POLITECNICO DI TORINO

Bachelor of Science in Mathematics for Engineering

Bachelor's Thesis

Computational strategies for the special functions of Mathieu

Supervisor
prof. Francesco Paolo Andriulli
Supervisor's signature

....

Candidate
Antonino Emanuele Scurria
Candidate's signature

....

Academic Year 2021-2022

Summary

In physical models we try to look for symmetries that may simplify the initial problem and allow us to proceed in our investigation. If we consider the Helmotz equation in an elliptical framework through the separation of variables we obtain two different differential equations whose solutions are the standard Mathieu functions and the modified Mathieu functions. Thus, the study of these special functions becomes relevant and the lack of a built-in implementation in both Mathematica and Matlab software increases the need for a correct computational scheme.Therefor, the aim of this thesis is to deeper investigate a computational scheme for these special functions and to highlight the difficulties encountered and the proposed solutions.

Acknowledgements

Special thanks to professor Andriulli and Ermanno who accompanied me during a wonderful and meaningful experience. You taught me a lot.

Contents

Chapter 1

Mathieu functions

Let consider the framework of the elliptical cylinder coordinates (ν, μ, z) :

$$x = a\cosh\mu\cos\nu \qquad y = a\sinh\mu\sin\nu \qquad z = z \tag{1.1}$$

where a is the semi-focal length of the elliptic system, $0 \leq \mu < \infty$, $0 \leq \nu \leq 2\pi$ and $-\infty < z < \infty$ (in this framework a and μ define a particular ellipse). These coordinate are linked to Cartesian coordinates (x, y, z) by:

$$\cosh\mu = \sqrt{\frac{x^2 + y^2 + a^2 + \sqrt{(x^2+y^2)^2 + a^4 - 2a^2(x^2 - y^2)}}{2a^2}} \qquad \cos\nu = \frac{x}{a\cosh\mu} \tag{1.2}$$

If we consider the Helmotz equation in which k is the wave number, that is

$$\nabla^2\psi + k^2\psi = 0 \tag{1.3}$$

separation of variables in the elliptic variables (μ, ν, z) leads us to seek for a solution of this type:

$$\psi = X(\nu)Y(\mu)Z(z) \tag{1.4}$$

If we substitute 1.4 in 1.3 we obtain:

$$\begin{aligned} \frac{d^2X}{d\nu^2} + (\alpha^2 - k^2a^2\cos^2\nu)X &= \frac{d^2X}{d\nu^2} + \left(\alpha^2 - k^2a^2\left(\frac{\cos 2\nu + 1}{2}\right)\right)X \\ &= \frac{d^2X}{d\nu^2} + ((\alpha^2 - 2q) - 2q\cos 2\nu)X = 0 \end{aligned} \tag{1.5}$$

$$\begin{aligned} \frac{d^2Y}{d\mu^2} - (\alpha^2 - k^2a^2\cosh^2\mu)Y &= \frac{d^2Y}{d\mu^2} - (\alpha^2 - k^2a^2\left(\frac{\cosh 2\mu + 1}{2}\right)Y \\ &= \frac{d^2Y}{d\mu^2} - ((\alpha^2 - 2q) - 2q\cosh 2\mu)X = 0 \end{aligned} \tag{1.6}$$

and

$$\frac{d^2Z}{dz^2} + (k^2 + \frac{4q}{k^2})Z = 0 \tag{1.7}$$

where α is a separation constant and $q = \frac{1}{4}(ka)^2$. Now let suppress the variable z and consider only a two dimensional framework depending on the variables ν and μ and define $\lambda = \alpha^2 - 2q$: equations 1.5 and 1.6 take the form

$$X''(\nu) + (\lambda - 2q\cos 2\nu)X(\nu) = 0 \tag{1.8}$$

and

$$Y''(\mu) - (\lambda - 2q\cosh 2\mu)Y(\mu) = 0 \tag{1.9}$$

Equations 1.8 and 1.9 are respectively the standard (or circumferential) Mathieu's equation and the modified (or radial) Mathieu's equation).

1.1 Angular and radial solutions

Let consider 1.8, it can be shown that, for a fixed k, there exists a countably infinite set of characteristic values $\lambda = a_r(q)$ which leads to even periodic solutions (ce_r) of 1.8. Moreover there is another countably infinite set $\lambda = b_r(q)$ related to odd periodic solutions (se_r) of 1.8. The solutions of 1.8 are commonly referred as angular Mathieu functions. If we consider a periodic solution $Q(\nu)$ of 1.8, we can express it through a Fourier expansion and we obtain:

$$Q(\nu) = \sum_{m=0}^{\infty} A_m \cos m\nu + B_m \sin m\nu \tag{1.10}$$

and if we substitute this expression in 1.8 we obtain

$$\sum_{m=0}^{\infty} (\lambda - m^2)A_m - q(A_{m-2} + A_{m+2})\cos(m\nu) \tag{1.11}$$

$$+\sum_{m=1}^{\infty} (\lambda - m^2)B_m - q(B_{m-2} + B_{m+2})\sin(m\nu) = 0 \tag{1.12}$$

$$A_{-m}, B_{-m} = 0 \quad m > 0 \tag{1.13}$$

Periodic solutions of the circumferential Mathieu's equation with even symmetry can be found in the form:

$$ce_{2n+p}(\nu, q) = \sum_{m=0}^{\infty} A_{2m+p}^{2n+p} \cos((2m+p)\nu), \quad n \geq 0, p = 0,1. \tag{1.14}$$

Considering periodic solutions with odd symmetry leads to :

$$se_{2n+p}(\nu,q) = \sum_{m=0}^{\infty} B_{2m+p}^{2n+p} \sin((2m+p)\nu), \quad n \geq 0, p = 1,2. \tag{1.15}$$

Recalling 1.9 the solutions of this equation (the modified one) are called radial Mathieu functions:in particular, we are interested in obtaining the radial Mathieu functions denoted as $Mc^{(1)}$ $Ms^{(1)}$ $Mc^{(2)}$ and $Ms^{(2)}$ and to obtain them we will use two different approaches: the first approach will be based on a built-in implementation of ce and se in Mathematica, whereas the second approach will deal with the implementation of ce and se themselves, exploiting the Matlab environment naturally oriented to vector and matrix calculus.

Chapter 2

Computational scheme in Mathematica

2.1 Connection formulas

The first approach to create correct a computational scheme for the radial Mathieu functions was based on 4 different formulas of the NIST (28.22.1-28.22.4):

$$Mc_n^{(1)}(\mu, q) = \sqrt{\frac{2}{\pi}} \frac{1}{g_{e,n}(\sqrt{q}) ce_n(0, q)} Ce_n(\mu, q) \tag{2.1}$$

$$Ms_n^{(1)}(\mu, q) = \sqrt{\frac{2}{\pi}} \frac{1}{g_{o,n}(\sqrt{q}) se'_n(0, q)} Se_n(\mu, q) \tag{2.2}$$

$$Mc_n^{(2)}(\mu, q) = \sqrt{\frac{2}{\pi}} \frac{1}{g_{e,n}(\sqrt{q}) ce_n(0, q)} \left(-f_{e,n}(\sqrt{q}) Ce_n(\mu, q) + \frac{2}{\pi C_n(q)} Fe_n(\mu, q) \right) \tag{2.3}$$

$$Ms_n^{(2)}(\mu, q) = \sqrt{\frac{2}{\pi}} \frac{1}{g_{o,n}(\sqrt{q}) se'_n(0, q)} \left(-f_{o,n}(\sqrt{q}) Se_n(\mu, q) - \frac{2}{\pi S_n(q)} Ge_n(\mu, q) \right) \tag{2.4}$$

