www.tubiad.org
ISSN:2148-3736
El-Cezerî Fen veMühendislikDergisi
Cilt: 4, No: 1, 2017 (32-45)

El-CezerîJournal of Science and Engineering
Vol: 4, No: 1, 2017 (32-45)

ECJSE

Research Paper / Makale

A Newton-Raphson Based Roots Finding Algorithm Design and its Applications to Circular Waveguides

Coşkun DENIZ[*]

Adnan Menderes University, Faculty of Engineering, Department of Electrical and Electronics Engineering, Aytepe Campus, 09010-Aydin/TURKEY, cdeniz@adu.edu.tr

Received/Geliş: 15.08.2016 **Revised/Düzeltme:** 12.12.2016 **Accepted/Kabul:** 21.12.2016

Abstract: Determination of zeros of first two kinds of Bessel functions and their derivatives by fast and reliable accurate calculations is essential to determine the necessaryTransverse Electric (TE) and Transverse Magnetic (TM) modes supported by the Circular Waveguides (CWGs). Here, a fast computational algorithm design based on the numerical Newton-Raphson (N-R) method to determine the first n zeros ofthese special functions is being presented. Our suggestion involves: scanning the given function in the selected domain according to the selectednumber of iteration steps (or, number of domain division) and finding their zeros by the N-R method in each step. One of the repeated roots and roots out of the domain are rejected and the remaining roots are re-sorted by the optional bubble sorting algorithm. Consequently, related TE and TM modes of the circular waveguide determining the electromagnetic wave behavior is obtained successfully. Our design running under the free "Wolfram CDF player" software has been opened to the users for free in the web page of our institution. Results of a sample application of our design to a specific CWG regarding these modes along with the cut-off frequencies and propagating electromagnetic wave frequencies are being presented.

Keywords: Bessel functions, Circular waveguides, Cylindrical waveguides, Lossless medium, TE modes, TM modes, Numerical root finding, Newton-Raphson method

Newton-Raphson Temelli Kökleri Bulma Algoritması Tasarımı ve Dairesel Dalgakılavuzlarına Uygulamaları

Özet: Dairesel dalgakılavuzlarının (DDK) desteklediği Transvers Elektrik (TE) ve Transvers Manyetik (TM) modların belirlenmesinde, ilk iki türden Bessel fonksiyonlarının ve türevlerinin sıfırlarının hızlı ve güvenilir doğruluklarda hesaplanarak tespiti elzemdir. Burada, bu özel fonksiyonlarınistenen aralıkta ilk n sıfırını bulan, Newton-Raphson (N-R) temelli, hızlı hesaplama yapabilen bir algoritma tasarımı sunulmaktadır. Önerimiz, fonksiyonun seçilen iterasyon adım sayısına göre (veya domain bölme sayısına göre), seçilen tanım aralığında taranmasınıve her adımda köklerin N-R yöntemiyle bulunmasını içermektedir. Bulunan tekrarlı köklerden biri ve girilen tanım aralığı dışında bulunan kökler atılmakta ve geriye kalan kökler, opsiyonel olarak konulan köpük (bubble) sıralama algoritmasına göre tekrardan sıralanmaktadır. Netice itibariyle, elektomanyetik dalga davranışlarını belirleyen, dairesel dalgakılavuzlarının TE ve TM modları başarıyla elde edilmektedir. Ücretsiz "Wolfram CDF player" altında çalışan tasarımımız, kurumumuzun ilgili internet adresinden kullanıcıların hizmetine ücretsiz olarak sunulmuştur. Tasarımımızın özel bir DDK'na örnek uygulamasının modlara, kesme frekansına ve ilerleyen dalga frekansına ilişkin sonuçları sunulmaktadır.

Anahtar kelimeler: Bessel fonksiyonları, Dairesel dalga klavuzları, Kayıpsız ortam, Silindirik dalga klavuzları, TE modlar, TM modlar, Numerik kök bulma, Newton-Raphson metodu

How to cite this article
Deniz, C., "*A Newton-Raphson Based Roost Finding Algorithm Design and its Applications to Circular Waveguides*" El-CezerîJournal of Science and Engineering, 2017, 4(1); 32-45.

1. Introduction

Special functions having infinite zeros in the entire domain implying finite zeros in a given subdomain; like trigonometric functions, Bessel functions, Airy functions, etc., have various fundamental applications in physical and engineering sciences where finding their zeros as accurate and as fast as possible in the desired domains is essential, i.e.; [1—14]. Since analytical or numerical solutions in finding roots of such functions are always not appropriate, various numerical and analytical approximation techniques have been improved [14—23]. For example, Bessel functions as we study here are defined exactly as infinite series:

$$J_m(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\Gamma(k+m+1)} \left(\frac{x}{2}\right)^{2k+m}, m\epsilon\square^+ \tag{1a}$$

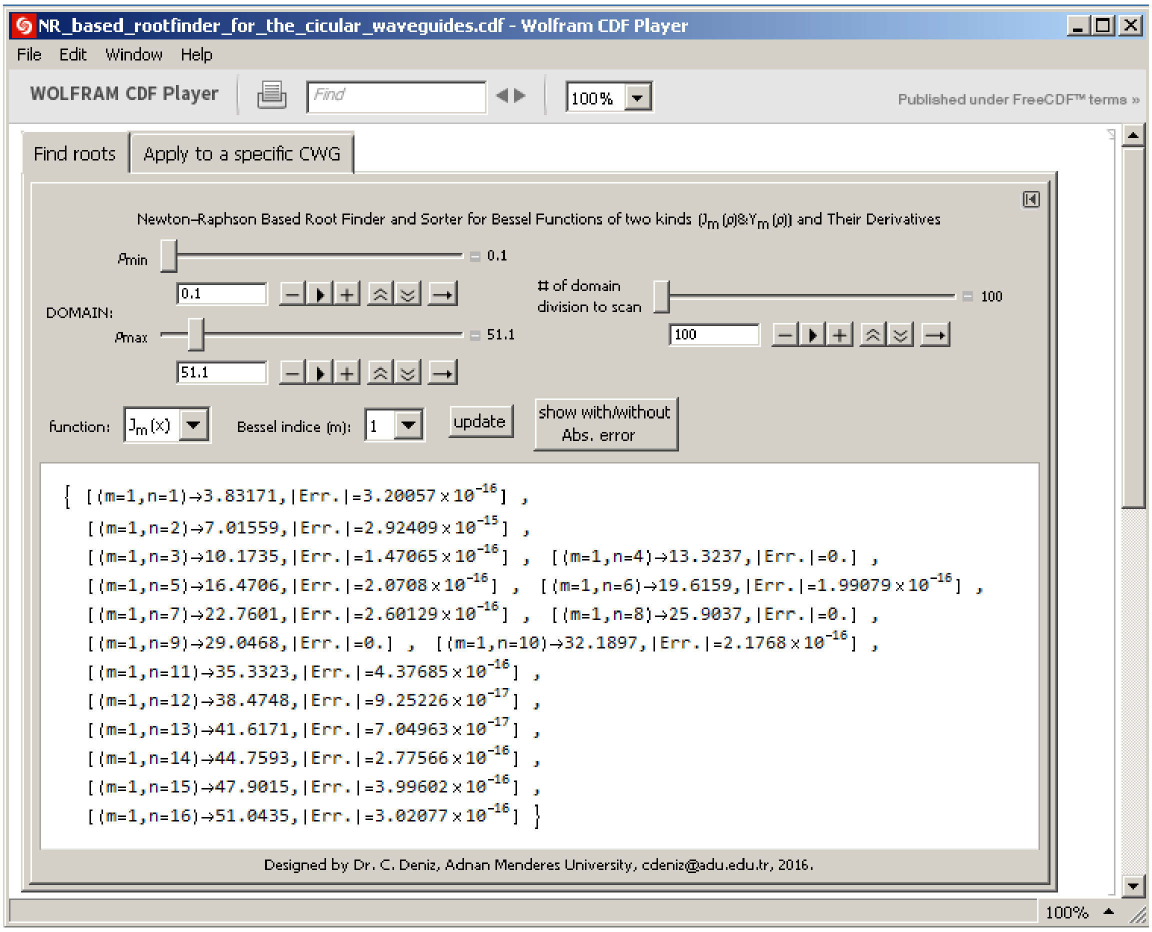
$$Y_m(x) = \begin{cases} \frac{J_m(x)\cos(m\pi) - J_{-m}(x)}{\sin(m\pi)}, \; if \; m\epsilon\square^+ \\ \lim_{p\to m}\left[\frac{J_m(x)\cos(p\pi) - J_{-m}(x)}{\sin(p\pi)}\right], if \; m\epsilon\square^+ \end{cases} \tag{1b}$$

