ULB
Faculté des Sciences Appliquées

SCK•CEN
THE BELGIAN NUCLEAR RESEARCH CENTRE
Reactor Physics & MYRRHA

Neutron Transport with Anisotropic Scattering

Theory and Applications

Gert Van den Eynde

Promotor : Prof. Dr. Ir. R. Beauwens
Co-Promotor: Prof. Dr. Ir. E. Mund

Thesis submitted in candidature for the degree of Doctor in Applied Sciences

May 2005

Understanding grows only logarithmically with
the number of floating point operations.

— J.P. BOYD, *Chebyshev and Fourier Spectral Methods, 2000*

This one's tricky. You have to use imaginary
numbers, like eleventeen. . .

— CALVIN, *Calvin and Hobbes by Bill Watterson*

Summary

This thesis is a blend of neutron transport theory and numerical analysis. We start with the study of the problem of the Mika/Case eigenexpansion used in the solution process of the homogeneous one-speed Boltzmann neutron transport equation with anisotropic scattering for plane symmetry. The anisotropic scattering is expressed as a finite Legendre series in which the coefficients are the "scattering coefficients". This eigenexpansion consists of a discrete spectrum of eigenvalues with its corresponding eigenfunctions and the continuous spectrum $[-1, +1]$ with its corresponding eigendistributions. In the general case where the anisotropic scattering can be of any (finite) order, multiple discrete eigenvalues exist and these have to be located to have the complete spectrum. We have devised a stable and robust method that locates all these discrete eigenvalues. The method is a two-step process: first the number of discrete eigenvalues is calculated and this is followed by the calculation of the discrete eigenvalues themselves, now being able to count them down and make sure none are forgotten.

During our numerical experiments, we came across what we called near-singular eigenvalues: discrete eigenvalues that are located extremely close to the continuum and hence lead to near-singular behaviour in the eigenfunction. Our solution method has been adapted and allows for the automatic detection of such a near-singular eigenvalue.

For the elements of the continuous spectrum $[-1, +1]$, there is no non-zero function satisfying the associated eigenequation but there is a non-zero distribution that does satisfy it. It is not feasible to compute a distribution as such but one can evaluate integrals in which this distribution appears. The continuum part of the eigenexpansion can hence only be characterised by its (angular) moments. Accurate and fast numerical quadrature is needed to evaluate these integrals. Several quadrature methods have been evaluated on a representative test function.

The eigenexpansion was proved to be orthogonal and complete and hence can be used to represent the infinite medium Green's function. The lat-

ter is the building block of the Boundary Sources Method, an integral solution method for the neutron transport equation. Using angular and angular/spatial moments of the Green's function, it is possible to solve with high accuracy slab problems. We have written a one-dimensional slab code implementing this Boundary Sources Method allowing for media with arbitrary order anisotropic scattering. Our results are very good and the code can be considered as a benchmark code for others.

As a final application, we have used our code to study the discrete spectrum of a well-known scattering kernel in radiative transfer, the Henyey-Greenstein kernel. This kernel has one free parameter which is used to fit the kernel to experimental data. Since the kernel is a continuous function, a finite Legendre approximation needs to be adopted. Depending on the free parameter, the approximation order and the number of secondaries per collision, the number of discrete eigenvalues ranges from two to thirty and even more. Bounds for the minimum approximation order are derived for different requirements on the approximation: non-negativity, an absolute and relative error tolerance.

Thank you!

I would like to thank everyone who has contributed from far and close to the realisation of this work.

First of all, I would like to thank my supervisors, Professor Beauwens and Professor Mund, for their valuable contributions, their advice and, above all, their friendship.

A thank you also goes out to my colleagues and friends at the Reactor Physics & MYRRHA department at SCK•CEN. Especially to Dr. Hamid Aït Abderrahim, head of department and MYRRHA project leader, who has always supported me and my work. I would also like to thank the neutronics team, and especially Dr. Edouard Malambu, who taught me the tricks of the trade. And for sure everyone for the friendly atmosphere and the usual non-sense during lunch…

I owe a lot of gratitude to the Library Team at SCK•CEN. Without their help in retrieving papers and books, this work would have been non-existent. It's nice to know you can rely on these professionals!

I would also like to thank the management of SCK•CEN and its board of directors for allowing people to specialise themselves and pursue a PhD thesis. I firmly believe this is a win-win situation.

Words do not suffice to describe the support and love of my wife and son. Family is the greatest gift one can ever imagine.

Thank you!

Gert,
Mol, April 15, 2005

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

In the beginning...

The starting point for this work was the lack of a time-dependent transport code that could handle the peculiarities of an Accelerator Driven System (ADS): the sub-criticality and the external source. SCK•CEN was in need for this kind of code to perform studies on the dynamic behaviour of their ADS in design, MYRRHA (`http://www.sckcen.be/myrrha`). It was decided to start a collaboration with the Service de métrologie nucléaire at the ULB under the direction of Professor R. Beauwens. He is the person responsible for the neutronics module in the code HEXNODYN, a likely candidate code for the problem at hand. The idea was to modify HEXNODYN so that it could handle subcritical systems driven by a time-dependent external source.

MYRRHA, *an Accelerator Driven System*

MYRRHA is an Accelerator Driven System (ADS) under development at Mol in Belgium. It is aiming to serve as a basis for the European XT-ADS (eXperimental demonstration of Transmutation in ADS) and to provide protons and neutrons for various R&D applications. It consists of a proton accelerator delivering a 350 MeV - 5 mA proton beam to a liquid Pb-Bi spallation target that in turn couples to a Pb-Bi cooled, subcritical fast nuclear core. The project started in 1997 and the aim is to put MYRRHA in service in 2014-2015.

The purpose of the Accelerator Driven System MYRRHA is to serve as a multi-purpose irradiation facility for

- ADS technological demonstration

- Waste transmutation studies of minor actinides and long lived fission products
- Structural material studies for nuclear reactors, fusion and ADS type reactors
- Radioisotope production for medical & industrial applications
- Proton beam applications.

MYRRHA consists of a sub-critical MOX-fuelled core that is driven by an external neutron source generated by the spallation of atoms in the liquid lead-bismuth target material by bombarding the latter with high energy (350 MeV - 5 mA) protons coming from a particle accelerator. The core will produce about 50 MW of thermal power for a $k_{eff} \approx 0.95$.

The HEXNODYN *code*

HEXNODYN, a three-dimensional multi-group transport and diffusion code in hexagonal-z geometry for fast reactor transient analysis has been jointly developed by Siemens-KWU (Federal Republic of Germany) and the Université Libre de Bruxelles (Belgium) under contract with the Joint Research Centre at Ispra (Italy) in the late eighties as part of a European effort in producing standardised software for reactor safety analysis. The code was foreseen to become the spatial-kinetics neutronics module of the EAC-2 code (European Accident Code). Unfortunately, the thermo-hydraulic part of EAC-2 was never fully completed and the coupling between the two wasn't really tested.

The main static part of HEXNODYN was developed in the late eighties by R. Wagner [Wag89]. The method is a nodal discrete ordinates method, falling in the family of interface current nodal methods [Bad90]. These methods are based on a spatial transverse integration of the multidimensional transport equation. This procedure leads to a set of coupled one-dimensional equations. The coupling between the different equations is assured by the so-called transverse leakage terms. The reasoning behind this approach is that a coupled set of one-dimensional equations is easier to solve than a single multi-dimensional equation. The work horse of these type of codes is thus the one-dimensional solver. If the one-dimensional solver is not accurate enough, one cannot expect a good result on the multi-dimensional problem.

The time-dependent part of the code is based on the quasi-static approach [ON85; Het93] in the SCK•CEN code CASSANDRE [ASDM84; AD86]. The merge of the two is reported in [BDM+90]. The next step was the addition of anisotropic sources [Bea94]. However, the external source was still

static. For the simulation of an Accelerator Driven System where beam trips – the sudden drop of the proton beam coming from the accelerator – happen quite often, this needed improvement. The implementation of the time-dependent external source was done during a Master's thesis [VdE01].

The need for anisotropic scattering

We used the DIF3D 8.0 code to study the reference MYRRHA core and some perturbations thereof [VdE03]. One of the calculations done was the assessment of the effect of anisotropic scattering on the k_{eff} calculation. We summarise the results in Table 1.1. It is clear that by ignoring the anisotropic scattering, the results are completely wrong. It is this behaviour that urged us to make sure HEXNODYN could treat anisotropic scattering.

Anisotropy	k_{eff}
isotropic scattering	0.999459
first order	0.947052
third order	0.947143

Table 1.1: *Effect of anisotropic scattering on k_{eff} for the* MYRRHA *core.*

The Boundary Sources Method

The Boundary Sources Method is an integral transport method which makes capital out of the exact knowledge of a transport kernel. This limits the method to simple geometries like planar, spherical or cylindrical geometry. The gist of the method is to represent the solution to an inhomogeneous finite slab problem as a convolution of a transport kernel with two sources located at the boundaries of the medium – the boundary sources – combined with a particular solution of the inhomogeneous problem in analogous infinite medium.

The transport kernel we decided to use is the infinite medium Green's function. The kernel can be built using the Mika/Case eigenexpansion. Associated with this eigenexpansion are a set of discrete eigenvalues and their corresponding eigenfunctions and the continuum $[-1, +1]$ which is also an "eigenvalue" but with a corresponding eigendistribution. The eigenexpansion has been proved orthogonal and complete and hence can be used to represent solutions to other problems, like the Green's function. Since there

is a distribution present, the (continuum part of) expansion cannot be numerically calculated unless it appears inside an integral expression.

Solving the finite slab problem boils down to the calculation of the boundary sources which, clearly, only have an angular dependency. A discretised version of the Boundary Sources equations is obtained by representing the boundary sources by a finite angular Legendre expansion of order M, which we also call the order of the method. To have a closed system of equations, one requires continuity of the first $M+1$ angular moments at internal boundaries together with boundary conditions at the outer left and right cells of a slab. This fits quite nicely with the expression of the Green's function using the eigenexpansion, since we can calculate angular and angular/spatial moments of this expansion.

As an improvement over Wagner's method, Vandevelde [VdV95] implemented a Boundary Sources Method for isotropic scattering in HEXNODYN as the basic one-dimensional solver module. In isotropic scattering, only one parameter determines the properties of a cell (if the spatial coordinate is expressed in free-mean-paths the total cross section equals one): c, the number of secondaries per collision. For the continuum part, one needs to evaluate an integral over ν where the only remaining parameters are c and x, the position in the cell, Vandevelde developed a two-dimensional approximation for the integral taking as input c and x. By this approach he avoided costly numerical quadrature.

In order to extend the capabilities of HEXNODYN to anisotropic scattering, it was clear that we needed to continue the work on the boundary sources method to extend it to anisotropic scattering. At first we were only looking at first order anisotropic scattering. However, we quickly decided to go for the general problem of arbitrary order anisotropic scattering. Both to allow for more eccentric applications but also from the point of view of academic research.

We soon saw that the results of the Boundary Sources Method we implemented were very promising. At that moment, we decided to try to go for gold and have the code as accurate as possible using classical IEEE double precision arithmetic, i.e. without resorting to symbolic multi-precision packages. The goal was now to create a highly accurate benchmark code. A goal which, I think, has been achieved.

As a consequence of the aim to allow for arbitrary order anisotropic scattering, a cell has a lot more parameters than c and the total cross section Σ_t, namely the scattering coefficients describing the scattering law applicable in the cell. Due to this more general problem, the approach of Vandevelde

is not possible anymore: one cannot build "cheap" approximations when so many parameters are at play.

The work. . .

The work can be divided in two major parts: the accurate and fast calculation of the angular and angular/spatial moments of the infinite medium Green's function transport kernel and two applications, a multi-cell slab Boundary Sources Method and an analysis of the discrete spectrum of the Henyey-Greenstein scattering kernel.

In the next chapter, we give an overview of the basic theory of the eigenexpansion and its properties. We give the derivation of the eigenexpansion and show how the discrete part and continuum part are defined. The discrete eigenvalues are the roots of the characteristic equation, a highly nonlinear equation. Several properties of the discrete eigenvalues are given. The general solution of the homogeneous one-speed Boltzmann neutron transport equation as well as the infinite medium Green's function are expressed using the eigenexpansion.

In the derivation of the eigenexpansion, a recurrence relation pops up defining a set of polynomials. These polynomials, their properties and evaluation methods are the subject of Chapter 3. It can be proved that these polynomials are orthogonal and their zeros are related to the discrete eigenvalues. Algorithms are proposed to evaluate these polynomials in a stable fashion.

The fourth chapter is fully devoted to the discrete spectrum. A two-step method is proposed to calculate all discrete eigenvalues. Indeed, as one can show using the argument principle, a classical result in complex analysis, the number of discrete eigenvalues is related to the number of roots of a certain polynomial, which we call the characteristic polynomial, in the interval $[0, +1]$. The first step of the process is to count this number of roots. In a second phase, the actual discrete eigenvalues are calculated by solving the characteristic equation. Since we know the number of discrete eigenvalues, we can do this in a structured fashion. During our numerical experiments, we came across what we call near-singular eigenvalues: discrete eigenvalues that are located so close to the continuum they give rise to near-singular behaviour in the discrete eigenfunction. We have modified our code so that the presence of such near-singular eigenvalues is detected and they can be calculated with the same accuracy as normal eigenvalues. Finally, expressions for discrete part of the Green's function are given together with the

expressions for angular and angular/spatial moments needed in the Boundary Sources Method.

The eigenexpansion is not complete without the continuum part. The calculation of the angular and angular/spatial moments of the continuum part of the Green's function is the subject of Chapter 5. This calculation requires the evaluation of integrals of which no closed expression exists and in contrary to the isotropic case there are too many parameters to allow for a good approximation. The only solution here is to go for numerical quadrature. Using a test function that has the same properties as the real integrals appearing in the eigenexpansion, we have selected numerical quadrature routines. The angular/spatial moments pose an extra problem due to round-off effects and need special treatment.

Chapter 6 presents the first application of the foregoing theory and calculational methods: the Boundary Sources Method in planar geometry (multicell slab) allowing for arbitrary order scattering in the media of the different cells. The method is summarised together with its discrete version and the continuity relations and boundary condition relations. This leads to a linear system to be solved. Five test cases show the power of the method where we compare with other deterministic transport codes and the stochastic Monte Carlo code nowadays considered as the reference, MCNPX.

The next chapter presents a study of the discrete spectrum of the Henyey-Greenstein scattering kernel, a kernel originating in radiative transfer and photon scattering. This kernel is a continuous function with one free parameter that can be used to fit experimental data. First of all, an expression for the Legendre coefficients is derived. Next to the free parameter, the order of approximation determines the scattering function used in the Boltzmann equation. We derive expressions for the minimal approximation order needed for three requirements on the approximation: non-negativity and an absolute and relative error tolerance. Non-negativity is very important, especially when using the scattering function in Monte Carlo codes where the scattering function is used as a probability density function and hence a negative value would lead to very strange results.

Finally, Chapter 8 gives more details about the programming and the resulting code, a C++ library. We describe the different classes and their inter-relations. We also provide the results of a profiling run showing where the code can be more optimised in the future.

CHAPTER 2

The homogeneous neutron transport equation with anisotropic scattering

SUMMARY– We present the basic homogeneous one-speed Boltzmann neutron transport equation with anisotropic scattering for plane symmetry. Solving this equation forms the building block of many modern day multi-group nodal codes. A formal solution is derived using the eigenexpansion first introduced by Mika and Case. The eigenexpansion consists of a discrete set of eigenvalues and corresponding eigenfunctions complemented by the continuum $[-1, +1]$ with corresponding eigendistributions. Depending on the problem, there may be a number of discrete eigenvalues. The eigenexpansion can be shown to be orthogonal and complete from which one can conclude that every solution to the homogeneous one-speed Boltzmann neutron transport equation with anisotropic scattering for plane symmetry can be expressed in a unique manner as a linear combination of the eigenfunctions and eigendistribution. Finally, the Green's function for an infinite medium is composed using the eigenexpansion.

2.1 The formal solution of the Boltzmann equation

In this section we shall retake the analysis first performed by Mika [Mik61] and later summarised in the classical textbook by Case and Zweifel [CZ67]. It will lead us to an eigenvalue decomposition of the general solution of the homogeneous one-speed Boltzmann neutron transport equation in plane geometry. The one assumption made is that the scattering kernel can be expressed as a *finite* series of Legendre polynomials. This is, however, a classical assumption, honoured by all modern day computer codes (since

infinite series cannot be represented on a computer).

The homogeneous one-speed Boltzmann neutron transport equation with anisotropic scattering is given by

$$\mu \frac{\partial \psi(x,\mu)}{\partial x} + \Sigma_t \psi(x,\mu) = c\Sigma_t \int_{4\pi} F(\mathbf{\Omega}' \cdot \mathbf{\Omega}) \psi(x,\mathbf{\Omega}') d\mathbf{\Omega}' \tag{2.1}$$

where we use the following notation:

x the spatial coordinate,

μ the angular coordinate, equal to $\mathbf{\Omega} \cdot \mathbf{1}_x$

Σ_t the total macroscopic cross section,

c the number of secondary neutrons per collision and

$F(\mathbf{\Omega}' \cdot \mathbf{\Omega})$ the scattering function or scattering kernel.

As mentioned before, we assume that we can express the scattering kernel as a finite series of Legendre polynomials:

$$F(\mathbf{\Omega}' \cdot \mathbf{\Omega}) \approx f(\mathbf{\Omega}' \cdot \mathbf{\Omega}) = \sum_{l=0}^{N} \frac{2l+1}{4\pi} f_l P_l(\mathbf{\Omega}' \cdot \mathbf{\Omega}) \tag{2.2}$$

where N denotes the order of the approximation and the f_l are the Legendre coefficients.

The scattering kernel needs to be normalised and this is typically done by requiring $f_0 \equiv 1$. Putting the Legendre expansion (2.2) of the scattering kernel in the transport Equation (2.1), we get

$$\mu \frac{\partial \psi(x,\mu)}{\partial x} + \Sigma_t \psi(x,\mu) = c\Sigma_t \sum_{l=0}^{N} \frac{2l+1}{4\pi} f_l \int_{4\pi} \psi(x,\mathbf{\Omega}') \, P_l(\mathbf{\Omega}' \cdot \mathbf{\Omega}) d\mathbf{\Omega}'. \tag{2.3}$$

By applying the addition theorem for spherical harmonics [MOS66] and simplifying further we arrive at

$$\mu \frac{\partial \psi(x,\mu)}{\partial x} + \Sigma_t \psi(x,\mu) = \frac{c\Sigma_t}{2} \sum_{l=0}^{N} (2l+1) f_l P_l(\mu) \int_{-1}^{+1} \psi(x,\mu') \, P_l(\mu') d\mu'. \tag{2.4}$$

The gist of this formal solution procedure is to introduce the *ansatz**:

$$\psi(x,\mu) = \phi_\nu(\mu) e^{-\frac{\Sigma_t x}{\nu}} \tag{2.5}$$

*The same result can be achieved by the classical approach of separation of variables

where we call the $\phi_\nu(\mu)$ the "eigenfunctions" and the corresponding ν the "eigenvalues". Equation (2.4) then becomes

$$(\nu - \mu)\phi_\nu(\mu) = \frac{c\nu}{2}\sum_{l=0}^{N}(2l+1)f_l P_l(\mu)\int_{-1}^{+1}\phi_\nu(\mu')P_l(\mu')d\mu'. \tag{2.6}$$

We multiply Equation (2.6) by the Legendre polynomial $P_k(\mu)$ and integrate over $\mu = -1 \rightarrow +1$. We use the short-hand

$$\phi_k(\nu) \equiv \int_{-1}^{+1} P_k(\mu)\phi_\nu(\mu)d\mu \tag{2.7}$$

and from the orthogonality properties of the Legendre polynomials (Eq.A.13) we have

$$\int_{-1}^{+1} P_l(\mu)P_k(\mu)d\mu = \frac{2}{2l+1}\delta_{lk} \tag{2.8}$$

and

$$\mu P_l(\mu) = \frac{1}{2l+1}\left((l+1)\,P_{l+1}(\mu) + lP_{l-1}(\mu)\right). \tag{2.9}$$

Using Equations (2.7)–(2.9) in the transport equation with ansatz Eq. (2.6), we find the following recurrence relation:

$$\nu\,(1 - cf_k)\,\phi_k(\nu) - \frac{k+1}{2k+1}\phi_{k+1}(\nu) - \frac{k}{2k+1}\phi_{k-1}(\nu) = 0. \tag{2.10}$$

All $\phi_k(\nu)$ are proportional to $\phi_0(\nu)$ and hence it is convenient to normalize the $\phi_k(\nu)$ so that $\phi_0(\nu) \equiv 1$. The higher order $\phi_k(\nu)$ are functions of k, ν, c and f_k, the Legendre coefficients of the scattering kernel. Using the definition of the $\phi_k(\nu)$ in Equation (2.6) we find

$$(\nu - \mu)\,\phi_\nu(\mu) = \frac{c\nu}{2}\sum_{l=0}^{N}(2l+1)\;f_l\;\phi_l(\nu)\;P_l(\mu). \tag{2.11}$$

Since the right-hand-side of this equation is a known (calculable) function, the general solution for the $\phi_\nu(\mu)$ is given by:

$$\phi_\nu(\mu) = \frac{c\nu}{2}P\frac{1}{\nu - \mu}\sum_{l=0}^{N}(2l+1)\;f_l\;\phi_l(\nu)\;P_l(\mu) + \lambda(\nu)\delta(\nu - \mu) \tag{2.12}$$

where P denotes the Cauchy principal value and the $\phi_\nu(\nu)$ are subject to the normalisation condition

$$\int_{-1}^{+1} \phi_\nu(\mu) d\mu \; = \; 1. \tag{2.13}$$

The solution (2.12) has two terms: the first one corresponds to values of ν not in the interval $\mu = -1 \ldots + 1$ (the Dirac term disappears) and one corresponding to this interval so that ν can become equal to μ. This makes that this general solution is not calculable except when it appears inside an integral. The solution is not a function as such, but a distribution [GC62]. So in general, we say that there exist discrete eigenvalues, when $\nu \notin [-1, +1]$, and there exists a continuum of eigenvalues $\nu = [-1, +1]$.

2.2 Scattering kernels

A scattering kernel is a function defined on $\mu \in [-1, +1]$ describing the probability distribution of the (cosine of) the angle between the incoming and outgoing direction of the scattered neutron. Throughout this text, we shall use two scattering kernels in the examples showing certain aspects of the theory, namely the Henyey-Greenstein scattering kernel [HG41] and the binomial scattering kernel [KSV70; Sie80].
The Henyey-Greenstein kernel is defined by

$$HG(\mu|g) = \frac{1 - g^2}{2\left(1 - 2\mu g + g^2\right)^{3/2}}$$

where g is a free parameter in the range $g \in [-1, +1]$. A more thorough study of this kernel is done in Chapter 7. An interesting fact that we need further on is that there exists a simple expression for the Legendre moments of this kernel, namely

$$f_l = g^l$$

where f_l denotes the k-th Legendre moment defined by

$$f_l = \int_{-1}^{+1} HG(\mu|g) P_l(\mu) d\mu.$$

For a proof of this property, see Chapter 7.

The binomial scattering kernel is given by

$$Bi^N(\mu) = \frac{N+1}{2^N}(1 + \mu)^N \tag{2.14}$$

where N is the order of the scattering law. The Legendre moments are given by the following first-order recurrence relation [Sie80]

$$f_l^N = \left(\frac{N+1-l}{N+1+l}\right) f_{l-1}^N \tag{2.15}$$

with the initial value $f_0^N = 1$. Since the scattering law is a polynomial of degree N in μ, we have $f_l^N = 0$ for $l > N$. The recurrence (2.15) can easily be solved to give a closed expression for the coefficients f_l:

$$f_l^N = \prod_{j=0}^{l-1} \left(\frac{N-j}{N+j+2}\right), \; l > 0. \tag{2.16}$$

2.3 The discrete eigenvalues and eigenfunctions

Definition

The discrete eigenfunctions are the solutions $\phi_\nu(\mu)$ when $\nu \notin [-1,+1]$ subject to the normalisation condition (2.13)

$$\int_{-1}^{+1} \phi_\nu(\mu) d\mu \; = \; 1. \tag{2.17}$$

Inserting the solution (2.12) in this condition we arrive at the definition of what we shall call the characteristic function $\Lambda(\nu)$:

$$\Lambda(\nu) \equiv 1 - \frac{c\nu}{2} \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) \int_{-1}^{+1} \frac{P_l(\mu)}{\nu - \mu} d\mu = 0. \tag{2.18}$$

The integral can be expressed in terms of the Legendre functions of the second kind $Q_l(\nu)$ (see Appendix A), which leads to our reference equation:

$$\Lambda(\nu) \equiv 1 - c\nu \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) Q_l(\nu) = 0. \tag{2.19}$$

The roots of Equation (2.19) are the discrete eigenvalues we are interested in. We denote them by ν_j. The corresponding discrete eigenfunctions can be written as:

$$\phi_{\nu_j}(\mu) = \frac{c\nu_j}{2} P \frac{\sum_{l=0}^{N} (2l+1) \; f_l \; \phi_l(\nu_j) \; P_l(\mu)}{\nu_j - \mu}. \tag{2.20}$$

General properties

Two interesting relations are the limits for $\nu \to 0$ and $\nu \to +\infty$. The limit for $\nu \to 1$ for the Legendre functions of the second kind $Q_l(\nu)$ tends to $+\infty$, so the sign of the limit $\nu \to 1$ for $\Lambda(\nu)$ is given by

$$\operatorname{sgn}\left[\lim_{\nu\to+1}\Lambda(\nu)\right] = \operatorname{sgn}\left[-\sum_{l=0}^{N}(2l+1)f_l\phi_l(1)\right]. \tag{2.21}$$

At the other side of the real line, the limit for $\nu \to +\infty$ is given by [Inö70]

$$\lim_{\nu\to+\infty}\Lambda(\nu) = \prod_{l=0}^{N}(1-cf_l). \tag{2.22}$$

From the symmetry properties of the characteristic equation $\Lambda(\nu) = 0$ one can deduce the following symmetry on the roots:

Property 2.1. *If ν_j is a (real or complex) root of $\Lambda(\nu)$, so are $-\nu_j$ and $\pm\nu_j^*$ where * denotes the complex conjugate.*

Proof. From the recurrence relation (2.10) one can easily deduce that

$$\phi_l(-\nu) = (-)^l\phi_l(\nu).$$

One also has the symmetry relation for the Legendre functions of the second kind $Q_l(\nu)$ (Eq. A.23)

$$Q_l(-\nu) = (-)^{l+1}Q_l(\nu).$$

Using both these relations in Equation (2.19), we have

$$\Lambda(-\nu_j) = 1 - c(-\nu_j)\sum_{j=0}^{N}(2l+1)f_l\phi_l(-\nu_j)Q_l(-\nu_j) = \Lambda(\nu_j).$$

Since the $\phi_l(\nu)$ are polynomials with real coefficients, one has

$$\phi_l(\nu^*) = \phi_l^*(\nu)$$

and the same goes for the Legendre $Q_l(\nu)$ function. Hence, the discrete eigenvalues appear in quartets (plus-minus and complex conjugates) unless they are strictly real (only a plus-minus pair) or strictly imaginary (a complex conjugate pair).

■

For two specific values of c, some more information on the discrete eigenvalues is well known:

Property 2.2. *If $c = 0$ (the fully absorbing case), there are no discrete eigenvalues.*

Proof. If $c = 0$, the characteristic function $\Lambda(\nu)$ reduces to the constant function 1 which clearly has no roots for any value of ν. ■

Property 2.3. *If $c = 1$ (the critical case), there is a double root at ∞ and, in the case of anisotropic scattering, possibly others.*

Proof. In the case that $c = 1$, the characteristic Equation (2.18) can be rewritten as

$$\Lambda(\nu) = 1 - \frac{1}{2}\sum_{l=0}^{N}(2l+1)f_l\phi_l(\nu)\nu\int_{-1}^{+1}\frac{P_l(\mu)}{\nu-\mu}d\mu = 0.$$

Note that

$$\nu\int_{-1}^{+1}\frac{P_l(\mu)}{\nu-\mu}d\mu = \int_{-1}^{+1}\frac{P_l(\mu)}{1-\mu/\nu}d\mu \xrightarrow[\nu\to\infty]{} \begin{cases} 1 & l = 0 \\ 0 & l \neq 0 \end{cases}.$$

So from the sum, only the term $l = 0$ remains when $\nu \to \infty$, simplifying Equation (2.18) to

$$\Lambda(\nu) = 1 - \frac{1}{2}\times 2 = 0$$

which proves that $\nu \to \infty$ indeed is a solution when $c = 1$.

