

A Matlab toolbox for the computation of Bessel functions with large complex order

Zoïs MOITIER
IRMAR, University of Rennes 1

April-June 2016

Document last modification: April 4, 2022

Abstract

We code a Matlab toolbox to compute the Bessel and Hankel functions with large complex order. First, we will show which formulas we use, then how to use the toolbox, and some relative errors with Mathematica.

Contents

1	Formulas use	1
1.1	Airy type asymptotic expansions	1
1.2	Far from the turning point	3
1.3	Near the turning point	3
2	How to use the toolbox	3
3	Numerical validation	4

1 Formulas use

1.1 Airy type asymptotic expansions

We want to compute the Bessel function of the first kind $J_\nu(z)$, the Bessel function of the second kind $Y_\nu(z)$, the Hankel function of the first kind $H_\nu^{(1)}(z)$, and the Hankel function of the second kind $H_\nu^{(2)}(z)$, for ν a large complex number and z a complex number. Those functions are solutions of this differential equation:

$$z^2 w''(z) + zw'(z) + (z^2 - \nu^2)w(z) = 0.$$

If ν is a real and z is a complex, Matlab already has functions for this (`besselj`, `bessely` and `besselh`) so, in this case, we use the Matlab functions, but they don't work with complex number ν . For complex ν , we will use Airy type asymptotic expansion to compute J , Y , $H^{(1)}$ and $H^{(2)}$.

We start with the formulas find in [1], which are:

$$J_\nu(\nu y) = \left(\frac{4\zeta}{1-y^2} \right)^{1/4} \left(\frac{\text{Ai}(\nu^{2/3}\zeta)}{\nu^{1/3}} A_\nu(\zeta) + \frac{\text{Ai}'(\nu^{2/3}\zeta)}{\nu^{5/3}} B_\nu(\zeta) \right) \quad (1.1a)$$

$$Y_\nu(\nu y) = - \left(\frac{4\zeta}{1-y^2} \right)^{1/4} \left(\frac{\text{Bi}(\nu^{2/3}\zeta)}{\nu^{1/3}} A_\nu(\zeta) + \frac{\text{Bi}'(\nu^{2/3}\zeta)}{\nu^{5/3}} B_\nu(\zeta) \right) \quad (1.1b)$$

$$H_\nu^{(1)}(\nu y) = 2e^{-i\pi/3} \left(\frac{4\zeta}{1-y^2} \right)^{1/4} \left(\frac{\text{Ai}(j\nu^{2/3}\zeta)}{\nu^{1/3}} A_\nu(\zeta) + \frac{j \text{Ai}'(j\nu^{2/3}\zeta)}{\nu^{5/3}} B_\nu(\zeta) \right) \quad (1.1c)$$

$$H_\nu^{(2)}(\nu y) = 2e^{i\pi/3} \left(\frac{4\zeta}{1-y^2} \right)^{1/4} \left(\frac{\text{Ai}(\bar{j}\nu^{2/3}\zeta)}{\nu^{1/3}} A_\nu(\zeta) + \frac{\bar{j} \text{Ai}'(\bar{j}\nu^{2/3}\zeta)}{\nu^{5/3}} B_\nu(\zeta) \right) \quad (1.1d)$$

where $\nu \in \mathbb{C}$, $y \in \mathbb{C}^*$, $j = e^{2i\pi/3}$ and ζ is defined by

$$\zeta = \begin{cases} \left(\frac{3}{2} \right)^{2/3} \left(\log \left(\frac{1 + \sqrt{1-y^2}}{y} \right) - \sqrt{1-y^2} \right)^{2/3} & \text{if } |y| \leq 1 \\ - \left(\frac{3}{2} \right)^{2/3} \left(\sqrt{y^2-1} - \arccos \left(\frac{1}{y} \right) \right)^{2/3} & \text{if } |y| > 1 \end{cases}$$

where we use the principal value for the logarithm and the power functions. To compute $A_\nu(\zeta)$ and $B_\nu(\zeta)$, we use the following asymptotic expansions for $|\nu| \rightarrow +\infty$, uniformly with respect to y ,

$$A_\nu(\zeta) \sim \sum_{k=0}^{+\infty} \frac{a_k(\zeta)}{\nu^{2k}} \quad \text{and} \quad B_\nu(\zeta) \sim \sum_{k=0}^{+\infty} \frac{b_k(\zeta)}{\nu^{2k}}. \quad (1.2)$$