where $J_m(x) \& Y_m(x)$ are the Bessel functions of first and second kind with index m and Γ is the gamma function. We can see that their exact numerical calculations as well as finding their zeros are impractical but they can be calculated by some approximations according to the situations such as asymptotic approximations, approximations for large Bessel indices, various numerical approximation techniques, etc., as given in fundamental textbooks [6—9,14—23].

Today, it is possible to retrieve nearly exact values of such special functions, their derivatives and zeros by entering very simple commands to the conventional mathematical and engineering software such as Mathematica, Maple, etc., whose principles are based on such improved approximation techniques [24—26]. Here we suggest a very fast and accurate numerical method based on the conventional Newton-Raphson (N-R) method given in [27—29] to find the zeros of the Bessel functions of the first two kinds and their derivatives in a desired domain. Our algorithm involves scanning these functions in the given domain with the given number of domain divisions (which also implies the iteration number for scanning the radius domain) and finding their zeros for each division by the numerical N-R method. One of the repeated roots are rejected as well as the roots out of the domain and the remaining roots are re-sorted by the optional bubble sorting algorithm given in [34—39].

A waveguide is a structure that guides waves, such as sound waves or electromagnetic waves where the latter is our case here. They enable a signal to propagate with minimal loss of energy by restricting expansion to one dimension or two. This is a similar effect to waves of water constrained within a canal. Without the physical constraint of a waveguide, signals will typically be radiated and decreased according to the inverse square law as they expand into three dimensional space. There are different types of waveguides for each type of wave. The original and most common is a hollow conductive metal pipe used to carry high frequency radio waves, particularly microwaves regarding the electromagnetic waveguides (EMWGs), or simply;| waveguides (WGs). It might also be filled with any medium or some combinations of media to form a dielectric waveguide. Our hallow-conductive metal pipe might be rectangular to be called a parallel plate waveguide (PPWG) or circular (cylindrical) to be called a circular (or cylindrical) waveguide (CWG) [9—14]. In EMWGs we have three common modes which determine the behavior of the electromagnetic wave to be guided: i) TE modes (Transverse Electric) where there is no electric field in the direction of propagation, ii) TM modes (Transverse Magnetic) where there is no magnetic field in the direction of propagation, iii) TEM modes (Transverse Electro Magnetic) where there is neither electric nor magnetic field in the direction of propagation. For the CWG only the first two of them are supported.

In CWG design and analyses, roots of such special functions are essential since they are used to define these modes to characterize the electromagnetic behavior of the CWG. Related electromagnetic wave parameters such as cut-off frequencies below which the wave can not propagate (severely attenuated), frequency bandwidth in which it can operate to transmit the electromagnetic wave, propagating wave frequencies, wavelengths and wave impedances (which give a relationship between the electric and magnetic components of the propagating electromagnetic wave), etc. are characterized by these modes [9—14]. We use only the first kinds of Bessel functions and their derivatives to find the related Transverse Magnetic (TM) and Transverse Electric (TE) modes supported by the CWG in the Electromagnetic Wave Theory, respectively [9—14]. Our design running under the free “Wolfram CDF player” software (where CDF stands for Computable Document Format [30—32]) is open to the users for free in the web page given in [33]. A general appearance of it while running is given in Figure 1 and Figure 2. Our design is made up of two parts: i) Finding roots: to find the roots of these special functions and their derivatives by the suggested algorithm (shown in Figure 1), ii) Apply to a specific CWG: Apply the roots of the required functions among them to determine the modes and find some of the electromagnetic parameters of a specific CWGwhose parameters are selected by the user (shown in Figure 2).



NR_based_rootfinder_for_the_cicular_waveguides.cdf - Wolfram CDF Player
File
Edit
Window
Help
WOLFRAM CDF Player
Find
100%
Published under FreeCDF™ terms »
Find roots
Apply to a specific CWG
update
show with/without Abs. error
100%

Newton-Raphson Based Root Finder and Sorter for Bessel Functions of two kinds ($J_m(\rho)$&$Y_m(\rho)$) and Their Derivatives