Let us start with analyzing 2.1 and 2.2:

$$g_{e,2n}(q) = (-1)^n \sqrt{\frac{2}{\pi}} \frac{ce_{2n}(\frac{1}{2}\pi, q)}{A_0^{2n}(q)} \tag{2.5}$$

$$g_{e,2n+1}(q) = (-1)^{n+1} \sqrt{\frac{2}{\pi}} \frac{ce'_{2n+1}(\frac{1}{2}\pi, q)}{\sqrt{q} A_1^{2n+1}(q)} \tag{2.6}$$

$$g_{0,2n+1}(q) = (-1)^n \sqrt{\frac{2}{\pi}} \frac{se_{2n+1}(\frac{1}{2}\pi, q)}{\sqrt{q} B_1^{2n+1}(q)} \tag{2.7}$$

$$g_{o,2n+2}(q) = (-1)^{n+1} \sqrt{\frac{2}{\pi}} \frac{se'_{2n+2}(\frac{1}{2}\pi, q)}{q B_2^{2n+2}(q)} \tag{2.8}$$

where A_i^j and B_i^j stand for the i-th coefficients of ce_j and se_j in the expansion presented in 1.14 and 1.15. Moreover Ce and Se are solutions of 1.9 (modified Mathieu's equation) so defined (substituting ν with $i\nu$ in 1.8 we obtain 1.9):

$$Ce_n(\mu, q) = ce_n(\pm i\mu, q) \tag{2.9}$$

$$Se_n(\mu, q) = \mp i se_n(\pm i\mu, q) \tag{2.10}$$

2.2 $Mc^{(1)}$ and $Ms^{(1)}$ implementation

We start with the definition of the coefficient of the expansions in 1.8 and 1.9 (Abramowitz and Stegun 20.5.5);

$$A_0^{2n} = \frac{1}{2\pi} \int_0^{2\pi} ce_{2n}(\mu, q) d\mu \tag{2.11}$$

$$A_m^n = \frac{1}{\pi} \int_0^{2\pi} ce_n(\mu, q) \cos(m\mu) d\mu \tag{2.12}$$

and

$$A_m^n = \frac{1}{\pi} \int_0^{2\pi} se_n(\mu, q) \cos(m\mu) d\mu \tag{2.13}$$

In Mathematica we have:

```
ce[n_, z_, q_] = MathieuC[MathieuCharacteristicA[n, q], q, z];
se[n_, z_, q_] = MathieuS[MathieuCharacteristicB[n, q], q, z];
A[m_?IntegerQ, n_?IntegerQ, q_] := (1 - KroneckerDelta[0, m] / 2) 1 / π NIntegrate[ce[n, x, q] Cos[m x], {x, 0, 2 π}] /;
  EvenQ[n] && EvenQ[m] || OddQ[n] && OddQ[m]
B[m_?IntegerQ, n_?IntegerQ, q_] := 1 / π NIntegrate[se[n, x, q] Sin[m x], {x, 0, 2 π}] /;
  EvenQ[n] && EvenQ[m] || OddQ[n] && OddQ[m]
```

Now we have to implement the coefficients $g_{e,n}$ and $g_{o,n}$ as presented in 2.5 2.6 2.8 2.7:

```
gen[n_?EvenQ, q_] := (-1)^(n / 2) Sqrt[2 / π] ce[n, π / 2, q] / A[0, n, q]
gen[n_?OddQ, q_] := (-1)^((n + 1) / 2) Sqrt[2 / π] Derivative[0, 1, 0][ce][n, π / 2, q] / (Sqrt[q] A[1, n, q])
gon[n_?OddQ, q_] := (-1)^((n - 1) / 2) Sqrt[2 / π] se[n, π / 2, q] / (Sqrt[q] B[1, n, q])
gon[n_?EvenQ, q_] := (-1)^(n / 2) Sqrt[2 / π] Derivative[0, 1, 0][se][n, π / 2, q] / (q B[2, n, q])
```

The next step consists in defining Ce and Se as in 2.9 and 2.10:

```
Ce[n_, z_, q_] = ce[n, I z, q];
Se[n_, z_, q_] = -I se[n, I z, q];
```

Now we have all we need to define $Mc^{(1)}$ and $Ms^{(1)}$:

```
Mc1[n_, z_, q_] := Sqrt[2 / π] 1 / (gen[n, q] × ce[n, 0, q]) Ce[n, z, q]
Ms1[n_, z_, q_] := Sqrt[2 / π] 1 / (gon[n, q] × Derivative[0, 1, 0][se][n, 0, q]) Se[n, z, q]
```

$Mc^{(1)}$ and $Ms^{(1)}$ so obtained have been numerically controlled and compared with tables obtained by Blanch and Clemm (Tables Relating to the Radial Mathieu Functions: Functions of the first kind, by G. Blanch and D. S. Clemm) and with "Accurate computation of Mathieu's functions by Malcom M. Bibby, Andrew F. Peterson" and, initially, the matching observed was amazing; after some tests I discovered some instability problems related to the argument μ (z in Mathematica) of the functions.

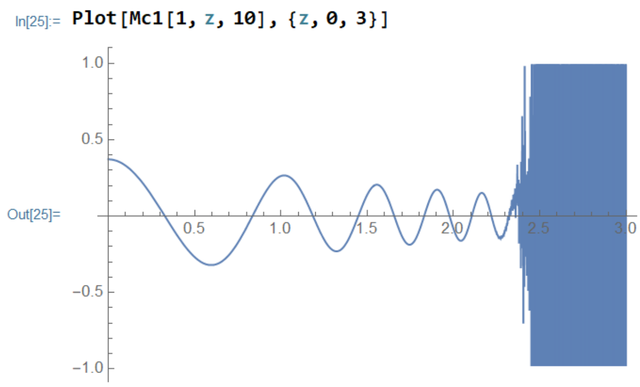

Figure 2.1. instability problems in the computation of $Mc_1^1(z{,}10)$

We started investigating the origin of the instability related to the argument μ and we discovered that it is due not to the algorithm implemented, but to the built-in functions of Mathematica: indeed, in the algorithm we exploited the built-in functions "MathieuC" and "MathieuS" to compute ce and se respectively (first two lines of the Mathematica code) computed in iz (we consider a real z, so iz is purely an imaginary number) and probably the algorithm, either the continued fraction method or the matrix formulation (the latter will be discussed later), used in the functions just mentioned is not conceived to work with complex numbers.

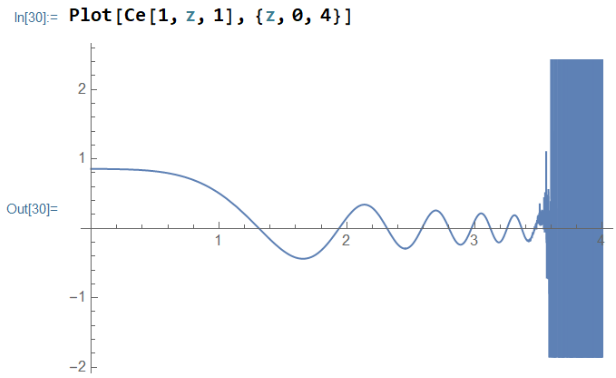

Figure 2.2. instability problems in the computation of $Ce_5(z,1)$

The connection formulas presented in 2.1 2.2 2.3 and 2.4 were very useful to implement the radial Mathieu functions $Mc^{(1)}$ $Ms^{(1)}$ $Mc^{(2)}$ and $Ms^{(2)}$ due to the fact that we have a built-in implementation in Mathematica of functions that computes both the characteristic values ("MathieuCarachteristicA" and"MathieuCarachteristicB") and the solutions *ce* and *se* but, since we are unable to use those functions to obtain a stable definition of *Ce* and *Se*, we have to move to another strategy.