■

In his paper [Cas74], K.M. Case distinguishes three classes of problems for the discrete eigenvalues and proves, under mild restrictions, some useful properties on the nature of these discrete eigenvalues. He considers the subcritical case ($c < 1$), the critical case ($c = 1$) and the supercritical case ($c > 1$).

Property 2.4. *When the following three conditions are fulfilled,*

(a) $\sum_{l=0}^{\infty} l^2|f_l| < \infty$ *which is a kind of convergence criterion on the scattering coefficients. If they are derived from a scattering kernel $F(\mu)$ which is a continuous function for $\mu \in [-1, +1]$ this requirement is fulfilled.*

(b) *$f(\mu)$ is continuous and $f(\mu) \geq 0$ for $\mu \in [-1, +1]$. Continuity is evidently assured when $f(\mu)$ is expanded in a finite series of Legendre polynomials. Non-negativity has to be imposed (for example, by the user of a neutronics code when he enters the scattering coefficients). A consequence hereof is that $|f_l| \leq 1$ for $l \geq 1$. Indeed, using the Mean Value Theorem for definite integrals (we have required continuity of $f(\mu)$ so this theorem is valid):*

$$|f_l| = |\int_{-1}^{+1} f(\mu)P_l(\mu)d\mu| = |P_l(\alpha)\int_{-1}^{+1} f(\mu)d\mu|$$

with $\alpha \in [-1, +1]$. The remaining integral is nothing else but the definition of f_0, which is by normalisation equal to one. The Legendre polynomial $P_l(c)$ is bounded by the interval $[-1, +1]$, so finally, we have $|f_l| \leq 1$.

(c) $(1 - cf_{2k+1}) \geq 0, \quad k = 0, 1, \ldots$ *Note that in the subcritical and critical case ($c \leq 1$) this requirement is always fulfilled.*

Case has proved that the discrete eigenvalues are

(i) are real or purely imaginary,
(ii) are simple,
(iii) occur in plus-minus pairs,
(iv) are equal to or larger than one in magnitude $|\nu_j| \geq 1$
(v) are finite in number

Additionally, when $c < 1$, (i) can be refined to "are real".

Proof. See [Cas74; GC80]. Chihara and Nevai revisited the former proofs and gave two alternative proofs on the finiteness of the discrete eigenvalues [CN82]. ■

In the case of subcritical problems, which 99% of the problems in neutron transport theory are, these properties have quite some consequences on the computational complexity of calculating the discrete eigenvalues. The fact that they are real and occur in plus-minus pairs restrict the search area to the positive real axis, compared to the complex plane otherwise. The fact that they are equal to or larger than one implies that they cannot lie in the continuum $[-1, +1]$ and hence are non-degenerate [See Mik61]. And finally, they are finite in number. A direct result of this fact is that, if we would be able to calculate the number of discrete eigenvalues, we could count them and check if we have found them all. In the other direction, one could say that we search for all the discrete eigenvalues but we have a stopping criterion: the number of discrete eigenvalues that exist.

In a later paper, D. Cacuci [Cac82] proved that the convergence criterion required by Case (Property 2.4, (a))

$$\sum_{l=0}^{\infty} l^2 |f_l| < \infty$$

is not satisfied by hydrogen at large lethargies. He provides a new convergence criterion, even weaker than that of Case:

$$\sum_{l=1}^{\infty} l \log l |f_l| < \infty.$$

Again, it is clear that for a continuous scattering kernel approximated by a finite Legendre series this convergence criterion is always fulfilled.

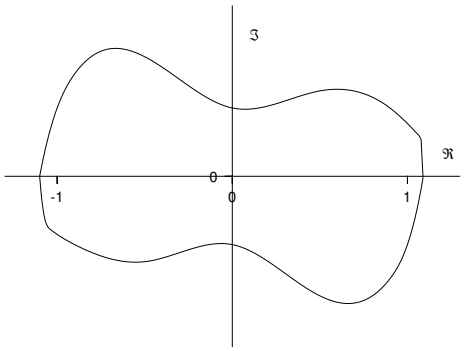


Figure 2.1: *Contour around the* $[-1, +1]$ *cut.*

The number of discrete eigenvalues

The number of discrete eigenvalues can be calculated by means of the argument principle, a classical theorem from complex analysis [Hen88]. Since the function $\Lambda(\nu)$ is holomorphic with no singularities in the cut plane, the number of its roots can be expressed as the change of argument following a contour in the complex plane encircling the cut $[-1, +1]$ on the real axis (see Figure 2.1 for a schematic view):

$$2M = \frac{1}{2\pi}\Delta_C \arg \Lambda(z). \tag{2.23}$$

The total number of roots is denoted by $2M$. We use this "doubling" because we know that the discrete eigenvalues appear in plus-minus pairs, so M denotes the number of positive, respectively negative, discrete eigenvalues. From Equations (A.4–A.5) in Appendix A, we have (where P denotes the Cauchy Principle Value)

$$Q_l^+(\nu) \equiv \lim_{\epsilon \to 0} Q_l(\nu + i\epsilon) = PQ_l(\nu) - \frac{i\pi}{2} \tag{2.24}$$

$$Q_l^-(\nu) \equiv \lim_{\epsilon \to 0} Q_l(\nu - i\epsilon) = PQ_l(\nu) + \frac{i\pi}{2} \tag{2.25}$$

and inserting this in the characteristic function $\Lambda(\nu)$ we find

$$\Lambda^+(\nu) \equiv \lim_{\epsilon \to 0} \Lambda(\nu + i\epsilon) = P\Lambda(\nu) + \frac{i\pi c\nu}{2} P_c(\nu) \tag{2.26}$$

$$\Lambda^-(\nu) \equiv \lim_{\epsilon \to 0} \Lambda(\nu - i\epsilon) = P\Lambda(\nu) - \frac{i\pi c\nu}{2} P_c(\nu) \tag{2.27}$$

where $P_c(\nu)$ is the polynomial of degree $2N$ given by

$$P_c(\nu) = \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) P_l(\nu). \tag{2.28}$$

We assume that neither $\Lambda^+(\nu)$ nor $\Lambda^-(\nu)$ vanish in the interval $[-1, +1]$. If this assumption is not fulfilled, we have so-called degenerate eigenvalues (discrete eigenvalues lying in the continuum). We shall not consider this case in this work (see [Mik61] for more information how to treat this peculiar problem).

Under the assumption above and using Equations (2.26–2.27) we can rewrite Equation (2.23) as

$$2M = \frac{1}{2\pi} \left[\Delta_{C_+} \arg \Lambda^+(\nu) + \Delta_{C_-} \arg \Lambda^-(\nu) \right] \tag{2.29}$$

where $\Delta_{C_+} \arg \Lambda^+(\nu)$ denotes the change in argument of $\Lambda^+(\nu)$ from -1 to $+1$ and $\Delta_{C_-} \arg \Lambda^-(\nu)$ is the change in argument of $\Lambda^-(\nu)$ from $+1$ to -1. We have the following symmetry relations for $\Lambda^+(\nu)$ and $\Lambda^-(\nu)$:

$$\Lambda^+(\nu) = \Lambda^-(-\nu) \tag{2.30}$$

$$\arg \Lambda^+(\nu) = -\arg \Lambda^+(-\nu) \tag{2.31}$$

$$\arg \Lambda^+(0) = \arg \Lambda^-(0) = 0. \tag{2.32}$$

Using these in Equation (2.29) we find the classical result

$$M = \frac{1}{\pi} \arg \Lambda^+(1). \tag{2.33}$$

Mika makes the following observation [Mik61, eq. 3.7 and further]: as the polynomial $P_c(\nu)$ is of the order at most $2N$, the number of roots of the imaginary part of $\Lambda^+(\nu)$ lying between 0 and 1 cannot exceed $N+1$. He uses, without mentioning it explicitly, the fact that the polynomial is symmetric around $\nu = 0$, $P_c(-\nu) = P_c(\nu)$, what can be seen directly from its definition, and thus its roots lie also symmetrical around the origin. So we have the following property:

Property 2.5. *The number of positive discrete eigenvalues is bounded by $N+1$, where N is the order of the anisotropy (the order of approximation in the Legendre approximation of the scattering kernel).*

At the time of Mika's paper, it was quite an arduous task to compute all roots of a polynomial (possibly of a high degree), certainly using the

computers available at the time. What Equation (2.33) really tells us, is the following:

Property 2.6. *The number of positive eigenvalues is bounded by the number of roots lying on the real line segment* $[0, 1]$ *of the polynomial* $\nu P_c(\nu)$.

This exact number of roots is almost always a sharper bound than the order $N + 1$. The question arises however, if and when the bound is reached exactly.

Theorem 2.1. *The bound in Property 2.6 is reached exactly when the polynomial* $\nu P_c(\nu)$ *and the characteristic equation* $\Lambda(\nu)$ *have alternating simple roots on the real line segment* $[0, 1]$.

Proof. A zero of $\nu P_c(\nu)$ makes the imaginary part of Equations (2.26)–(2.27) zero, hence the argument an integer multiple of π. A zero of $\Lambda(\nu)$ in $[0, +1]$ makes the real part zero which makes the argument become an odd multiple of $\pi/2$. Having the roots alternating indeed makes the argument go from 0 to 2π and round again.

■

All our numerical experiments have shown no other cases but those who fulfil the conditions of Theorem 2.1. We have never come across an example where the bound was not reached. Unfortunately, we were unable to establish a firm proof for the fact that the bound is always reached under the aforementioned conditions. We feel, however, confident enough to make the following conjecture:

Conjecture 2.2. *For subcritical cases* $c < 1$ *for which the conditions of Property 2.4 are fulfilled, the bound in Property 2.6 is reached exactly.* The number of discrete eigenvalues is equal to the number of roots of the polynomial $\nu P_c(\nu)$ lying in the real interval $[0, +1]$.

So instead of working with Property 2.5, we work with the result of Conjecture 2.2. Calculating the number of roots of the polynomial $\nu P_c(\nu)$ on the real line $[0, +1]$ gives us the exact number of positive discrete eigenvalues. Solving the characteristic equation $\Lambda(\nu)$ becomes much easier when one knows the number of roots one has to find. The better part of Chapter 4 is devoted to the automatic and robust calculation of all discrete eigenvalues.

2.4 The continuum eigendistribution

In addition to the discrete eigenvalues and eigenfunctions, there is the continuum $\nu \in [-1, +1]$ with the corresponding eigendistribution

$$\phi_\nu(\mu) = \frac{c\nu}{2} P \frac{1}{\nu - \mu} \sum_{l=0}^{N} (2l+1)\ f_l\ \phi_l(\nu)\ P_l(\mu) + \lambda(\nu)\delta(\nu - \mu). \tag{2.34}$$

An expression for $\lambda(\nu)$ can be found by applying the normalisation condition (2.13) to Equation (2.34) above. One finds that

$$\lambda(\nu) = 1 - \frac{c\nu}{2} \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) P Q_l(\nu) \tag{2.35}$$

which is nothing less but the Cauchy principle value of the characteristic function $\Lambda(\nu)$.

As we shall see further, we can express the full solution of the Boltzmann equation using a linear combination of all discrete eigenfunctions plus an integral term where the continuum part appears in the integrand. The better part of Chapter 5 is dedicated to efficient and robust calculation schemes for this integral term. Evidently, this is heavily based on numerical quadrature, since there is no closed form for the appearing integrals.

2.5 Orthogonality properties of the eigenfunctions

In his classical paper [Mik61], Mika also proved orthogonality of the eigenfunctions:

Property 2.7. *For two arbitrary eigenfunctions either from the discrete or continuous spectrum we have*

$$\int_{-1}^{+1} \mu \phi(\nu, \mu) \phi(\nu', \mu) d\mu = 0, \quad \nu \neq \nu'. \tag{2.36}$$

Proof. From [Mik61]: Write Equation (2.11) for ν and ν':

$$\left(1 - \frac{\mu}{\nu}\right) \phi(\nu, \mu) = \frac{c}{2} \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) P_l(\mu), \tag{2.37}$$

$$\left(1 - \frac{\mu}{\nu'}\right) \phi(\nu', \mu) = \frac{c}{2} \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu') P_l(\mu). \tag{2.38}$$

When we multiply the first equation by $\phi(\nu', \mu)$, the second by $\phi(\nu, \mu)$, integrating over $\mu = -1 \to +1$ and using the definition of $\phi_l(\nu)$ (Eq. 2.7) and subtracting both sides, we get:

$$\left(\frac{1}{\nu}-\frac{1}{\nu'}\right)\int_{-1}^{+1}\mu\phi(\nu,\mu)\phi(\nu',\mu)d\mu=0 \tag{2.39}$$

leading to the orthogonality relation (2.36) we want to prove. ■

The normalisation integrals for the discrete spectrum are defined as

$$M_{j\pm}=\int_{-1}^{+1}\mu\left[\phi_{j\pm}(\mu)\right]^2 d\mu \tag{2.40}$$

which can be worked out by direct integration to give

$$\begin{aligned}M_{j\pm}=\pm c\nu_j\Bigg[&\frac{c\nu_j^2}{2\left(\nu_j^2-1\right)}\sum_{k=0}^{N}\sum_{l=0}^{N}\theta_{kl}\left(2k+1\right)f_k\phi_k\left(\nu_j\right)\left(2l+1\right)f_l\phi_l\left(\nu_j\right)\\&-\frac{1}{2}\sum_{k=0}^{N}\left(2k+1\right)f_k\left[\phi_k\left(\nu_j\right)\right]^2\\&-\nu_j\sum_{k=1}^{N}\left(4k+1\right)f_{2k}\phi_{2k}\left(\nu_j\right)\sum_{s=1}^{k}(4s-1)\phi_{2s-1}\left(\nu_j\right)\\&-\nu_j\sum_{k=0}^{N}\left(4k+3\right)f_{2k+1}\phi_{2k+1}\left(\nu_j\right)\sum_{s=0}^{N}\left(4s+1\right)\phi_{2s}\left(\nu_j\right)\Bigg],\end{aligned} \tag{2.41}$$

where

$$\theta_{kl}=\begin{cases}1 & \text{when } (k+l) \text{ is even,}\\ \nu_j & \text{when } (k+l) \text{ is odd.}\end{cases} \tag{2.42}$$

They also obey the equation

$$M_{j\pm}=\pm\frac{c}{2}\nu_j^2\sum_{l=0}^{N}(2l+1)f_l\phi_l(\nu_j)P_l(\nu_j)\left.\frac{\partial\Lambda(\nu)}{\partial\nu}\right|_{\nu=\nu_j}. \tag{2.43}$$

The normalisation integral for the continuum is given by

$$M(\nu)=\int_{-1}^{+1}\mu\phi\left(\nu,\mu\right)\phi\left(\nu',\mu\right)d\mu=M(\nu)\delta(\nu-\nu') \tag{2.44}$$

$$=\nu\left[\left[P\Lambda\left(\nu\right)\right]^2+\frac{c^2\pi^2\nu^2}{4}\left[P_c\left(\nu\right)\right]^2\right] \tag{2.45}$$

$$=\nu\Lambda^+(\nu)\Lambda^-(\nu). \tag{2.46}$$

The eigenexpansion is also complete, i.e. it can be used as a basis for the representation of general functions. For a more functional analytic approach on the completeness, one can consult [Han73].

2.6 The general solution

The general solution of the homogeneous one-speed Boltzmann neutron transport equation can be summarised as

$$\begin{aligned}\psi(x,\mu) = &\sum_{j=1}^{M} a_{j+}\phi(+\nu_j,\mu)e^{-\frac{\Sigma_t x}{\nu_j}} \\ &+ \sum_{j=1}^{M} a_{j-}\phi(-\nu_j,\mu)e^{+\frac{\Sigma_t x}{\nu_j}} \\ &+ \int_{-1}^{+1} A(\nu)\phi(\nu,\mu)e^{-\frac{\Sigma_t x}{\nu}}\end{aligned} \tag{2.47}$$

where $2M$ denotes the total number of discrete eigenvalues. Note that, typically, the positive and negative discrete eigenvalues are separated in the summation (we take the ν_j as the positive eigenvalues and just negate them to have their negative counterparts). It will be shown that this is useful when composing the infinite medium Green's function in the next section. The constants $a_{j\pm}$ and the function $A(\nu)$ have to be determined by means of appropriate boundary conditions.

2.7 The infinite medium Green's function

An interesting application of the former eigendecomposition is the expression of the infinite medium Green's function using the eigenfunctions and eigendistribution, hence determining the coefficients $a_{j\pm}$ and the function $A(\nu)$. This is not only interesting as such, but we shall see later on in Chapter 6 that this Green's function is the building block of the Boundary Sources Method.

The infinite medium Green's function is defined as the solution to the non-homogeneous Boltzmann transport equation

$$\mu\frac{\partial\psi(x,\mu)}{\partial x}+\Sigma_t\psi(x,\mu)=$$
$$\frac{c\Sigma_t}{2}\sum_{l=0}^{N}(2l+1)f_lP_l(\mu)\int_{-1}^{+1}\psi(x,\mu')\;P_l(\mu')d\mu'+\delta(x-x_0)\delta(\mu-\mu_0). \quad (2.48)$$

Physically, it can be interpreted as the neutron distribution from a plane source of unit strength located at $x = x_0$ and emitting in the direction $\mu = \mu_0$. Equation (2.48) can be converted into a homogeneous equation by replacing the source term by a jump condition at $x = x_0$. So we seek the solution of the homogeneous equation

$$\mu\frac{\partial\psi(x,\mu)}{\partial x}+\Sigma_t\psi(x,\mu)=\frac{c\Sigma_t}{2}\sum_{l=0}^{N}(2l+1)f_lP_l(\mu)\int_{-1}^{+1}\psi(x,\mu')\;P_l(\mu')d\mu' \quad (2.49)$$

subject to the boundary condition

$$\psi(0+,\mu)-\psi(0-,\mu)=\frac{1}{\mu}\delta(\mu-\mu_0)\delta(x-x_0). \quad (2.50)$$

We also assume to be in a subcritical medium, hence $c < 1$, so we known that the solution must vanish at infinity:

$$\lim_{|x|\to\infty}\psi(x,\mu)=0. \quad (2.51)$$

To satisfy the limit condition (2.51), we must separate our general solution in two cases:

$$\psi(x,\mu)=\begin{cases}\displaystyle\sum_{j=1}^{M}a_{j+}\phi(\nu_j,\mu)e^{-\Sigma_t(x-x_0)/\nu_j} & \\ \qquad\displaystyle+\int_0^{+1}A(\nu)\phi(\nu,\mu)e^{-\Sigma_t(x-x_0)/\nu}d\nu & x\geq x_0,\\ \displaystyle\sum_{j=1}^{M}a_{j-}\phi(-\nu_j,\mu)e^{+\Sigma_t(x-x_0)/\nu_j} & \\ \qquad\displaystyle-\int_{-1}^{0}A(\nu)\phi(\nu,\mu)e^{-\Sigma_t(x-x_0)/\nu}d\nu & x\leq x_0.\end{cases} \quad (2.52)$$

The remaining condition to apply is the boundary condition at $x = x_0$. Putting Equation (2.52) in this condition we find

$$\sum_{j=1}^{M} a_{j+}\phi(\nu_j,\mu) + \sum_{j=1}^{M} a_{j-}\phi(\nu_j,\mu) + \int_{-1}^{+1} A(\nu)\phi(\nu,\mu)d\mu = \frac{\delta(x-x_0)\delta(\mu-\mu_0)}{\mu}. \quad (2.53)$$

The latter equation is the full-range eigenexpansion of the function

$$\Psi(\mu) = \frac{\delta(x-x_0)\delta(\mu-\mu_0)}{\mu}$$

and hence the coefficients can be obtained by using the orthogonality relations. One finds

$$a_{j\pm} = \frac{1}{M_{j\pm}}\phi(\pm\nu_j,\mu_0), \quad (2.54)$$

$$A(\nu) = \frac{1}{M(\nu)}\phi(\nu,\mu_0) \quad (2.55)$$

where $M_{j\pm}$ and $M(\nu)$ are given by Equations (2.40)-(2.44). Substituting (2.54)-(2.55) in Equation (2.52) we can rewrite the infinite medium Green's function as

$$G(x_0,\mu_0|x,\mu) = \begin{cases} \displaystyle\sum_{j=1}^{M} \frac{\phi(\nu_j,\mu_0)\phi(\nu_j,\mu)}{M_{j+}} e^{-\Sigma_t(x-x_0)/\nu_j} \\ \qquad \displaystyle + \int_0^{+1} \frac{\phi(\nu,\mu_0)\phi(\nu,\mu)}{M(\nu)} e^{-\Sigma_t(x-x_0)/\nu} d\nu \quad x \geq x_0, \\ \displaystyle\sum_{j=1}^{M} \frac{\phi(-\nu_j,\mu_0)\phi(-\nu_j,\mu)}{M_{j-}} e^{+\Sigma_t(x-x_0)/\nu_j} \\ \qquad \displaystyle - \int_{-1}^{0} \frac{\phi(\nu,\mu_0)\phi(\nu,\mu)}{M(\nu)} e^{-\Sigma_t(x-x_0)/\nu} d\nu \quad x \leq x_0. \end{cases} \quad (2.56)$$

It is the above form we shall use throughout the next chapters. In the next chapter, we analyse the transport polynomials $\phi_k(\nu)$. In Chapter 4 and Chapter 5 we study respectively the discrete and continuum part of the above equation.

CHAPTER 3

The transport polynomials and related half-range functions

SUMMARY– The transport polynomials $\phi_k(\nu)$ and their half-range counterparts $\phi_k^{\pm}(\nu)$ are the (half-range) Legendre moments of the Mika/Case eigenfunction or eigendistribution. In this chapter, we review their definition and orthogonality relation and we describe our algorithms to calculate them.

3.1 The transport polynomials

The transport polynomials appear in the solution of the one-speed homogeneous neutron transport Equation (Eq.2.4). They are defined as the Legendre moments of the complete eigenfunction $\phi_\nu(\mu)$ by Equation (2.7)

$$\phi_k(\nu) \equiv \int_{-1}^{+1} \phi_\nu(\mu) P_k(\mu) d\mu.$$

So the eigenfunction can be expressed as an infinite Legendre series

$$\phi_\nu(\mu) = \sum_{k=0}^{\infty} \frac{2k+1}{2} \phi_k(\nu) P_k(\mu). \tag{3.1}$$

As shown in Chapter 2, they also obey the three-term recurrence (2.10), which forms the starting point of the analysis presented in this chapter. Therefore, let us retake the recurrence:

$$\nu \left(1 - c f_k\right) \phi_k(\nu) - \frac{k+1}{2k+1} \phi_{k+1}(\nu) - \frac{k}{2k+1} \phi_{k-1}(\nu) = 0. \tag{3.2}$$

The $\phi_k(\nu)$ are all proportional to $\phi_0(\nu)$ and this value is conveniently chosen to be equal to 1. Hence, the first few polynomials are given by

$$\begin{aligned}\phi_0(\nu) &= 1,\\ \phi_1(\nu) &= \nu(1-cf_0) = \nu(1-c),\\ \phi_2(\nu) &= \frac{3}{2}\nu^2(1-cf_1)(1-c) - \frac{1}{2}.\end{aligned} \tag{3.3}$$

From the recurrence relation it is clear that when $c \equiv 0$, the $\phi_k(\nu)$ reduce to the Legendre orthogonal polynomials. Also for $\nu = 0$, one has

$$\phi_k(0) = P_k(0). \tag{3.4}$$

The recurrence relation also allows us to derive the following symmetry relation

$$\phi_k(-\nu) = (-1)^k \phi_k(\nu) \tag{3.5}$$

indicating that the odd degree polynomials are odd functions and the even degree polynomials are even functions.

Orthogonality

It is well known that a set of orthogonal polynomials $P_n(x)$ obeys a three-term recurrence [Sze59; Buy92]

$$xP_n(x) = a_{n+1}P_{n+1}(x) + b_nP_n(x) + a_nP_{n-1}(x) \quad P_{-1}(x) = 0, P_0(x) = 1 \tag{3.6}$$

where $\{a_n\}_{n=0}^{\infty}$ is a sequence of positive real numbers and $\{b_n\}_{n=0}^{\infty}$ a sequence of real numbers. By Favard's theorem [See for example Chi90] the converse is also valid: a three-term recurrence gives rise to a set of orthogonal polynomials over a certain orthogonality interval with a certain weight function. The coefficients are uniquely determined by:

$$a_n = \int_{\mathcal{S}} w(x)xP_n(x)P_{n-1}(x)dx \tag{3.7}$$

$$b_n = \int_{\mathcal{S}} w(x)x\left[P_n(x)\right]^2 dx \tag{3.8}$$

where $\mathcal{S}$ denotes the orthogonality interval or spectrum. Based on this theory, it is clear that the polynomials $\phi_k(\nu)$ also form an orthogonal set. Both Beauwens and Inönü [Bea68; Inö70] proved orthogonality of the $\phi_k(\nu)$. Beauwens uses the so-called $\mathcal{K}$-transform to achieve this result. We shall, later in this chapter when we discuss the half-range functions, come back to this transform. For completeness, we give here Inönü's proof.

Lemma 3.1. *[See CZ67, Appendix G] Given the singular integral equation*

$$g(t) = A(t)f(t) + P\int_a^b \frac{B(t')f(t')}{t'-t}dt', \tag{3.9}$$

then a sufficient condition for the existence of a solution is that $g(t)$, $A(t)$ and $B(t)$ satisfy the Hölder conditions defined by, for example in the case of $g(t)$,

$$|g(t) - g(t')| < C|t - t'|^{\mu}$$

for $a < t, t' < b$ with some $\mu > 0$ and C a constant and

$$|g(t) - g(c)| < \left(\frac{D}{|t-c|^{\alpha}}\right)$$

for $\alpha < 1$, c an endpoint of the interval (a or b) and $a < t < b$ and finally, if $A(t) \pm iB(t) \neq 0$ in $[a, b]$.

Theorem 3.2. *The polynomials $\phi_k(\nu)$ form an orthogonal set on the interval $\mathcal{S} = [-1, +1] \cup \{\nu_j, -\nu_j\}_{j=1,\ldots,M}$ with respect to the weight function $w(\nu)$ defined by*

$$w(\nu) = \frac{\nu}{M(\nu)}U_{(-1,+1)}(\nu) + \sum_{j=1}^{M}\frac{\nu_j}{M_j}\left[\delta\left(\nu - \nu_j\right) + \delta\left(\nu + \nu_j\right)\right] \tag{3.10}$$

where the ν_j are the positive discrete eigenvalues, M_j the corresponding normalisation factor, $M(\nu)$ the continuum normalisation factor and where $U_{(-1,+1)}(\nu)$ denotes the characteristic function of the real interval $[-1, +1]$:

$$U_{(a,b)}(t) = \begin{cases} 0 & \text{when } t \notin [a, b], \\ 1 & \text{when } t \in [a, b]. \end{cases} \tag{3.11}$$

Proof. Mika [Mik61] proved that any function $\psi(\mu)$ satisfying the Hölder condition on the interval $[-1, +1]$ can be expanded in terms of the discrete and continuum eigenfunctions:

$$\psi(\mu) = \int_{-1}^{+1} A(\nu)\phi_\nu(\mu)d\nu + \sum_{j=1}^{M}\left\{a_{j+}\phi_{\nu+}(\mu) + a_{j-}\phi_{\nu-}(\mu)\right\} \tag{3.12}$$

where the coefficients $a_{j\pm}$ and the function $A(\nu)$ are defined as

$$a_{\nu\pm} = \frac{1}{M_{j\pm}}\int_{-1}^{+1} \mu\phi_{\nu\pm}(\mu)\psi(\mu)d\mu \tag{3.13}$$

$$A(\nu) = \frac{1}{M(\nu)}\int_{-1}^{+1} \mu\phi_\nu(\mu)\psi(\mu)d\mu. \tag{3.14}$$

Let us consider the expansion for the function

$$\psi(\mu) = \frac{P_k(\mu)}{(1-cf_k)}, \quad k < N.$$

Using Equations (3.1, 3.13, 3.14) together with the recurrence relation (A.12) for the Legendre polynomials and the recurrence relation for the transport polynomials (3.2) one finds

$$A(\nu) = \frac{\nu}{M(\nu)}\phi_k(\nu) \tag{3.15}$$

$$a_{j\pm} = \frac{\nu_{j\pm}}{N_{j\pm}}\phi_k(\nu_{j\pm}). \tag{3.16}$$

Putting this in Equation (3.12), multiplying by $P_l(\mu)$ and integrating over μ from -1 to $+1$, one finds

$$\begin{aligned}\frac{2\delta_{k,l}}{(2l+1)(1-cf_l)} &= \int_{-1}^{+1} P_l(\mu)d\mu\int_{-1}^{+1}\frac{\nu}{N(\nu)}\phi_k(\nu)\phi_\nu(\mu)d\nu \\ &+ \int_{-1}^{+1} P_l(\mu)d\mu\sum_{j=1}^{M}\left[\frac{\nu_j}{N_{j+}}\phi_k(\nu_j)\phi_{j+}(\mu) - \frac{v_j}{N_{j-}}\phi_k(-\nu_j)\phi_{j-}(\mu)\right].\end{aligned} \tag{3.17}$$

We are allowed to switch the order of integration since the first term on the right-hand-side contains only a single principal value integral. Doing this leads to the final orthogonality relation for the transport polynomials:

$$\frac{2\delta_{k,l}}{(2l+1)(1-cf_l)} = \int_{-1}^{+1} \phi_k(\nu)\phi_l(\nu)\frac{\nu}{M(\nu)}d\nu + \sum_{j=1}^{M}\left[\phi_k(\nu_j)\phi_l(\nu_j)\left(\frac{\nu_j}{N_{j+}}\right) + \phi_k(-\nu_j)\phi_l(-\nu_j)\left(\frac{-\nu_j}{N_{j-}}\right)\right]. \quad (3.18)$$

■

Relations between $\phi_k(\nu)$, $P_l(\nu)$ and $Q_n(\nu)$

Starting from the recurrence relations (Eq.3.2) for the transport polynomials $\phi_k(\nu)$ and the Legendre polynomials $P_l(\nu)$ and the Legendre functions of the second kind $Q_n(\nu)$ (Eq.A.12), one can derive several relations linking these three functions. Two of them will be of interest later on, so we present them here.