DOMAIN: ρmin 0.1; ρmax 51.1

# of domain division to scan 100

function: $J_m(x)$ Bessel indice (m): 1

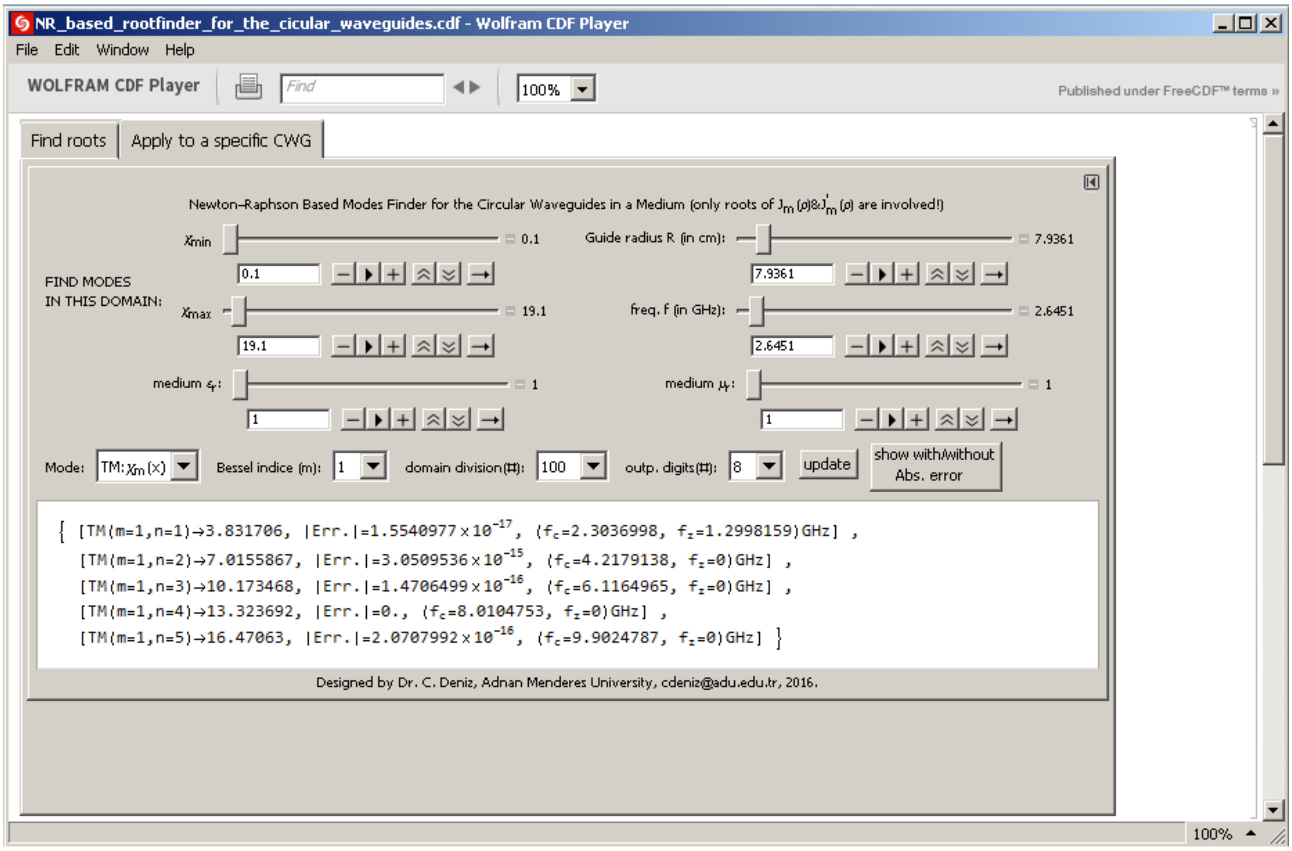
```
{ [(m=1,n=1)→3.83171,|Err.|=3.20057 x 10^-16] ,
  [(m=1,n=2)→7.01559,|Err.|=2.92409 x 10^-15] ,
  [(m=1,n=3)→10.1735,|Err.|=1.47065 x 10^-16] ,  [(m=1,n=4)→13.3237,|Err.|=0.] ,
  [(m=1,n=5)→16.4706,|Err.|=2.0708 x 10^-16] ,  [(m=1,n=6)→19.6159,|Err.|=1.99079 x 10^-16] ,
  [(m=1,n=7)→22.7601,|Err.|=2.60129 x 10^-16] ,  [(m=1,n=8)→25.9037,|Err.|=0.] ,
  [(m=1,n=9)→29.0468,|Err.|=0.] ,  [(m=1,n=10)→32.1897,|Err.|=2.1768 x 10^-16] ,
  [(m=1,n=11)→35.3323,|Err.|=4.37685 x 10^-16] ,
  [(m=1,n=12)→38.4748,|Err.|=9.25226 x 10^-17] ,
  [(m=1,n=13)→41.6171,|Err.|=7.04963 x 10^-17] ,
  [(m=1,n=14)→44.7593,|Err.|=2.77566 x 10^-16] ,
  [(m=1,n=15)→47.9015,|Err.|=3.99602 x 10^-16] ,
  [(m=1,n=16)→51.0435,|Err.|=3.02077 x 10^-16] }
```

Designed by Dr. C. Deniz, Adnan Menderes University, cdeniz@adu.edu.tr, 2016.

Figure 1. An appearance of the “Find roots” part of the designed software while running (“show with error” is activated here)

Since finding these modes are essential, we find them first and apply to a specific CWG to show its success as a sample application in the second part of our design. Here, parameters of a specific CWG regarding guide radius, guide permittivity and permeability (we assume lossless media) and applied wave frequency are selected by the user via the user control panel as shown in Figure 2. Then, it calculates the related cut-off and propagating wave frequencies among these electromagnetic parameters. We intentionally calculate only these two essential parameters for simplification here to show its success with a sample application, however, other parameters can be added when required.

In Section 2, we present and discuss the fundamentals of the circular waveguides.In Section 3, we present the designed algorithm to find and sort the zeros of these special functions and their derivatives. In Section 4 we present the results of sample applications of our design which involves a specific CWG where cut off frequencies and propagating wave frequencies for the associated modes are calculated.


NR_based_rootfinder_for_the_cicular_waveguides.cdf - Wolfram CDF Player
File
Edit
Window
Help
WOLFRAM CDF Player
Find
100%
Published under FreeCDF™ terms »
Find roots
Apply to a specific CWG
update
show with/without Abs. error
100%

Newton-Raphson Based Modes Finder for the Circular Waveguides in a Medium (only roots of $J_m(\rho)$&$J'_m(\rho)$ are involved!)

FIND MODES IN THIS DOMAIN: x_{min} 0.1; x_{max} 19.1

Guide radius R (in cm): 7.9361

freq. f (in GHz): 2.6451

medium ε_r: 1

medium μ_r: 1

Mode: TM: $x_m(x)$ Bessel indice (m): 1 domain division(#): 100 outp. digits(#): 8

{ [TM(m=1,n=1)→3.831706, |Err.|=1.5540977×10^{-17}, (f_c=2.3036998, f_z=1.2998159)GHz] ,
[TM(m=1,n=2)→7.0155867, |Err.|=3.0509536×10^{-15}, (f_c=4.2179138, f_z=0)GHz] ,
[TM(m=1,n=3)→10.173468, |Err.|=1.4706499×10^{-16}, (f_c=6.1164965, f_z=0)GHz] ,
[TM(m=1,n=4)→13.323692, |Err.|=0., (f_c=8.0104753, f_z=0)GHz] ,
[TM(m=1,n=5)→16.47063, |Err.|=2.0707992×10^{-16}, (f_c=9.9024787, f_z=0)GHz] }

Designed by Dr. C. Deniz, Adnan Menderes University, cdeniz@adu.edu.tr, 2016.

Figure 2. An appearance of the “Apply to a specific CWG” part of the designed software while running (“show with error” is activated here)

2. Application to Circular Waveguides in Electromagnetic Wave Theory

The circular waveguide is occasionally used as an alternative to the rectangular waveguide. Like other waveguides constructed from a single, enclosed conductor, the circular waveguide supports theTransverse Electric (TE) and Transverse Magnetic (TM) modes only. Field configurations of some low lying TE^z and/or TM^z modes in a circular waveguide are given as an example in Figure 3.