Where the sequences $(a_k(\zeta))_{k \in \mathbb{N}}$ and $(b_k(\zeta))_{k \in \mathbb{N}}$, are defined as

$$a_k(\zeta) = \sum_{s=0}^{2k} \mu_s \zeta^{-3s/2} u_{2k-s} \left((1-y^2)^{-1/2} \right), \quad (1.3a)$$

$$b_k(\zeta) = -\zeta^{1/2} \sum_{s=0}^{2k+1} \lambda_s \zeta^{-3s/2} u_{2k-s+1} \left((1-y^2)^{-1/2} \right). \quad (1.3b)$$

The sequences $(\lambda_s)_{s \in \mathbb{N}}$ et $(\mu_s)_{s \in \mathbb{N}}$ are defined by

$$\lambda_s = \frac{\Gamma(3s + \frac{1}{2})}{9^s \sqrt{\pi} \Gamma(2s+1)} \quad \text{and} \quad \mu_s = -\frac{2\Gamma(3s + \frac{3}{2})}{9^s \sqrt{\pi} (6s-1) \Gamma(2s+1)}$$

and the quantities $(u_k)_{k \in \mathbb{N}}$ are the Debye polynomial sequences defined by induction by

$$u_0(t) = 1$$

$$u_{k+1}(t) = \frac{1}{2} t^2 (1-t^2) u'_k(t) + \frac{1}{8} \int_0^t (1-5x^2) u_k(x) dx$$

with $t \in \mathbb{C}$.

1.2 Far from the turning point

Practically, the sequences $(\lambda_s)_{s \in \mathbb{N}}$ et $(\mu_s)_{s \in \mathbb{N}}$ are computed by induction from $\lambda_0 = 1$ through the relations

$$\mu_s = -\frac{6s+1}{6s-1}\lambda_s \quad \text{and} \quad \lambda_{s+1} = \frac{(6s+1)(6s+5)}{48(s+1)}\lambda_s.$$

The sums [Eq. \(1.2\)](#) converge rapidly, if y is not too close of 1, so we can only take the first three terms and have a very good accuracy. If $y \sim 1$, the sums [Eq. \(1.2\)](#) converge very slowly, and we need another method to compute $A_\nu(\zeta)$ and $B_\nu(\zeta)$.

1.3 Near the turning point

In this case we can't use the exact formulas [Eq. \(1.2\)](#) because of the too slow decay of the sum's coefficients. Instead, we develop in Taylor series the coefficients $(a_k(\zeta))_{k \in \mathbb{N}}$ and $(b_k(\zeta))_{k \in \mathbb{N}}$ as follows

$$a_k(\zeta) = \sum_{s=0}^{+\infty} a_{k,s} \eta^s \quad \text{and} \quad b_k(\zeta) = 2^{1/3} \sum_{s=0}^{+\infty} b_{k,s} \eta^s$$

where $\eta = 2^{-1/3}\zeta$. The way the sequences $(a_{k,s})_{(k,s) \in \mathbb{N}^2}$ and $(b_{k,s})_{(k,s) \in \mathbb{N}^2}$ can be computed is given in [\[2\]](#). But, we don't need to go far to have good accuracy and, in practice, we use the following expansions:

$$a_0(\zeta) = 1,$$

$$a_1(\zeta) = \frac{-1}{225} - \frac{71}{38500}\eta + \frac{82}{73125}\eta^2 + \frac{5246}{3898125}\eta^3 + \frac{185728}{478603125}\eta^4,$$

$$a_2(\zeta) = \frac{151439}{218295000} + \frac{68401}{147262500}\eta - \frac{1796498167}{4193689500000}\eta^2 - \frac{583721053}{830718281250}\eta^3,$$

$$a_3(\zeta) = \frac{-887278009}{2504935125000} - \frac{3032321618951}{9708942993750000}\eta + \frac{14632757586911}{39373025737500000}\eta^2,$$