Theorem 3.3. *Between the Legendre functions $P_l(\nu)$, $Q_n(\nu)$ and the transport polynomials $\phi_k(\nu)$ the following relations hold:*

$$\sum_{l=0}^{N}(2l+1)f_l P_l(\nu)\phi_l(\nu) = \left(\frac{N+1}{c\nu}\right)\left[P_{N+1}(\nu)\phi_N(\nu) - P_N(\nu)\phi_{N+1}(\nu)\right] \quad (3.19)$$

$$\sum_{l=0}^{N}(2l+1)f_l Q_l(\nu)\phi_l(\nu) = \frac{1}{c\nu} + \left(\frac{N+1}{c\nu}\right)\left[Q_{N+1}(\nu)\phi_N(\nu) - Q_N(\nu)\phi_{N+1}(\nu)\right] \quad (3.20)$$

Proof. Start from

$$(x-y)\sum_{l=0}^{N}(2l+1)P_l(x)\phi_l(y) = \sum_{l=0}^{N}(2l+1)xP_l(x)\phi_l(y) - \sum_{l=0}^{N}(2l+1)P_l(x)y\phi_l(y) \quad (3.21)$$

and use the recurrence relation (A.12) to expand $x\,P_l(x)$ and the recurrence relation (3.2) to expand $y\,\phi_l(y)$:

$$(x-y)\sum_{l=0}^{N}(2l+1)P_l(x)\phi_l(y) = \sum_{l=0}^{N}(2l+1)\left[\frac{l+1}{2l+1}P_{l+1}(x)+\frac{l}{2l+1}P_{l-1}(x)\right]\phi_l(y) - \sum_{l=0}^{N}(2l+1)\left[ycf_l\phi_l(y)+\frac{l+1}{2l+1}\phi_{l+1}(y)+\frac{l}{2l+1}\phi_{l-1}(y)\right]P_l(x). \quad (3.22)$$

Simplifying further leads to

$$(x-y)\sum_{l=0}^{N}(2l+1)P_l(x)\phi_l(y) = \sum_{l=0}^{N}(l+1)P_{l+1}(x)\phi_l(y)+\sum_{l=0}^{N}lP_{l-1}(x)\phi_l(y) - \sum_{l=0}^{N}(2l+1)ycf_l\phi_l(y)P_l(x)-\sum_{l=0}^{N}(l+1)\phi_{l+1}(y)P_l(x)-\sum_{l=0}^{N}\phi_{l-1}(y)P_l(x). \quad (3.23)$$

Separating out the Nth term from the two summations having a factor $(l+1)$, we find

$$(x-y)\sum_{l=0}^{N}(2l+1)P_l(x)\phi_l(y) = (N+1)P_{N+1}(x)\phi(y)+\sum_{l=0}^{N-1}(l+1)P_{l+1}(x)\phi_l(y)+\sum_{l=0}^{N}lP_{l-1}(x)\phi_l(y) - cy\sum_{l=0}^{N}(2l+1)f_lP_l(x)\phi_l(y)-(N+1)\phi_{N+1}(y)P_N(x) - \sum_{l=0}^{N-1}(l+1)\phi_{l+1}(y)P_l(x)-\sum_{l=0}^{N}l\phi_{l-1}(y)P_l(x). \quad (3.24)$$

We shift the indices of the two sums with upper limit $N-1$ by one and take into account that the other two sums have no contribution for $l=0$ (since they have the factor l) and thus their lower limit can be set to 1. This leads us to

$$(x-y)\sum_{l=0}^{N}(2l+1)P_l(x)\phi_l(y) =$$

$$(N+1)\Big[P_{N+1}(x)\phi(y)-(N+1)\phi_{N+1}(y)P_N(x)\Big]-cy\sum_{l=1}^{N}(2l+1)f_lP_l(x)\phi_l(y)$$
$$+\sum_{l=1}^{N}lP_l(x)\phi_{l-1}(y)+\sum_{l=1}^{N}lP_{l-1}(x)\phi_l(y)$$
$$-\sum_{l=1}^{N}l\phi_l(y)P_{l-1}(x)-\sum_{l=1}^{N}l\phi_{l-1}(y)P_l(x). \quad (3.25)$$

The latter four sums cancel each other out. By setting $x = y = \nu$, we find what was to be proved. The proof for the second relation follows the same reasoning.

■

Zeros of the transport polynomials

Zeros of orthogonal polynomials have some interesting properties [Sze59; Ask75]. For completeness, we summarise here the most important facts about the zeros of the $\phi_k(\nu)$ [Inö70]. We use the notation ν_1 for the largest root of the characteristic equation $\Lambda(\nu)$. Hence, the orthogonality interval is bounded by $-\nu_1$ and $+\nu_1$.

- All zeros of $\phi_k(\nu)$ are real and simple and are situated between $-\nu_1$ and $+\nu_1$.
- Any two zeros of $\phi_k(\nu)$ are separated by a zero of $\phi_{k-1}(\nu)$.
- The two zeros of $\phi_k(\nu)$ which have the largest absolute value approach $\pm\nu_1$ in the limit for $k \to \infty$.
- Outside the interval $[-1,+1]$, $\phi_k(\nu)$ can have at most one root between two roots of $\Lambda(\nu)$.
- The roots of the orthogonal polynomial system $\phi_k(\nu)$ cover the interior of $[-1,+1]$ everywhere densely, while outside this interval the $\phi_k(\nu)$ have a limited number of zeros.

The $\phi_k(\nu)$ polynomials belong to the so-called Nevai class of orthogonal polynomials [Nev79; VA90] and hence all properties of this class are valid for the transport polynomials. One such property is that the asymptotic zero distribution is regular: it is an inverted semicircle in the interval $[-1,+1]$.

Asymptotic behaviour

Of interest to us is the asymptotic value of the fraction $\phi_{k+1}(\nu)/\phi_k(\nu)$ for $k \to \infty$ and $|\nu| > 1$. According to the Poincaré-Perron theorem, this fraction is in the limit equal to one of the two roots of the characteristic equation corresponding to the three-term recurrence defining the Legendre polynomials and the transport polynomials. This characteristic equation is given by

$$\lambda^2 - 2\nu\lambda + 1 = 0. \tag{3.26}$$

The roots of this equation can easily be expressed as

$$\lambda_{1,2} = \nu \pm \sqrt{\nu^2 - 1} \tag{3.27}$$

and they are different in modulus when $|\nu| > 1$. One can show that the limiting value for $\phi_{k+1}(\nu)/\phi_k(\nu)$ is equal to the root with the positive sign, hence

$$\lim_{k\to\infty} \frac{\phi_{k+1}(\nu)}{\phi_k(\nu)} = \lambda_1 = \nu + \sqrt{\nu^2 - 1}. \tag{3.28}$$

From this, we have

$$\lim_{k\to\infty} \frac{\phi_{k+1}(\nu)}{\phi_k(\nu)} = \lim_{k\to\infty} \frac{P_{k+1}(\nu)}{P_k(\nu)} = \lambda_1 = \nu + \sqrt{\nu^2 - 1}. \tag{3.29}$$

Also of interest is the asymptotic relation between the characteristic equation $\Lambda(\nu)$, the transport polynomials $\phi_k(\nu)$ and the Legendre polynomials $P_k(\nu)$. Inönü showed that for $|\nu| > 1$, one has

$$\Lambda(\nu) = \lim_{k\to\infty} \frac{\phi_k(\nu)}{P_k(\nu)} \tag{3.30}$$

and this equation leads to an interesting fact we shall use later on in our computational method of calculating all discrete eigenvalues. From Equation (3.30) one can derive the following set of properties.

Property 3.1. *The roots of the polynomials $\phi_k(\nu)$ and $\Lambda(\nu)$ have the following inter-relationships:*

(i) The number of roots of $\Lambda(\nu)$ outside the real interval $[-1, +1]$ must be finite.

(ii) All roots of $\Lambda(\nu)$ are real and simple (for $0 < c < 1$).

(iii) The roots of $\Lambda(\nu)$ are reached as the limits of the roots of the polynomials $\phi_k(\nu)$ for $k \to \infty$ and the approach to the limiting value is from the left for $\nu > 1$ (and from the right for $\nu < -1$).

k	$\phi_k(\nu)$ Eq. (3.2) forward	Relative error	$\phi_k(\nu)$ Eq. (3.2) backward	Relative error
5	3.128406×10^{-03}	$\mathcal{O}(10^{-13})$	3.128406×10^{-03}	$\mathcal{O}(10^{-16})$
10	1.346186×10^{-05}	$\mathcal{O}(10^{-08})$	1.346186×10^{-05}	$\mathcal{O}(10^{-16})$
15	6.530479×10^{-08}	$\mathcal{O}(10^{-04})$	6.532548×10^{-08}	$\mathcal{O}(10^{-15})$
20	$-2.723637 \times 10^{-09}$	$\mathcal{O}(10^{+01})$	3.341311×10^{-10}	$\mathcal{O}(10^{-15})$

Table 3.1: *Accuracy of forward and backward direction of recurrence (3.2) for $\nu = \nu_1$.*

Calculation scheme

The evaluation of the transport polynomials $\phi_k(\nu)$ is based on the recurrence equation since closed expressions are too complicated to build on-the-fly. In the interval $[-1, +1]$, the recurrence is forward stable and can be used without problems. Outside of this interval, the recurrence is forward stable in all points except in the roots of the characteristic equation $\Lambda(\nu)$.

This can be intuitively seen by retaking the third item in Property 3.1: in the limit, the roots of the $\phi_k(\nu)$ approach the roots of the characteristic equation $\Lambda(\nu)$. Firstly, for k increasing, the value of $\phi_k(\nu_j)$ goes to zero (where ν_j is a root of $\Lambda(\nu)$). Secondly, the behaviour of the $\phi_k(\nu)$ is such that they oscillate with large amplitudes (when k increases, the amplitude increases) outside the interval $[-1, +1]$ and cross the ν-axis sharply in their roots. When evaluating such a polynomial in a root of $\Lambda(\nu)$ which is, for k high enough, very close to a root of $\phi_k(\nu)$ itself, we are adding and subtracting extremely large values expecting a small result. In infinite precision arithmetic this result is exact, but in finite precision we suffer from large cancellation errors.

The solution for this problem is to use the backward direction of the recurrence equation with adequate initial values. This approach is called Miller's (improved) algorithm and is described in more detail in Appendix C.

To conclude, we give an example of both approaches to show the disastrous results of using the forward direction. Let us consider the problem characterised by $c = 0.8$ and a 5-th order approximation of the Henyey-Greenstein scattering kernel with free parameter $g = 0.25$. The only root of $\Lambda(\nu)$ is located at $\nu_1 \approx 1.576178$. Table 3.1 shows the result of the forward and backward approach in evaluating the recurrence for $\nu = \nu_1$, both with relative errors compared to MAPLE [Map04] results.

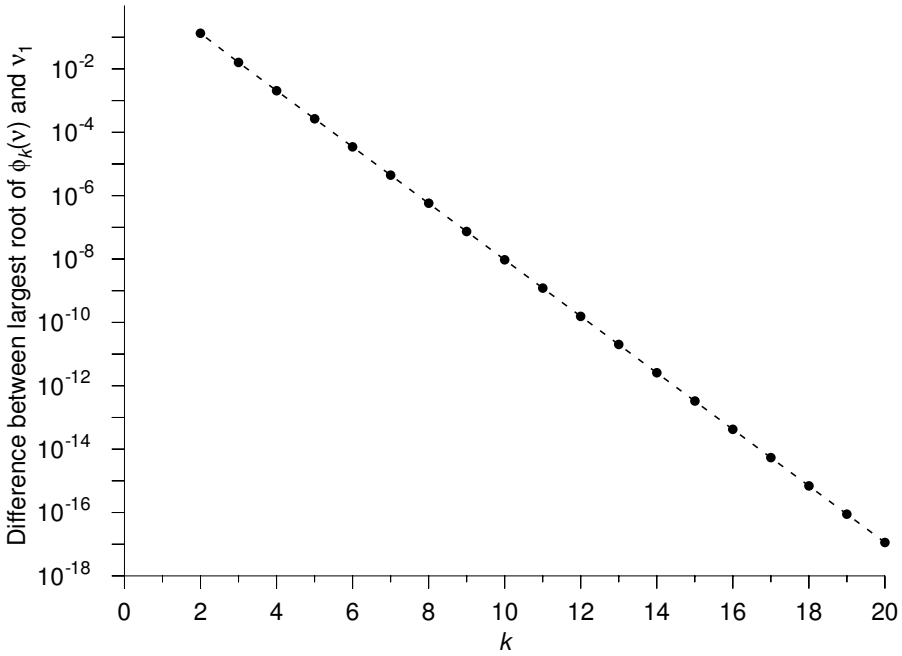


Figure 3.1: *Graphical representation of the first argument why the forward direction of Equation (3.2) is unstable for ν a root of $\Lambda(\nu)$.*

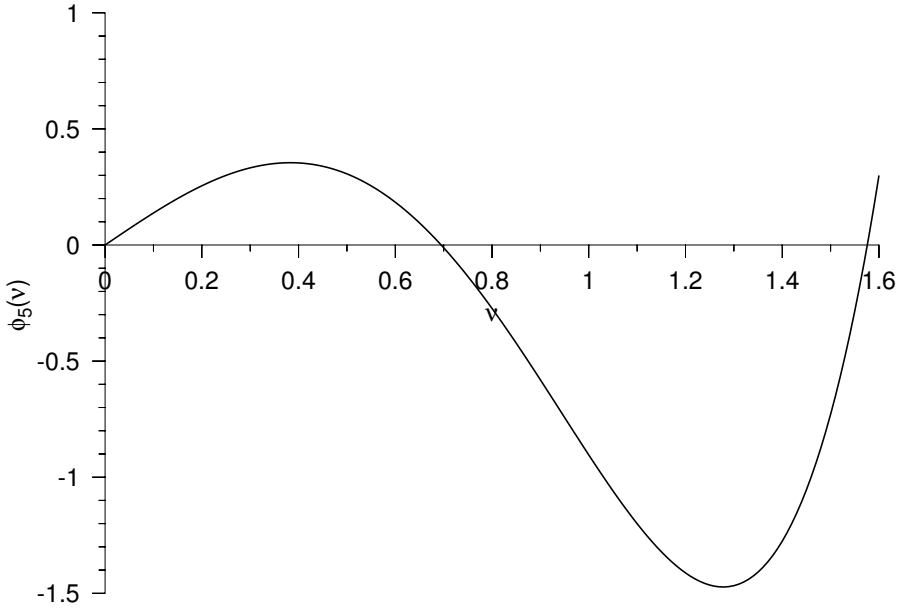
Figure 3.1 shows the first argument in a graphical fashion: the largest root of $\phi_k(\nu)$ approaches the root of $\Lambda(\nu)$, ν_1. The second argument is illustrated in Figure 3.2 where we see the large oscillations around ν_1.

3.2 The transport half-range functions

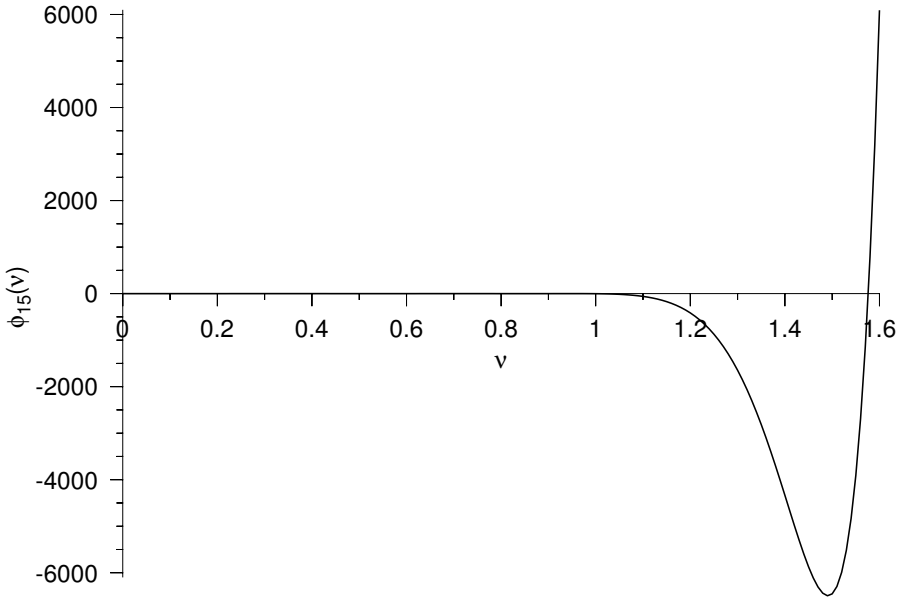
In the development of the Boundary Sources Method (see Chapter 6), one could use next to the transport polynomials also the so-called half-range functions. The Yvon polynomials (also called the shifted Legendre polynomials) are defined as

$$P_k^{\pm}(z) = U(\pm z)P_k(2z \mp 1) \tag{3.31}$$

where $U(\pm z)$ is the characteristic function of the interval of respectively $[0, +1]$ and $[-1, 0]$. The $P_k^+(z)$ form an orthogonal set on $[0, +1]$, while the $P_k^-(z)$ form an orthogonal set on $[-1, 0]$. When one considers these functions on the full interval $[-1, +1]$, they are no longer polynomials, the point $z = 0$ is a singularity. Starting from the recurrence relation for the Legendre



a. *Behaviour of* $\phi_5(\nu)$



b. *Behaviour of* $\phi_{15}(\nu)$

Figure 3.2: *Graphical representation of the second argument why the forward direction of Equation (3.2) is unstable for* ν *a root of* $\Lambda(\nu)$.

polynomials, it is easy to see that the Yvon functions obey the following recurrence relation:

$$(2n+1)(2\mu \mp 1)P_n^{\pm} = (n+1)P_{n+1}^{\pm}(\mu) + nP_{n-1}^{\pm}(\mu). \tag{3.32}$$

The corresponding functions of the second kind are defined using the Legendre functions of the second kind as

$$Q_k^{\pm}(z) = Q_k(2z \mp 1) \tag{3.33}$$

In the same fashion how we arrived at the transport polynomials $\phi_k(\nu)$ as the moments of the complete eigenfunction in Equation (2.7) with respect to the Legendre polynomials, we would now like to find an expression for the moments of the eigenfunction with respect to the Yvon functions. Hence we are looking for the $\phi_k^{\pm}(\nu)$ defined as (where we substituted the definition of the Yvon functions already and adapted the integration intervals accordingly):

$$\phi_k^{+}(\nu) = \int_0^{+1} \phi_\nu(\mu) P_k(2\mu - 1) d\mu \tag{3.34a}$$

$$\phi_k^{-}(\nu) = \int_{-1}^{0} \phi_\nu(\mu) P_k(2\mu + 1) d\mu. \tag{3.34b}$$

Using the techniques of the $\mathcal{K}$-transform, one can show that these too obey a three-term recurrence equation. As before, it is this recurrence equation that is used as the basic tool for evaluating these half-range functions (indeed, they are no longer polynomials).

The $\mathcal{K}$-transform

The anisotropic $\mathcal{K}$-transform is defined for a function $h(\mu)$ defined on $\mu \in [-1, +1]$ as [Bea68]:

$$H(z) = h(z) - \frac{cz}{2} \sum_{l=0}^{\infty} (2l+1) f_l \phi_l(\nu) P \int_{-1}^{+1} \frac{h(z) - h(\mu)}{z - \mu} P_l(\mu) d\mu. \tag{3.35}$$

We shall use the short-hand

$$H(z) = \mathcal{K}(z, \mu) h(\mu). \tag{3.36}$$

The $\mathcal{K}$-transform of a function $h(\mu)$ is defined, assuming the integrals exists, for all $z \in [-1, +1]$ and also for $z = \pm\nu_j$ where the ν_j are the roots of $\Lambda(\nu)$.

Based on this transformation, Beauwens [Bea68] proved both orthogonality of the eigenfunctions $\phi_\nu(\mu)$ and the orthogonality of the transport polynomials $\phi_k(\nu)$. He also showed, based on the fact that the $\mathcal{K}$-transform of the Legendre polynomials are nothing but the transport polynomials $\phi_k(\nu)$, that the $\mathcal{K}$-transform of a function developed in Legendre polynomials

$$h(\mu) = \sum_{l=0}^{\infty} \frac{2l+1}{2} h_l P_l(\mu) \tag{3.37}$$

is given by

$$H(z) = \sum_{l=0}^{\infty} \frac{2l+1}{2} h_l \phi_l(z). \tag{3.38}$$

The $\mathcal{K}$-transform of the Yvon functions

We are looking for an expression for the $\mathcal{K}$-transform of the Yvon functions defined as

$$\phi_k^{\pm}(z) = \mathcal{K}(z, \mu) P_k^{\pm}(z). \tag{3.39}$$

Using the definitions of the $\mathcal{K}$-transform (3.35) and the Yvon functions (3.31) we have

$$\begin{aligned}
\phi_k^{\pm}(z) &= P_k^{\pm}(z) - \frac{cz}{2} \sum_{l=0}^{\infty} (2l+1) f_l \phi_l(z) \int_{-1}^{+1} \frac{P_k^{\pm}(z) - P_k^{\pm}(\mu)}{z-\mu} P_l(\mu) d\mu \\
&= P_k^{\pm}(z) - \frac{cz}{2} \sum_{l=0}^{\infty} (2l+1) f_l \phi_l(z) P_k^{\pm}(z) \int_{-1}^{+1} \frac{P_l(\mu)}{z-\mu} d\mu \\
&\quad + \frac{cz}{2} \sum_{l=0}^{\infty} (2l+1) f_l \phi_l(z) \int_{-1}^{+1} \frac{P_k^{\pm}(\mu) P_l(\mu)}{z-\mu} d\mu
\end{aligned}$$

which we can rewrite as

$$\begin{aligned}\phi_k^{\pm}(z) = P_k^{\pm}(z)\left[1 - cz\sum_{l=0}^{\infty}(2l+1)f_l\phi_l(z)Q_l(z)\right] \\ + \frac{cz}{2}\sum_{l=0}^{\infty}(2l+1)f_l\phi_l(z)\int_{-1}^{+1}\frac{P_k^{\pm}(\mu)P_l(\mu)}{z-\mu}d\mu.\end{aligned} \tag{3.40}$$

Since we have made the assumption that we approximate the scattering kernel by a finite Legendre series of order N, the f_l for $l > N$ are zero and the summations in Equation (3.40) have N as upper limit. Unfortunately, the above equation is very complex to evaluate. Let us try to find, as in the case of the full-range transport polynomials $\phi_k(z)$, a recurrence relation honoured by the half-range functions $\phi_k^{\pm}(z)$. Take the following three expressions $(2k+1)(2z \mp 1)\phi_k^{\pm}(z)$, $-(k+1)\phi_{k+1}^{\pm}(z)$ and $-k\phi_{k-1}^{\pm}(z)$ and add them together using the definition of Equation (3.40). One finds

$$\begin{aligned}(2k+1)(2z\mp1)\phi_k^{\pm}(z) - (k+1)\phi_{k+1}^{\pm}(z) - k\phi_{k-1}^{\pm}(z) = \\ \left[(2k+1)(2z\mp1)P_k^{\pm}(z) - (k+1)P_{k+1}^{\pm}(z) - kP_{k-1}^{\pm}(z)\right] \\ \times\left[1 - cz\sum_{l=0}^{\infty}(2l+1)f_l\phi_l(z)Q_l(z)\right] \\ + \frac{cz}{2}\sum_{l=0}^{\infty}(2l+1)f_l\phi_l(z) \\ \times\int_{-1}^{+1}\frac{P_l(\mu)}{z-\mu}\left[(2k+1)(2z\mp1)P_k^{\pm}(\mu) - (n+1)P_{k+1}^{\pm}(\mu) - kP_{k-1}^{\pm}(\mu)\right]d\mu.\end{aligned} \tag{3.41}$$

From the recurrence Equation (3.32) of the Yvon polynomials we see that the first term becomes zero and we are left with only the infinite sum containing the integral term. Again using the recurrence (3.32) we can replace

$$-(n+1)P_{k+1}^{\pm}(\mu) - kP_{k-1}^{\pm}(\mu)$$

by

$$-(2k+1)(2\mu\mp1)P_k^{\pm}(\mu)$$

which leads us to

$$(2k+1)(2z\mp 1)\phi_k^{\pm}(z) - (k+1)\phi_{k+1}^{\pm}(z) - k\phi_{k-1}^{\pm}(z) = \frac{cz}{2}\sum_{l=0}^{\infty}(2l+1)f_l\phi_l(z)$$
$$\times \int_{-1}^{+1}\frac{P_l(\mu)}{z-\mu}\left[(2k+1)(2z\mp 1)P_k^{\pm}(\mu) - (2k+1)(2\mu\mp 1)P_k\pm(\mu)\right]d\mu \quad (3.42)$$

and simplifying this we finally arrive at

$$(2k+1)(2z\mp 1)\phi_k^{\pm}(z) - (k+1)\phi_{k+1}^{\pm}(z) - k\phi_{k-1}^{\pm}(z) = cz\sum_{l=0}^{\infty}(2l+1)f_l\phi_l(z)(2k+1)\int_{-1}^{+1}P_k^{\pm}(\mu)P_l(\mu)d\mu. \quad (3.43)$$

Suppose for an instant that we would have done the same analysis for the Legendre polynomials instead of the Yvon polynomials, we would have z instead of $(2z\mp 1)$ in the left-hand-side and $P_k(z)$ for $P_k^{\pm}(z)$. From that we see that by grace of the orthogonality relation of the Legendre polynomials, the summation is reduced to a single term and we would have derived the recurrence relation for the transport polynomials $\phi_k(\nu)$, the full-range moments. Now however, the integral doesn't disappear for all but one term and we are stuck with a non-homogeneous recurrence equation. This surely limits the use of the half-range functions in the concept of the Boundary Sources Method.

CHAPTER 4

The discrete eigenfunctions

SUMMARY– The discrete eigenvalues are the solution to the characteristic equation, which is a highly non-linear equation. We describe our solution procedure to obtain all solutions to this equation which are all discrete eigenvalues in the Mika/Case eigenfunction expansion. This solution procedure is a two-step process: first we calculate an upper bound on the number of discrete eigenvalues and use this as input to locate all discrete eigenvalues. As a special case, we study near-singular eigenvalues that can occur for certain combinations of the parameter c and scattering kernel. Finally, we show how to calculate angular and spatial moments of the discrete eigenfunctions which are needed to build a Green's kernel.

4.1 Introduction

The discrete eigenvalues are the solution of the highly non-linear equation

$$\Lambda(\nu) \equiv 1 - c\nu \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) Q_l(\nu) = 0. \tag{4.1}$$

We assume in the remainder of this text that the conditions of Property 2.4 in Chapter 2 are fulfilled and only real eigenvalues exist and their number is finite.

Solution methods have been proposed by Kaper et al. [KSV70] who use a step-by-step evaluation of Equation (2.33), expressing the number of positive discrete eigenvalues as the change of argument of $\Lambda(\nu)$. They use a Gaussian quadrature to evaluate the integral form of the characteristic Equation (2.18) instead of using the properties of the Legendre functions of the second kind $Q_n(\nu)$. Both this and the fact that the argument can change

quite rapidly for $\nu \to 1$ (and hence small steps are needed), their computational cost is high. Siewert uses in one paper [Sie80] the Wiener-Hopf factorisation technique. This requires the a-priori knowledge of the number of discrete eigenvalues and leads to coupled non-linear equations to be solved. On top of that, the non-linear equations have different forms depending on the number of discrete eigenvalues, making this technique not fit for an automated and robust algorithm. Building this system of non-linear equations also requires numerical quadrature.