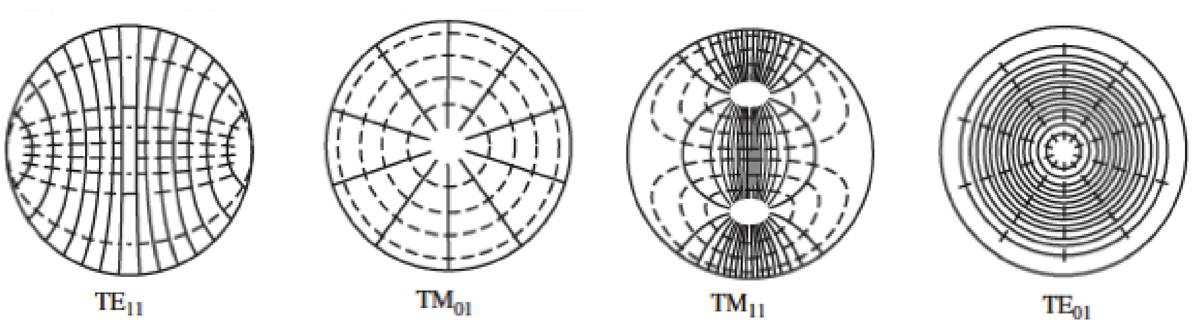


Figure 3: Field configurations of some low lyingTE^z and/or TM^z modes in a circular waveguide (Solid curves: for the electric field component, dashed curves: for the magnetic field components. Figures are from [12, p. 492])

Transverse Electric and Magnetic (TEM) modes supported by some other kinds of waveguides such as coaxial or parallel plate waveguides are not supported by the circular waveguides [12—14]. In other words, TEM does not exist in the CWG as in common to all one-conductor-type EMWGs [12—14]. By saying TE or TM mode, transverse to z-direction (TE^z or TM^z) is conventionally implied, as we follow here. If any other direction in which the electric or magnetic field component of the electromagnetic wave were transverse, it would be specifically defined, such as TE^x and TM^x for TE and TM modes transverse to the x-direction [12].

These modes have characteristic cut-off frequencies, below which electromagnetic energy is severely attenuated. Circular waveguide's round cross section makes it easy to machine, and it is often used to feed conical horns. Further, the TE_{0n} modes of circular waveguide have very low attenuation. General properties of the TE and TM modes can be summarized as follows [9—14]:

2.1. TE modes

The transverse electric to z (TE^z) modes can be derived by letting the vector potential $\boldsymbol{A}$ and $\boldsymbol{F}$ be equal to the followings:

$$\boldsymbol{A} = 0 \tag{2a}$$

$$\boldsymbol{F} = \boldsymbol{a_z} F_z(\rho, \emptyset, z) \tag{2b}$$

from which we have

$$\nabla^2 F_z(\rho, \emptyset, z) + \beta^2 F_z(\rho, \emptyset, z) = 0 \tag{3a}$$

whose solution gives:

$$F_z(\rho, \emptyset, z) = \left[A_1 J_m(\beta_\rho \rho) + B_1 Y_m(\beta_\rho \rho)\right] \times [C_2 \cos(m\emptyset) + D_2 \sin(m\emptyset)] \times [A_3 e^{-j\beta_z z} + B_3 e^{j\beta_z z}] \tag{3b}$$

where

$$\beta_\rho^2 + \beta_z^2 = \beta^2 \tag{4}$$

and J_m&Y_m are the Bessel functions of first and second kind respectively. The constants $\{A_1, B_1, C_2, D_2, A_3, B_3, m, \beta_\rho, \beta_z\}$ can be calculated by using the following boundary conditions:

$$E_\emptyset(\rho = a, \emptyset, z) = 0 \tag{5a}$$

$$E_z(\rho = a, \emptyset, z) = 0 \tag{5b}$$

from which we get

$$F_z(\rho, \emptyset, z) = A_{mn} J_m(\beta_\rho \rho) \times [C_2 \cos(m\emptyset) + D_2 \sin(m\emptyset)] \times A_3 e^{-j\beta_z z} \tag{6}$$

Then, the electric field component $E_\emptyset^+$ can be calculated from

$$E_\emptyset^+(\rho, \emptyset, z) = \frac{1}{\varepsilon} \frac{\partial F_z^+(\rho, \emptyset, z)}{\partial \rho} \tag{7a}$$

and by applying the boundary condition for $E_\emptyset^+$ in (5a), we get:

$$E_{\emptyset}^{+}(\rho = a, \emptyset, z) = 0 \Rightarrow J'_{m}(\beta_\rho) = 0 \Rightarrow \beta_\rho = \frac{\chi'_{mn}}{a} \tag{7b}$$

where χ'_{mn} represents the nth zero ($n = 1, 2, 3, ...$) of the derivative of the Bessel function $J_m(x)$ of the first kind of order m ($m = 0, 1, 2, 3, ...$). The smallest value of χ'_{mn} corresponds to $\chi'_{11} = 1.8412$ ($m = 1, n = 1$).

Using (4) and (7b), β_z of the mn mode can be written as follows:

$$(\beta_z)_{mn} = \begin{cases} \sqrt{\beta^2 - \beta_\rho^2} = \sqrt{\beta^2 - \left(\frac{\chi'_{mn}}{a}\right)^2}, \beta > \beta_\rho = \frac{\chi'_{mn}}{a} \\ 0, \qquad \beta = \beta_c = \beta_\rho = \frac{\chi'_{mn}}{a} \\ -j\sqrt{\beta_\rho^2 - \beta^2} = j\sqrt{\left(\frac{\chi'_{mn}}{a}\right)^2 - \beta^2}, \beta < \beta_\rho = \frac{\chi'_{mn}}{a} \end{cases} \tag{8a}$$

where Cut-off is defined when $\beta_{z(mn)} = 0$, namely:

$$\beta_c = \omega_c\sqrt{\mu\varepsilon} \Rightarrow (f_c)_{mn} = \frac{\chi'_{mn}}{2\pi a\sqrt{\mu\varepsilon}} \tag{8b}$$

where $(f_c)_{mn}$ is the cut-off frequency above which the related TE mode propogates with the guide wavelength:

$$\lambda_g = \frac{2\pi}{(\beta_z)_{mn}} \tag{8c}$$

2.2. TM modes

Similarly, the transverse magnetic to z (TM^z) modes can be derived by letting the vector potential $\boldsymbol{A}$ and $\boldsymbol{F}$ be equal to the followings:

$$\boldsymbol{F} = 0 \tag{9a}$$

$$\boldsymbol{A} = \boldsymbol{a_z}A_z(\rho, \emptyset, z) \tag{9b}$$

from which we have

$$\nabla^2 A_z(\rho, \emptyset, z) + \beta^2 A_z(\rho, \emptyset, z) = 0 \tag{10a}$$

whose solution gives:

$$A_z(\rho, \emptyset, z) = \left[A_1 J_m(\beta_\rho\rho) + B_1 Y_m(\beta_\rho\rho)\right] \times [C_2\cos(m\emptyset) + D_2\sin(m\emptyset)] \times [A_3 e^{-j\beta_z z} + B_3 e^{j\beta_z z}] \tag{10b}$$

where

$$\beta_\rho^2 + \beta_z^2 = \beta^2 \tag{11}$$

and J_m&Y_m are the Bessel functions of first and second kind respectively. The constants $\{A_1, B_1, C_2, D_2, A_3, B_3, m, \beta_\rho, \beta_z\}$ can be calculated by using the following boundary conditions:

$$E_{\emptyset}(\rho = a, \emptyset, z) = 0 \tag{12a}$$

$$E_z(\rho = a, \emptyset, z) = 0 \tag{12b}$$

from which we get

$$A_z^{+}(\rho, \emptyset, z) = B_{mn} J_m(\beta_\rho\rho) \times [C_2\cos(m\emptyset) + D_2\sin(m\emptyset)] \times A_3 e^{-j\beta_z z} \tag{13a}$$