Chapter 3

Computational scheme in Matlab

3.1 Eigenvalue problem algorithm

The strategy illustrated in the previous chapter, through the connection formulas, was suitable to Mathemtica since we have a built-in implementation of *ce* and *se*; despite this we had instability problems related to imaginary arguments. For this reason we have to work on the implementation of *ce* and *se* itself. Indeed we changed the initial strategy based on the connection formulas and started to exploit other formulas presented in "Abramowitz and Stegun":

$$Mc_{2n}^{(j)}(\mu,q)=\sum_{m=0}^{\infty}(-1)^{n+m}\frac{A_{2m}^{2n}}{\epsilon_s A_{2s}^{2n}}(\mathcal{J}_{m-s}(u1)\mathcal{Z}_{m+s}^{(j)}(u2)+\mathcal{J}_{m+s}(u1)\mathcal{Z}_{m-s}^{(j)}(u2)) \tag{3.1}$$

where $\epsilon_0=2$ and $\epsilon_j=1$, for s=1,2,3...∞ (s arbitrary) and $n\geq 0$

$$Mc_{2n+1}^{(j)}(\mu,q)=\sum_{m=0}^{\infty}(-1)^{n+m}\frac{A_{2m+1}^{2n+1}}{A_{2s+1}^{2n+1}}(\mathcal{J}_{m-s}(u1)\mathcal{Z}_{m+s+1}^{(j)}(u2)+\mathcal{J}_{m+s+1}(u1)\mathcal{Z}_{m-s}^{(j)}(u2)) \tag{3.2}$$

$$Ms_{2n}^{(j)}(\mu,q)=\sum_{m=1}^{\infty}(-1)^{n+m}\frac{B_{2m}^{2n}}{B_{2s}^{2n}}(\mathcal{J}_{m-s}(u1)\mathcal{Z}_{m+s}^{(j)}(u2)-\mathcal{J}_{m+s}(u1)\mathcal{Z}_{m-s}^{(j)}(u2)) \tag{3.3}$$

$$Ms_{2n+1}^{(j)}(\mu,q)=\sum_{m=0}^{\infty}(-1)^{n+m}\frac{B_{2m+1}^{2n+1}}{B_{2s+1}^{2n+1}}(\mathcal{J}_{m-s}(u1)\mathcal{Z}_{m+s+1}^{(j)}(u2)-\mathcal{J}_{m+s+1}(u1)\mathcal{Z}_{m-s}^{(j)}(u2)) \tag{3.4}$$

where s arbitrary,$n\geq 0$ ($n>0$ in the third case) , $u1=\sqrt{q}\exp(-\mu)$, $u2=\sqrt{q}\exp(\mu)$, $p=0{,}1, \mathcal{J}_p$ stands for the p-th Bessel function of the first kind and

$\mathcal{Z}_p^{(j)}$ stands for the Bessel function of the first kind ($\mathcal{J}_p$) if $j = 1$, the Bessel function of the second kind ($\mathcal{Y}_p$) if $j = 2$ and the Hankel functions $\mathcal{H}_p^{(1)}$ and $\mathcal{H}_p^{(2)}$ respectively for $j = 3,4$ (we will consider only $j = 1,2$ but I illustrated the other cases for the sake of completeness). Although s is arbitrary, it can be chosen, according to 20.4.13 of the "Handbook of Mathematical functions (Abramowitz and Stegun)", as the index of the greatest (in modulus) Fourier coefficient to improve series convergence.

Thus we move to Matlab to simplify the implementation of these formulas thanks to an environment naturally prepared for matrix and vector calculus. Indeed, if we consider the expansion in Fourier series as presented in 1.14 and 1.15 substitute it in 1.8, we obtain a homogeneous set of three-term recursion relations with their arbitrary normalization condition (we use the standard normalization condition shown in the NIST) for ce_{2n} ce_{2n+1} se_{2n+1} se_{2n+2}

- for ce_{2n}

$$\begin{aligned} \lambda A_0^{2n} - qA_2^{2n} &= 0 \quad m = 0 \\ (\lambda - 4)A_2^{2n} - q(A_0^{2n} + A_4^{2n}) &= 0 \quad m = 1 \\ (\lambda - (2m)^2)A_{2m}^{2n} - q(A_{2m-2}^{2n} + A_{2m+2}^{2n}) &= 0, \quad m \geq 2 \end{aligned}$$

  with the normalization condition

$$2(A_0^{2n})^2 + \sum_{m=1}^{\infty}(A_{2m}^{2n})^2 = 1$$

- for ce_{2n+1}

$$\begin{aligned} (\lambda - 1 - q)A_1^{2n+1} - qA_3^{2n+1} &= 0 \quad m = 0 \\ (\lambda - (2m+1)^2)A_{2m+1}^{2n+1} + A_{2m+3}^{2n+1} &= 0, \quad m \geq 1 \end{aligned}$$

  with the normalization condition

$$\sum_{m=0}^{\infty}(A_{2m+1}^{2n+1})^2 = 1$$

- for se_{2n+1}

$$\begin{aligned} (\lambda - 1 + q)B_1^{2n+1} - qB_3^{2n+1} &= 0 \quad m = 0 \\ (\lambda - (2m+1)^2)B_{2m+1}^{2n+1} - q(B_{2m-1}^{2n+1} + B_{2m+1}^{2n+1}) &= 0, \quad m \geq 1 \end{aligned}$$

  with the normalization condition

$$\sum_{m=0}^{\infty}(B_{2m+1}^{2n+1})^2 = 1$$

- for se_{2n+2}

$$(\lambda-4)B_2^{2n+2}-qB_4^{2n+2}=0 \quad m=0$$
$$(\lambda-(2m+2)^2)B_{2m+2}^{2n+2}-q(B_{2m}^{2n+2}+B_{2m+4}^{2n+2})=0, \quad m\geq 1$$

with the normalization condition

$$\sum_{m=0}^{\infty}(B_{2m+2}^{2n+2})^2=1$$

So, for a fixed value of q, we have to solve an infinite order eigenvalue problem related to a tridiagonal matrix where λ is the eigenvalue parameter and A_{2m+p}^{2n+p} and B_{2m+p}^{2n+p} (p as already mentioned) are the eigenvector components. If we assume that the coefficients of the Fourier series can be neglected for $m \geq M$ then we have to solve a finite order eigenvalue problem, namely:

$$\begin{bmatrix} 0 & \sqrt{2}q & & & & & 0 \\ \sqrt{2}q & 4 & q & & & & \\ & q & 16 & q & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & q & (2m)^2 & q \\ 0 & & & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \sqrt{2}A_0^{2n} \\ A_2^{2n} \\ A_4^{2n} \\ \vdots \\ \vdots \\ A_{2m}^{2n} \end{bmatrix} = \lambda_{2m} \begin{bmatrix} \sqrt{2}A_0^{2n} \\ A_2^{2n} \\ A_4^{2n} \\ \vdots \\ \vdots \\ A_{2m}^{2n} \end{bmatrix} \tag{3.5}$$

$$\begin{bmatrix} (1+q) & q & & & & & 0 \\ q & 9 & q & & & & \\ & q & 25 & q & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & q & (2m+1)^2 & q \\ 0 & & & & & & \ddots \end{bmatrix} \begin{bmatrix} A_1^{2n+1} \\ A_3^{2n+1} \\ A_5^{2n+1} \\ \vdots \\ \vdots \\ A_{2m+1}^{2n+1} \end{bmatrix} = \lambda_{2m+1} \begin{bmatrix} A_1^{2n+1} \\ A_3^{2n+1} \\ A_5^{2n+1} \\ \vdots \\ \vdots \\ A_{2m+1}^{2n+1} \end{bmatrix} \tag{3.6}$$