Our solution method is two-step process: firstly we calculate the upper bound on the number of discrete eigenvalues using Property 2.6 (in fact we use Conjecture 2.2) and use this number to calculate all the discrete eigenvalues using a fast and stable evaluation of $\Lambda(\nu)$ making use of the recurrence equations for the Legendre functions of the second kind (Eq.A.24) and the transport polynomials (Eq.3.2) and their corresponding evaluation methods.

4.2 The characteristic polynomial

From Conjecture 2.2, we take that the number of positive discrete eigenvalues is equal to the number of roots on the real line $[0, 1]$ of the polynomial $\nu P_c(\nu)$ where $P_c(\nu)$ is the characteristic polynomial defined by

$$P_c(\nu) = \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) P_l(\nu). \tag{4.2}$$

This N-term sum can be simplified using relation (3.19) of Theorem 3.3:

$$P_c(\nu) = \frac{N+1}{c\nu} \left[P_{N+1}(\nu)\phi_N(\nu) - P_N(\nu)\phi_{N+1}(\nu) \right]. \tag{4.3}$$

The latter form reduces the computation time and reduces the risk of roundoff during the summation of N terms. We must take care evaluating this form close to zero since we have a factor ν in the denominator. For small values of ν, our code reverts to the definition in Equation (4.2).

Since we are interested only in the number of roots of $\nu P_c(\nu)$ in $[0, +1]$ and not in the roots themselves, the application of Sturm sequences [Act90; Gle97] came to mind. In order to apply this method we need the following (a) the coefficients of the polynomial (b) polynomial operations (addition, subtraction, multiplication and most computationally complex, polynomial division) . It is the latter (and the numerical errors that go along) next to the known instability of the Sturm method for high degree polynomials that made us decide to abandon this approach.

A classical routine to locate all roots of a polynomial is the Jenkins-Traub algorithm [JT70; Jen75]. It is a complex (in the sense of difficult to grasp, not in the sense of complex coefficients) three-stage method and is well known to generally also give good results [Goe94] for polynomials up to degree 50. As is typical with almost all algorithms in numerical analysis, there always exist (engineered) counterexamples for which an algorithm behaves badly. In the case of polynomial root finding, this example is the classical Wilkinson polynomial [Hig02].

First approach: The companion matrix method

It is a well known fact that the roots of a polynomial

$$p(x) = \sum_{k=0}^{n} a_k x^k$$

can be computed as the eigenvalues of the companion matrix [GVL96; EM95; Goe94] defined as

$$C_{p(x)} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0/a_n \\ 1 & 0 & \cdots & 0 & -a_1/a_n \\ 0 & 1 & \cdots & 0 & -a_2/a_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1}/a_n \end{bmatrix}. \tag{4.4}$$

This method is robust and stable [TT94; Mos86; Uhl92] under the condition that the companion matrix is balanced before calling the eigenvalue calculation routine. It has the advantage that it guarantees to find all roots. To apply this method, we need the coefficients of the polynomial, so the first step is to "build" the polynomial by interpolation.

Polynomial interpolation can be done using different techniques: Lagrange, Newton, etc. These methods are so-called "value"-methods. They interpolate the polynomial and return a value for this polynomial at a certain point. However, we are interested in the "coefficient"-problem: we need as a final result the coefficients of the different powers of ν in the polynomial. We do this by solving the corresponding Vandermonde system. Given the $n+1$ distinct interpolation points α_i and the corresponding function values f_i the coefficient vector a is the solution of the dual Vandermonde system

$$V^T a = f \tag{4.5}$$

with V defined as

$$V_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_n \\ \vdots & \vdots & & \vdots \\ \alpha_0^n & \alpha_1^n & \cdots & \alpha_n^n \end{bmatrix}. \tag{4.6}$$

This Vandermonde system is a linear system of size $n+1$, where n is the degree of the polynomial. Due to the structure of the Vandermonde matrix and the fact that it depends only on $n+1$ parameters it is possible to reduce the cost of solving this linear system compared to classical Gauss LU-factorisation.

From the defining Equation (4.2), we see that the polynomial only has even powers of ν. So instead of working with a polynomial of degree $2N$, we substitute $y = \nu^2$ and have a polynomial of size N to solve for. Evidently, this also reduces the computational cost significantly.

The algorithm we implemented in our code for solving this Vandermonde system is described in [Hig02, Chap. 22, Algorithm 22.2]. The cost of the algorithm is $6n^2$ flops. Unfortunately, Vandermonde matrices are often extremely ill-conditioned, even worse so for large matrices. This ill-conditioning is a consequence of the monomials being a poor basis for the polynomials. However, in some cases the aforementioned algorithm can accurately calculate a solution even when the condition number is awfully high [BP70]. Higham writes down the following heuristic: "... systems with large normed solution that are solved to high accuracy by the fast algorithms".

The ill-conditioning can be mitigated a little bit by choosing the interpolation points in an intelligent fashion. It is very unwise to take equidistant interpolation points, the asymptotic behaviour of the condition number of V_n is then given by [Hig02, Table 22.1, p.418]

$$\kappa_\infty(V_n) \sim \frac{\sqrt{2}}{4\pi} 8^n$$

which grows very fast with n. A much better choice are the Chebychev nodes (roots or maxima of the Chebychev polynomial $T_{n+1}(x)$) for which the condition number is given by

$$\kappa_\infty(V_n) \sim \frac{3^{3/4}}{4} \left(1+\sqrt{2}\right)^n.$$

There is only one set of points for which the Vandermonde matrix V_n is perfectly conditioned: the roots of unity. This would lead us to the use of a FFT algorithm, but we would need the complex evaluation of $\nu P_c(\nu)$.

There is one more technique to improve the condition number of the matrix V_n: re-order the interpolation points using the Leja ordering [Lej57]. This ordering is such that for the interpolation points α_i holds

$$\alpha_0 = \max_{0 \le i \le n} |\alpha_i| \tag{4.7a}$$

$$\prod_{k=0}^{j-1} |\alpha_j - \alpha_k| = \max_{i \ge j} |\alpha_i - \alpha_k|, \quad j = 1, \ldots, n-1. \tag{4.7b}$$

For a given set of $n+1$ points α_i, the Leja ordering can be computed in n^2 flops using Algorithm 4.1 [Hig02]. This re-ordering is equivalent to the pivoting done in Gaussian elimination with partial pivoting (GEPP), but can be calculated beforehand.

Input: n, degree of polynomial
Input: α_i, $i = 0, \ldots, n$, Chebychev points
Output: α_i, $i = 0, \ldots, n$, Leja-ordered Chebychev points

$t \leftarrow \max_{0 \le i \le n} |\alpha_i|$;
if $i \neq 1$ **then**
  $\alpha_1 \leftrightarrow \alpha_i$;
endif
$p_i \leftarrow 1$, $i = 1, \ldots, n$;
for $k \leftarrow 2$ **to** $n-1$ **do**
  for $i \leftarrow k$ **to** n **do**
    $p_i \leftarrow p_i\,(\alpha_i - \alpha_{k-1})$;
  endfor
  $t \leftarrow \max_{k \le i \le n} |p_i|$;
  $i \leftarrow i + k - 1$;
  if $i \neq k$ **then**
    $\alpha_i \leftrightarrow \alpha_k$;
    $p_i \leftrightarrow p_k$;
  endif
endfor

Algorithm 4.1: Algorithm for Leja ordering.

When the interpolation process is finished, we can build the companion matrix and start the eigenvalue calculation. The matrix is first bal-

anced [TT94; GVL96] and then all eigenvalues are calculated. We use the well-known LAPACK library [ABB+99] for all linear algebra operations and related calculations. The companion matrix is first balanced using the DGEBAL routine and the eigenvalues are calculated using the DGEEV routine, both from LAPACK.

In the case of low degree of anisotropy, the Vandermonde approach remains stable. In our computer experiments we could go up to degree 10 in anisotropy. For higher degrees of anisotropy, the "coefficient problem" becomes so ill-conditioned that we have large relative errors on the polynomial coefficients and hence the roots of this polynomial are no good approximations for the roots of the real $P_c(\nu)$.

Second approach: The companion matrix method revisited

The main source of trouble in the previous approach is clearly the Vandermonde matrix. The calculation of the polynomial coefficients from interpolation values is an ill-posed problem. This is mainly due to the badness of the basis functions x^k to represent polynomials. A much better set of basis functions for polynomials is the Chebyshev basis where the Chebyshev polynomials $T_k(x)$ form the basis functions.

Suppose we have a polynomial $p(x)$ of degree D that we express using the Chebyshev basis:

$$p(x) = \sum_{k=0}^{D} a_k x^k = \sum_{k=0}^{D} c_k T_k(x). \tag{4.8}$$

The question that arises is, of course, how to use this representation to calculate all roots of $p(x)$. As far as we know, there is no expression available for the companion matrix of $p(x)$ using the Chebyshev coefficients c_k. However, there exists a nice trick [Boy02] called "degree-doubling".

Theorem 4.1. *Theorem 5.1 in [Boy02]. Define a polynomial $h(z)$ of twice the degree by*

$$h(z) = \sum_{k=0}^{2D} b_k z^k \tag{4.9}$$

where

$$b_k = \begin{cases} c_{k-D}, & k > D \\ 2c_0, & k = D \\ c_{D-k}, & k < D. \end{cases} \tag{4.10}$$

Then the roots x_k of $p(x)$ on the real interval $[-1, +1]$ are related to the roots z_k of $h(z)$ on the unit disk in the complex z-plane through

$$x_k = \Re(z_k) \tag{4.11}$$

Proof. See [Boy02]. ■

Since we are only interested in the roots on $[0, +1]$, this theorem gives us what we need to get at a companion matrix: indeed, we have the power-basis coefficients of $h(z)$, so we can build the companion matrix and calculate its eigenvalues.

Let us apply the above on the characteristic polynomial $P_c(\nu)$. The first step to be done is building the Chebyshev expansion. We evaluate $P_c(\nu)$ using either Equation (4.2) or Equation (4.3). We can calculate the Chebyshev coefficients c_k using the following theorem:

Theorem 4.2. *For a polynomial $p(x)$ of degree D, the coefficients c_k in Equation (4.8) are given by*

$$c_k = \frac{2}{D+1} \sum_{j=0}^{D} p\left(\cos\theta_j\right) \cos k\theta_j,\ k = 1, \ldots, D \tag{4.12a}$$

$$c_0 = \frac{1}{D+1} \sum_{j=0}^{D} p\left(\cos\theta_j\right) \tag{4.12b}$$

with

$$\theta_j = \frac{\pi\left(j + \frac{1}{2}\right)}{D+1}. \tag{4.13}$$

Proof. See [Buy92]. ■

Building this Chebyshev expansion has a second advantage: we need to do this only once and can use Smith's algorithm [Buy92, Algorithm 1.2]

(sometimes called Clenshaw's recurrence [PTVF96]) to evaluate

$$P_c(\nu) = \sum_{k=0}^{2N} c_k T_k(x).$$

This is much cheaper compared to the repeated use of Equations (4.2)–(4.3). It can even be made cheaper by noting from its definition (4.2) that $P_c(\nu)$ is an even function. As a consequence, the c_k for k odd are zero. Hence, we need half the storage and half the computation time.

Now that we have the Chebyshev coefficients c_k, we can perform the degree doubling as defined in Theorem 4.1. This leads us to a polynomial $h(z)$ of degree $4N$ (double of the degree of $P_c(\nu)$). However, again because $P_c(\nu)$ is an even function, the b_k with k odd are zero. By setting $u = z^2$, we again halve the degree needed, back to $2N$. We build the companion matrix using the even b_k coefficients (i.e. the companion matrix belonging to the polynomial $\tilde{h}(u)$) and calculate its eigenvalues which are the roots of $\tilde{h}(u)$. Let us denote them by u_k. Since we have $u = z^2$, the roots of $h(z)$ are given by the positive and negative square root of each u_k:

$$z_k = \{+\sqrt{u_1}, -\sqrt{u_1}, \ldots, +\sqrt{u_{2N}}, -\sqrt{u_{2N}}\}. \tag{4.14}$$

Applying the main result of Theorem 4.1, we have that all roots z_k that lie on the unit disk, i.e. $|z_k| = 1$, give rise to a root of $P_c(\nu)$, $\nu_k = \Re(z_k) \in [-1, +1]$. Hence, the final step in the approach is to loop over all roots and see if they lie on the unit disk. The complete algorithm is summarized in Algorithm 4.2.

We have also implemented this approach in our code and, as could be expected, it is much more stable than the Vandermonde approach. We get about 10 correct digits on the roots of $P_c(\nu)$ in $[0, +1]$ for an anisotropy degree of 50.

We conclude this section with an example of the Henyey-Greenstein scattering kernel with $N = 34$, $g = 0.70$ and $c = 0.90$. Figure 4.1 shows the roots u_k of the polynomial $\tilde{h}(u)$. We clearly see 4 roots lying on the unit circle, two by two symmetrical around the real axis, hence giving rise to 2 roots for $P_c(\nu)$ on the real line segment $[0, +1]$, as shown in Figure 4.2.

Third approach: The Newton-Maehly solver

In the case of very high degree anisotropy, where the companion matrix solution has lost accuracy and computation time would be too high (calculating the eigenvalues of a matrix of size n is an $\mathcal{O}(n^3)$ process, the balancing

Input: c, N, f_k, $k = 0, \ldots, N$
Output: All roots of $P_c(\nu)$ lying on the real line segment $[0, +1]$
for $j = 0$ **to** $2N$ **do**
  $\theta_j \leftarrow \frac{\pi(j+\frac{1}{2})}{2N+1}$;
endfor
$c_0 \leftarrow \frac{1}{2N+1} \sum_{j=0}^{2N} P_c(\cos\theta_j)$;
for $k = 1$ **to** $2N$ **do**
  $c_k \leftarrow \frac{2}{2N+1} \sum_{j=0}^{2N} P_c(\cos\theta_j) \cos k\theta_j$;
endfor
$b_0 \leftarrow 2c_0$;
for $k = 1$ **to** $2N$ **do**
  $b_k \leftarrow c_{|k-2N|}$;
endfor
Build companion matrix C taking into account $u = \nu^2$;
Balance C using DGEBAL (LAPACK);
Calculate eigenvalues $\lambda_j, j = 1, \ldots, 2N$ of C using DGEEV (LAPACK);
for $j = 1$ **to** $2N$ **do**
  if λ_j *on the unit circle* ($|\lambda_j| = 1$) **then**
    Add $\Re\left(\sqrt{\lambda_j}\right)$ to the list of roots of $P_c(\nu)$;
  endif
endfor

Algorithm 4.2: The Chebyshev companion matrix method.

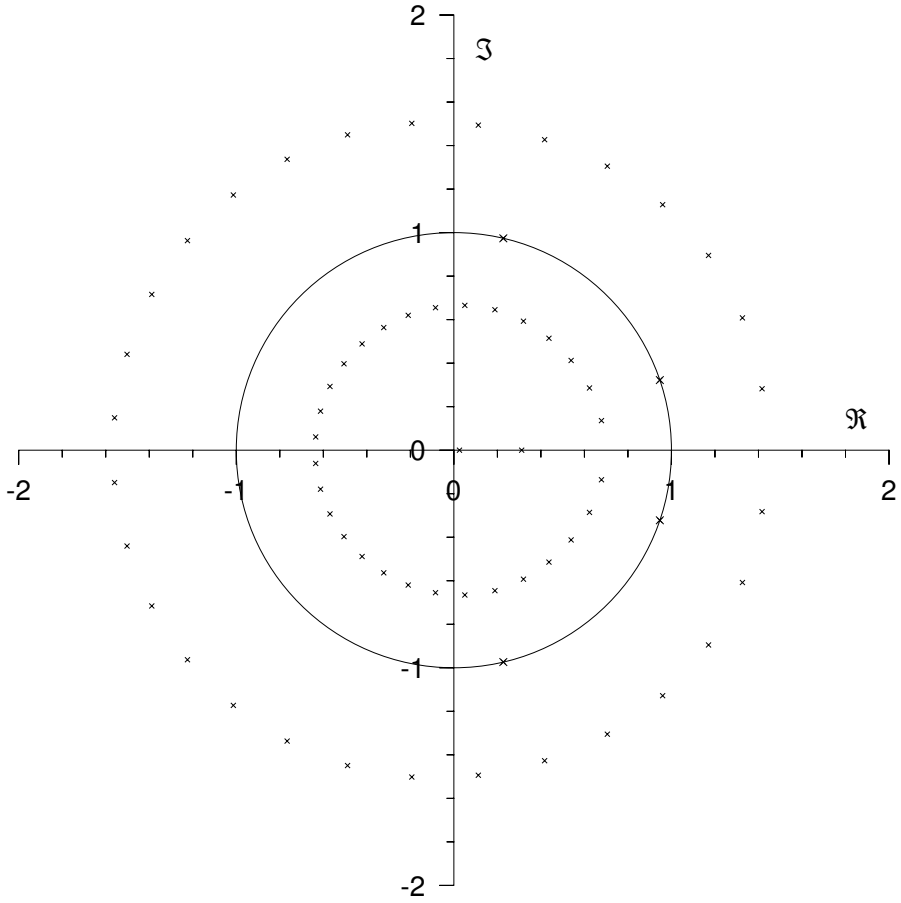


Figure 4.1: *Roots of the double degree polynomial close to the unit circle for* $HG_{34}(\mu|0.70)$*,* $c = 0.90$*.*

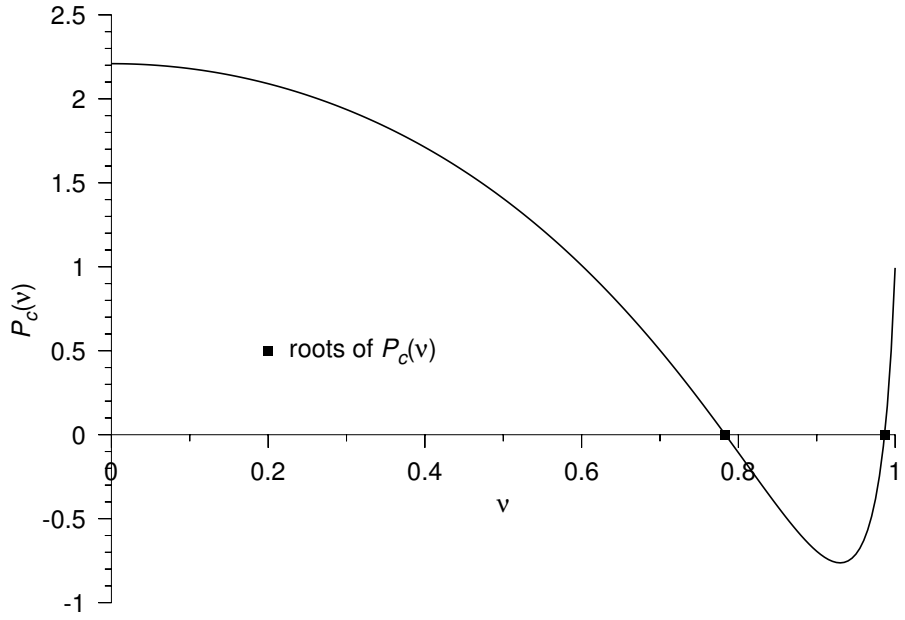


Figure 4.2: *Characteristic polynomial for* $HG_{34}(\mu|0.70), c = 0.90$.

needed beforehand an $\mathcal{O}(n^2)$ process), we use another approach: we apply the classical Newton-Raphson iteration

$$x_{k+1} = x_k - \frac{f(x)}{f'(x)} \tag{4.15}$$

which is a quadratically convergent method. We use Maehly's implicit deflation trick [BS62] to eliminate the roots already found. Maehly showed that the derivative of the reduced polynomial

$$P_j(x) = \frac{P(x)}{(x - x_1) \cdots (x - x_j)} \tag{4.16}$$

can be written as

$$P_j'(x) = \frac{P'(x)}{(x - x_1) \cdots (x - x_j)} - \frac{P(x)}{(x - x_1) \cdots (x - x_j)} \sum_{i=1}^{j} (x - x_i)^{-1} \tag{4.17}$$

and hence the Newton-Raphson iteration (4.15) can be rewritten as

$$x_{k+1} = x_k - \frac{P(x_k)}{P'(x_k) - P(x_k) \sum_{i=1}^{j} (x - x_i)^{-1}}. \tag{4.18}$$

It is the latter iteration we use for solving high order anisotropy cases. The question arises, of course, which starting point should be chosen to start the iteration. From computational experiments, we have seen that the roots of the characteristic polynomial have the tendency to cluster near $\nu = 1$. We use the so-called "box" approach in the Newton-Raphson solver: to limit computation time and the probability of divergence, we limit our iteration points to the segment $[0, +1]$. As soon as the iteration ends up outside this box, we restart the routine with a different starting point (a little bit further away from $\nu = 1$). Based on some heuristics based on our computer experiments, we have set some limits on the maximum number of "out-of-box" events. The method is summarised in Algorithm 4.3.

This approach works quite good. The main disadvantage is that we rely on an iterative process and hence loose the guarantee that we find all roots in $[0, +1]$. The parameters appearing in Algorithm 4.3 like *maxiter* and *maxnoconvergence* are based on computer experiments and heuristics. They are not optimal (if such a thing would be possible) with respect to computational cost.

Conclusion

Since the Chebyshev approach with degree doubling gives accurate results up to an anisotropy degree of 50, we prefer to use this method for those cases. Using the Chebyshev approach and the companion matrix eigenvalues, we have a guarantee that we have located all roots that lie on the real line segment $[0, +1]$. The Vandermonde approach has been abandoned in favour of this Chebyshev method. For higher degrees, the code switches automatically to the Newton-Raphson iteration with Maehly implicit deflation. We loose however the guarantee that we find all roots.

4.3 The characteristic equation

Now that we have found the number of positive discrete eigenvalues M, we can start the real work on calculating them. We have assumed that these discrete eigenvalues are real, so our search can be limited to the real segment $]+1, +\infty[$.

One would be tempted to use the quadratically convergent Newton-Raphson method to attack the problem. However, the typical form of $\Lambda(\nu)$ doesn't allow this so easily. Close to $\nu = +1$, the derivative of $\Lambda(\nu)$ becomes very large (in absolute value) due to the singularity of the Legendre functions of the second kind $Q_k(\nu)$ which results in very small steps in the

Input: c, N, f_k, $k = 0, \ldots, N$
Output: All roots of $P_c(\nu)$ lying on the real line segment $[0, +1]$

$maxnoconvergence \leftarrow 10$;
$maxiter \leftarrow 60$;
$maxstep \leftarrow 1.0e10$;
$tol \leftarrow 1.0e-12$;
$\delta \leftarrow 0$;
$j \leftarrow 0$;
while $j < maxnoconvergence$ **do**
  $\nu_0 \leftarrow 1 - \delta$;
  $iter \leftarrow 0$;
  $converged \leftarrow false$;
  while $iter < maxiter$ **and not** $converged$ **do**
    $iter \leftarrow iter + 1$;
    $step \leftarrow NewtonRaphsonMaehly(\nu)$;
    if $|step| > maxstep$ **then**
      NewtonRaphsonMaehly step too large $converged \leftarrow false$;
    else
      $\nu \leftarrow \nu + step$;
    endif
    if $|\nu| > 1.25$ **then**
      out of box $converged \leftarrow false$;
    else
      if $|P_c(\nu)| \leq tol \times |\nu P_c'(\nu)|$ **then**
        convergence, add root to list and restart $converged \leftarrow true$;
        $\delta \leftarrow 0$;
        $j \leftarrow 0$;
      endif
    endif
    if not $converged$ **then**
      no convergence, shift initial point $j \leftarrow j + 1$;
      $\delta \leftarrow 10^{-4 \times \left(\frac{maxnoconvergence - j}{maxnoconvergence - 1}\right)}$;
    endif
  endwhile
endwhile

Algorithm 4.3: The Newton-Maehly method.

Newton-Raphson iteration (4.15). On the other side of the axis, for $\nu \to \infty$, we have

$$\lim_{\nu\to\infty} \Lambda(\nu) = \prod_{l=0}^{N} (1 - cf_l) \tag{4.19}$$

and this (constant) limit is reached quite rapidly, hence the derivative of $\Lambda(\nu)$ approaches zero which results in a singularity in the Newton-Raphson iteration. Finally, the choice of a starting point is not evident.

Our approach is again two-fold: first we apply a bracketing procedure. The outcome of this procedure is a set of brackets dividing the real line in different segments and these brackets have the property that they contain exactly one root of the characteristic equation. The second step consists of solving the characteristic equation in each segment for the single root in that segment.

An upper bound for the largest root

We cannot bracket the infinite interval $]+1, +\infty[$. In one way or another, we need to stop the process somewhere. We need to have a bound on the largest root of the characteristic equation and use this as end-point of our initial interval. Let us call this bound ξ so we have $\nu_1 \leq \xi$. Our aim is to limit the bracketing process to the interval $]+1, \xi[$.

A property of the transport polynomials $\phi_k(\nu)$ comes to our aid. From Property 3.1, item (iii), we know that the roots of $\Lambda(\nu)$ are reached by the roots of the $\phi_k(\nu)$ for $k \to \infty$ and the approach is from the left (for $\nu > 1$, the case we are interested in). An important consequence of this property is the fact that the spectrum of the transport polynomials, i.e. the interval containing all roots, is bounded. This can also be deduced from the recurrence Equation (3.2), taking into account the convergence properties on the Legendre coefficients mentioned in Chapter 2, as proved in [Chi90]. Suppose we would be able to get an upper bound τ on this spectrum, i.e. an upper bound on the largest root x_{kk} of $\phi_k(\nu)$ for $k \to \infty$:

$$x_{kk} \leq \tau \quad \forall k. \tag{4.20}$$

However, due to the direction of the approach in the limit (from the left)

$$\lim_{k\to\infty} x_{kk} = \nu_1 \text{ and } x_{kk} \leq \nu_1 \tag{4.21}$$

the upper bound τ is not necessarily an upper bound for ν_1. The bound could be located in the gap between x_{kk} and ν_1. However, in our computer

experiments, we have never come across this situation. Even so, to make our code robust we have added a check to see if at the position $\nu = \tau$, the characteristic equation is already close to its asymptotic value. If not, the bound for the bracketing process ξ is shifted to the right until we are in the asymptotic regime.

What remains to derive is, of course, the upper bound τ on the spectrum of the transport polynomials. In his paper [Fre86], Freud provides such a bound for the case of polynomials orthogonal with respect to a symmetric weight function, which is the case here. The general case has been treated later in [KL99].

Let us rewrite the recurrence (3.2) in its orthonormal form:

$$\nu\tilde{\phi}_k(\nu) = a_{k+1}\tilde{\phi}_{k+1}(\nu) + a_k\tilde{\phi}_{k-1}(\nu) \tag{4.22}$$

with the coefficients a_k given by

$$a_k = \frac{k}{\sqrt{(1-cf_k)\,(1-cf_{k-1})\,(2k+1)\,(2k-1)}}. \tag{4.23}$$

From [Fre86, Theorem 4]: the largest root x_{kk} of the polynomial $\tilde{\phi}_k(x)$ belonging to the orthonormal system (4.22) can be bounded by

$$x_{kk} \leq 2\cos\left(\frac{\pi}{k+1}\right) \max_{j\leq k-2} a_{j+1}, \tag{4.24}$$

so if we take the limit $k \to \infty$, we find

$$\lim_{k\to\infty} x_{kk} = \tau \leq 2 \max_j a_{j+1}. \tag{4.25}$$

Given the fact that $|f_k| \leq 1$, this simplifies to

$$x_{kk} \leq \tau \leq \frac{2}{\sqrt{3}} \frac{1}{(1-c)}. \tag{4.26}$$

We have reduced our infinite interval to the interval $]+1, \xi[$, with ξ given by

$$\xi = \frac{2}{\sqrt{3}} \frac{1}{(1-c)} \tag{4.27}$$

or, when needed, shifted to arrive in the asymptotic region of $\Lambda(\nu)$. It is this interval that we feed to the bracketing process described in the next section. To conclude, we give some examples for the Henyey-Greenstein scattering kernel for $N = 10$, $g = 0.25, 0.50$ and $c = 0.20, 0.80$ in Table 4.1. The first

g	c	ν_1	$x_{50,50}$	ξ
0.25	0.20	$1.003342 \times 10^{+00}$	$1.003337 \times 10^{+00}$	$1.443376 \times 10^{+00}$
0.25	0.80	$1.576178 \times 10^{+00}$	$1.576178 \times 10^{+00}$	$5.773503 \times 10^{+00}$
0.50	0.20	$1.022964 \times 10^{+00}$	$1.022964 \times 10^{+00}$	$1.443376 \times 10^{+00}$
0.50	0.80	$1.845585 \times 10^{+00}$	$1.845585 \times 10^{+00}$	$5.773503 \times 10^{+00}$

Table 4.1: *An upper bound for the largest root, an example.*

entry clearly shows the approach from the left by the largest root of ϕ_{50} to the real discrete eigenvalue.