Then, the electric field component E_z^{+} can be calculated from

$$E_z^{+}(\rho, \emptyset, z) = -j\frac{1}{\omega\mu\varepsilon}\left(\frac{\partial^2}{\partial\rho^2} + \beta^2\right)A_z^{+}(\rho, \emptyset, z) \tag{13b}$$

and by applying the boundary condition in (12b) to (13b), we get

$$E_z(\rho = a, \emptyset, z) = 0 \Rightarrow J_m(\beta_\rho) = 0 \Rightarrow \beta_\rho = \frac{\chi_{mn}}{a} \tag{13c}$$

where χ_{mn} represents the nth zero ($n = 1, 2, 3, ...$) of the Bessel function $J_m(x)$ of the first kind of order m ($m = 0, 1, 2, 3, ...$). The smallest value of χ_{mn} corresponds to $\chi_{01} = 2.4049$ ($m = 0, n = 1$).

Using (13c) and (11), β_z of the mn mode can be written as follows:

$$(\beta_z)_{mn} = \begin{cases} \sqrt{\beta^2 - \beta_\rho^2} = \sqrt{\beta^2 - \left(\frac{\chi_{mn}}{a}\right)^2}, \beta > \beta_\rho = \frac{\chi_{mn}}{a} \\ 0, \quad \beta = \beta_c = \beta_\rho = \frac{\chi_{mn}}{a} \\ -j\sqrt{\beta_\rho^2 - \beta^2} = j\sqrt{\left(\frac{\chi_{mn}}{a}\right)^2 - \beta^2}, \beta < \beta_\rho = \frac{\chi_{mn}}{a} \end{cases} \tag{14}$$

where Cutoff is defined when $\beta_{z(mn)} = 0$, namely:

$$\beta_c = \omega_c\sqrt{\mu\varepsilon} \Rightarrow (f_c)_{mn} = \frac{\chi_{mn}}{2\pi a\sqrt{\mu\varepsilon}} \tag{15}$$

where $(f_c)_{mn}$ is the cut-off frequency above which the related TM mode propogates with the guide wavelength given in eq. (8c). Since the cutoff frequencies of the TE_{0n} and TM_{1n} modes are identical ($\chi'_{0n} = \chi_{1n}$); they are referred to also as degenerate modes.

3. Finding Oscillatory Zeros via Newton-Raphson (N-R) Method in a Given Domain

3.1. The Newton-Raphson (N-R) method

In numerical analysis, Newton's method (also known as the Newton–Raphson method), named after Isaac Newton and Joseph Raphson, is a method for finding successively better approximations to the roots (or zeroes) of a real-valued functions:

$$x : f\ (x) = 0 \tag{16}$$

The Newton–Raphson method in one variable is implemented as follows [27—29]:
The method starts with a function f defined over the real numbers x, the function's derivative f', and an initial guess x_0 for a root of the function f. If the function satisfies the assumptions made in the derivation of the formula and the initial guess is close, then a better approximation x_1 is

$$x_1 = x_0 - \frac{f(x_0)}{f\prime(x_0)} \tag{17}$$

Geometrically, $(x_1, 0)$ is the intersection of the x-axis and the tangent of the graph of $f$at $(x_0, f(x_0)$. The process is repeated as

$$x_{n+1} = x_n - \frac{f(x_n)}{f\prime(x_n)} \tag{18}$$

until a sufficiently accurate value is reached. This algorithm is first in the class of Householder's methods, succeeded by Halley's method. The method can also be extended to complex functions and to systems of equations. Today, it is very easy to find the zero of such functions, $f(x)$, around x_0 by applying the Newton-Raphson method in any computational program such as mathematica with the following simple command [24]:

$$FindRoot[f, \{x, x_0\}, Method -> \{"Newton", "StepControl" -> None\}] \tag{19a}$$

Here, mathematica uses Newton-Raphson method starting from x_0 and finds the nearest zero to it. Or, alternatively [24];

$$FindRoot[f, \{x, x_0, x_{min}, x_{max}\} \tag{19b}$$

finds roots between x_{min} and x_{max}starting from x_0. Since the default method in the simpest case in mathematica is Newton-Raphson Method, method option may not be specified in these commands when used in “FindRoot“ given in (19a)—(19b) where iteration continues until $f'(x_n)$ in (18) vanishes to the default “working precision value of Mathematica. “Working precision” value can also be set to any specific value along with the other optional parameters like “accuracy goal”and “precision goal” if desired [24].

3.2. The Bubble Sorting

Bubble sorting, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted. Each pair of adjacent items in the list are compared and swapped if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list. Main algorithm of the conventional Bubble sorting given below is widely studied in the literature(also including introductory textbooks) [34—36]:

```
begin BubbleSort(list)
  for all elements of list
    if list[i] > list[i+1]
      swap(list[i], list[i+1])
    end if
  end for
  return list
end BubbleSort
```

Although the algorithm is simple and popular, it is normally known to be too slow and impractical for most problems involving large datum to be sorted even when compared to the insertion sorting algorithm [34—36]. It is practical if the element number is not too large as our case here. We also prefer it here since it needs tiny code size and requires relatively little memory. Advantages and disadvantages of the bubble sorting is well analyzed in details in [36].

3.3. Extensions of the N-R method to find roots in a given domain

Newton-Raphson method given in Section 3.1 above finds only one root around $\mathbf{x_0}$. So, we can extend it to find zeros in a given domain $[\boldsymbol{a}, \boldsymbol{b}]$ by the following designed algorithm:

```
Step 1: Set precision: $m = 0.01$and input the followings:
function: $f$, domain: $[x_{min}, x_{max}]$, domain division number: $n$
Step 2: For $i = 1$ to $n$
            $\Delta x = \frac{x_{max}-x_{min}}{n}$
                $y_i$ ←Find root by the N-R Method $\{f, x_{0i} \leftarrow i \times \Delta x\}$
                    If $x_{min} \le x_i \le x_{max}$ then $x_i \leftarrow y_i$
                    Else $x_i \leftarrow \{\}$
                    End if
  End For
Step 3: Remove one of the repeated roots out the precision $m$.
Step 4: Apply bubble sorting (optional)
```