$$b_0(\zeta) = 2^{1/3} \left(\frac{1}{70} + \frac{2}{225}\eta + \frac{138}{67375}\eta^2 - \frac{296}{511875}\eta^3 - \frac{38464}{63669375}\eta^4 - \frac{117472}{797671875}\eta^5 \right),$$

$$b_1(\zeta) = 2^{1/3} \left(\frac{-1213}{1023750} - \frac{3757}{2695000}\eta - \frac{3225661}{6700443750}\eta^2 + \frac{90454643}{336992906250}\eta^3 + \frac{48719944786}{142468185234375}\eta^4 \right),$$

$$b_2(\zeta) = 2^{1/3} \left(\frac{16542537833}{37743205500000} + \frac{115773498223}{162820783125000}\eta + \frac{548511920915149}{1721719224225000000}\eta^2 - \frac{212557317961}{883884251250000}\eta^3 \right),$$

$$b_3(\zeta) = 2^{1/3} \left(\frac{-9597171184603}{25476663712500000} - \frac{430990563936859253}{568167343994250000000}\eta - \frac{3191320338955050557}{7777535495585625000000}\eta^2 \right).$$

The way we choose if we are near or far from the turning point to compute $A_\nu(\zeta)$ and $B_\nu(\zeta)$, is as follows: we check if the sequences $(a_k(\zeta))_{k \in \mathbb{N}}$ and $(b_k(\zeta))_{k \in \mathbb{N}}$ are decreasing and if they are, we use the first method, if they are not, we use the second method.

2 How to use the toolbox

It is governed by the CeCILL-C license under French law and abiding by the rules of distribution of free software. It can be used, modified and/or redistributed under the

terms of the CeCILL-C license as circulated by CEA, CNRS and INRIA at the following URL: <http://www.cecill.info>.

The toolbox contains the following Matlab scripts:

- **abk**: computes the sequences $(a_k(\zeta))_{k \in \mathbb{N}}$ and $(b_k(\zeta))_{k \in \mathbb{N}}$ when y is far from the turning point.
- **abktp**: computes the sequences $(a_k(\zeta))_{k \in \mathbb{N}}$ and $(b_k(\zeta))_{k \in \mathbb{N}}$ when y is near the turning point.
- **besselc**: computes all the Bessel and Hankel functions.
Usage `[j,y,h1,h2] = besselc(nu, z)` for ν and z complex or real numbers.
- **besselhc**: computes the Hankel functions.
Usage `h = besselhc(nu, l, z)` for ν and z complex or real numbers, $l = 1$ for $H^{(1)}$ and $l = 2$ for $H^{(2)}$.
- **besseljc**: computes the Bessel function of the first kind.
Usage `j = besseljc(nu, z)` for ν and z complex or real numbers.
- **besselyc**: computes the Bessel function of the second kind.
Usage `y = besselyc(nu,z)` for ν and z complex or real numbers.
- **debye**: computes the Debye polynomial sequence.
- **lambdamu**: computes the sequences λ and μ .

The reason I have one script that compute all of them is because, like you can see in Eq. (1.1) for computing the two Bessel and Hankel functions we use the same $A_\nu(\zeta)$ and $B_\nu(\zeta)$. So if you need to compute all of them it faster to use the **besselc** script than calling the three other, but if you need only one of them it faster to call for example **besseljc** than **besselc**.

Since we need to compute the Airy functions and we use the Matlab functions **airy** to do that, and since they can return **Inf** et **NaN**, my script can also return **Inf** and **NaN**.

3 Numerical validation

For numerical validation, we use Mathematica as a reference. The first test we did was to compare the values for 2000 random $\nu = r_\nu e^{i\theta_\nu}$ with $r_\nu \in [400, 40000]$, $\theta_\nu \in [-\frac{\pi}{10^4}, \frac{\pi}{10^4}]$ and for each value of ν we took a random $z_\nu \in [0.999|\nu|, 1.001|\nu|]$. For the 2000 couples (ν, z_ν) we computed the result of the two Bessel and Hankel functions with Mathematica and with my script, and we compute the relative error between them. The results are shown Fig. 2. The maximum of the relative error is -7.707 in logarithmic scale. So we have 7 digits accurate but if we look in a smaller zone we can have more digits correct. We did a comparison in three other zone.