$$\begin{bmatrix} 1-q & q & & & & & 0 \\ q & 9 & q & & & & \\ & q & 25 & q & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & q & (2m+1)^2 & q \\ 0 & & & & & & \ddots \end{bmatrix} \begin{bmatrix} B_1^{2n+1} \\ B_3^{2n+1} \\ B_5^{2n+1} \\ \vdots \\ \vdots \\ B_{2m+1}^{2n+1} \end{bmatrix} = \lambda_{2m+1} \begin{bmatrix} B_1^{2n+1} \\ B_3^{2n+1} \\ B_5^{2n+1} \\ \vdots \\ \vdots \\ B_{2m+1}^{2n+1} \end{bmatrix} \tag{3.7}$$

$$\begin{bmatrix} 4 & q & & & & & 0 \\ q & 16 & q & & & & \\ & q & 36 & q & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & q & (2m)^2 & q \\ 0 & & & & & & \ddots \end{bmatrix} \begin{bmatrix} B_2^{2n+2} \\ B_4^{2n+2} \\ B_6^{2n+2} \\ \vdots \\ \vdots \\ B_{2m}^{2n+2} \end{bmatrix} = \lambda_{2m} \begin{bmatrix} B_2^{2n+2} \\ B_4^{2n+2} \\ B_6^{2n+2} \\ \vdots \\ \vdots \\ B_{2m}^{2n+2} \end{bmatrix} \tag{3.8}$$

respectively for ce_{2n} ce_{2n+1} se_{2n+1} and se_{2n+2} . So, to obtain the coefficients of the Fourier expansion, we have to truncate the infinite eigenvalue problem to an N-dimensional problem: in this way we will obtain N different eigenvectors which will correspond to the coefficient A and B of the first N orders of ce and se respectively. Moreover the eigenvalues obtained will correspond to the characteristic values a_n and b_n.

3.2 Matlab implementation

The first step in the algorithm consists in creating the matrix as defined in 3.5, 3.6, 3.7 and 3.8: in the code below we split the 4 different cases through the "flag"variables "ftype" ("1" stands for ce whilst "2" stands for se) and the "flag" variable "order" ("1" stands for an even index whereas "2" stands for an odd one).

```
    if nargin<4, ntrms=50; end
if nargin==0, q=10; order=1; ftype=1; end
v=diag(q*ones(ntrms-1,1),1); v=v+v.';
if ftype==1; % even valued
  if order==1 % even order
    v=v+diag((2*(0:ntrms-1)).^2);
    v(1,2)=sqrt(2)*q; v(2,1)=v(1,2);
  else           % odd order
    v=v+diag((1+2*(0:ntrms-1)).^2);
    v(1,1)=1+q;
  end
else % ftype==2, odd valued
  if order==1  % even order
    v=v+diag((2*(1:ntrms)).^2);
  else           % odd order
    v=v+diag((1+2*(0:ntrms-1)).^2);
    v(1,1)=1-q;
  end
```

```
end
```

Once we have obtained the desired matrix we compute the eigenvalues through the built-in function "eig": the variable "a"is a vector containing the eigenvalues, the variable "c" is the matrix containing the eigenvectors (that are the coefficients A_m or B_m). The last line of this part of the code represents just what we have to do to obtain A_0 in 3.5 since, from the computation of the eigenvectors, we obtain $\sqrt{2}A_0$.

```
[c,a]=eig(v);
a=diag(a);
if order==1 && ftype==1
   c(1,:)=c(1,:)/sqrt(2);
end
```

Here we impose the normalization conditions in accordance with what stated above and remembering, in the case of ce_{2n}, that we have to multiply A_0 by 2.

```
      % Normalize the series coefficients
cv=sum(conj(c).*c);
if order==1 && ftype==1
   cv=cv+conj(c(1,:)).*c(1,:);
end
c=ones(ntrms,1)*(1./sqrt(cv)).*c;
```

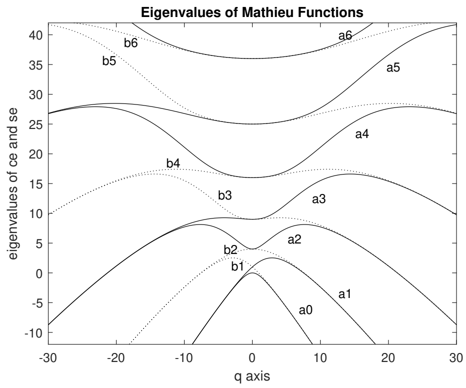

Figure 3.1. First 6 characteristic values a_n (continuous lines) and b_n(dotted lines) as function of q

After having obtained the coefficient A_m^n we are able to compute respectively ce_n through the Fourier expansion ("ntrms" is an integer that indicates how many

Fourier coefficients we desire to compute ce_n and "x" is a vector containing the points where we want to evaluate ce_n).

```
    if nargin<4, ntrms=50; end
if mod(n,2)==0, order=1; n=2*(0:ntrms-1); % even order
else order=2; n=1+2*(0:ntrms-1); end      % odd order
[a,c]=matue(q,order,1,ntrms);
v=cos(x(:)*n)*c(:,1+fix(m/2));
```

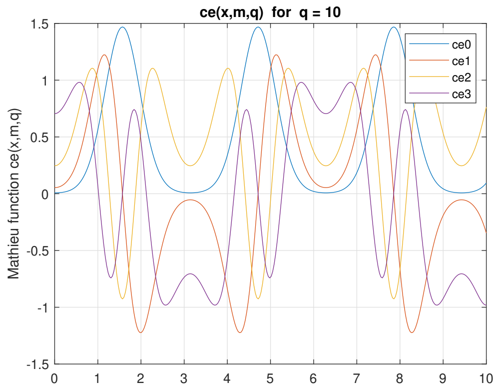

Figure 3.2. Plot of ce_m with $0 \leq m \leq 3$ with q=10

Analogous to ce_n

```
    if nargin<4, ntrms=50; end
if mod(m,2)==0
   order=1; n=2*(1:ntrms);       % even order
   [a,c]=matue(q,order,2,ntrms);
   v=sin(x(:)*n)*c(:,m/2);
else
   order=2; n=1+2*(0:ntrms-1); % odd order
   [a,c]=matue(q,order,2,ntrms);
   v=sin(x(:)*n)*c(:,1+fix(m/2));
end
```

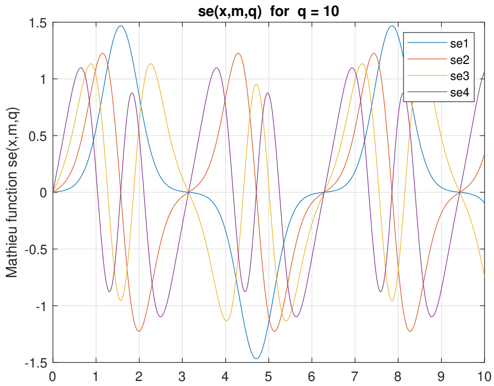

Figure 3.3. Plot of se_m with $1 \leq m \leq 4$ with q=10

The code below computes $Mc_n^{(1)}$ exploiting 3.1 and 3.2: "z" is the vector of points where we want to evaluate the function.

```
    if nargin<4, ntrms=50; end
z=z(:); nz=length(z); q2=sqrt(q);
k=0:ntrms-1; sgn=cos(pi*k)';
mh=1+fix(n/2); u1=q2*exp(-z); u2=q2*exp(z);
if mod(n,2)==0 % even index
   r=n/2; [a,c]=matue(q,1,1,ntrms);
   u=c(:,mh); [ubig,s]=max(abs(u));
   p0=(-1)^r/u(s)*sgn;
   if s==1, p0=p0/2; end; s=s-1;
   f=(bes(k-s,u1).*bes(k+s,u2)+...
       bes(k+s,u1).*bes(k-s,u2))*...
       (p0.*u);
else                % odd index
   [a,c]=matue(q,2,1,ntrms); r1=(n+1)/2;
   u=c(:,mh); [ubig,s]=max(abs(u));
   p0=-(-1)^r1/u(s)*sgn; s=s-1;
   f=(bes(k-s,u1).*bes(k+s+1,u2)+...
       bes(k+s+1,u1).*bes(k-s,u2))*...
       (p0.*u);
end
```