Here in Step 1, the function $\boldsymbol{f}$, domain: $[\boldsymbol{x_{min}}, \boldsymbol{x_{max}}]$, number of domain division (or, scanning iteration number): $\boldsymbol{n}$ and working precision: $\boldsymbol{m}$ is inserted by the user. In Step 2, we scan the given function $\boldsymbol{f}$, in the given domain: $[\boldsymbol{x_{min}}, \boldsymbol{x_{max}}]$ with scanning step number: $\boldsymbol{n}$. In each step with step number $\boldsymbol{i}$, we find the $\boldsymbol{i}$th zero $\boldsymbol{x_i}$ around $\boldsymbol{x_{0i}} \leftarrow \boldsymbol{i} \times \Delta \boldsymbol{x} = \boldsymbol{i} \times \frac{x_{max}-x_{min}}{n}$ by the Newton-Raphson method and assign it to $\boldsymbol{x_i}$ if the root is in the domain $\boldsymbol{x_{min}} \le \boldsymbol{x_i} \le \boldsymbol{x_{max}}$. In Step 3, we remove the repeated roots out the precision $\boldsymbol{m}$. Since different initial guess values (=$\boldsymbol{x_{0i}}$) in each iteration in step 2 can give very close values in our N-R method, we have chosen $\boldsymbol{m} = \boldsymbol{0.01}$ in our calculationsto distinguish them here. As an example, if two roots $\boldsymbol{x_i}$&$\boldsymbol{x_{i'}}$ are very close to each other with satisfying the condition:$|\boldsymbol{x_i} - \boldsymbol{x_{i'}}| < m$, it assumes that roots $\boldsymbol{x_i}$ and $\boldsymbol{x_{i'}}$ with $\boldsymbol{i} \ne \boldsymbol{i'}$ are repeated and one of them is rejected. In Step 3, we sort the roots by the conventionalBubble sorting given in[34—36].This part is optional since we scan the domain in the ascending order, however, it guarantees the roots found to be sorted when running with very small or large domain division values where some of the roots might not be found in the ascending order.

4.Results

4.1. Zeros of Trigonometric functions

Our suggestion has been tested first for the trigonometric functions with

$$f(x) = Sinx \tag{20a}$$

whose exact zeros are:

$$x_n = n\pi, n = 0, \pm1, \pm2, \ldots \tag{20b}$$

These results have also been obtained numerically by our algorithm. Some of these results for the Sine function is given in Table 1 as follows:

Table 1. Results for $f = Sinx, x_{min} = -13,\ x_{max} = 13, n = 100, m = 0.01$

Root # (n)	1	2	3	4	5	6	7	8	9
Exact	-12.5664	-9.42478	-6.28319	-3.14159	0	3.14159	6.28319	9.42478	12.5664
Our algorithm	-12.5664	-9.42478	-6.28319	-3.14159	0	3.14159	6.28319	9.42478	12.5664
Abs. Error	0	0	0	0	0	0	0	0	0

Similarly, we obtained the numerical values of zeros of the cosine function:

$$f(x) = Cosx \Rightarrow x_n = \left(n + \frac{1}{2}\right)\pi, n = 0, \pm1, \pm2, \ldots \tag{21}$$

In both cases we obtain the zeros with zero error.

4.2. Zeros of Bessel functions and their derivatives

Results for Bessel functions of first two kinds and their derivatives for various index values are given in Table 2a—2d as follows:

Table 2a.Results for $f = J_\nu(x), x_{min} = 0,\ x_{max} = 53, n = 100, m = 0.01$

ν n	**0**	**1**	**2**	**3**	**4**
1	2.404825558	3.83170597	5.135622302	6.380161896	7.588342435
2	5.52007811	7.01558667	8.41724414	9.76102313	11.06470949
3	8.653727913	10.17346814	11.61984117	13.01520072	14.37253667
4	11.79153444	13.32369194	14.79595178	16.22346616	17.61596605
5	14.93091771	16.47063005	17.95981949	19.40941523	20.82693296
6	18.07106397	19.61585851	21.11699705	22.58272959	24.01901952
7	21.21163663	22.76008438	24.27011231	25.7481667	27.19908777
8	24.35247153	25.90367209	27.42057355	28.90835078	30.37100767
9	27.49347913	29.04682853	30.5692045	32.06485241	33.53713771
10	30.63460647	32.18967991	33.71651951	35.21867074	36.69900113
11	33.77582021	35.33230755	36.86285651	38.37047243	39.8576273
12	36.91709835	38.47476623	40.00844673	41.52071967	43.01373772
13	40.05842576	41.61709421	43.15345378	44.66974312	46.16785351
14	43.19979171	44.759319	46.29799668	47.81778569	49.32036069
15	46.34118837	47.90146089	49.44216411	50.96502991	52.4715514

Table 2b. Results for $f = J_\nu'(x), x_{min} = 0,\ x_{max} = 53, n = 100, m = 0.01$

ν n	0	1	2	3	4
1	3.83170597	1.841183781	3.054236928	4.201188941	5.317553126
2	7.01558667	5.331442774	6.706133194	8.015236598	9.282396285
3	10.17346814	8.536316366	9.969467823	11.34592431	12.68190844
4	13.32369194	11.7060049	13.17037086	14.58584829	15.96410704
5	16.47063005	14.86358863	16.34752232	17.78874787	19.1960288
6	19.61585851	18.01552786	19.51291278	20.97247694	22.40103227
7	22.76008438	21.16436986	22.67158177	24.14489743	25.58975968
8	25.90367209	24.31132686	25.82603714	27.31005793	28.76783622
9	29.04682853	27.45705057	28.97767277	30.47026881	31.93853934
10	32.18967991	30.60192297	32.12732702	33.62694918	35.10391668
11	35.33230755	33.7461829	35.27553505	36.78102068	38.26531699
12	38.47476623	36.88998741	38.42265482	39.93310862	41.4236665
13	41.61709421	40.03344405	41.56893494	43.08365266	44.57962314
14	44.759319	43.17662897	44.71455353	46.23297108	47.73366752
15	47.90146089	46.31959756	47.85964161	49.38130009	50.88615915

Table 2c. Results for $f = Y_\nu(x), x_{min} = 0,\ x_{max} = 53, n = 100, m = 0.01$

ν n	0	1	2	3	4
1	0.893576966	2.197141326	3.384241767	4.527024661	5.645147894
2	3.957678419	5.429681041	6.793807513	8.097553763	9.361620615
3	7.08605106	8.596005868	10.02347798	11.39646674	12.73014447
4	10.22234504	11.74915483	13.20998671	14.62307774	15.99962709
5	13.36109747	14.89744213	16.37896656	17.81845523	19.22442896
6	16.50092244	18.04340228	19.53903999	20.99728475	22.4248106
7	19.6413097	21.18806893	22.69395594	24.16623576	25.61026705
8	22.78202805	24.33194257	25.84561372	27.32879985	28.78589366
9	25.92295765	27.47529498	28.9950804	30.4869896	31.95468668
10	29.06403025	30.61828649	32.14300226	33.64204938	35.11852953
11	32.20520412	33.7610178	35.28979387	36.79479103	38.27866809
12	35.34645231	36.90355532	38.43573349	39.94576723	41.43596063
13	38.48775665	40.04594464	41.58101487	43.09536751	44.59101823
14	41.62910447	43.1882181	44.72577712	46.24387443	47.74428809
15	44.77048661	46.33039925	47.8701227	49.39149802	50.8961052

Table 2d. Results for $f = Y_\nu'(x), x_{min} = 0,\ x_{max} = 53, n = 100, m = 0.01$