Unfortunately, the bound obtained by Equation (4.27) usually largely over-estimates the real position of the largest root of $\phi_k(\nu)$. This becomes important for higher order scattering problems where we risk to have overflow for $\phi_k(\xi)$ and underflow for $Q_k(\xi)$ for too large values of ξ. We encountered these type of problems in the Cloud C_1 problem (a $N = 299$ problem!) described later on in this chapter.

Therefore, we modified our code to loop over the different a_k's of Equation (4.23) and keep track of the maximum value. This approach reduced the upper bound significantly, but still not enough for the Cloud C_1 problem.

The final solution we adopted was to explicitly calculate the largest root of a higher order $\phi_k(\nu)$. The roots of an orthogonal polynomial can easily be calculated as the eigenvalues of a symmetric tridiagonal matrix [See Sze59; Buy92; PTVF96] with its elements defined by the recurrence coefficients. The calculation of these eigenvalues can be done by a modified QL-algorithm as indicated in [GVL96; PTVF96]. At the moment we use, quite arbitrarily, the largest root of $\phi_{20}(\nu)$. Computational experiments showed this to be a sufficiently high order, while still keeping the computational cost low. The approach here can be fastened by avoiding the full QL algorithm and focus on the largest eigenvalue, which is the only one we're interested in.

The bracketing procedure

The purpose of this procedure is to produce a set of brackets that divide the real line segment $]+1, \xi]$ in smaller segments each containing exactly one root. It boils down to a recursively refined grid of evaluation points on $]+1, \xi[$. In each iteration step, the series of points is evaluated for sign changes and those are recorded as the brackets. This bracketing works since

we know that the roots of the characteristic equation are simple and hence a sign change occurs for every root. When the whole interval is scanned, we compare the number of segments containing one root we found to the real number of discrete eigenvalues we found using the characteristic polynomial approach. When they are equal the recursion stops, if not the grid is refined once more. During the recursion, the previous evaluations are, of course, retained. We use the C++ `map` [Str97] structure for the storage of these evaluations since this structure has efficient insertion properties (insertion in a sorted structure).

The main issue to deal with here is the grid and refinement specifications. We have opted for a logarithmically spaced grid. We have based this choice on the general shape of $\Lambda(\nu)$. Typically, the characteristic equation oscillates (in the case of multiple eigenvalues) for values of ν close to one, while for larger values the function quickly goes to its asymptotic state. Hence, we can expect a clustering of the discrete eigenvalues close to $\nu = 1$ and need not have a very fine grid for larger ν.

In the first level of the recursion we start with an $n = M + 1$ point grid (M is the number of positive discrete eigenvalues) defined by

$$x_0 = 1.0,\ x_{k+1} = h \times x_k,\ h = \xi^{1/n}. \tag{4.28}$$

If the number of segments containing one root is not equal to M, we refine the grid by doubling n. The procedure is indicated in Algorithm 4.4. Take for example the problem of the Henyey-Greenstein scattering kernel with $N = 34$, $g = 0.70$ and $c = 0.90$. The characteristic polynomial $P_c(\nu)$ has in this case two roots in $[0, +1]$, hence we know that there are three positive discrete eigenvalues. The upper bound ξ is given by

$$\xi = \frac{2}{\sqrt{3}} \frac{1}{(1-c)} \approx 11.5470.$$

Our algorithm needs four grids to find the correct brackets. Figure 4.3 shows the first grid, the three positive discrete eigenvalues and the brackets it found. Figure 4.4 shows the final grid with the correct brackets. Table 4.2 shows the real line segments obtained (together with the root in that segment). As a second example, and justification for the logarithmic approach, we take the binomial scattering law of order $N = 20$, for different values of c. Table 4.3 shows the recursion depth needed together with the map size, i.e. the number of function evaluations.

Input: ξ, $\Lambda(\nu)$, the number of positive discrete eigenvalues M
Output: a set of brackets dividing $[1, \xi]$ in segments containing exactly one root of $\Lambda(\nu)$

$n \leftarrow (M+1)/2$;
$Nroots \leftarrow 0$;
repeat
  $n \leftarrow n \times 2$;
  $h \leftarrow \xi^{1/n}$;
  $\nu_0 \leftarrow 1.0$;
  for $j = 1$ **to** M **do**
    $\nu_j \leftarrow h \times \nu_{j-1}$;
    Add $\Lambda(\nu_j)$ to *map*;
  endfor
  Loop over *map* entries counting the sign changes $NRoots$;
until $NRoots = M$;

Algorithm 4.4: The logarithmic bracketing procedure.

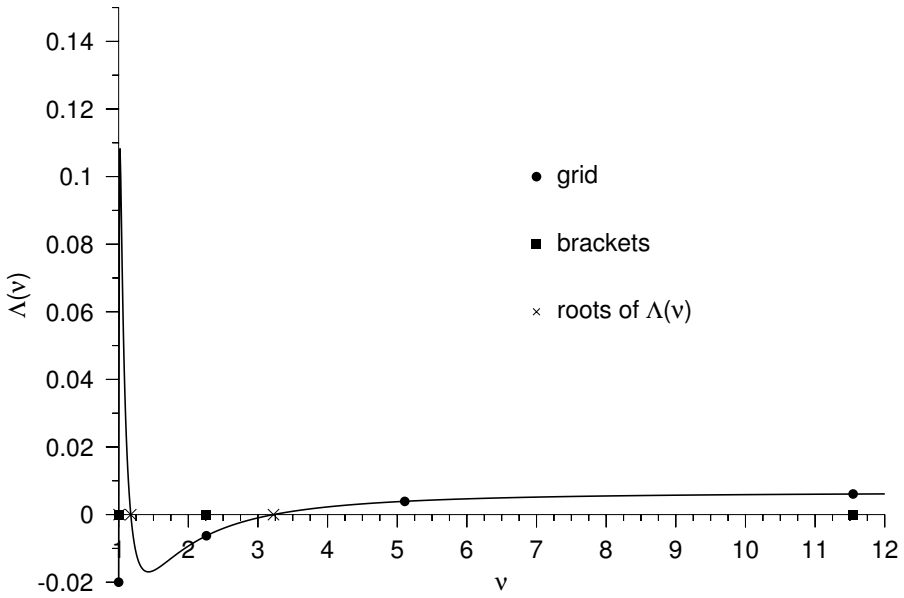


Figure 4.3: *Grid, brackets and roots of $\Lambda(\nu)$ in the first recursion level.*

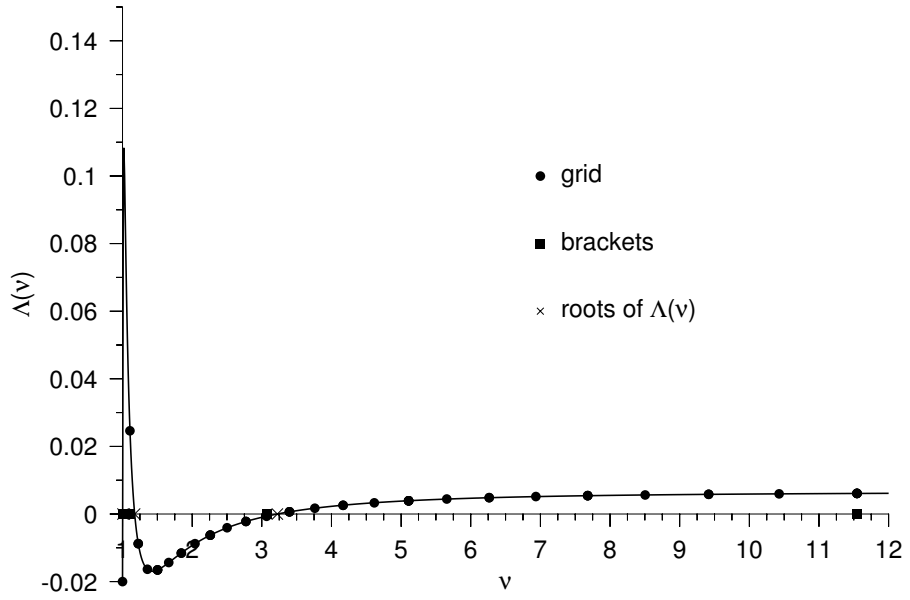


Figure 4.4: *Grid, brackets and roots of $\Lambda(\nu)$ in the final (4^{th}) recursion level.*

Left boundary	Right boundary	Root
$+1.000000 \times 10^{+00}$	$+1.107311 \times 10^{+00}$	$+1.004979 \times 10^{+00}$
$+1.107311 \times 10^{+00}$	$+3.068776 \times 10^{+00}$	$+1.174433 \times 10^{+00}$
$+3.068776 \times 10^{+00}$	$+1.154701 \times 10^{+01}$	$+3.223535 \times 10^{+00}$

Table 4.2: *Boundaries of the real line segments in the example.*

		Linear bracketing		Log bracketing	
c	M	Recursion depth	Map size	Recursion depth	Map size
0.1	1	1	2	1	2
0.3	2	2	7	2	5
0.5	2	2	7	1	3
0.7	2	2	5	1	3
0.9	3	4	36	2	9
0.95	3	5	82	3	19
0.99	3	7	341	3	17
0.999	3	10	2975	3	19

Table 4.3: *Comparison of linear versus logarithmic bracketing.*

Calculating the root in a bracketed interval

We now have a set of real line segments, each containing exactly one discrete eigenvalue. As mentioned before, the use of the quadratically convergent Newton-Raphson iteration is not a good idea due to the shape of $\Lambda(\nu)$. In our opinion, the best routine for the job is the Van Wijngaarden-Dekker-Brent (VWDB) routine [PTVF96]. Starting from a bracketed root, the method applies a clever combination of bisection and inverse quadratic interpolation. Its convergence ratio is super-linear with order ≈ 1.618. Heuristically, this means that we multiply the number of correct digits by 1.618 per iteration step.

Retaking the first example of the previous section, the Henyey-Greenstein scattering kernel, we show in Table 4.4 the roots of $\Lambda(\nu)$ found by the Van Wijngaarden-Dekker-Brent algorithm, the number of iterations needed and the relative error compared to MAPLE calculations.

The second example, the binomial scattering law with order $N = 20$, was also solved by Siewert [Sie80] in the case of $c = 0.95$. In Table 4.5, we compare the results from our approach to the results by Siewert and to MAPLE calculations.

Root	Iterations	Relative error
$+1.004979 \times 10^{+00}$	14	$\mathcal{O}(10^{-16})$
$+1.174433 \times 10^{+00}$	12	$\mathcal{O}(10^{-16})$
$+3.223535 \times 10^{+00}$	7	$\mathcal{O}(10^{-15})$

Table 4.4: *Roots found by the VWDB routine for example 1.*

g	Root of $P_c(\nu)$
0.4850	$+9.946419 \times 10^{-01}$
0.4825	$+9.964251 \times 10^{-01}$
0.4800	$+9.982365 \times 10^{-01}$

Table 4.6: *Roots of $P_c(\nu)$ for $HG(\mu|g)$, $N = 5$, $g = 0.4850, 0.4825, 0.4800$, $c = 0.80$.*

VWDB	Iterations	Relative error	Siewert	Relative error
$+1.019559 \times 10^{+00}$	12	$\mathcal{O}(10^{-12})$	$1.019559 \times 10^{+00}$ †	$\mathcal{O}(10^{-08})$
$+1.666787 \times 10^{+00}$	12	$\mathcal{O}(10^{-12})$	$1.666787 \times 10^{+00}$	$\mathcal{O}(10^{-08})$
$+7.480699 \times 10^{+00}$	8	$\mathcal{O}(10^{-12})$	$7.480699 \times 10^{+00}$	$\mathcal{O}(10^{-08})$

† In the original paper [Sie80], this result was 1.019586. After communication with the author it became clear there was a misprint and the result lacked a '5' digit at position 6.

Table 4.5: *Roots found by the VWDB routine for example 2.*

4.4 Near-singular eigenvalues

As an example of the occurrence of near-singular eigenvalues, we use a fifth-order approximation of the Henyey-Greenstein scattering kernel with the parameter $g = 0.4850$, 0.4825 and 0.4800. The number of secondaries per collision is taken to be $c = 0.8$.

Let us first see how the characteristic polynomial behaves in the interval of interest $[0, 1]$. Figure 4.5 shows that $P_c(\nu)$ for this problem has one root for all three values of g, indicating according to our Conjecture 2.2, two positive discrete eigenvalues. Table 4.6 shows the roots of $P_c(\nu)$ for each value of g.

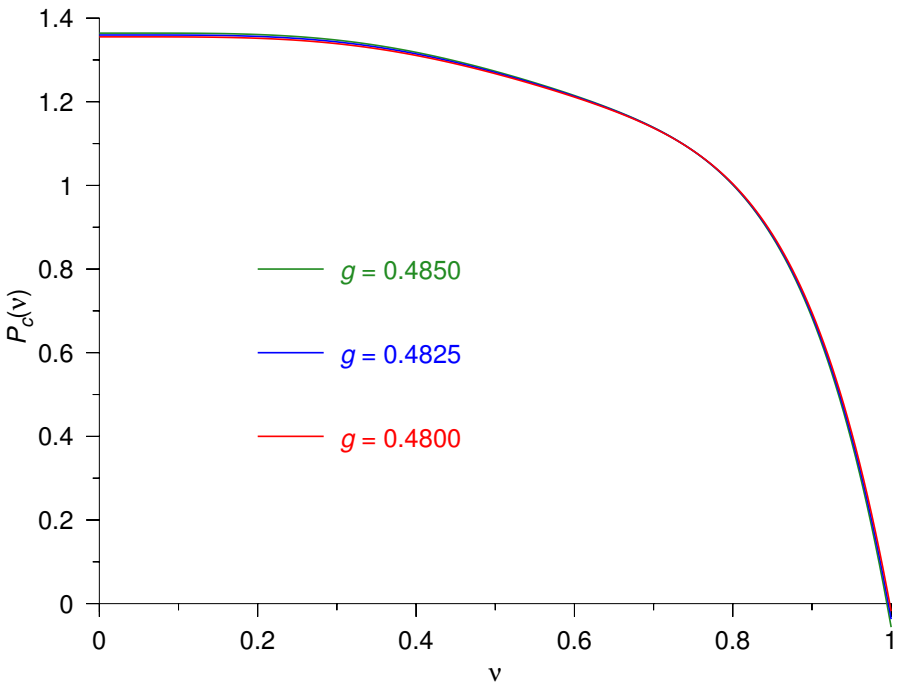


Figure 4.5: *Characteristic polynomial* $P_c(\nu)$ *for* $HG(\mu|g)$, $N = 5$, $g = 0.4850, 0.4825, 0.4800$, $c = 0.80$.

However, when plotting the characteristic equation $\Lambda(\nu)$, it seems that there is only one root (See Figure 4.6). Either there is no second root and our conjecture is proved false or we have missed the root. From Figure 4.6 it is clear that beyond $\nu = 2$, the function $\Lambda(\nu)$ is reaching for its asymptotic value, Equation (2.22),

$$\lim_{\nu\to+\infty} \Lambda(\nu) = \prod_{l=0}^{N}(1 - cf_l) = \begin{cases} +8.444809 \times 10^{-02} & \text{g} = 0.4850 \\ +8.518548 \times 10^{-02} & \text{g} = 0.4825 \\ +8.592294 \times 10^{-02} & \text{g} = 0.4800 \end{cases} \tag{4.29}$$

and hence we shouldn't expect a root to pop up there. However, looking at the sign of $\lim_{\nu\to1} \Lambda(\nu)$ by means of Equation (2.21) it is easily calculated that in all three cases, the limit is $+\infty$! So somewhere very close to $\nu = 1$, the characteristic equation must bend back up, cross the ν-axis and have another root, before shooting away to $+\infty$ at $\nu = 1$.

It is these type of roots very close to $\nu = 1$ that we have called near-singular roots. We need to accurately calculate the characteristic equation for values $\nu = 1 + \delta$, where δ can be very small. Since we work in IEEE double precision, as soon as δ is equal to or smaller than *macheps* , being about 2.2×10^{-16}, there is no difference anymore between 1 and $1 + \delta$. But even when δ is moderately small, there is a large loss of precision when calculating the eigenfunctions (2.20) for such an eigenvalue. Indeed, there is a term $\nu_j - \mu$ in the denominator of Equation (2.20). When ν_j is of the form $1 + \delta$ and μ is approaching 1, we suffer from catastrophic cancellation.

Accurate calculation of the characteristic equation

The idea presented here is to accurately calculate the solutions of the characteristic equation using not $\nu = 1 + \delta$ but δ itself. We therefore need approximations on one hand for $\phi_k(1 + \delta)$ and on the other hand for $Q_k(1 + \delta)$.

Backward and forward error bounds on $\phi_k(1 + \delta)$

Since the $\phi_k(\nu)$ are polynomials of degree k in ν, we apply no approximations nor tricks when $\nu = 1 + \delta$. As long as $\nu = 1 + \delta$ can be represented in IEEE double precision, we use the value of ν. We evaluate the polynomials using the classical Horner scheme. Higham [Hig02] provides a backward error bound for this algorithm: the computed value y is the exact value at

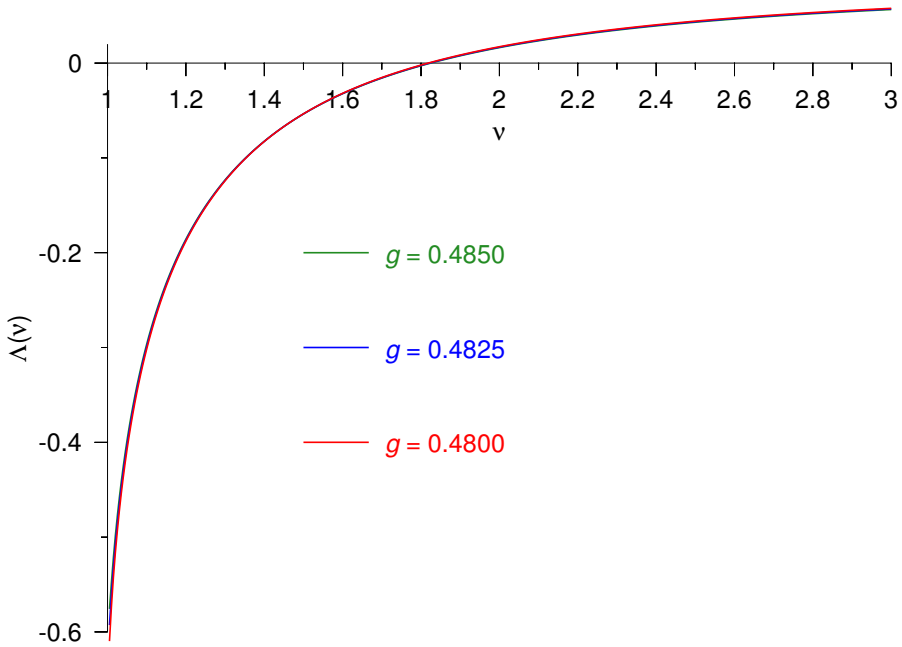


Figure 4.6: *Characteristic equation* $\Lambda(\nu)$ *for* $HG(\mu|g)$, $N = 5$, $g = 0.4850, 0.4825, 0.4800$, $c = 0.80$.

ν of a polynomial obtained by making relative perturbations of size at most γ_{2k} to the coefficients of $\phi_k(\nu)$, with γ_{2k} given by

$$\gamma_{2k} = \frac{2ku}{1-2ku} \tag{4.30}$$

where u denotes the unit roundoff and is of the size 1.1×10^{-16} for IEEE double precision. So as long as ku remains a lot smaller than unity, there is no instability.

Of course, as soon as δ becomes smaller than u, there is no difference anymore between 1 and $1+\delta$ (the value $1+\delta$ is rounded to 1). The error that we make in this case can be estimated. We first note the following lemma:

Lemma 4.3. *(From [Hig02, Lemma 3.4, p. 68], where we changed some notation in order not to cause confusion with our notation) If* $|\alpha_i| \leq \beta$ *and* $\rho_i = \pm 1$ *for* $i = 1, \ldots, n$ *and* $n\beta < 0.01$, *then*

$$\prod_{i=1}^{n}(1+\alpha_i)^{\rho_i} = 1 + \eta_n \tag{4.31}$$

where $|\eta_n| \leq 1.01n\beta$.

Theorem 4.4. *The absolute error between $\phi_k(1+\delta)$ and $\phi_k(1)$ can be bounded by*

$$|\phi_k(1+\delta) - \phi_k(1)| \leq 1.01k\delta \sum_{l=0}^{k} |a_l| \tag{4.32}$$

where a_l denote the coefficient of the powers ν^l in $\phi_k(\nu)$ when $k\delta \leq 0.01$.

Proof. If we assume the condition for the lemma above to hold, $k\delta \leq 0.01$, and see that $\alpha_i = \delta, \forall i$, we have

$$\begin{aligned}
\phi_k(1+\delta) &= \sum_{l=0}^{k} a_l (1+\delta)^l \\
&= \sum_{l=0}^{k} a_l (1+\eta_k), \qquad |\eta_l| \leq 1.01k\delta \\
&= \sum_{l=0}^{k} a_l + \sum_{l=0}^{k} a_l \eta_l \\
&= \phi_k(1) + \sum_{l=0}^{k} a_l \eta_l
\end{aligned}$$

and thus

$$\begin{aligned}
|\phi_k(1+\delta) - \phi_k(1)| &= |\sum_{l=0}^{k} a_l \eta_l| \\
&\leq \sum_{l=0}^{k} |a_l||\eta_l| \\
&\leq 1.01k\delta \sum_{l=0}^{k} |a_l|
\end{aligned}$$

what was to be proved. ■

An accurate calculation of $Q_k(1+\delta)$

The most critical part in calculating $\Lambda(\nu)$ for values of $\nu = 1+\delta$ is clearly the accurate calculation of the Legendre functions of the second kind, since

they become dominant close to unity due to the singularity there. The algorithm used to evaluate the Legendre functions of the second kind, the improved Miller (algorithm C.3), fails for small δ. The evaluation of the continued fraction using the modified Lentz algorithm does not converge quickly enough and introduces a lot of roundoff error. Hence, the initial values are wrongly calculated and thus one cannot expect the Miller algorithm to work properly. For extremely small δ we loose a lot of digits due to the fact that we cannot represent the argument $1+\delta$ with enough precision. We clearly need an algorithm that can calculate the Legendre function of the second kind based on δ and not on $1+\delta$ for its argument. Luckily, there exists an expansion for $Q_n(z)$ for $z \to 1$. We start from the expansion (A.32) [WFS04a], where we replace z by $1+\delta$. This gives us:

$$\begin{aligned} Q_n(1+\delta) = &\sum_{k=0}^{n} \frac{(k+n)!\psi(k+1)}{k!^2(n-k)!} \left(\frac{\delta}{2}\right)^k \\ &+ (-1)^k \left[\frac{1}{2}\left(\log\left(\frac{2+\delta}{\delta}\right)\right) - \psi(n+1)\right] \sum_{k=0}^{\infty} \frac{(-n)_k (n+1)_k}{k!^2} \left(\frac{\delta}{2}\right)^k \end{aligned} \tag{4.33}$$

in which $\psi(n)$ denotes the Euler Psi-function (or Digamma function)

$$\psi(n+1) = \sum_{m=1}^{n} \frac{1}{m} - \gamma, \quad n \in \mathbb{N}_0$$

and $(n)_k$ denotes the Pochhammer symbol

$$(n)_k = n(n+1)\cdots(n+k-1) = \frac{\Gamma(n+k)}{\Gamma(n)}.$$

Let us reorder the summations so that we have one summation from $k=0$ up to $k=n$ and the second one from $k=n+1$ to $k=\infty$:

$$\begin{aligned} Q_n(1+\delta) = &\sum_{k=0}^{n} \left\{ \frac{(k+n)!\psi(k+1)}{k!^2(n-k)!} + \right. \\ &\left. (-1)^k \left[\frac{1}{2}\left(\log\left(\frac{2+\delta}{\delta}\right)\right) - \psi(n+1)\right] \frac{(-n)_k (n+1)_k}{k!^2} \right\} \left(\frac{\delta}{2}\right)^k + \\ &\sum_{k=n+1}^{\infty} (-1)^k \left[\frac{1}{2}\left(\log\left(\frac{2+\delta}{\delta}\right)\right) - \psi(n+1)\right] \frac{(-n)_k (n+1)_k}{k!^2} \left(\frac{\delta}{2}\right)^k. \end{aligned} \tag{4.34}$$

Let us define the functions $t(k,n)$, $u(k,n)$ and $v(n,\delta)$ as

$$t(k,n) = \frac{(k+n)!\psi(k+1)}{k!^2(n-k)!}, \tag{4.35}$$

$$u(k,n) = \frac{(-n)_k(n+1)_k}{k!^2}, \tag{4.36}$$

and

$$v(n,\delta) = \frac{1}{2}\left(\log\left(\frac{2+\delta}{\delta}\right)\right) - \psi(n+1). \tag{4.37}$$

Using these definitions above, we can rewrite our expression for $Q_n(1+\delta)$ as

$$Q_n(1+\delta) = \sum_{k=0}^{n}\left\{t(k,n) + (-1)^k u(k,n)v(n,\delta)\right\}\left(\frac{\delta}{2}\right)^k + \sum_{k=n+1}^{\infty}\left\{(-1)^k u(k,n)v(n,\delta)\right\}\left(\frac{\delta}{2}\right)^k. \tag{4.38}$$

Both $t(k,n)$ and $u(k,n)$ can be calculated based on values $t(k-1,n)$ and $u(k-1,n)$ and hence a two-term recurrence can be derived which greatly reduces the cost of evaluation. The $v(n,\delta)$ is no function of the running index k and can be calculated once before the summation starts.

The relations between the previous and current value of $t(k,n)$ and $u(k,n)$ are given by

$$t(0,n) = -\gamma, \tag{4.39a}$$

$$t(k,n) = \frac{(n+k)(n-k+1)}{k^2}\frac{\psi(k+1)}{\psi(k)}t(k-1,n), \tag{4.39b}$$

and

$$u(0,n) = 1, \tag{4.40a}$$

$$u(k,n) = -\frac{(n+k)(n-k+1)}{k^2}u(k-1,n). \tag{4.40b}$$

The relations above give us an efficient algorithm to calculate $Q_n(1+\delta)$ since the subsequent terms in the summation have a factor $(\delta/2)^k$ which

when δ is sufficiently small, quickly goes to zero. However, we should be wary of the numerical stability of the calculation and more precisely in the summation. Let us first prove some basic properties of the $t(k,n)$ and $u(k,n)$.

Lemma 4.5. *The $t(k,n)$ for $k = 1,\ldots,n$ are always positive. Only $t(0,n)$ is negative.*

Proof. By definition $t(0,n) = -\gamma < 0$. Also by its definition, only $\psi(1)$ is negative. So $\psi(k+1)/\psi(k)$ is negative for $k = 1$ and positive for $k > 1$. The factor $(n+k)(n-k+1)/k^2$ is always positive for $k = 0,\ldots,n$. From these two facts, it follows that $t(1,n)$ is positive and from there on, the higher order $t(k,n)$ are positive. ■

Lemma 4.6. *The $u(k,n)$ are positive for k even, negative for k odd and zero for $k > n$.*

Proof. The factor $(n+k)(n-k+1)/k^2$ is positive for $k \leq n$. The minus sign in Equation (4.40b) makes the $u(k,n)$ changing sign over and over. For $k = n+1$, the factor $(n+k)(n-k+1)/k^2$ becomes zero and hence $u(n+1,n) = 0$. By the recurrence, the $u(k,n)$ remain zero for $k > n+1$. ■

From Lemma 4.6 we know that $u(k,n) = 0$ for $k \geq n+1$. Hence the second summation in Equation (4.38) vanishes completely. This leaves us with

$$Q_n(1+\delta) = \sum_{k=0}^{n} \left\{ t(k,n) + (-1)^k u(k,n) v(n,\delta) \right\} \left(\frac{\delta}{2}\right)^k. \tag{4.41}$$

The previous expression for $Q_n(1+\delta)$ is still an exact formula. There were no approximations made to arrive at this result. Hence, if we evaluate this expression, using the simplified evaluations of $t(k,n)$ and $u(k,n)$ using their recurrences (4.39-4.40) we find a correct value for $Q_n(1+\delta)$.