In zone 1 we have 1600 points with $\Re(\nu) \in [395, 405]$, $\Im(\nu) \in [-10^{-5}, -10^{-6}]$ and $z = 414.93$. The result are shown in figure 3. The maximum of relative error is -10.634 in logarithmic scale.

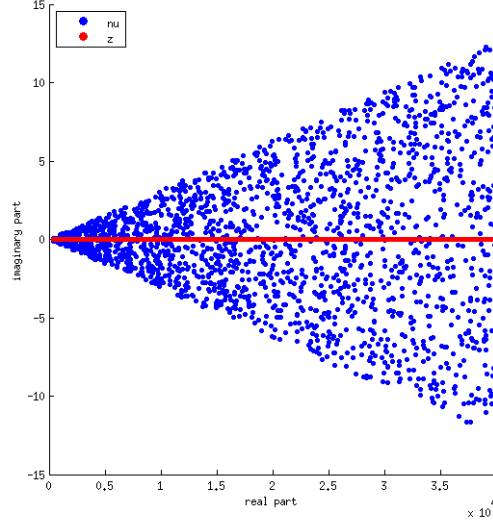


Figure 1: In blue we have the 2000 ν that we take and in red it is the z .

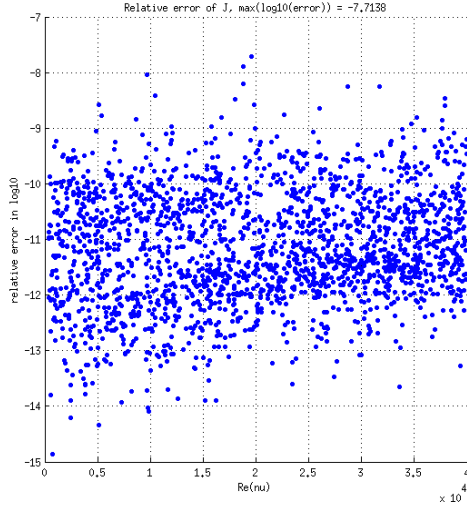
In zone 2 we have 1600 points with $\Re(\nu) \in [3100, 3200]$, $\Im(\nu) \in [-0.7, 0.5]$ and $z = 3160.2$. The results are shown in figure 4. The maximum of relative error is -9.727 in logarithmic scale.

In zone 3 we have 1600 points with $\Re(\nu) \in [412.93, 416.93]$, $\Im(\nu) \in [-10^{-5}, -10^{-6}]$ and $z = 414.93$. The results are shown in figure 5. The maximum of relative error is -8.814 in logarithmic scale.

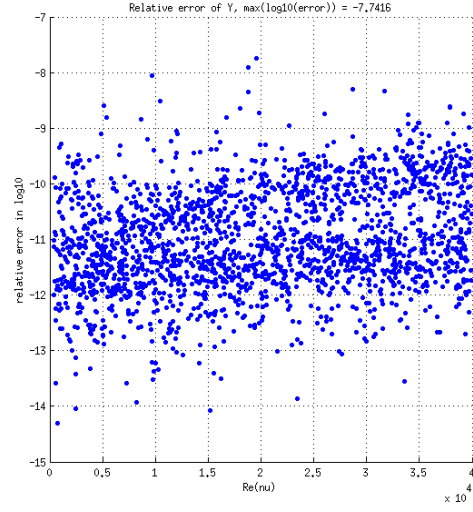
To conclude, if $|\nu| \geq 400$ we have 7 digits correct and we can have more the greater $|\nu|$ is and z not too close to ν .