ν n	0	1	2	3	4
1	2.197141326	3.683022857	5.002582931	6.253633208	7.464921737
2	5.429681041	6.941499954	8.350724701	9.698787984	11.00516915
3	8.596005868	10.12340466	11.57419547	12.97240905	14.33172352
4	11.74915483	13.28575816	14.76090931	16.1904472	17.58443602
5	14.89744213	16.44005801	17.93128594	19.38238845	20.80106234
6	18.04340228	19.59024176	21.0928945	22.55979186	23.99700412
7	21.18806893	22.73803472	24.24923168	25.72821319	27.17988669
8	24.33194257	25.88431462	27.40214584	28.89067842	30.35396061
9	27.47529498	29.02957582	30.55270888	32.04898401	33.5217971
10	30.61828649	32.17411823	33.70158627	35.20426661	36.68504838
11	33.7610178	35.31813446	36.84921342	38.35728168	39.84482697
12	36.90355532	38.46175387	39.99588738	41.50855144	43.00191052
13	40.04594464	41.60506662	43.14181784	44.65844873	46.15685955
14	43.1882181	44.74813745	46.2871571	47.80724696	49.31008861
15	46.33039925	47.89101407	49.43201847	50.95515126	52.46191104

Note that we preferred to use symbol ρ for the domain of the Bessel functions rather than x in our design in [33], since cylindrical coordinates are essential in the CWGs which we use in the second part of the user console.

4.3. Applications to Circular Waveguides

Table 3a.Results of the TM modes for application to a specific CWG:$\chi_{min} = 0, \chi_{max} = 13, domain\ div.: n = 100, outp.digits = 10, R = 5cm, \varepsilon_r = \mu_r = 1$, and $f = 5GHz$.

$\boldsymbol{\nu}$ \ $\boldsymbol{n}$	CWG	**0**	**1**	**2**	**3**	**4**
1	$\chi'_{\nu n}$	2.404825558	3.83170597	5.135622302	6.380161896	7.588342435
	f_c	2.294850557	3.656478347	4.900765322	6.088390916	7.241320189
	f_z	4.442258538	3.410302934	0.991211007	0	0
2	$\chi'_{\nu n}$	5.52007811	7.01558667	8.41724414	9.76102313	11.06470949
	f_c	5.267639594	6.694757099	8.032315417	9.314642092	10.55870961
	f_z	0	0	0	0	0
3	$\chi'_{\nu n}$	8.653727913	10.17346814	11.61984117	13.01520072	14.37253667
	f_c	8.257984558	9.708225588	11.08845459	12.42000299	13.71526666
	f_z	0	0	0	0	0

Table 3b. Results of the TE modes for application to a specific CWG:$\chi_{min} = 0,\ \chi_{max} = 13, domain\ div.: n = 100, outp.digits = 10, R = 5cm,\ \varepsilon_r = \mu_r = 1$, and $f = 5GHz$.

$\boldsymbol{\nu}$ \ $\boldsymbol{n}$	CWG	**0**	**1**	**2**	**3**	**4**
1	χ_{vn}	3.8170597	1.841183781	3.054236928	4.201188941	5.317553126
	f_c	3.656478347	1.756984665	2.914563717	4.009064504	5.074376274
	f_z	3.410302934	4.681132864	4.062673792	2.987875802	0
2	χ_{vn}	7.01558667	5.331442774	6.706133194	8.015236598	9.282396285
	f_c	6.694757099	5.087630734	6.399455231	7.648692069	8.857903317
	f_z	0	0	0	0	0
3	χ_{vn}	10.17346814	8.536316366	9.969467823	11.34592431	12.68190844
	f_c	9.708225588	8.145942355	9.513554408	10.82706421	12.10195249
	f_z	0	0	0	0	0

5. Conclusions

We calculated zeros of the Bessel functions of first and second kinds with integer indices and their derivatives successfully by the suggested algorithm based on the conventional N-R root finding.In the first part of the user console of our design running in Mathematica cdf player (The “find roots” part), user selects the Bessel function of either kind (among the first two kinds and their derivatives) and selects its index value. Selected function is being scanned by the iteration number (which is equal to the number of domain division also selected by the user) through the domain (also selected by the user). After the application of the N-R root-finding for each division has been completed, repeated roots and roots out of the domain are rejected. We set the distinguish parameter by m=0.001 as optimum to decide such two close roots (as a consequence of the Newton-Raphson process with close initial guesses in each iteration step) to be repeated if they deviate more than this value as explained above. Finally, we sort the roots by the conventional Bubble sorting [34—36]. This final step is optional here since it becomes necessary for the number of division values which might be set to an inappropriate value causing some of the roots to be not in the correct ascending order. Since the domain is scanned from minimum to maximum in the ascending order, we normally get the roots in the ascending order. Actually, Bubble sorting has been known to be impractical when compared with the other sorting methods and there has been much effort for the enhancement of it, i.e., [37—39]. However, it works here very well without any modification since it needs little memory and we do not have too much roots to be processed to spend much time in sorting. It is obvious that, when this optional part is removed by using the appropriate domain division values, it becomes very fast even for too much root values to be processed. Moreover, by applying some optimization and enhancement to the bubble sorting, or by using other sorting algorithms, speed can be increased for huge root values when this optional part is used.

In the second part of the user console of our design, the TM and TE modes supported by the CWGs has been obtained successfully by using the results for the first kinds of Bessel functions and their derivatives obtained in the first part of the user console. User selects i) the CWG parameters: guide radius, wave frequency, relative permittivity and permeability of the guide medium (we study lossless and linear medium here), mode type to be calculated (TM or TE) and ii) Our roots finding parameter: Number of domain division. Output digit numbers to be displayed is also added for the cases where user wishes to round off the results to any specific digit number. In both parts of the user console, there is also a button with name "Show with/without Abs. Error" to activate or inactivate the appearance of the absolute error values calculated according to the following formula:

$$\boldsymbol{Abs.Err.} = \begin{cases} |J'_m(\chi'_{mn})|, \boldsymbol{for\ the\ TM\ modes} \\ |J_m(\chi_{mn})|, \boldsymbol{for\ the\ TE\ modes} \end{cases}, \boldsymbol{m = 0, 1, 2, \dots, n = 1, 2, 3, \dots} \quad (22)$$

which would give zero when really the exact values (which might have huge numbers of digits) were used. We apply the related formulas given in Section 2 to find the related electromagnetic parameters of the CWG under question. Results are in a great consistence with the results given in [12—14] where the Newton-Raphson method is used, namely with a very slight difference of about $\boldsymbol{\sim 10^{-16}}$ or less (even zero for some of the values) which is within the default working precision value of the Mathematica. Moreover, degenerate $\boldsymbol{TE_{0n}}$ and $\boldsymbol{TM_{1n}}$ modes where cutoff frequencies and wavelengths are common as given in [12—14] are also apparently seen in our results given in Table 2a and Table 2b with $\boldsymbol{\chi'_{0n} = \chi_{1n}}$. We also obtain $\boldsymbol{(f_c^{TM})_{0n} = (f_c^{TE})_{1n}, (f_z^{TM})_{0n} = (f_z^{TE})_{1n}}$ for the same selected wave frequency values in a specific CWG as shown in Table 3a and Table 3b.In Table 3a and 3b, results of the selected CWG parameters are given for the TE and TM modes with indices up to: $\boldsymbol{m_{max} = 4 \& n_{max} = 3}$. However, results for much more index values can be obtained in the selected intervals via the user console. Table 3a and Table 3b shows that, the selected CWG when used with the selected frequency value (=5GHz), modes with $\boldsymbol{f_z = 0}$ are not supported (wave can not propagate since severely attenuated) since the selected wave frequency is smaller than the cut-off frequency values of the related mode. For much detailed CWG analyses, other sorting algorithms are required to find the other electromagnetic parameters such as correct order of the all supported modes which will be necessary to determine the operating band-width of each mode. However, they can be calculated manually by using these results for the modes around the low lying dominant mode values as in [12—14]. Our design running under the free Wolfram cdf player, which is freely downloadable via [30] is freely open to the users via [33].