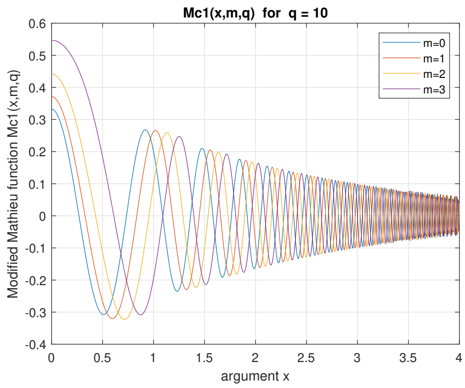

Figure 3.4. Plot of $Mc_m^{(1)}$ with $0 \leq m \leq 3$ with q=10

Analogous to $Mc^{(1)}$

```
    if nargin<4, ntrms=50; end
z=z(:); z1=sqrt(q)*exp(-z);
z2=sqrt(q)*exp(z);
if mod(m,2)==0 % even order
  k=1:ntrms; [a,c]=matue(q,1,2,ntrms);
  u=c(:,m/2); r=m/2;
  [ubig,s]=max(abs(u)); p0=(-1)^r/u(s);
  f=(bes(k-s,z1).*bes(k+s,z2) -...
      bes(k+s,z1).*bes(k-s,z2))*...
      (p0*cos(k*pi)'.*u);
else                % odd order
  k=0:ntrms-1; [a,c,vsave]=matue(q,2,2,ntrms);
  u=c(:,1+fix(m/2)); [ubig,s]=max(abs(u));
  r=(m-1)/2; p0=(-1)^r/u(s); s=s-1;
  f=(bes(k-s,z1).*bes(k+s+1,z2) -...
      bes(k+s+1,z1).*bes(k-s,z2))*...
      (p0*cos(k*pi)'.*u);
end
```

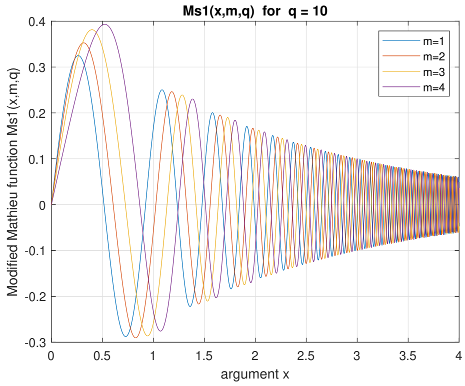

Figure 3.5. Plot of $Ms_m^{(1)}$ with $1 \leq m \leq 4$ with q=10

Analogous to $Mc^{(1)}$

```
     if nargin<4, ntrms=50; end
z=z(:); nz=length(z); q2=sqrt(q);
k=0:ntrms-1; sgn=cos(pi*k)';
mh=1+fix(m/2); u1=q2*exp(-z); u2=q2*exp(z);
if mod(m,2)==0 % even index
   r=m/2; [a,c]=matue(q,1,1,ntrms);
   u=c(:,mh); [ubig,s]=max(abs(u));
   p0=(-1)^r/u(s)*sgn;
   if s==1, p0=p0/2; end; s=s-1;
   f=(bes(k-s,u1).*besy(k+s,u2)+...
       bes(k+s,u1).*besy(k-s,u2))*...
       (p0.*u);
else                  % odd index
   [a,c]=matue(q,2,1,ntrms); r1=(m+1)/2;
   u=c(:,mh); [ubig,s]=max(abs(u));
   p0=-(-1)^r1/u(s)*sgn; s=s-1;
   f=(bes(k-s,u1).*besy(k+s+1,u2)+...
       bes(k+s+1,u1).*besy(k-s,u2))*...
       (p0.*u);
end
```

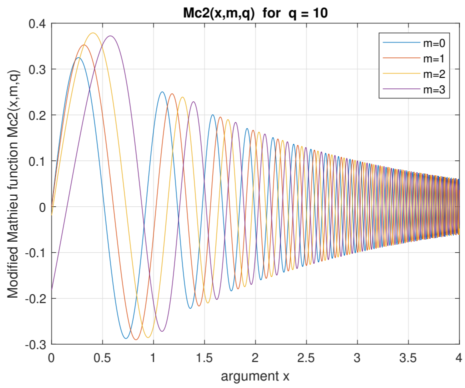

Figure 3.6. Plot of $Mc_m^{(2)}$ with $0 \leq m \leq 3$ with q=10

Analogous to $Mc^{(1)}$

```
    if nargin<4, ntrms=50; end
z=z(:); z1=sqrt(q)*exp(-z);
z2=sqrt(q)*exp(z);
if mod(m,2)==0 % even order
  k=1:ntrms; [a,c]=matue(q,1,2,ntrms);
  u=c(:,m/2); r=m/2;
  [ubig,s]=max(abs(u)); p0=(-1)^r/u(s);
  f=(bes(k-s,z1).*besy(k+s,z2) -...
     bes(k+s,z1).*besy(k-s,z2))*...
     (p0*cos(k*pi)'.*u);
else                % odd order
  k=0:ntrms-1; [a,c]=matue(q,2,2,ntrms);
  u=c(:,1+fix(m/2)); [ubig,s]=max(abs(u));
  r=(m-1)/2; p0=(-1)^r/u(s); s=s-1;
  f=(bes(k-s,z1).*besy(k+s+1,z2) -...
     bes(k+s+1,z1).*besy(k-s,z2))*...
     (p0*cos(k*pi)'.*u);
end
```

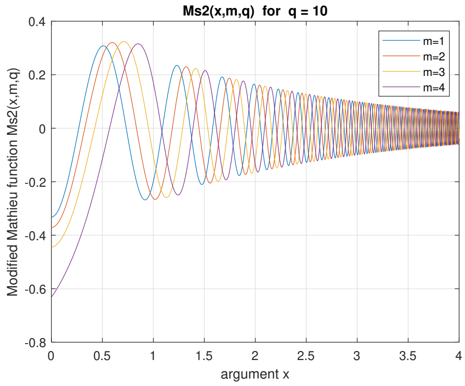

Figure 3.7. Plot of $Ms_m^{(2)}$ with $1 \leq m \leq 4$ with q=10

Chapter 4

Physical application

The Mathieu functions we have worked for have important applications in physical scenarios in which we consider elliptical boundaries: the modeling of situations of this type have been investigated far less than situations presenting spherical symmetry and thus we do often see approximate solutions that exploit numerical analysis. For this reason we are now presenting a brief physical example of an exact analytic solution on elliptical boundaries to check the consistency of the developed computational scheme.

4.1 Single layer singular values

According to 'Spectral decompositions and nonnormality of boundary integral operators in acoustic scattering' of Betcke, Phillips and Spence and considering an ellipse with boundary Γ parametrized by $\gamma(\nu) = (a\cosh(\mu)\cos(\nu), a\sinh(\mu)\sin(\nu))$, $\nu \in [0, 2\pi]$, we can write the spectral decomposition of the single layer operator as:

$$\lambda_k^c = \frac{i\pi}{2} Mc_k^{(1)}(\mu, q) Mc_k^{(3)}(\mu, q) \tag{4.1}$$

$$\lambda_k^s = \frac{i\pi}{2} Ms_k^{(1)}(\mu, q) Ms_k^{(3)}(\mu, q) \tag{4.2}$$

where $Mc_i^{(j)} and Ms_l^{(j)}$ stand for the radial Mathieu functions of the i-th and l-th order respectively ($0 \leq i < \infty$, $0 < l < \infty$) and j-th kind ($1 \leq j \leq 4$) ($q = \frac{1}{4}ka$, as stated above).We have also to underline that

$$Mc_j^{(3)} = Mc_j^{(1)} + iMc_j^{(2)} \tag{4.3}$$

and

$$Ms_j^{(3)} = Ms_j^{(1)} + iMs_j^{(2)}. \tag{4.4}$$

Thus we have all the ingredients to obtain the spectral decomposition of S considering the previous formulas 3.1 -3.4 with $j = 1{,}2$ (because we need only $Mc^{(1)}$,$Ms^{(1)}$,$Mc^{(2)}$ and $Ms^{(2)}$). So we consider a numerical solutions of the eigenvalues of the single layer operator obtained through a Galerkin-type method.