References

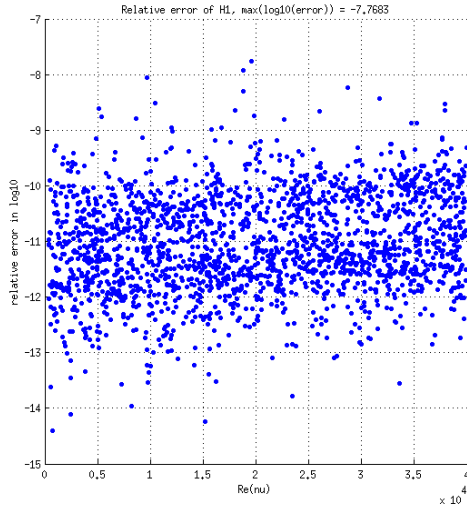
- [1] F. W. J. Olver. *Asymptotics and special functions*. AKP Classics. A K Peters, Ltd., Wellesley, MA, 1997. [doi:10.1201/9781439864548](https://doi.org/10.1201/9781439864548).
- [2] N. M. Temme. Numerical algorithms for uniform airy-type asymptotic expansions. *Numerical Algorithms*, 15(2):207–225, 1997. [doi:10.1023/A:1019197921337](https://doi.org/10.1023/A:1019197921337).



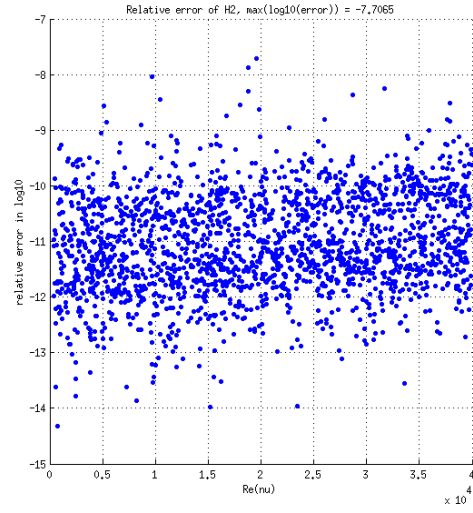
(a) Relative error of J



(b) Relative error of Y

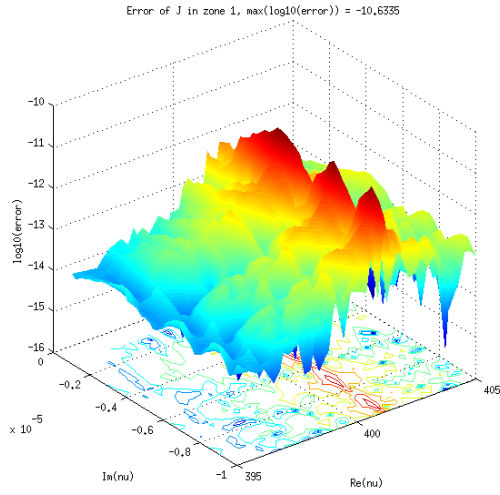


(c) Relative error of H1

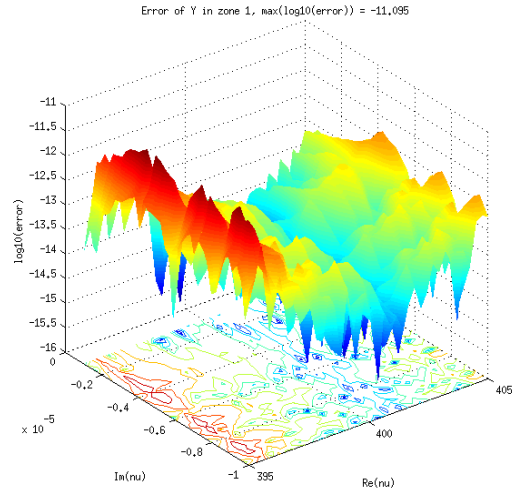


(d) Relative error of H2

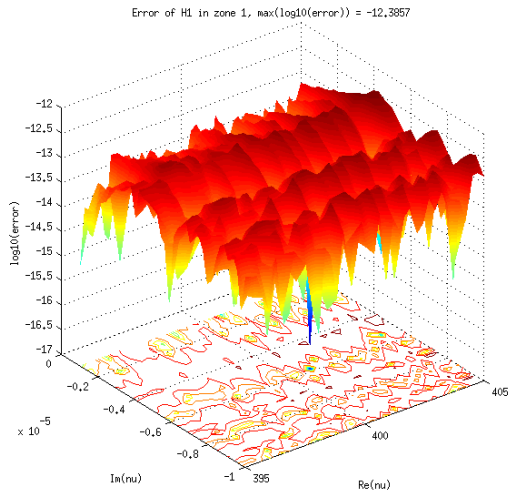
Figure 2: I plot the projection of $(\Re(\nu), \Im(\nu), \text{relative error})$ on the plan $\Im(\nu) = 0$.



