boundary condition (2) we find that $J_n(\sqrt{\lambda}) = 0$. Therefore, $\sqrt{\lambda_{m,n}}$ of Δ in the unit disc with Dirichlet boundary conditions corresponds to the mth zero of J_n.

For this project, Bessel function zeros were calculated using Newton's method, and the derivative $J_n'(x)$ used in the iteration was calculated using the recurrence relation $J_n'(x) = -J_{n+1}(x) + \frac{n}{x}J_n(x)$. Finally, the eigenfunctions $u_{m,n}(r,\theta) = J_n(\sqrt{\lambda_{m,n}}r)\cos(n\theta)$ were plotted. A few such functions appear in the following figures.

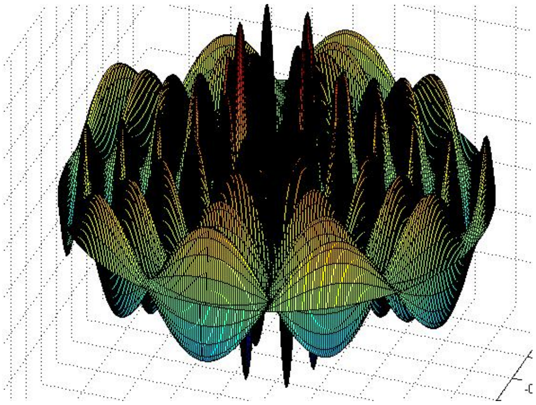

Figure 7: Eigenfunction $u_{10,5}$

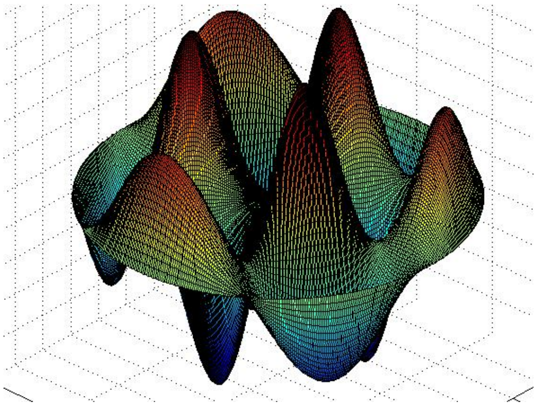

Figure 8: Eigenfunction $u_{2,3}$

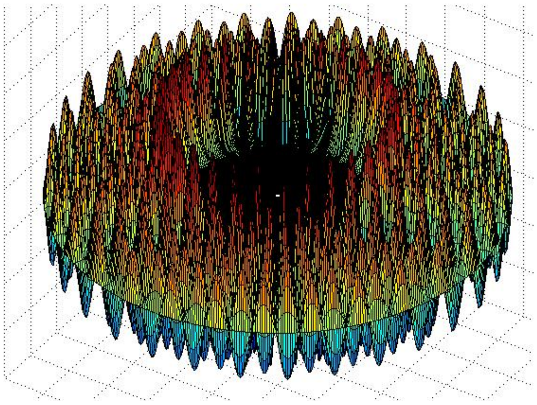

Figure 9: Eigenfunction $u_{10,30}$

References

[1] ”Chapter 10 Bessel Functions.” NIST Digital Library of Mathematical Functions. N.p., n.d. Web. 30 May 2014. <http://dlmf.nist.gov/10>.

[2] Press, William H. Numerical Recipes: The Art of Scientific Computing. Cambridge, UK: Cambridge UP, 2007. Print.