The discretized operators used are the following

$$[\mathcal{S}]_{mn} =< \lambda_m, \mathcal{S}^k \lambda_n >$$

$$[\mathcal{G}]_{mn} =< \lambda_m, \lambda_n >$$

in which $< f, g >= \int_\gamma f(\mathbf{r})g(\mathbf{r})d\mathbf{r}$, λ are hat functions defined on the curvilinear abscissa (the support of these are functions are parts of the geometry, they are not straight segments) and the operator $\mathcal{S}$ is defined as

$$(\mathcal{S}^k j)(\mathbf{r}) = \int_\gamma \frac{i}{4} H_0^{(1)}(k||\mathbf{r} - \mathbf{r'}||)j(\mathbf{r'})d\mathbf{r'}.$$

Then we compute $[\mathcal{U}, \mathcal{S}, \mathcal{V}] = svd(\mathcal{G}^{-1/2}\mathcal{S}\mathcal{G}^{-1/2})$ to retrieve the singular values $diag(\mathcal{S})$. Note that the singular values are reordered as a function of the frequency of the singular vectors $\mathcal{V}$

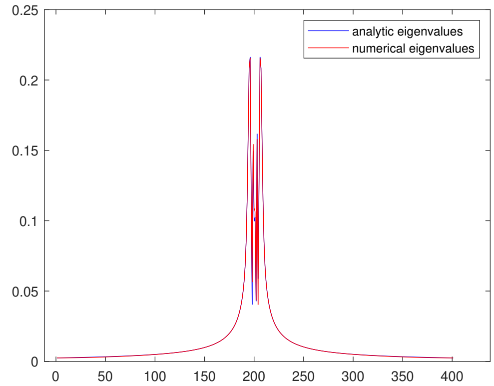

Figure 4.1. Plot the relative error between the analytic eigenvalues obtained through $Mc^{(1)}$ $Mc^{(2)}$ $Ms^{(1)}$ and $Ms^{(2)}$ and the ones obtained through a Galerkin-type method (the mismatch in the central part is due to problems related to the discretization of the mesh).

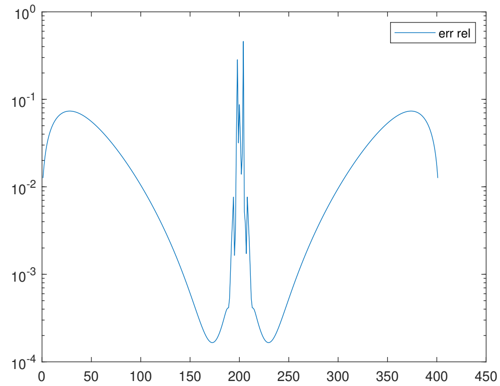

Figure 4.2. Plot the relative error between the analytic eigenvalues obtained through $Mc^{(1)}$ $Mc^{(2)}$ $Ms^{(1)}$ and $Ms^{(2)}$ and the ones obtained through a Galerkin-type method.

Chapter 5

Large-order instability problems

Through the analysis of the spectral decomposition of the single layer operator we discovered instability problems related to large orders of the radial Mathieu functions $Mc^{(2)}$ and $Ms^{(2)}$. Indeed,considering 3.1 -3.4 with j=2, we notice that the terms of the series are made up of a coefficient (C_k) that multiplies a summation of two products between the Bessel function of the first kind and the Bessel function of the second kind, namely:

$$Mc_{2n}^{(2)}(\mu, q) = \sum_{k=0}^{\infty} C_k(\mathcal{J}_{k-s}(u1)\mathcal{Y}_{k+s}(u2) + \mathcal{J}_{k+s}(u1)\mathcal{Y}_{k-s}(u2))$$

for Mc_{2n}, analogously for Mc_{2n+1}, Ms_{2n} and Ms_{2n+1}. For this reason, when we consider a high order eigenvalue, that corresponds to a high order $Mc^{(2)}$ and $Ms^{(2)}$, we have to deal with higher orders of $\mathcal{J}$ and $\mathcal{Y}$ (if we set s to improve series convergence): in the first case ($\mathcal{J}$) we are dealing with something that goes rapidly to 0 whereas, in the second case ($\mathcal{Y}$), we have something that rapidly grows to ∞.Through a debug analysis we discovered that the problem is an overflow error related to the evaluation itself of $\mathcal{Y}$ and not to the summation: this is due to the fact that we have to evaluate functions $\mathcal{Y}$ with larger and larger orders (since the coefficients with a relevant magnitude are concentrated in the neighbourhood of the diagonal of the matrix of the Fourier coefficients) that assume very large values farther and farther from 0 ($\mathcal{Y}$ has a singularity in 0); so we understand that considering a high frequency, that is a large k, and mostly a "circular" ellipse, that is a large μ, gives us stability since $\mathcal{Y}$ is evaluated in $u2 = \sqrt{q}\exp{(\mu)}$ (see 3.1 3.2 3.3 3.4).

5.1 Strategy based on large-order asymptotic expansion of $\mathcal{J}$ and $\mathcal{Y}$

A possible solution is that we could set s equal to 1 and use an asymptotic formulation of $\mathcal{J}$ and $\mathcal{Y}$ for high order. Indeed, according to 9.3.1 of 'Handbook of mathematical functions', for large ν we have:

$$\mathcal{J}_\nu(u) \sim \sqrt{\frac{1}{2\pi\nu}}\left(\frac{eu}{2\nu}\right)^\nu \tag{5.1}$$

and

$$\mathcal{Y}_\nu(u) \sim -\sqrt{\frac{2}{\pi\nu}}\left(\frac{eu}{2\nu}\right)^{-\nu}. \tag{5.2}$$

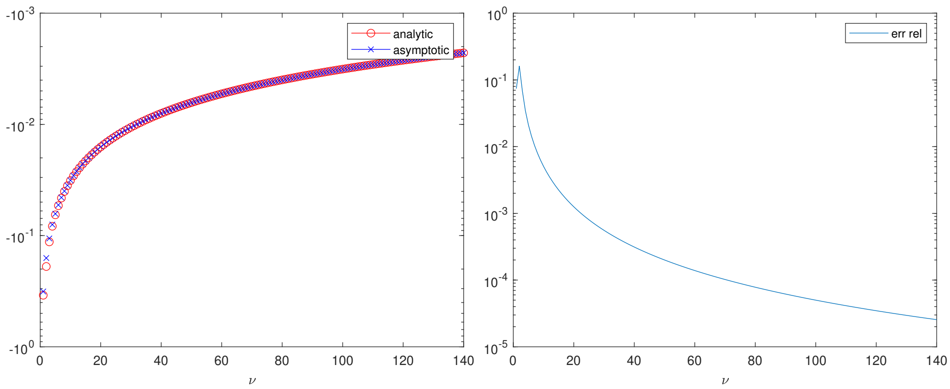

Figure 5.1. approximation of the product between $\mathcal{J}_\nu(\mu)$ and $\mathcal{Y}_\nu(\mu)$ with $\mu = 1$