However, as mentioned before, we should take care of the numerical stability when evaluating Equation (4.41). Let us take as an illustrative example the evaluation of $Q_{20}(1+\delta)$ for $\delta = 10^{-01}$ and $\delta = 10^{-06}$. Table 4.7 shows the results compared to MAPLE high-precision results.

The source for the instability is the possibility of having an alternating sign in the consecutive terms in the sum of Equation (4.41). Indeed, Figure 4.7 shows, as a function of the summation index k, the two contributions $t(k,n)$ and $(-1)^k u(k,n) v(\delta,n)$ for both examples, $n = 20, \delta = 10^{-06}$ on top and $n = 20, \delta = 10^{-01}$ below. We clearly see that for the case of $\delta = 0^{-01}$ we

δ	Equation 4.41	MAPLE	Relative error
10^{-01}	$4.5335980129 \times 10^{-05}$	$4.5335980203 \times 10^{-05}$	$\mathcal{O}(10^{-9})$
10^{-06}	3.6575674025	3.6575674025	$\mathcal{O}(10^{-16})$

Table 4.7: *Comparison of Equation (4.41) for $Q_{20}(1+\delta)$ with* MAPLE *results.*

are adding and subtracting large numbers, which is the cause for the instability of the summation. Figure 4.8 shows the total term (including the convergence factor $(\delta/2)^k$) in the summation for consecutive k. In Figure 4.8b we see that we are adding and subtracting numbers of order of magnitude 10^{+1} to arrive at a final result of order of magnitude 10^{-5}. Clearly, this is not very good for the stability of the evaluation.

It is, of course, important to know for which values of n and δ this summation is numerically stable. In the next paragraph, we shall derive a sufficient condition, linking n and δ, for stability and give numerical values for both n and δ.

Theorem 4.7. *A sufficient condition on n and δ for the numerical stability of the summation of Equation (4.41) is*

$$\frac{1}{2}\log\left(1+\frac{2}{\delta}\right) \geq \sum_{m=1}^{n}\frac{1}{m}. \tag{4.42}$$

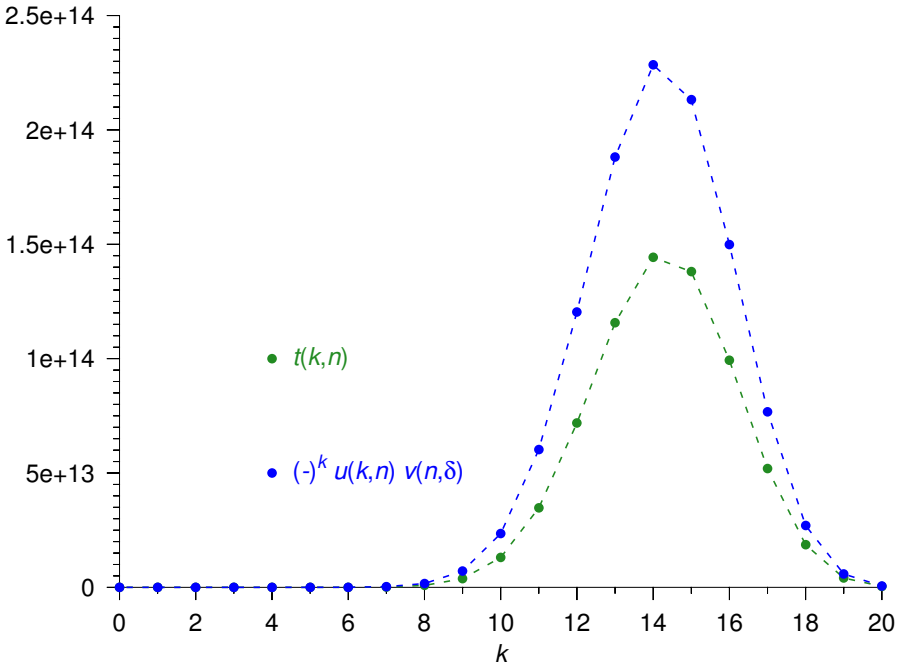
Proof. From [Hig02, p.90, item 3] : a sufficient condition for stable summation of n terms is that all terms have the same sign. The relative error on this type of sum is bound by $n \times u$, where u is the unit roundoff. To have all terms positive in the sum of Equation (4.41) means

$$t(k,n) + (-1)^k u(k,n) v(n,\delta) \geq 0 \quad k = 0,\ldots,n. \tag{4.43}$$

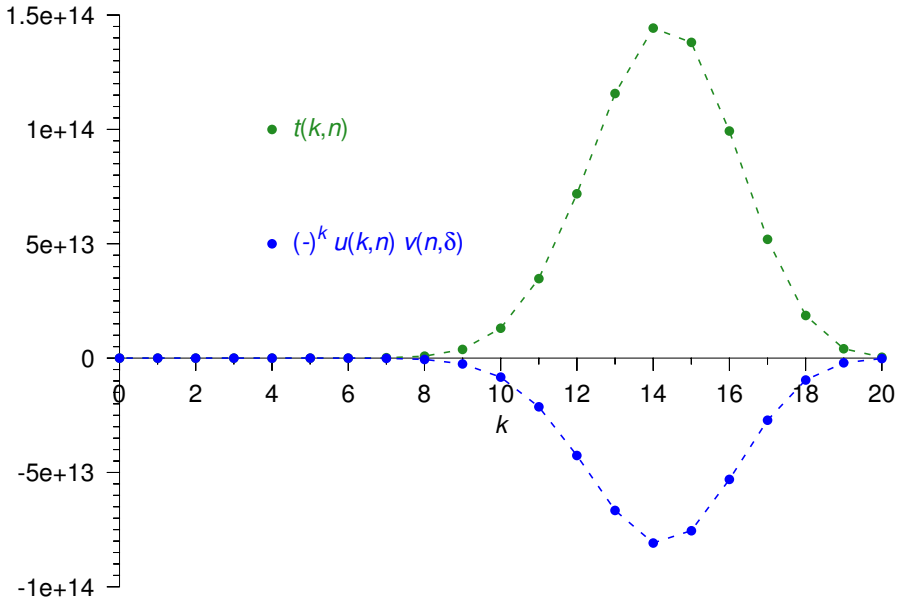
We know from Lemma 4.6 that $(-1)^k u(k,n)$ is always positive for $k = 0,\ldots,n$. We also know from Lemma 4.5 that $t(k,n)$ is positive for $k = 1,\ldots,n$ and only $t(0,n)$ is negative. Hence, to have the requirement (4.43) fulfilled for $k = 1,\ldots,n$, $v(n,\delta)$ has to be positive. However, this is not sufficient since $t(0,n)$ is negative. So the final requirement (taking into account that $u(0,n) = 1$) is

$$v(n,\delta) \geq -t(0,n) \tag{4.44}$$

which when inserting the definitions of $v(n,\delta)$, $t(0,n)$ and $\psi(n+1)$ leads to

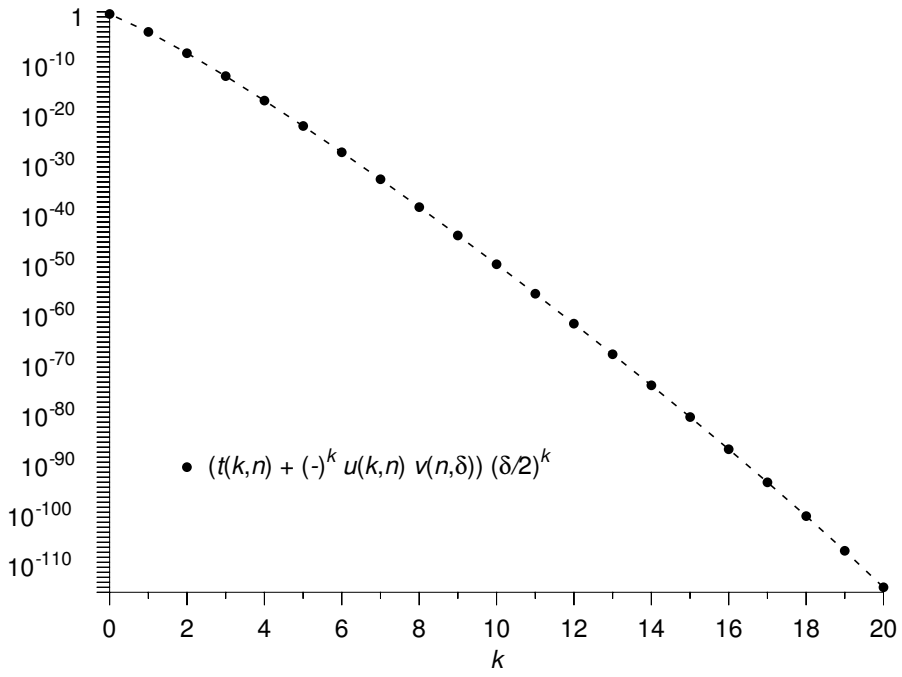


a. $n = 20, \delta = 1.0 \times 10^{-06}$

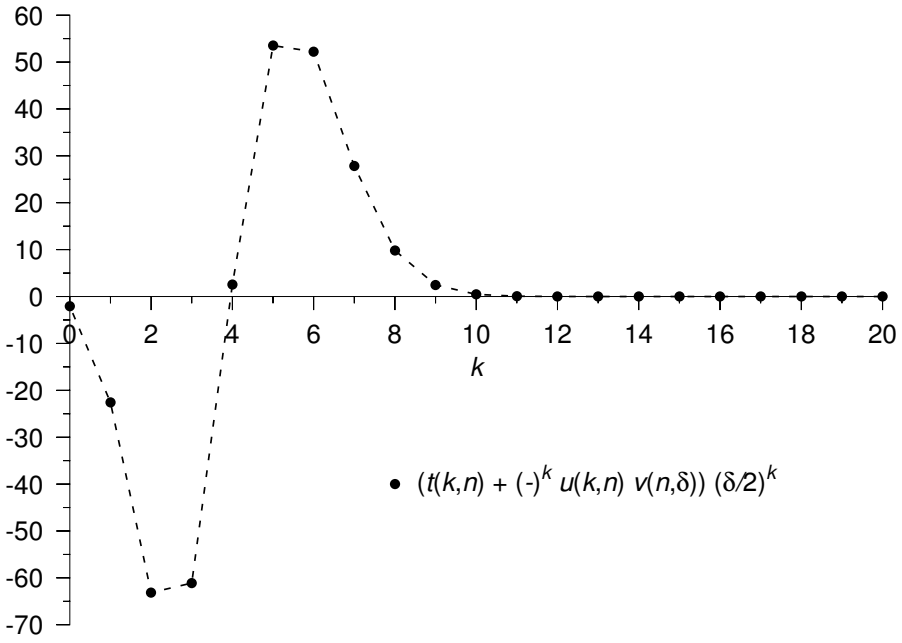


b. $n = 20, \delta = 1.0 \times 10^{-01}$

Figure 4.7: *The terms $t(k,n)$ and $(-1)^k u(k,n)v(\delta,n)$ in Equation (4.41).*



a. $n = 20, \delta = 1.0 \times 10^{-06}$



b. $n = 20, \delta = 1.0 \times 10^{-01}$

Figure 4.8: *The total term, including the convergence factor, in Equation (4.41).*

n	δ
1	3.1304×10^{-01}
2	1.0479×10^{-01}
5	2.1010×10^{-02}
10	5.7306×10^{-03}
20	1.5011×10^{-03}
50	2.4724×10^{-04}
100	6.2423×10^{-05}

Table 4.8: *Limiting values on δ as a function of n for stability in the summation of Equation (4.41).*

$$\frac{1}{2}\log\left(\frac{2+\delta}{\delta}\right) - \psi(n+1) \geq \gamma$$
$$\frac{1}{2}\log\left(\frac{2+\delta}{\delta}\right) - \sum_{m=1}^{n}\frac{1}{m} + \gamma \geq \gamma$$
$$\frac{1}{2}\log\left(1+\frac{2}{\delta}\right) \geq \sum_{m=1}^{n}\frac{1}{m}$$

what was to be proved.

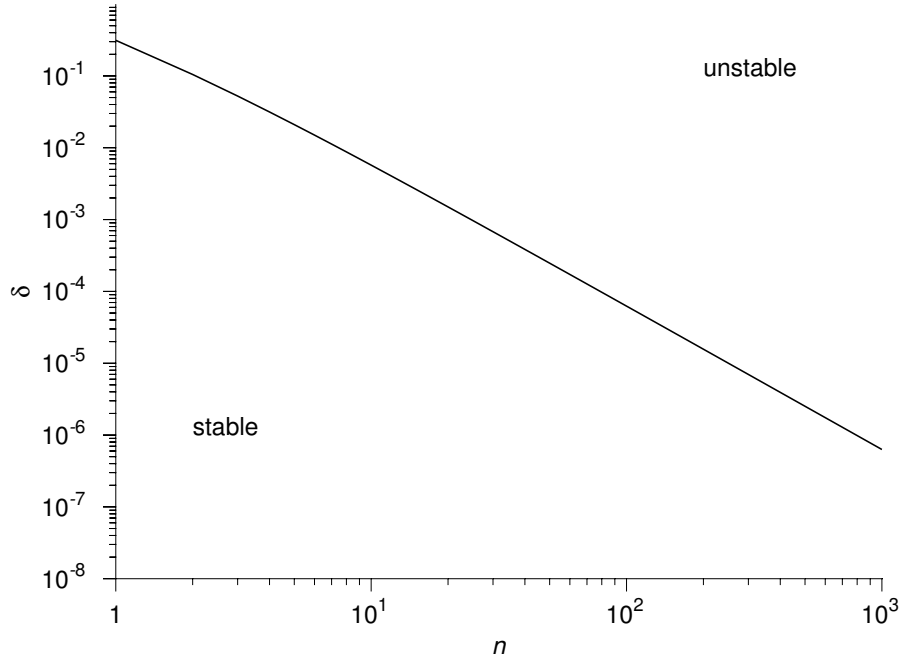
■

Equation (4.42) has no closed solution for n as a function of δ, which is the most interesting relation, since we want to know what the maximum order is that can be calculated in a stable manner for a given δ. Table 4.8 summarizes some explicit results, while Figure 4.9 shows the stable and unstable region in a graphical fashion.

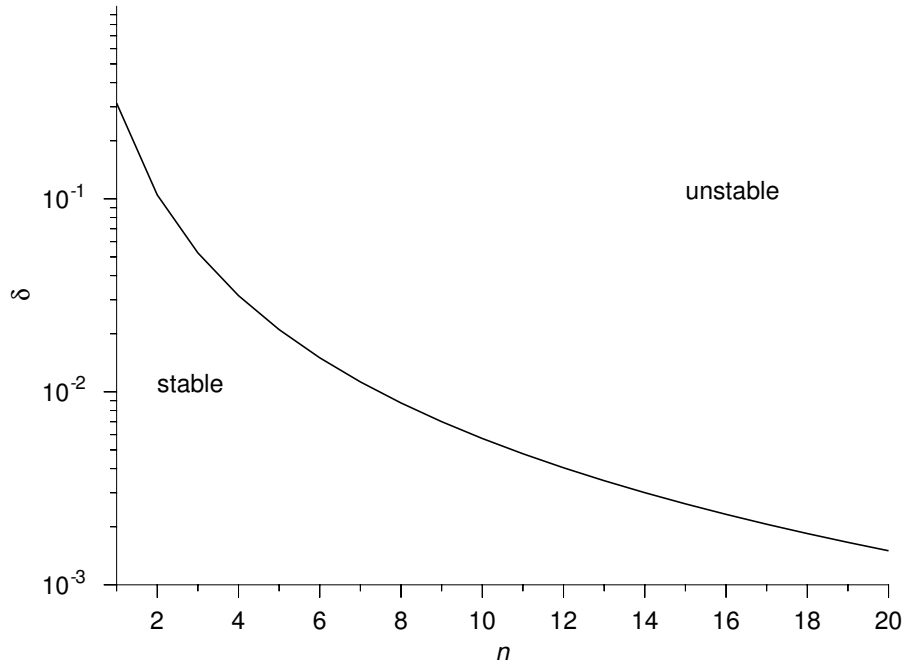
Both from Table 4.8 and Figure 4.9 we see that the values of δ remain reasonable, even for large orders. For example, for $n = 100$, we need to have δ smaller than 6.242300×10^{-05}. If δ would be larger, the classical algorithm C.3 would have no problem at all in calculating $Q_n(1+\delta)$.

A modified solving algorithm

In the previous section, we have shown how to calculate with sufficient precision, the components of the characteristic Equation (4.1), namely the



a. *Logarithmic plot,* $n = 1, \ldots, 1000$.



b. *Linear plot,* $n = 1, \ldots, 20$.

Figure 4.9: *Stability regions for Equation (4.41).*

transport polynomials $\phi_k(\nu)$ and the Legendre functions of the second kind $Q_n(\nu)$ for $\nu = 1 + \delta$. In order to solve this equation for the near-singular roots, we need to modify our solving algorithm also. Instead of solving for ν, we need to solve for δ. This poses again some problems with respect to the limitation of IEEE 754 double precision arithmetic.

Indeed, because of the limitations of IEEE 754 double precision, we cannot start the VWDB routine with, for example $\delta_{left} = 1.00 \times 10^{-200}$ and $\delta_{right} = 1.00 \times 10^{-7}$. Severe cancellation would occur in the algorithm in the inverse interpolation process. The bisection process would suffer from very slow convergence since in IEEE 754 double precision:

$$\frac{(\delta_{left} + \delta_{right})}{2} = \frac{(1.00 \times 10^{-200} + 1.00 \times 10^{-7})}{2} = 5.00 \times 10^{-08}.$$

Hence we would only halve the current value in the iteration to get to the next value. Our solution method is based on a two-step process: an order of magnitude solver and a detailed solver.

The order of magnitude solver

The idea of the first step is to get a reasonable idea on the order of magnitude of δ. We propose a logarithmic bisection method, i.e. a classical bisection where the calculation of the midpoint is not $(x_{left} + x_{right})/2$ but $\sqrt{x_{left} \cdot x_{right}}$. This improves the convergence ratio and provides us a good estimate for the VWDB routine.

The detailed solver

Once we have a good order of magnitude on δ, we can use the VWDB routine using δ as argument. Since we are already close to the solution, we don't have to worry about cancellation errors anymore.

The example revisited

Our general solution algorithm for the characteristic equation $\Lambda(\nu) = 0$ detects the occurrence of near-singular eigenvalues. The indication for this occurrence is a convergence of the first bracketed segment to $[1, 1 + macheps]$: the Van Wijngaarden-Dekker-Brent routine breaks down at this moment since the inverse quadratic interpolation process breaks down and it can't do a decent bisection due to the limits of representing the midpoint of the segment. When this phenomena occurs, we start the modified solver using the initial bracket for δ being $[10^{-200}, 10^{-10}]$.

g	δ	Relative error
0.4850	1.645221×10^{-15}	$\mathcal{O}(10^{-11})$
0.4825	2.026105×10^{-22}	$\mathcal{O}(10^{-13})$
0.4800	9.541390×10^{-44}	$\mathcal{O}(10^{-12})$

Table 4.9: *Near-singular roots for $HG(\mu|g)$, $N = 5$, $g = 0.4850, 0.4825, 0.4800$, $c = 0.80$.*

Applying the modified solving routine to our initial problem, the Henyey-Greenstein scattering kernel $HG(\mu|g)$ with $N = 5$, $g = 0.4850$, $g = 0.4825$ and $g = 0.4800$, we find the values in Table 4.9 for δ and corresponding relative errors compared to MAPLE high precision calculations. Figures 4.10 and 4.11 show respectively the order of magnitude of δ and the relative error of δ for both the logarithmic bisection procedure (the order of magnitude solver) and the Van Wijngaarden-Dekker-Brent method on δ (the detailed solver). It is clear from both plots that indeed, the logarithmic bisection reduces the order of magnitude of δ and when this is done, the VWDB routine refines the solution (hence, reduces the relative error).

Normalisation integrals

After the determination of the near-singular roots $\nu_j = 1 + \delta_j$, we also need to establish an approximation for the normalisation integrals (2.40). Let us retake their definition here for clarity:

$$\begin{aligned} M_{j\pm} = \pm c\nu_j \Bigg[& \frac{c\nu_j^2}{2\left(\nu_j^2 - 1\right)} \sum_{k=0}^{N} \sum_{l=0}^{N} \theta_{kl} \left(2k+1\right) f_k \phi_k\left(\nu_j\right) \left(2l+1\right) f_l \phi_l\left(\nu_j\right) \\ & - \frac{1}{2} \sum_{k=0}^{N} \left(2k+1\right) f_k \left[\phi_k\left(\nu_j\right)\right]^2 \\ & - \nu_j \sum_{k=1}^{N} \left(4k+1\right) f_{2k} \phi_{2k}\left(\nu_j\right) \sum_{s=1}^{k} (4s-1) \phi_{2s-1}\left(\nu_j\right) \\ & - \nu_j \sum_{k=0}^{N} \left(4k+3\right) f_{2k+1} \phi_{2k+1}\left(\nu_j\right) \sum_{s=0}^{N} \left(4s+1\right) \phi_{2s}\left(\nu_j\right) \Bigg], \end{aligned} \tag{4.45}$$

where

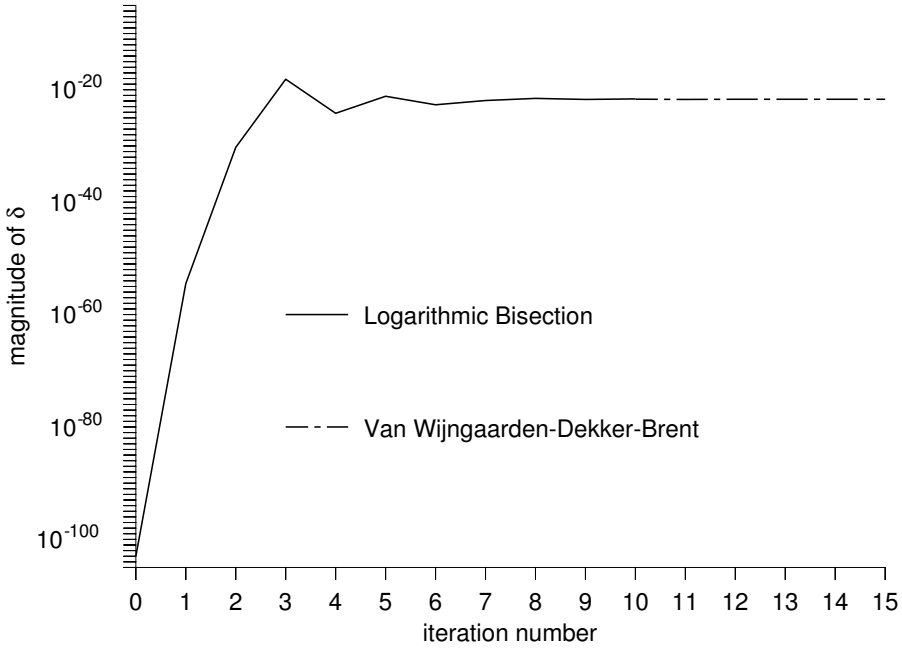


Figure 4.10: *Order of magnitude of δ during the modified solver algorithm.*

$$\theta_{kl} = \begin{cases} 1 & \text{when } (k+l) \text{ is even,} \\ \nu_j & \text{when } (k+l) \text{ is odd.} \end{cases} \tag{4.46}$$

Using for the different summation terms a shorthand S_1, S_2, S_3 and S_4, we can rewrite Equation (4.45) as

$$M_{j\pm} = \frac{\pm c\nu_j}{2}\left[\frac{c\nu_j^2}{\nu_j^2-1}S_1 - \frac{1}{2}S_2 - \nu_j S_3 - \nu_j S4\right]. \tag{4.47}$$

The terms $S1, S2, S3, S4$ can be easily calculated using $\nu_j = 1 + \delta_j$ since they only contain polynomials in ν and the relative error remains small. Replacing $\nu_j = 1 + \delta_j$ in Equation (4.47) and developing to a first order approximation in δ_j we find

$$M_{j\pm} \approx \frac{1}{4}c^2 S_1 \frac{1}{\delta_j} + \left(\frac{5}{8}c^2 S_1 - cS_4 - \frac{1}{2}cS_2 - cS_3\right) + \left(\frac{7}{16}c^2 S_1 - 2cS_4 - \frac{1}{2}cS_2 - 2cS_3\right)\delta_j. \tag{4.48}$$

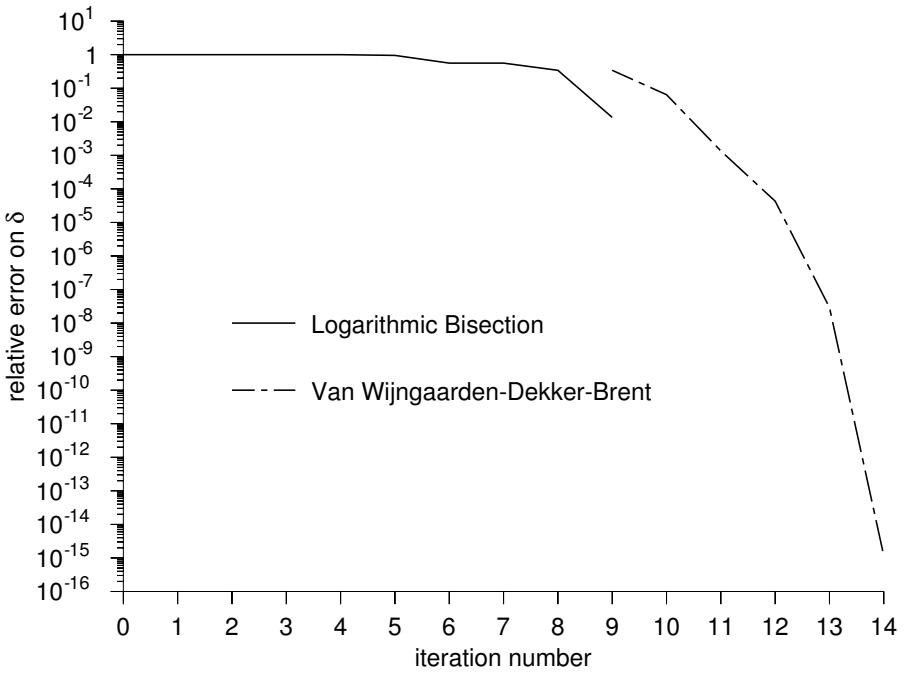


Figure 4.11: *Relative error of δ during the modified solver algorithm.*

g	M_{j+}	Relative error
0.4850	$3.075492 \times 10^{+11}$	$\mathcal{O}(10^{-11})$
0.4825	$1.085109 \times 10^{+18}$	$\mathcal{O}(10^{-13})$
0.4800	$5.472104 \times 10^{+38}$	$\mathcal{O}(10^{-12})$

Table 4.10: *Normalisation integrals for the near-singular roots of $HG(\mu|g)$, $N = 5$, $g = 0.4850, 0.4825, 0.4800$, $c = 0.80$.*

Table 4.10 shows the results of using the approximation (4.48) for the three cases in our example. The relative errors compared to MAPLE results are satisfactory.

4.5 Two challenging problems

In this section, we shall try to apply our code to the two challenging problems presented in [GS89]. The first one is a binomial scattering law of order $N = 59$. The second one is a problem where the scattering function is given

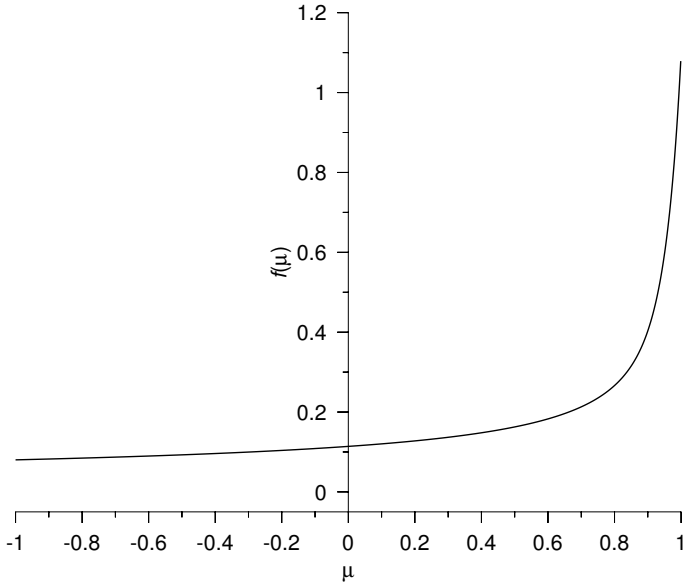


Figure 4.12: *The* 59*-th order Binomial scattering law* $Bi^{59}(\mu)$.

by the Cloud C_1 phase function posed by the Radiation Commission of the International Association of Meteorology and Atmospheric Physics.