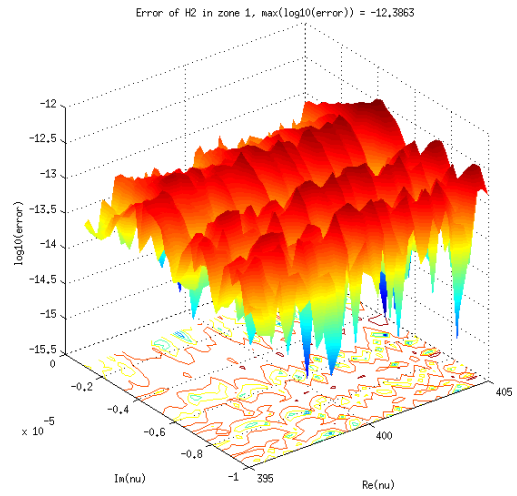
(a) Relative error of J in zone 1



(b) Relative error of Y in zone 1

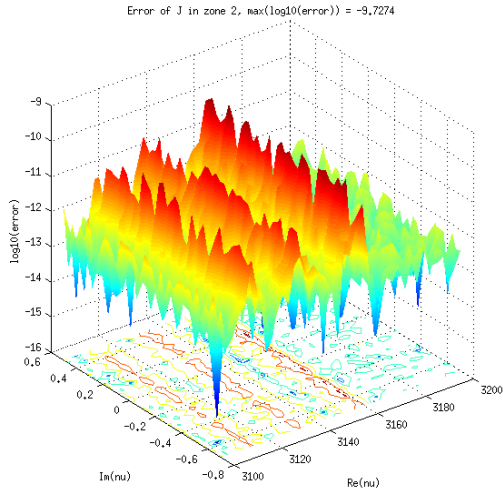


(c) Relative error of H1 in zone 1

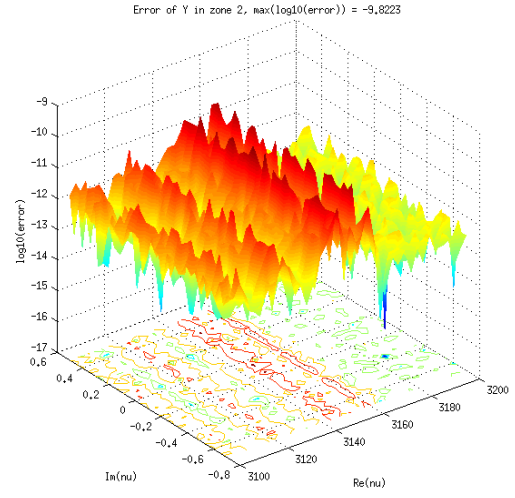


(d) Relative error of H2 in zone 1

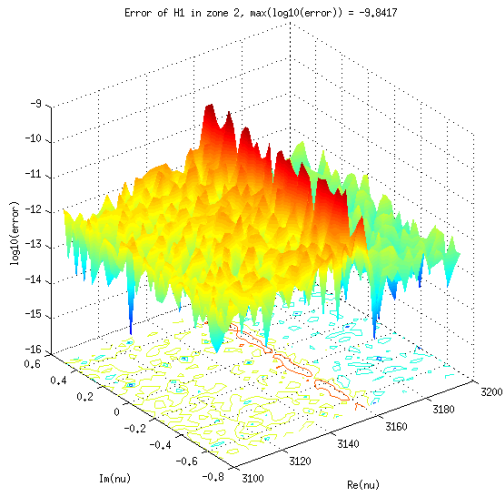
Figure 3: The relative error of the Bessel and Hankel functions.