Moreover if we consider a larger μ the approximation becomes less efficient but, in that case, we can exploit that $Mc_n^{(1)}(\mu, q)$ and $Mc_n^{(2)}(\mu, q)$, as μ goes to infinity, are asymptotically defined as the elliptic analogues of $\mathcal{J}_n$ and $\mathcal{Y}_n$ respectively (NIST Digital Library, eq. 28.20.11-12); same thing for $Mc_n^{(3)}(\mu, q)$ and $Mc_n^{(4)}(\mu, q)$ respectively with the Hankel functions $\mathcal{H}_n^{(1)}$ and $\mathcal{H}_n^{(2)}$ (NIST Digital Library, eq. 28.20.9-10,28.20.15), namely:

$$Mc_n^{(1)}(\mu, q) \sim \mathcal{J}_n(2q^{1/2}\cosh\mu) + \exp(\Im(2q^{1/2}\cosh\mu))\mathcal{O}(\operatorname{sech}\mu^{3/2})$$

$$Mc_n^{(2)}(\mu, q) \sim \mathcal{Y}_n(2q^{1/2}\cosh\mu) + \exp(\Im(2q^{1/2}\cosh\mu))\mathcal{O}(\operatorname{sech}\mu^{3/2})$$

$$Mc_n^{(3)}(\mu, q) \sim \mathcal{H}_n^{(1)}(2q^{1/2}\cosh\mu)(1 + \mathcal{O}(\operatorname{sech}\mu))$$

$$Mc_n^{(4)}(\mu, q) \sim \mathcal{H}_n^{(2)}(2q^{1/2}\cosh\mu)(1 + \mathcal{O}(\operatorname{sech}\mu)).$$

We observe analogous relations for $Ms_n^{(j)}$ with $0 \leq j \leq 4$ by exploiting NIST Digital Library eq. 28.20.16. Moreover, although we gain a more and more stability considering a large μ we have to underline that this leads to loss of the physical meaning of what we are doing: a large μ corresponds to a less and less elliptical ellipse and, for this reason, we have to focus on small values of μ just because the original aim was to give an analytic solutions to problems that are just approximated in a circular framework.

Considering 5.1 5.2 we can avoid any overflow errors through a trivial simplification by hand : indeed, setting s=1, we have

$$Mc_{2n}^{(2)}(\mu, q) = \sum_{k=0}^{\infty} C_k(\mathcal{J}_{k-1}(u1)\mathcal{Y}_{k+1}(u2) + \mathcal{J}_{k+1}(u1)\mathcal{Y}_{k-1}(u2))$$

and , exploiting 5.1 and 5.2 we can simplify, for a sufficient magnitude of k, terms of the series ($\mathcal{S}_k$ stands for the k-th term of the series)

$$\mathcal{S}_k(\mu, q) = C_k \left(-\frac{4k}{\pi} u2^{-2} \left(\frac{u1}{u2}\right)^{k-1} - \frac{1}{\pi 4k^3} u1^2 \left(\frac{u1}{u2}\right)^{k-1} \right)$$

and, using the definition of u1 and u2, it turns to

$$\mathcal{S}_k(\mu, q) = C_k \left(-\frac{4k}{q\pi} \exp(-2\mu k) - \frac{q}{\pi 4k^3} \exp(-2\mu k) \right). \tag{5.3}$$

analogously for Mc_{2n+1}, Ms_{2n} and Ms_{2n+1}. The problem of this approach lies in the fact that, considering 3.1 -3.4, we divide by A_{2s}^{2n},A_{2s+1}^{2n+1},B_{2s}^{2n} and B_{2s+1}^{2n+1}, but, if we consider s=1, this coefficient goes rapidly to 0 as n increases : with a small q,the only coefficients of the Fourier series that are not seen as 0 by Matlab are concentrated where $s = n$, but even when q is big , if we consider a large n, we see that ,for small s, all the coefficients are equal to 0. Moreover, if we consider $s = r$ to ensure we do not divide by 0, we can't use the asymptotic expansions of the Bessel functions for both the functions $\mathcal{J}$ and $\mathcal{Y}$ and consequently we cannot simplify by hand the terms as in 5.3 to avoid computational errors.

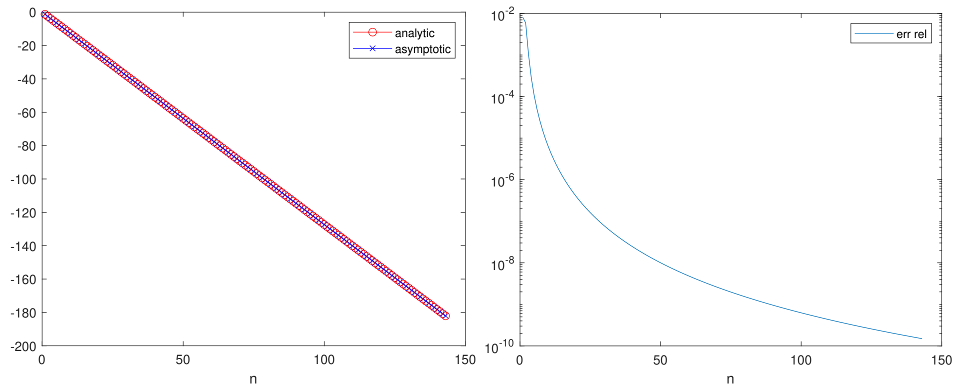

Figure 5.2. approximation of the product between $\mathcal{J}_{k-1}(\mu)$ and $\mathcal{Y}_{k+1}(\mu)$ with $\mu = 1$ using 5.1 and 5.2

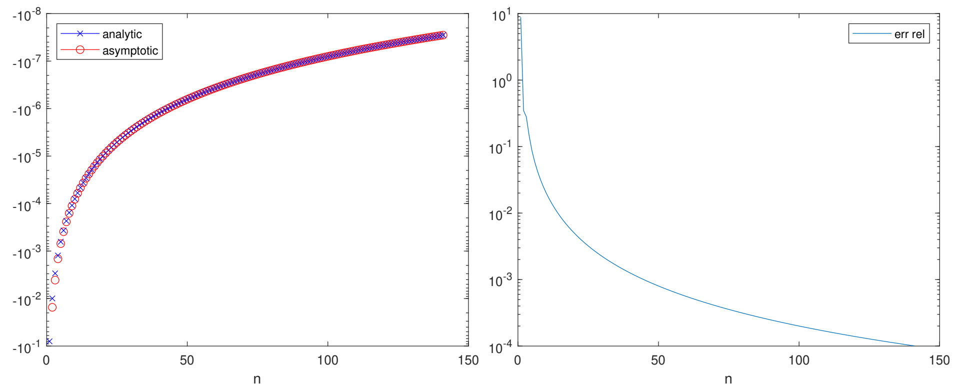

Figure 5.3. approximation of the product between $\mathcal{J}_{k+1}(\mu)$ and $\mathcal{Y}_{k-1}(\mu)$ with $\mu = 1$ using 5.1 and 5.2

5.2 $Ms_n^{(1)} Ms_n^{(2)}$ and $Mc_n^{(1)} Mc_n^{(2)}$ experimental large-order asymptotic expansion

Another interesting thing we observed is related to an asymptotic expansion for large order of the product $Mc_n^{(1)} Mc_n^{(2)}$ and $Ms_n^{(1)} Ms_n^{(2)}$: during the various test we performed we noticed a pattern which made us think that the spectrum of S goes to 0 as $\mathcal{O}(1/n)$. So we carried out some tests to check this fact and also to evaluate the constant factor. The importance of those two products derives by the fact that in the spectral decomposition of S, recalling 4.1 4.2 4.3 4.4 we can write :

$$\lambda_n^c = \frac{i\pi}{2}\left(\left(Mc_n^{(1)}(\mu, q)\right)^2 + iMc_n^{(1)}(\mu, q) Mc_n^{(2)}(\mu, q)\right) \tag{5.4}$$

and

$$\lambda_n^s = \frac{i\pi}{2}\left(\left(Ms_n^{(1)}(\mu,q)\right)^2 + iMs_n^{(1)}(\mu,q)Ms_n^{(2)}(\mu,q)\right). \tag{5.5}$$

and, observing that $\left(Mc_n^{(1)}(\mu,q)\right)^2$ and $\left(Ms_n^{(1)}(\mu,q)\right)^2$ go rapidly to 0 as n increases, we understand that, to fully characterize the spectrum of S for large orders, we have only to know the second term of the sum, that is $Mc_n^{(1)}Mc_n^{(2)}$ and $Ms_n^{(1)}Ms_n^{(2)}$.