The Binomial problem with $N = 59$

This problem is characterised by the Binomial scattering law, Equation (2.14) in Chapter 2. The order of the kernel is $N = 59$. In [GS89], three values for c are treated: $c = 0.9796585$, $c = 0.8115228$ and $c = 0.5678$. In order to give a view on the problem to handle, Figures 4.12, 4.13 and 4.14 show respectively a plot of the scattering function $Bi^{59}(\mu)$, the characteristic polynomial and the characteristic equation. The authors of [GS89] claim to have correct values up to the last digit. We verified this using MAPLE and it is indeed the case. In Table 4.11, we give our results and give relative errors between them and the MAPLE results. We see that our code gets between 11 and 12 significant digits, which is, given the fact we use IEEE double precision, quite good.

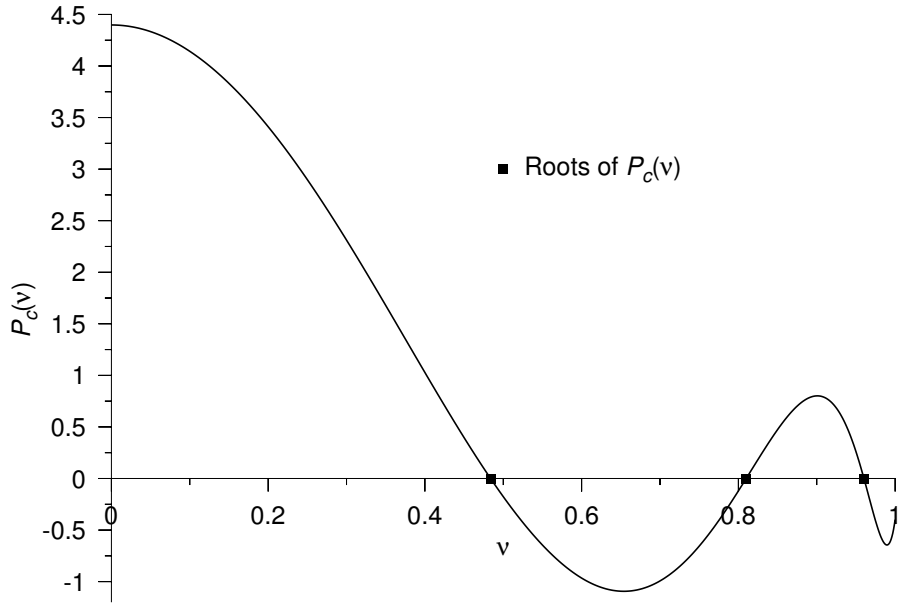


Figure 4.13: *The characteristic polynomial for the Bi^{59}, $c = 0.8783585$ problem.*

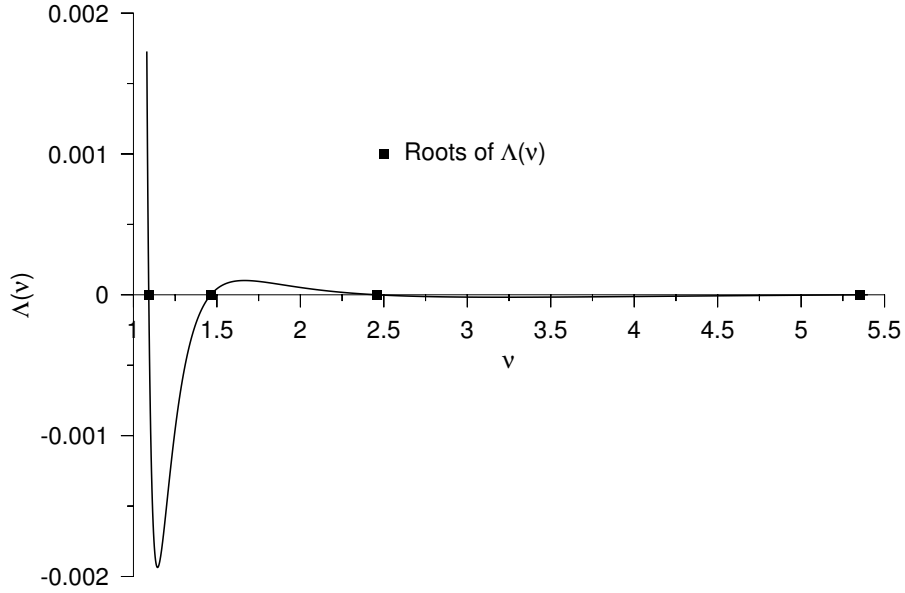


Figure 4.14: *The characteristic equation for the Bi^{59}, $c = 0.8783585$ problem.*

$c = 0.8783585$	
ν_j	Rel. Err.
1.09174545945395	$\mathcal{O}(10^{-11})$
1.46329528265439	$\mathcal{O}(10^{-11})$
2.46093515689889	$\mathcal{O}(10^{-11})$
5.35109107175738	$\mathcal{O}(10^{-12})$
$c = 0.8115228$	
ν_j	Rel. Err.
1.05554731130312	$\mathcal{O}(10^{-11})$
1.34029970030093	$\mathcal{O}(10^{-11})$
2.05024247394391	$\mathcal{O}(10^{-11})$
3.78316553362341	$\mathcal{O}(10^{-12})$
$c = 0.5678000$	
ν_j	Rel. Err.
1.00000000000115[†]	$\mathcal{O}(10^{-16})$
1.09160665761278	$\mathcal{O}(10^{-11})$
1.36796962642348	$\mathcal{O}(10^{-12})$
1.91200150082061	$\mathcal{O}(10^{-12})$

[†] In [GS89], this near-singular root is given as indicated. However, our code switches to the near-singular root solver and calculates $\delta = +1.1469713683188269 \times 10^{-12}$, which is correct up to 5 digits (compared to MAPLE results).

Table 4.11: *Positive discrete eigenvalues for the Bi^{59} problem.*

The Cloud C_1 problem

The Cloud C_1 problem is one of five test problems posed by the Radiation Commission of the International Association of Meteorology and Atmospheric Physics. An $N = 299$ order Legendre expansion is given in [GS85]. We solve the problem for $c = 0.9$, as in [GS89].

Figures 4.15, 4.16 and 4.17 show respectively a plot of the scattering function for the Cloud C_1 problem, the characteristic polynomial and the

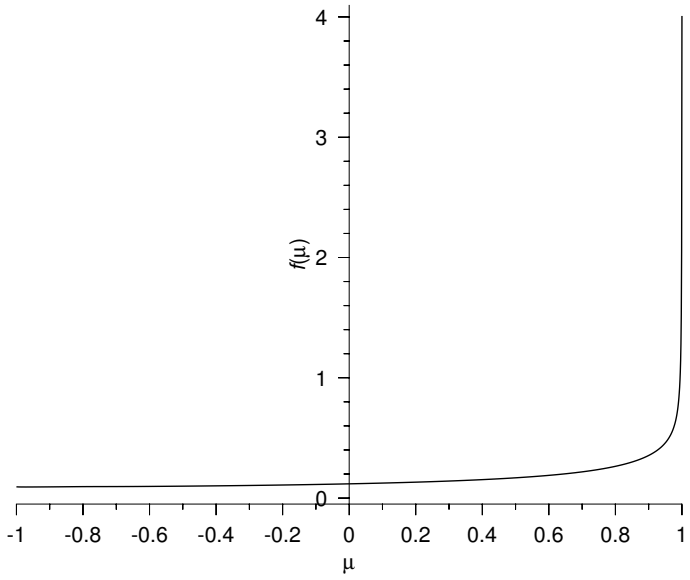


Figure 4.15: *The scattering function for the Cloud C_1 problem.*

characteristic equation. Note that for the characteristic equation, we used a logarithmic y-axis where we plot the absolute value of $\Lambda(\nu)$. The peaks indicated the positions of the roots. Exact locations are shown on the ν-axis. Again, the authors of [GS89] claim 16 correct digits. We verified this using MAPLE. In Table 4.12, we show our results and the ones presented in the above cited paper, together with a relative error between them. As one can see, our code performs as well as the one from Garcia and Siewert.

Conclusions

It is clear that high-order problems are suffering from numerical trouble in the calculation of the discrete eigenvalues. The main reason is the stable calculation of the orthogonal polynomials $\phi_k(\nu)$ for high values of k. As shown in Chapter 3, when ν is a root of the characteristic equation, the $\phi_k(\nu)$ form a decreasing series for k increasing. When ν is not a root of $\Lambda(\nu)$, the $\phi_k(\nu)$ form an increasing series for k increasing. The problem pops up for values of ν close to a root of $\Lambda(\nu)$. During the search for a root, one comes closer and closer to the real root and hence at a certain moment the series becomes decreasing and the backward approach should be used. However, for increasing k, the calculated root needs to be very close to the real root in order to keep the series decreasing. The slightest deviation is enough to

Our results	Results in [GS89]	Rel. Err
1.00038997910029e+00	1.00038997910029e+00	$\mathcal{O}(10^{-16})$
1.00293299482387e+00	1.00293299482387e+00	$\mathcal{O}(10^{-16})$
1.00748496308909e+00	1.00748496308909e+00	$\mathcal{O}(10^{-16})$
1.01387932280389e+00	1.01387932280389e+00	$\mathcal{O}(10^{-16})$
1.02205217731183e+00	1.02205217731182e+00	$\mathcal{O}(10^{-16})$
1.03198875034556e+00	1.03198875034556e+00	$\mathcal{O}(10^{-16})$
1.04370927267972e+00	1.04370927267972e+00	$\mathcal{O}(10^{-16})$
1.05726113819117e+00	1.05726113819117e+00	$\mathcal{O}(10^{-16})$
1.07271728978241e+00	1.07271728978241e+00	$\mathcal{O}(10^{-16})$
1.09017893701578e+00	1.09017893701578e+00	$\mathcal{O}(10^{-16})$
1.10977520869889e+00	1.10977520869889e+00	$\mathcal{O}(10^{-16})$
1.13167071524314e+00	1.13167071524314e+00	$\mathcal{O}(10^{-16})$
1.15607420331418e+00	1.15607420331418e+00	$\mathcal{O}(10^{-16})$
1.18325281654462e+00	1.18325281654462e+00	$\mathcal{O}(10^{-16})$
1.21355133540278e+00	1.21355133540278e+00	$\mathcal{O}(10^{-16})$
1.24743006482968e+00	1.24743006482968e+00	$\mathcal{O}(10^{-16})$
1.28551647469471e+00	1.28551647469471e+00	$\mathcal{O}(10^{-16})$
1.32872107056041e+00	1.32872107056041e+00	$\mathcal{O}(10^{-16})$
1.37843997083342e+00	1.37843997083342e+00	$\mathcal{O}(10^{-16})$
1.43703887813286e+00	1.43703887813286e+00	$\mathcal{O}(10^{-16})$
1.50930231753875e+00	1.50930231753875e+00	$\mathcal{O}(10^{-16})$
1.61065759135053e+00	1.61065759135053e+00	$\mathcal{O}(10^{-16})$
1.92895463219683e+00	1.92895463219683e+00	$\mathcal{O}(10^{-16})$
4.28215602543729e+00	4.28215602543729e+00	$\mathcal{O}(10^{-16})$

Table 4.12: *Positive discrete eigenvalues for the Cloud C_1 problem.*

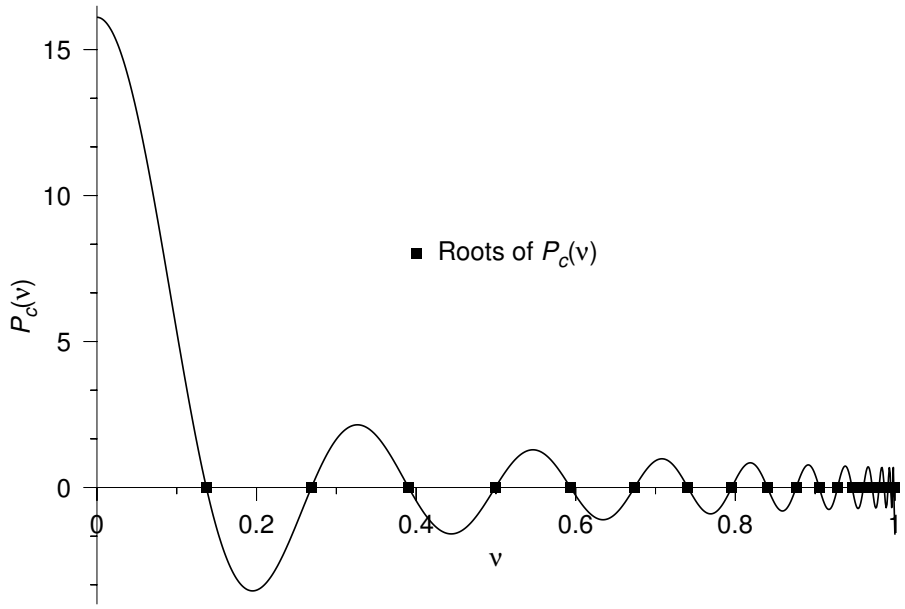


Figure 4.16: *The characteristic polynomial for the Cloud C_1 problem for $c = 0.9$.*

have the series increasing again from a certain order k onward. It is this arbitrarily located cross-over point that makes the evaluation of the $\phi_k(\nu)$ nearly impossible for high orders k. On the other hand, the effect of the above issue doesn't detain our code to achieve very good results for the Cloud C_1 problem. More research is needed to either overcome the problem and correctly evaluate the $\phi_k(\nu)$ or either to try to classify the problems that are hindered by the above numerical trouble.

The approach by Garcia and Siewert in [GS89] is to use a related Sturm sequence and avoid the explicit calculation of the characteristic equation. It is definitely worthwhile to implement this method in our code and see if it improves the code.

4.6 The discrete part of the infinite medium Green's function

Later in this work, when deriving the Boundary Sources Method (Chapter 6), we shall need both angular and angular/spatial moments of the infinite medium Green's function. The angular moments are defined as

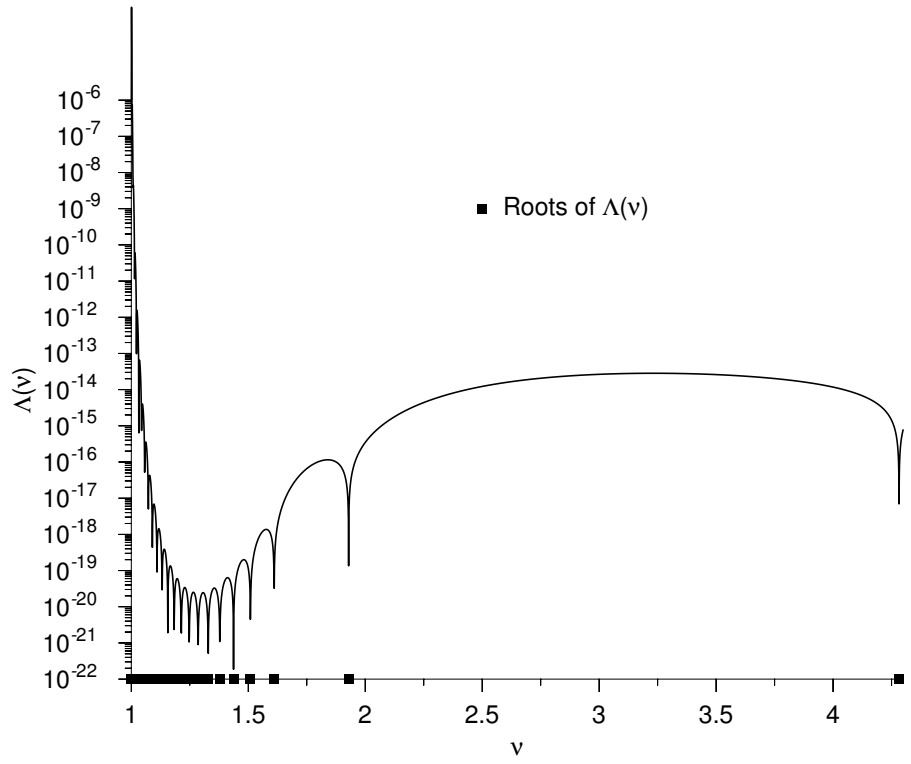


Figure 4.17: *The characteristic equation for the Cloud C_1 problem for $c = 0.9$.*

$$[G]_{p(\mu),q(\mu_0)}(x_0|x) = \int_{-1}^{+1}\int_{-1}^{+1} p(\mu)G(x_0,\mu_0|x,\mu)q(\mu_0)d\mu d\mu_0 \qquad (4.49)$$

with $p(\mu)$ and $q(\mu_0)$ functions defined on $[-1,+1]$. We shall use both the Legendre polynomials and the Yvon polynomials for $p(\mu)$ and $q(\mu_0)$. Beauwens has proved [Bea68] that these moments can be expressed using the $\mathcal{K}$-transforms of $p(\mu)$ and $q(\mu_0)$.

In this section we shall give expressions for the moments of the discrete part of the Green's function. From Chapter 2, Equation (2.56), we have for the discrete part of the infinite medium Green's function

$$G_d(x_0,\mu_0|x,\mu) = \begin{cases} \displaystyle\sum_{j=1}^{M} \frac{\phi(\nu_j,\mu_0)\phi(\nu_j,\mu)}{M_{j+}} e^{-\Sigma_t(x-x_0)/\nu_j} & x \geq x_0 \quad (4.50a) \\ \displaystyle\sum_{j=1}^{M} \frac{\phi(-\nu_j,\mu_0)\phi(-\nu_j,\mu)}{M_{j-}} e^{+\Sigma_t(x-x_0)/\nu_j} & x \leq x_0\,. \quad (4.50b) \end{cases}$$

The full-range Legendre angular moments are then given by

$$[G_d]_{m,n}(x_0|x) = \begin{cases} \displaystyle\sum_{j=1}^{M} \frac{\phi_m(\nu_j)\phi_n(\nu_j)}{M_{j+}} e^{-\Sigma_t(x-x_0)/\nu_j} & x \geq x_0 \quad (4.51a) \\ \displaystyle\sum_{j=1}^{M} \frac{\phi_m(-\nu_j)\phi_n(-\nu_j)}{M_{j-}} e^{+\Sigma_t(x-x_0)/\nu_j} & x \leq x_0\,. \quad (4.51b) \end{cases}$$

Property 4.1. *The following symmetry relation holds for the full-range Legendre angular moments of the Green's function: for $|x - x_0|$ fixed, we have*

$$\underbrace{[G_d]_{m,n}(x_0|x)}_{x\leq x_0} = (-)^{m+n}\underbrace{[G_d]_{m,n}(x_0|x)}_{x\geq x_0}.$$

Proof. Based on the symmetry properties of the different components of $[G_d]_{m,n}(x_0|x)$,

$$M_{j-} = -M_{j+},\ \phi_k(-\nu_j) = (-)^k\phi(+\nu_j)$$

the above relation is easily proved. ■

The half-range Yvon angular moments are given by

$$[G_d]_{m^\pm,n}(x_0|x) = \begin{cases} \sum_{j=1}^{M} \frac{\phi_m^\pm(\nu_j)\phi_n(\nu_j)}{M_{j+}} e^{-\Sigma_t(x-x_0)/\nu_j} & x \geq x_0 \quad (4.52a) \\ \sum_{j=1}^{M} \frac{\phi_m^\pm(-\nu_j)\phi_n(-\nu_j)}{M_{j-}} e^{+\Sigma_t(x-x_0)/\nu_j} & x \leq x_0\,. \quad (4.52b) \end{cases}$$

We also need spatial moments of the angular moments defined above. These spatial moments are required to represent the spatial behaviour in a cell when applying the Boundary Sources Method on that cell. We also use Legendre moments as the spatial moments, hence we use the Legendre basis for the representation of the spatial component in the slab, which is definitely an improvement over the classical power basis. We define the spatial moments as

$$[G_d]^k_{m,n}(x) = \int_a^b \tilde{P}_k(x_0)\,[G_d]_{m,n}(x_0|x)dx_0 \tag{4.53}$$

where $\tilde{P}_k(x_0)$ denote the transformed Legendre polynomials on the interval $[a,b]$, i.e. the set of polynomials orthogonal on the interval $[a,b]$ with respect to the weight function $w(x_0) \equiv 1$. When evaluating the above integral, we must take care of a possible switch of the relative positions of x and x_0, causing sign changes in $\left[G_d^\pm\right]_{m,n}(x_0|x)$ so switching the evaluation from Equation (4.51a) to Equation (4.51b) and vice versa. We can distinguish between the following cases:

1) $x \notin [a,b]$
   a) $x \leq a$, use Equation (4.51a) and multiply by $(-)^{m+n}$
   b) $x \geq b$, use Equation (4.51a)
2) $x \in [a,b]$, use $\int_a^x [Eq.(4.51a)] + (-)^{m+n} \int_x^b [Eq.(4.51a)]$

Since the spatial variable only appears in the exponential term, we can separate this term during the spatial integration. What we need in our code is hence a routine that returns the value of the following integral:

$$EA_k(x) = \int_c^d \tilde{P}_k(x_0) e^{-\alpha|x-x_0|} \tag{4.54}$$

where $\alpha = \Sigma_t/\nu_j$ is a positive real number. We know that ν_j is larger than unity, so α cannot be larger than Σ_t, the total cross-section. In real-life applications, this cross-section doesn't take on extreme values, so α has no extreme behaviour.

Since there is no closed expression for such an integral, we use a numerical quadrature routine to evaluate $EA_k(x)$. We shall discuss the evaluation of this integral more in detail in Chapter 5 because in the continuum case, ν is bounded by the interval $[0, +1]$ (if $x \geq x_0$) and α can become very large. We want to use the same (robust) routine for both the discrete and continuum calculations so we introduce our calculation scheme later on.

Effect of near-singular roots on the angular moments

It is interesting to see what the effect is of the presence of a near-singular eigenvalue on the convergence behaviour of the angular moments of the discrete part of the infinite medium Green's function. Indeed, the angular moments can be used to construct a double Legendre expansion for the discrete part of the Green's function:

$$G_d(x_0, \mu_0|x, \mu) \approx \sum_{m_1=0}^{M_1} \frac{2m_1+1}{2} \left(\sum_{m_2=0}^{M_2} \frac{2m_2+1}{2} [G_d]_{m,n}(x_0|x) P_{m_2}(\mu) \right) P_{m_1}(\mu_0) \tag{4.55}$$

where the approximation orders are M_1 and M_2. Since the discrete part of the Green's function is continuous on the approximation interval $[-1, +1] \times [-1, +1]$, we know that this approximation must converge for $M_1, M_2 \to \infty$.

Let us see what happens in case of a near-singular eigenvalue. We retake the example we used before: a 5-th order approximation of the Henyey-Greenstein with parameter $g = 0.485$ and $c = 0.8$. For this problem, we have a normal root at ≈ 1.8249 and a near-singular root $1 + \delta$ with $\delta \approx 1.645221 \times 10^{-15}$. In Figures 4.18, 4.19 and 4.20 we show the magnitude, i.e. $|[G_d]_{m,n}(x_0|x)|$, of the angular moments for $m = 0$, $m = 10$ and $m = 20$ while we took $x = x_0 = 0$. Note the different ranges in the y-axis. We clearly see that the normal root term converges smoothly but the near-singular one almost stays constant. There is, however, a very slow convergence. As a consequence, the total discrete component converges quickly up to where the normal and near-singular part cross each other.

4.7 Conclusions

We have developed a robust method to calculate all discrete eigenvalues in the Mika/Case eigenexpansion. These eigenvalues are the roots of the

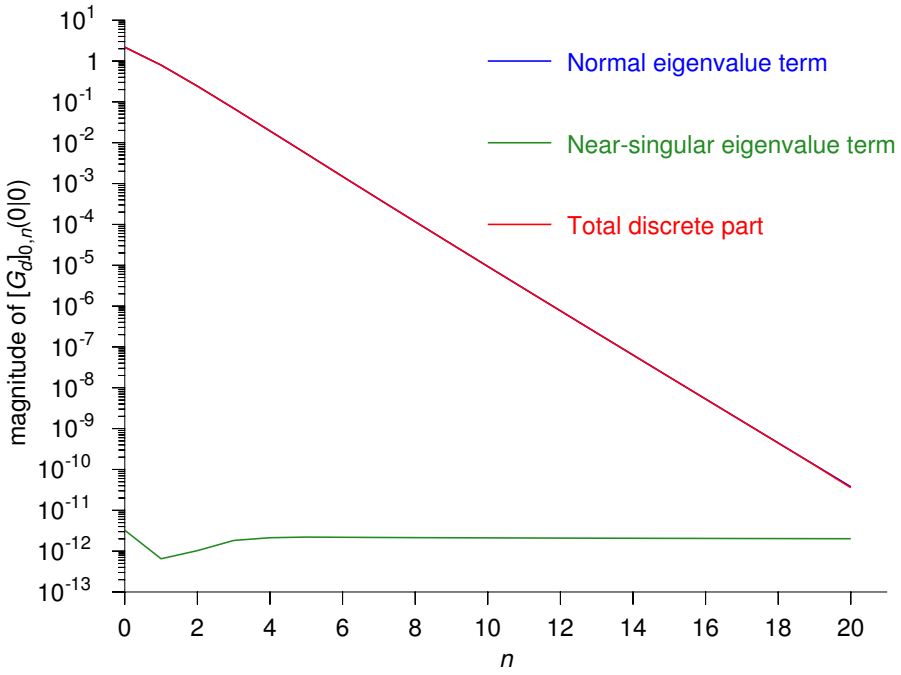


Figure 4.18: *Convergence of the angular moments in the presence of a near-singular eigenvalue.*

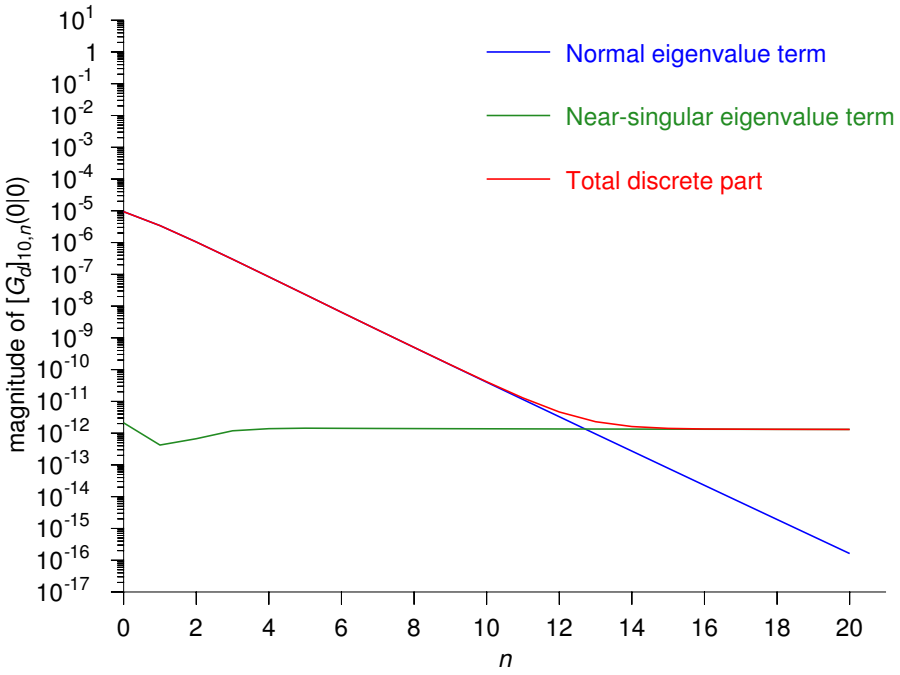


Figure 4.19: *Convergence of the angular moments in the presence of a near-singular eigenvalue.*

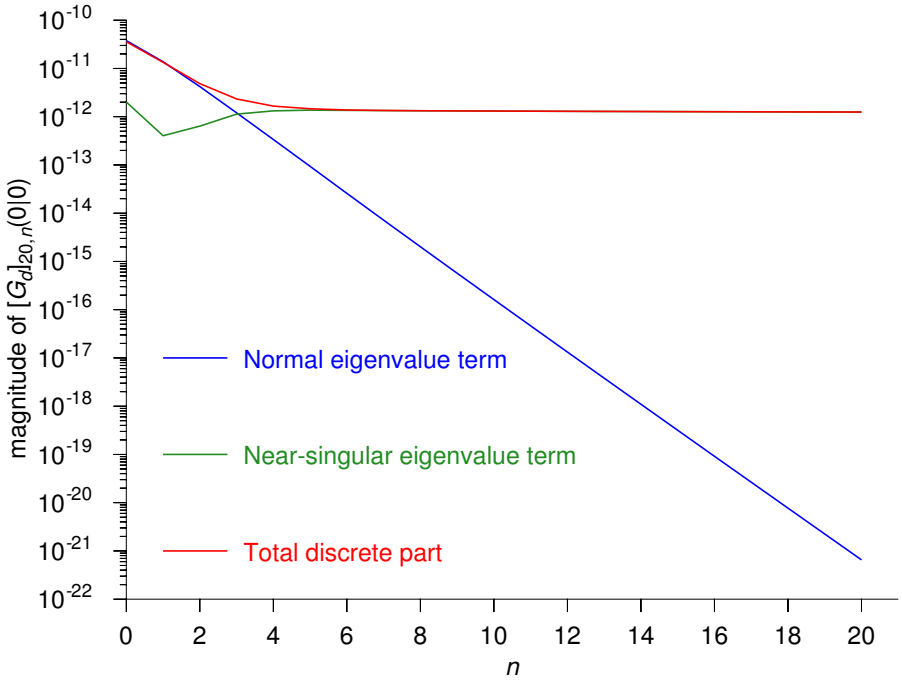


Figure 4.20: *Convergence of the angular moments in the presence of a near-singular eigenvalue.*

characteristic equation $\Lambda(\nu) = 0$. The method is a two-step process: we first calculate the number of positive discrete eigenvalues followed by the real solution process. Knowing in advance the number of roots one is looking for, the solution process can be terminated when all roots are found.