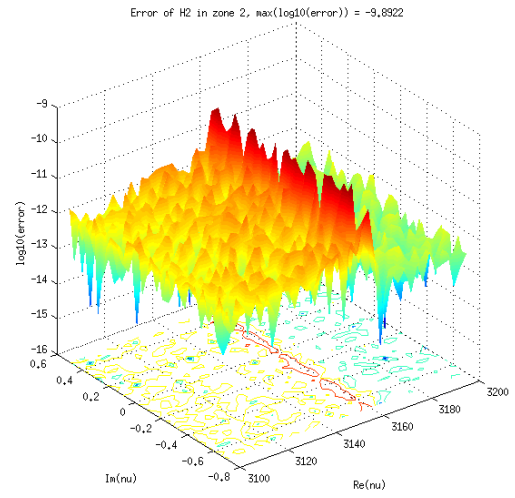
(a) Relative error of J in zone 2



(b) Relative error of Y in zone 2

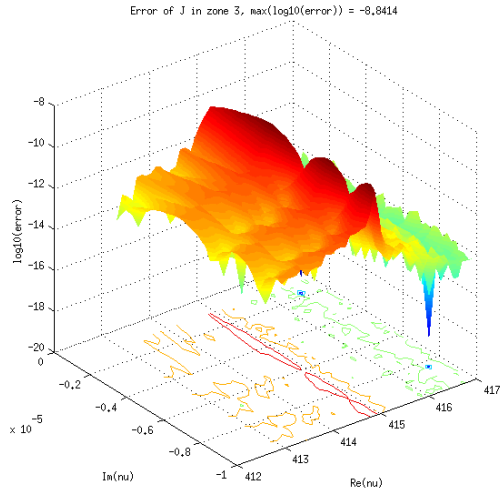


(c) Relative error of H1 in zone 2

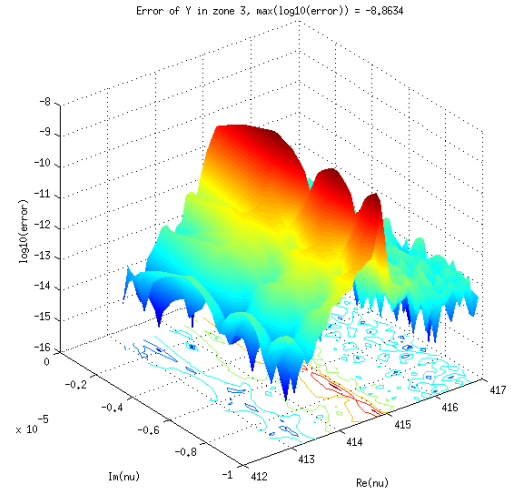


(d) Relative error of H2 in zone 2

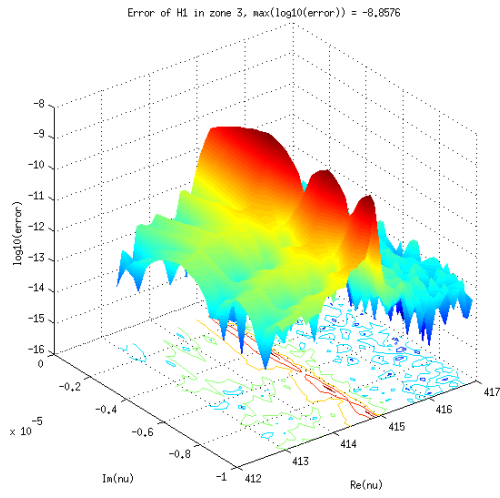
Figure 4: The relative error of the Bessel and Hankel functions.



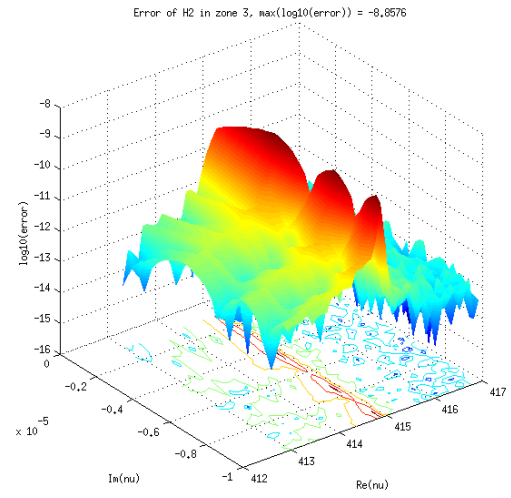
(a) Relative error of J in zone 3



(b) Relative error of Y in zone 3



(c) Relative error of H1 in zone 3



(d) Relative error of H2 in zone 3

Figure 5: The relative error of the Bessel and Hankel functions.