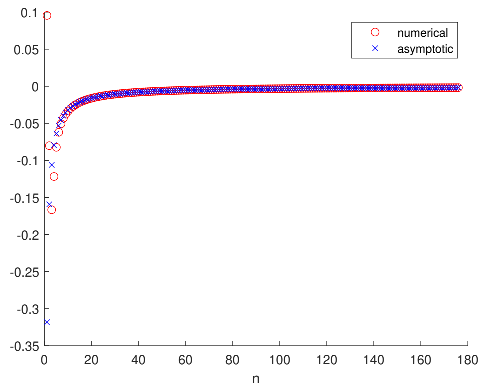

Figure 5.4. plot of $Ms_n^{(1)}Ms_n^{(2)}$ and $-\frac{1}{n}$ with $\mu = 1.1$ and $k = 4$

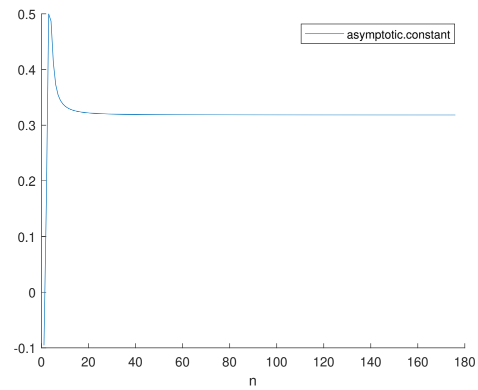

Figure 5.5. computation of the asymptotic constant: 3.183562116059216e-01 last value computed

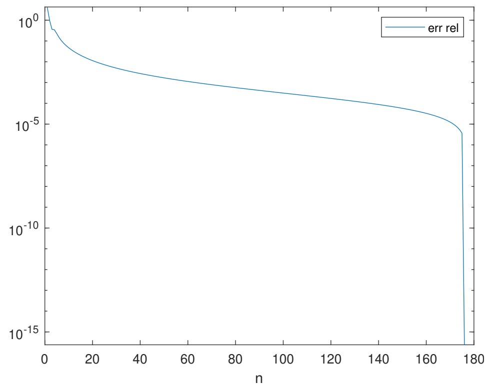

Figure 5.6. relative error between $Ms_n^{(1)}Ms_n^{(2)}$ and $-\frac{0.3183562116059216}{n}$

I tested this asymptotic expansion with values of μ and k around 1: if we consider a very large k (about $\geq 10^2$) we observe that $\left(Mc_n^{(1)}(\mu,q)\right)^2$ and $\left(Ms_n^{(1)}(\mu,q)\right)^2$ go to 0 for large orders but always more rapidly than $Mc_n^{(1)}(\mu,q)Mc_n^{(2)}(\mu,q)$ and $Ms_n^{(1)}(\mu,q)Ms_n^{(2)}(\mu,q)$ (moreover $\left(Mc_n^{(1)}(\mu,q)\right)^2$ and $\left(Ms_n^{(1)}(\mu,q)\right)^2$ are small if compared to $Mc_n^{(1)}(\mu,q)Mc_n^{(2)}(\mu,q)$ and $Ms_n^{(1)}(\mu,q)Ms_n^{(2)}(\mu,q)$).

To summarize

- large order instability of the product of the Bessel functions of the second kind and , as a consequence, large order instability of $Mc_n^{(2)}(\mu,q)$ and $Ms_n^{(2)}(\mu,q)$;
- simplification using the large order asymptotic expansion for the Bessel functions of the first and second kind;
- impossibility of using the asymptotic expansion since we cannot set s equal to a small value;
- possible large order asymptotic expansion for the products $Mc_n^{(1)}(\mu,q)Mc_n^{(2)}(\mu,q)$ and $Ms_n^{(1)}(\mu,q)Ms_n^{(2)}(\mu,q)$;
- larger μ and k (wavenumber) larger the orders which correspond to a good approximation of $Mc^{(1)}(\mu,q)Mc^{(2)}(\mu,q)$ and $Ms^{(1)}(\mu,q)Ms^{(2)}(\mu,q)$ with $-\frac{0.3183562116059216}{n}$.

Appendix A

Ce_n and Se_n implementation

As previously observed the Mathematica built-in implementation of Ce_n presents instability problem. So, after having developed and implemented a computational scheme for the radial functions in Matalab we noticed that,recalling the definition 2.9, we can use the Fourier expansion of ce_n substituting ν with $i\nu$ (it is interesting to observe that the modified Mathieu's equation is just a rotation of the angular equation in the complex plane) and we obtain:

$$Ce_{2n+p}(\nu, q) = \sum_{m=0}^{\infty} A_{2m+p}^{2n+p} \cosh((2m+p)\nu), \quad n \geq 0, p = 0{,}1 \tag{A.1}$$

Recalling 2.10, we can reason in the same way for Se_n obtaining:

$$Se_{2n+p}(\nu, q) = \sum_{m=0}^{\infty} B_{2m+p}^{2n+p} \sinh((2m+p)\nu), \quad n \geq 0, p = 1{,}2 \tag{A.2}$$

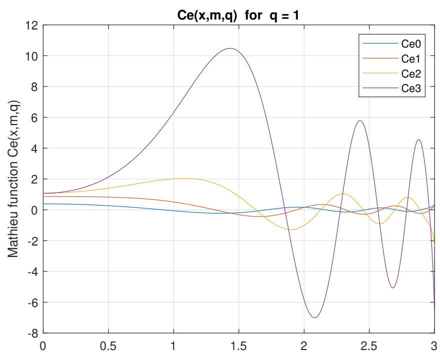

Figure A.1. Plot of Ce_m with $0 \leq m \leq 3$ with q=1

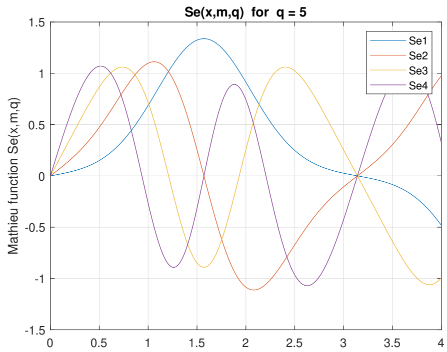

Figure A.2. Plot of Se_m with $1 \leq m \leq 4$ with q=5

Bibliography

[1] Malcom M. Bibby and Andrew F. Peterson. *Accurate computation of Mathieu functions.* Morgan & Claypool publisher, 1963.

[2] G. Blanch and D. S. Clemm. *Tables Relating to the Radial Mathieu Functions, Volume 1.* U.S. Government Printing Office, 1962.

[3] G. Blanch and D. S. Clemm. *Tables Relating to the Radial Mathieu Functions, Volume 2.* U.S. Government Printing Office, 1963.

[4] D.B.Hodge. The calculation of the eigenvalues and eigenfunctions of Mathieu's equation. January 1, 1972.

[5] Irene Stegun and Milton Abramowitz. *Handbook of mathematical functions.* United States Department of Commerce, National Bureau of Standards, 1972.

[6] Howard B. Wilson and Robert W. Scharstein. Computing elliptic membrane high frequencies by mathieu and galerkin method.