References

[1] Korenev, B.G.,"Bessel functions and their applications", Taylor and Francis, 2002, NY.

[2] Werner,A. and Eliezer C.J., "The Lengthening Pendulum",Journal of The Australian Mathematical Society (J. Aust. Math. Soc.), 1969, 9(3-4), pp.331—336.

[3] McMillan, M., Blasing, D., and Whitney, H.M., "Radial forcing and Edgar Allan Poe's lengthening pendulum," American Journal of Physics, 2013, 81(9), pp. 682–687.

[4] Asadi-Zeydabadi, M., "Bessel function and damped simple harmonic motion",Journal of Applied Mathematics and Physics (JAMP), 2014, 2, pp. 26—34.

[5] Bell, W.W., "Special functions for scientists and engineers", D. Van NostrandCompant Ltd., 1968, London, pp. 92—110.

[6] Boas, L.M., "Mathematical methods in the physical sciences", Wiley, 2006, 3rd ed., NY, pp. 587—606.

[7] Vallée,O. and Soares, M., “Airy functions and applications to physics”, Imperial College Press (distributed by World Scientific), 2004, NJ, pp. 115—176.

[8] Watson, G.N.,“A treatise on the theory of Bessel functions”, Cambridge University Press, 1995, 2nd ed., NY.

[9] ArfkenH.J. and Weber, G.B.,“Mathematical methods for physicists”, Elsevier Academic Press, 2005, 6th ed., pp. 675—686.

[10] Sekeljic, N., “Asymptotic expansion of Bessel functions; applications to electromagnetics”, Dynamics at the Horsetooth, 2010, Focussed Issue: Asymptotics and Perturbations 2A, pp.1—11.

[11] Niedziela, J.,“Bessel function and their applications”, Knoxville: University of Tennessee, 2008.

[12] Balanis, C., “Advanced engineering electromagnetics”, Wiley, 1989, 2nd ed., NY, pp. 483—500.

[13] Cheng, D. K., “Field and wave electromagnetics”, Addison-Wesley, 1989, 2nd ed., London, pp., 562—572.

[14] Beattie, C.L., “Table of first 700 zeros of Bessel functions—$J_l(x)$ and $J_l’(x)$”, Bell System Technical Journal, 1958, 37, pp. 689—697.

[15] MillaneR.P. and Eads, J.L.,“Polynomial approximations to Bessel functions”, IEEE Transactions On Antennas And Propagation, 2003, 51(6), pp. 1398—1400.

[16] Abuelma’atti, M.T., “Trigonometric approximations for some Bessel functions”, Active and Passive Elec. Comp., 1999, 22, pp. 75-85.

[17] Blachman, N. M. and Mousavinezhad, S. H. “Trigonometric approximations for Bessel functions”, EEE Transactions on Aerospace and Electronic Systems, 1986, AES-22(1): 2—7.

[18] Waldron, R. A. “Formulas for computation of approximate values of some Bessel functions”, Proceedings of the IEEE, 69, pp. 1686-1588, (1981).

[19] Luke, Y.L., “Approximation of special Functions”, Academic Press, 1975, NY.

[20] Newman, J.N., “Approximations for the Bessel and Struve functions”, Mathematics of Computation, 1984, 43(168), pp. 551—556.

[21] Harrison, J.,“Fast and accurate Bessel function computation”, Computer Arithmetic-proc. of 19th EEE symp. on Computer Arithmetic, Portland-Oregon, pp. 104—113, (2009).

[22] Vrahatism, M. N., Ragos, O., Zafiropoulos, F. A., Grapsa, T. N., “Locating and computing zeros of Airy functions”, Z. Angew. Math. Mech., 1996, 76 (7), pp. 419—422.

[23]Abramowitz A. andStegunI.A., “Handbook of mathematical functions, with formulas, Graphs, and Mathematical Tables”, 3rd printing with corrections, Vol. 55 of NBS Applied mathematics series, superintendent of documents, US Government Printing Office, 1965, Washington DC, pp. 355—479.

[24] Wolfram, S., “The mathematica book”, Wolfram Media Inc., 2003, 5th edition, USA, pp. 29—35&pp. 102—110.

[25] http://reference.wolfram.com/mathematica/ref/BesselJ.html

[26] Richards, D., “Advanced mathematical methods with maple”, Cambridge University Press, 2002, UK, pp. 325—331.

[27] Chapra, S. and Canale, R.,“Numerical methods for engineers”, WCB/McGraw-Hill, 2014, 7th Edition, NY, pp. 148—154.

[28] Hamming, R.W., “Numerical methods for scientists and engineers (Dover Books on Mathematics)”, Dover Publications, 1987, 2nd Revised ed., NY, pp. 68—72.

[29] Hoffman, J.D., “Numerical methods for engineers and scientists”, Marcel Dekkel, 2001, 2nd edition, NY-Basel, pp. 141—154.

[30] https://www.wolfram.com/cdf-player/

[31] https://en.wikipedia.org/wiki/Computable_Document_Format

[32] http://www.wolfram.com/events/siam-2016/files/CDF-4.pdf

[33] A Wofram-CDF circular waveguide analyzer design via finding roots of Bessel type special functions based on the Newton-Raphson method*(after downloaded and unzipped, it should be opened by the free wolfamcdf player, which is downloadable via [30])*

[34] https://en.wikipedia.org/wiki/Bubble_sort

[35] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C.,"Introduction to algorithms", MIT Press, 2009, 3rd ed., USA, p. 40.

[36] Astrachan, O.,"Bubble Sort: An archaeological algorithmic analysis", SIGCSE '03 Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, NY, pp. 1—5, (2003).

[37] Khairullah, Md., "Enhancing worst sorting algorithms", International Journal of Advanced Science and Technology, 2013, 56, pp. 13—26.

[38] Rohil, H. and Manisha, "Run time Bubble sort–An enhancement of Bubble sort", International Journal of Computer Trends and Technology (IJCTT), 2014,14(1), pp. 36—38.

[39] Arora, N., Kumar, S., and Tamta, V.K., "A novel sorting algorithm and comparison with Bubble sort and Insertion sort", International Journal of Computer Applications, 2012, 45(1): pp. 31—32.