As conjectured in Chapter 2, the number of positive discrete eigenvalues is equal to the number of roots of the polynomial $\nu P_c(\nu)$ on the real line segment $[0, +1]$. Two methods for the calculation of these roots have been implemented: one based on the eigenvalues of a corresponding companion matrix and one based on the Newton-Raphson iteration with Maehly's implicit deflation. The first method needs the polynomial to be built. Classical polynomial interpolation using a Vandermonde solver proved to be unstable for even small degrees of anisotropy. The method we have chosen is based on the Chebyshev expansion of the polynomial $P_c(\nu)$ and the "degree-doubling" trick. The eigenvalues of the corresponding companion matrix that lie on the unit circle have the same real part than the roots of $P_c(\nu)$ on the real line segment $[0, +1]$. This method has proved to be stable up to degrees of anisotropy of 50.

For higher degrees, we resort to the Newton-Raphson iteration with the implicit deflation technique of Maehly. We use a so-called "box" method to avoid convergence to roots far away from $[0, +1]$ or divergence. The method has computationally been proved stable and robust, but of course, we loose the exactness of the previous procedure where we can guarantee that we found all roots of $P_c(\nu)$. With the latter method, we cannot give this guarantee.

The second step is the localising of the roots of $\Lambda(\nu)$, the discrete eigenvalues themselves. This also is a two-step process: the first stage is the bracketing procedure. The outcome of this procedure is a set of intervals each containing exactly one root of $\Lambda(\nu)$. We then apply the derivative-free Van Wijngaarden-Dekker-Brent non-linear equation solving routine on each interval. This method is guaranteed to converge to the root in each interval.

During our computational experiments, we came across what we called the near-singular eigenvalues: discrete eigenvalues that lie extremely close to $\nu = 1$, so close even that we rewrote them as $\nu_j = 1 + \delta_j$. We have shown examples where δ_j ranges from 10^{-15} to 10^{-44}. These discrete eigenvalues cannot be computed using the above method: the Van Wijngaarden-Dekker-Brent routine suffers heavily from cancellation and loss of accurate digits. We therefore developed a solving routine that solves not for ν_j but for δ_j. This required accurate evaluation routines for $\Lambda(\nu)$ based on δ. These routines were developed and a requirement for stability was derived.

Both for the normal eigenvalues and the near-singular ones, the normalisation integrals were calculated. They are needed in the expression for the infinite medium Green's function. This function is the building block of the Boundary Sources Method. For this BSM, we also need angular and angular/spatial moments of this transport kernel.

CHAPTER 5

The continuum eigendistributions

SUMMARY– The continuum eigendistributions complete the full orthogonal and complete decomposition by Mika and Case. They are distributions and hence can only be characterised by angular and angular/spatial moments. We use numerical quadrature routines to accurately evaluate these moments.

5.1 Introduction

The continuum eigendistributions form the second component in the orthogonal decomposition. They are given by

$$\phi_\nu(\mu) = \frac{c\nu}{2} P \frac{1}{\nu - \mu} \sum_{l=0}^{N} (2l+1)\, f_l\, \phi_l(\nu)\, P_l(\nu) + \lambda(\nu)\delta(\nu - \mu) \tag{5.1}$$

where the parameter $\nu \in [-1, +1]$ and the normalisation function $\lambda(\nu)$ is given by

$$\lambda(\nu) = 1 - \frac{c\nu}{2} \sum_{l=0}^{N} (2l+1) f_l \phi_l(\nu) P Q_l(\nu) = \frac{1}{2} \left[\Lambda^+(\nu) + \Lambda^-(\nu) \right]. \tag{5.2}$$

We are interested in the angular moments of the eigendistributions since they form a part of the infinite medium Green's function.

5.2 The continuum part of the infinite medium Green's function

As in the case of the discrete part of the infinite medium Green's function, we are interested in the full-range and half-range Legendre angular moments keeping in mind their use in the Boundary Sources Method. The continuum part of the infinite medium Green's function is given by (see Equation (2.56))

$$G_c(x_0,\mu_0|x,\mu) = \begin{cases} \displaystyle\int_0^{+1} \frac{\phi(\nu,\mu_0)\phi(\nu,\mu)}{M(\nu)} e^{-\Sigma_t(x-x_0)/\nu} d\nu & x \geq x_0 \quad (5.3a) \\ \displaystyle -\int_{-1}^{0} \frac{\phi(\nu,\mu_0)\phi(\nu,\mu)}{M(\nu)} e^{-\Sigma_t(x-x_0)/\nu} d\nu & x \leq x_0. \quad (5.3b) \end{cases}$$

The full-range Legendre moments are then given by

$$[G_c]_{m,n}(x_0|x) = \begin{cases} \displaystyle\int_0^{+1} \frac{\phi_m(\nu)\phi_n(\nu)}{M(\nu)} e^{-\Sigma_t(x-x_0)/\nu} d\nu & x \geq x_0 \quad (5.4a) \\ \displaystyle -\int_{-1}^{0} \frac{\phi_m(\nu)\phi_n(\nu)}{M(\nu)} e^{-\Sigma_t(x-x_0)/\nu} d\nu & x \leq x_0. \quad (5.4b) \end{cases}$$

The half-range moments can easily be defined accordingly. In a previous work [VdV95], Vandevelde treated the isotropic case for this problem. Since in the isotropic case only two parameters are present (c and the spatial parameter $\Sigma_t(x-x_0)$), he was able to perform a two-dimensional minimax approximation by modifying the classical Remez algorithm. As a benchmark to validate his two-dimensional approximation, he used classical Gauss-Legendre quadrature and an adaptive Simpson rule [DR84].

Since the evaluation of the full-range Legendre moments of the infinite medium Green's function are the basic operations in the Boundary Sources Method this evaluation should be as fast and accurate as possible. The first step towards a method with those characteristics is an analysis of the integrand in question and a proposal for an "equivalent" function to be used in the numerical tests. The second step is the choice of an appropriate quadrature routine.

5.3 Analysis of the integrand

Looking at the integrand for $x > x_0$ in Equation (5.4), we can distinguish the following behaviour:

1. the numerator contains
   (a) a product of two highly oscillating polynomials $\phi_m(\nu)$ and $\phi_n(\nu)$ which is odd or even depending on $(m+n)$,
   (b) an exponential function in $(-1/\nu)$ (since the remainder in the exponential function is always positive) that vanishes for $x = x_0$ or $\Sigma_t \equiv 0$;
2. the denominator contains the product of
   (a) the polynomial ν and
   (b) a function that doesn't have any roots in $[0, +1]$ (we required non-degenerate eigenvalues [See Mik61]) but has a logarithmic term $\log(1-\nu)$ in $P\Lambda(\nu)$ (see Equations (2.44)-(2.45)).

From the above it is clear that the integrand does not show singular behaviour except at the two boundary points $\nu = 0$ and $\nu = +1$. For numerical quadrature, not only singularities of the integrand itself are important, but also the singularities of the consecutive derivatives of the integrand, with the first derivative having the largest influence. This comes from the fact that most error expansions for quadrature rules indeed have terms containing derivatives of the integrand in their series development [DR84].

Integrand behaviour at $\nu = 0$

For $\nu = 0$, three factors are of importance: the value of Σ_t, the value of $(x - x_0)$ and the parity of $(m+n)$. If $(m+n)$ is odd, the polynomial in the numerator is also odd and hence the factor ν in the denominator is cancelled out. However, when $(m+n)$ is even, the polynomials do not cancel each other out and the singularity can only disappear thanks to the exponential function present in the numerator. Indeed, when ν approaches zero, the exponential function $\exp(-1/\nu)$ also approaches zero but faster and hence the singularity disappears. Of course, the exponential term only exists in the non-vacuum case ($\Sigma_t \neq 0$) and for $x \neq x_0$.

Integrand behaviour at $\nu = +1$

At the other end of the integration interval, the only actor having any influence on the singular character of the integrand is the function $\log(1-\nu)$ in the denominator. For $\nu \to 1$, this function goes to $-\infty$ and hence the integrand goes to zero. However, the first derivative of the integrand goes

to $+\infty$ for $\nu \to 1$, hence toughening the job for the numerical quadrature routine.

5.4 Appropriate quadrature rules

Given the above characteristics, three types of quadrature routines are advised: adaptive Gaussian quadrature with a "singularity" treatment, so-called "double-exponential" formulae and "sinc" formulae. The latter inherently can handle boundary singularities, the former in some cases, but this can be improved using variable transformations.

Adaptive Gaussian quadrature

The most advanced implementation of Gaussian quadratures of all kinds is the FORTRAN 77 code QUADPACK written by Piessens et al. [PdDKÜK83] which is freely available from Netlib (`http://netlib.org`). The routine we chose from the collection is DQAGSE, which is an adaptive Gauss-Kronrod quadrature routine in double precision using Wynn's ϵ-algorithm [Wyn66; Wen89; Bre78] to improve convergence in case of singularities at the boundaries.

To overcome the difficulty of singularities at the boundaries of the integration interval, variable transforms can be applied. The basic idea behind these transforms is to choose the transform so that the end-point singularities are cancelled out. For more on the theory behind variable transforms and numerical quadrature, we refer to [DR84].

The two most common variable transformations are the TANH-rule [SS64] and the IMT-rule [IMT87; MI82]. Typically, they are used in conjunction with the trapezoid rule since in that case a lot of contributions in the error expansion cancel each other out. However, we can expect an improvement even for the DQAGSE routine if we smoothen the integrand and its derivatives at the end-point singularities.

The TANH*-transform*

The TANH-transform is attributed to Sag and Szekeres [SS64]. The transformation function is given by

$$\varphi(t) = \frac{1}{2}\left(1 + \tanh\left(\frac{1}{1-t} - \frac{1}{t}\right)\right) \tag{5.5}$$

and its derivative by

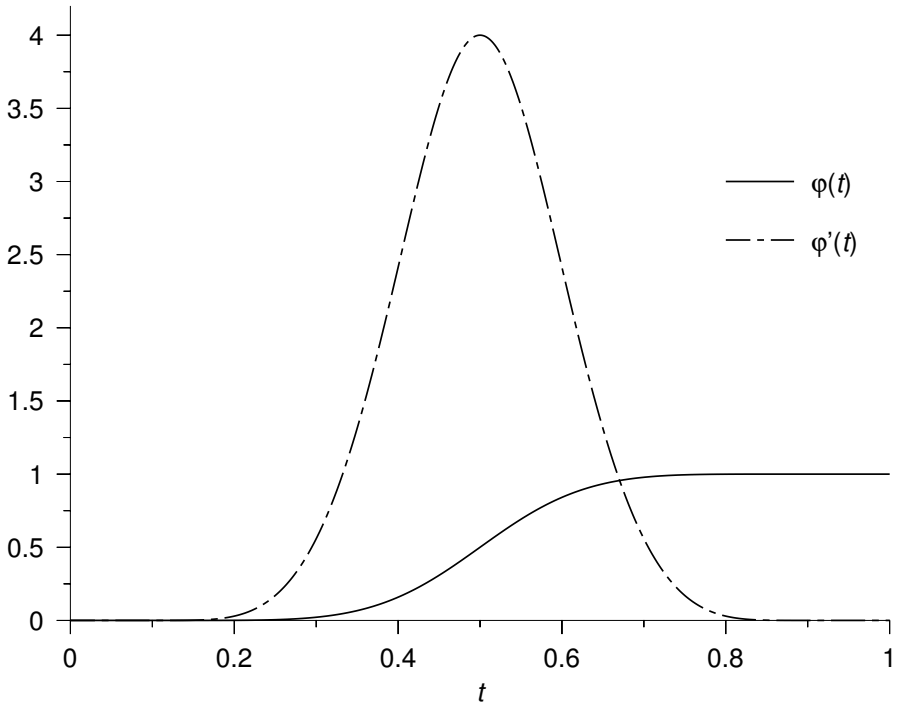


Figure 5.1: *The* TANH*-transform.*

$$\varphi'(t) = \frac{1}{2}\left[(1-t)^{-2} + t^{-2}\right]\left[1 - \tanh^2\left(\frac{1}{1-t} - \frac{1}{t}\right)\right]. \tag{5.6}$$

Figure 5.1 depicts both $\varphi(t)$ and $\varphi'(t)$. Applying the transformation $\nu = \varphi(t)$ in the integral one has

$$\int_0^1 f(\nu)d\nu = \int_0^1 f\left(\varphi(t)\right)\varphi'(t)dt. \tag{5.7}$$

From the transformation and Figure 5.1, we see that

$$\varphi(0) = 0, \varphi(1) = 1, \varphi'(t) > 0 \text{ for } t \in (0,1).$$

We also have that

$$\lim_{t\to 0}\frac{d^n}{dt^n}\varphi(t) = 0, \lim_{t\to 1}\frac{d^n}{dt^n}\varphi(t) = 0, n \in \mathbb{N}.$$

It is clear that the transformed integrand behaves smoothly at the two end-points, thanks to the multiplication with $\varphi'(t)$. It is of utmost importance to take special care in the evaluation of the integrand close to the end-points to avoid round-off.

The IMT*-transform*

Let us define the the auxiliary function $\Phi(t)$ as

$$\Phi(t) = \exp\left(-\frac{c}{1-t^2}\right), \tag{5.8}$$

where c is a positive real number (typically $c = 4$). There have been studies on the influence of this parameter (and other parameterisations) on the quality of the transformation [MI82]. The IMT-transform [IMT87] is then defined by

$$\varphi(t) = \frac{2}{\gamma}\int_{-1}^{t}\Phi(u)du \tag{5.9}$$

where γ is a normalisation factor

$$\gamma = \int_{-1}^{+1}\Phi(u)du. \tag{5.10}$$

The transformed integral is again

$$\int_0^1 f(v)dv = \int_0^1 f\left(\varphi(t)\right)\varphi'(t)dt \tag{5.11}$$

and has the interesting property, as can be seen in Figure 5.2, that at the end-points of the integration interval the integrand and all its derivatives vanish, hence the end-point singularities have been eliminated.

The major drawback of this transformation is the cost of evaluation of Equation (5.9). Indeed, we would need a quadrature rule to evaluate the transformation and this would increase the cost rather extensively. Luckily, a Chebyshev approximation for $\varphi(t)$ has been developed [DP76] and this limits the overhead to an acceptable level.

Double exponential quadrature

The double exponential method transforms the finite interval $[0,1]$ to an infinite interval $[-\infty,+\infty]$ by the variable transform

$$\varphi(t) = \frac{1}{2}\tanh\left(\frac{\pi}{2}\sinh t\right) + \frac{1}{2} \tag{5.12}$$

and then applies the uniformly divided trapezoidal formula [TM74] on the infinite integral. The name double-exponential comes from the fact that the transformed integrand shows "double-exponential" behaviour at the

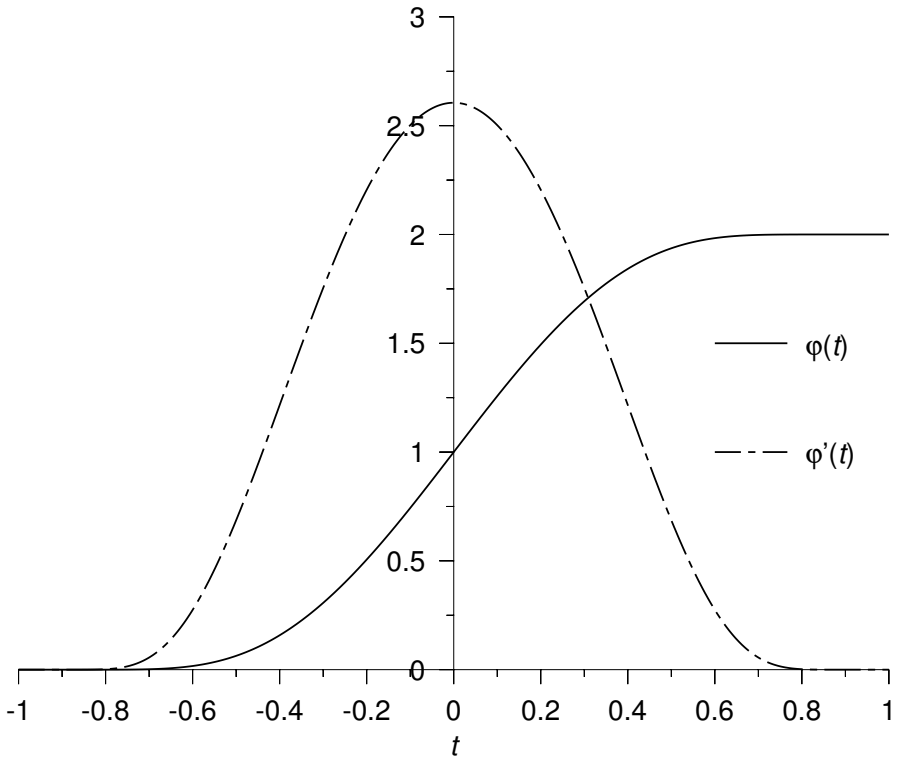


Figure 5.2: *The* IMT*-transform.*

end-points. By the application of the trapezoidal rule, the integral is transformed into an infinite sum composed of weighted function evaluations. The weights can be computed beforehand, depending on the accuracy needed, hence the preparatory work has to be done only once if one sticks to the same accuracy. We use the implementation INTDE by T. Ooura [OM91], which is freely available.

Sinc quadrature

The whole range of SINC methods [Ste81] are based on Whittaker's cardinal function $C(f,h)$ corresponding to a function f defined on the real line

$$C(f,h) = \sum_{k=-\infty}^{+\infty} f(kh)S(k,h) \tag{5.13}$$

where $h > 0$ is the step size and

$$S(k,h)(x) = \frac{\sin\left[(\pi/h)\,(x-kh)\right]}{(\pi/h)\,(x-kh)}. \tag{5.14}$$

As one can see, the expression in Equation (5.13) is closely related to the infinite range trapezoidal rule for numerical quadrature. The SINC-quadrature can be seen to be optimal in a certain sense [Sik82; SS84]. We use the implementation INTHP described in [SS84].

Comparison of the methods

To compare the different quadrature strategies mentioned above, we need some kind of test function that resembles as much as possible the functions we need to integrate. We choose the following:

$$f(m,n,\alpha) = \frac{P_m(x)P_n(x)e^{-\alpha/x}}{x\left[x^{20}\left(\log(1-x)\right)^{20} + x^{20} + 10\right]}. \tag{5.15}$$

The function above has the same kind of oscillatory behaviour (thanks to the Legendre polynomials) and the same singular behaviour at the two endpoints of integration. We noted the number of function evaluations needed to arrive at a minimum relative accuracy of $\mathcal{O}(10^{-12})$ for different combinations of m, n and α. These results are summarised in Table 5.1. Results annotated with † did not achieve the requested accuracy. From this table, several conclusions can be drawn:

(i) the effect of the variable transformation becomes more important for increasingly singular behaviour ($\alpha \to 0$)
(ii) both IMT as TANH perform well, IMT has a small advantage
(iii) the SINC method cannot handle highly oscillatory integrands
(iv) the INTDE method outperforms all others

Of course, function evaluations are not the only cost. When using variable transforms, these come into play too. We have run some tests and estimated the cost of both the TANH and IMT transforms and also the overhead in the different quadrature routines themselves. These estimates have been obtained using the VALGRIND profiling framework [SN04]. VALGRIND is an award-winning suite of tools for debugging and profiling x86-Linux programs. With the tools that come with Valgrind, one can automatically detect many memory management and threading bugs, making your programs more stable. One can also perform detailed profiling, to speed up and reduce memory use of your programs. The VALGRIND distribution currently includes five tools: two memory error detectors, a thread error detector, a cache profiler and a heap profiler.

We have normalised the results so that the overhead per function evaluation for DQAGSE equals 100. Those results are indicated in Table 5.2. So when we use the DQAGSE routine together with the IMT-transform, we need to add both overheads together to arrive at the total overhead. We clearly see the effect of the Chebyshev approximation used for the ITM-transformation: it is cheaper than the TANH-transformation (which uses the exact formula). The three quadrature routines have about the same overhead per function evaluation.

Final choice of quadrature routine

Given the analysis in the previous section on the test function defined there, we have taken the double-exponential method INTDE as our method of choice. In the case this method should fail (and then the routine returns an error code as such), we use the DQAGSE method with the IMT-transformation as a backup solution.

5.5 An accurate method for the spatial integral

As in the discrete part, we also need angular/spatial moments of the infinite medium Green's function:

α	m	n	DQAGSE	DQAGSE IMT	DQAGSE TANH	INTDE	SINC
10	0	0	609	315	357	80	†
	1	0	609	315	357	89	†
	50	25	1365	1239	1491	327	†
	50	26	1365	1281	1449	329	†
1	0	0	609	273	273	78	121
	1	0	609	273	273	78	121
	50	25	1701	1701	2037	322	†
	50	26	1701	1659	1701	325	†
10^{-4}	0	0	1449	399	357	301	161
	1	0	1365	315	357	154	161
	50	25	2163	1575	1953	315	625
	50	26	1701	1029	987	313	641
10^{-8}	0	0	2205	483	1071	322	197
	1	0	2205	357	399	82	193
	50	25	1911	1953	1953	336	†
	50	26	2625	1071	1659	335	785
0	0	0	/	/	/	/	/
	1	0	483	315	483	85	229
	50	25	1701	2499	2625	354	897
	50	26	/	/	/	/	/

Table 5.1: *Function evaluations for different quadrature methods.*

DQAGSE	IMT	TANH	INTDE	SINC
100	230	287	95	107

Table 5.2: *Overhead per function evaluation relative to* DQAGSE.

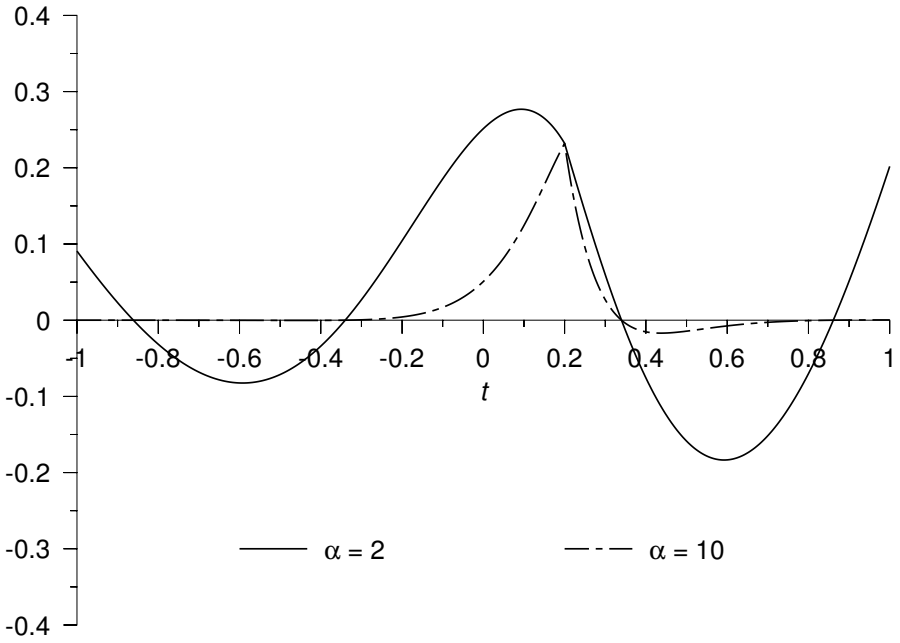


Figure 5.3: *Integrand of Equation (5.17) for* $k = 4$, $x = 0.2$ *and two values of* α.

$$[G_c]^k_{m,n}(x) = \int_a^b \tilde{P}_k(x_0)\,[G_c]_{m,n}(x_0|x)dx_0 \tag{5.16}$$

where the $\tilde{P}_k(x_0)$ are the transformed Legendre polynomials orthogonal on the interval $[a,b]$. Our method of working is to treat the spatial integral as the inner integral so we need a routine to evaluate the following generic integral

$$EA_k(x,\alpha) = \int_c^d \tilde{P}_k(x_0)e^{-\alpha|x-x_0|}dx_0 \tag{5.17}$$

where $\alpha = \Sigma_t/\nu$ is a positive real number. In contrast with the discrete case where α could not be larger than Σ_t and hence remains small, we here have $\alpha \in [0,+\infty[$. The larger α, the more peaked the integrand becomes at $x_0 = x$ (the quicker the decay left and right from $x_0 = x$). Figure 5.3 shows the integrand behaviour for different values of α. We therefore need to take special measures for large values of α.

In the following, we shall derive a stable method for the whole range of α. In our next analysis, we assume the orthogonality interval $[a,b]$ to be $[-1,+1]$, hence we work with the classical Legendre polynomials and we

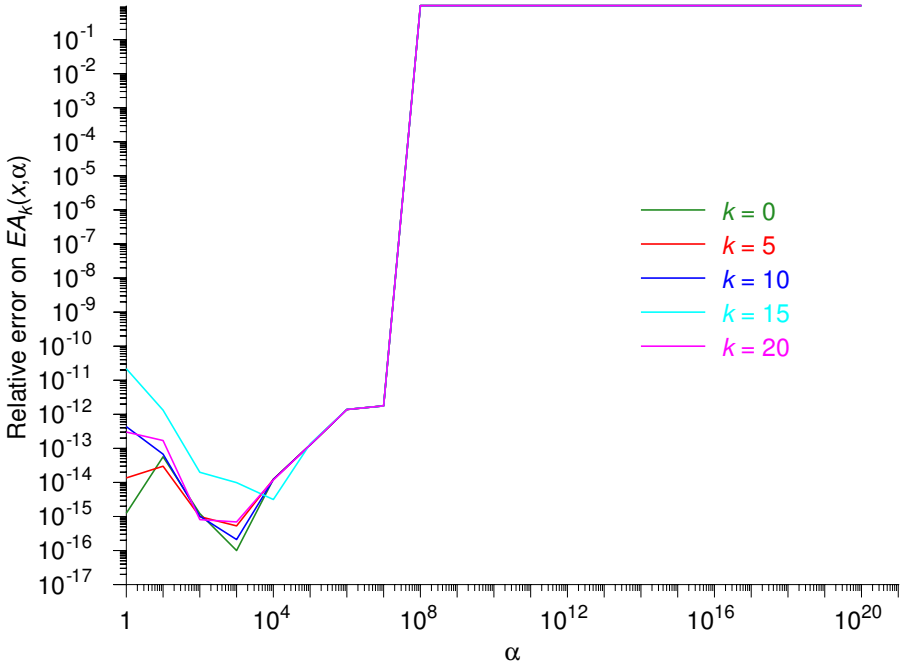


Figure 5.4: *Relative error for the numerical quadrature approach for* $EA_k(x,\alpha)$, $x = 0.2$.

assume the integration interval $[c,d]$ to be equal to the orthogonality interval. These two assumptions only avoid variable shifts and do not reduce the computational difficulty, merely the notational complexity.

First approach: Numerical quadrature

We apply the INTDE double-exponential routine we discussed in the previous section. It is clear that we split the problem in two sub-problems given the singularity (in the derivatives of the integrand) for $x_0 = x$. Figure 5.4 shows the relative error compared to MAPLE results for $EA_k(x,\alpha)$, $x = 0.2$ and different values of k as a function of α. It is clear from this figure that we loose accuracy from $\alpha = 10^{+7}$ onwards.

Second approach: A series expansion

The starting point for this series approach is the identity [AN04]

$$EA_2(\alpha,k,l) = \int_{-1}^{+1} dx \int_{-1}^{+1} dx_0 P_k(x) e^{-\alpha|x-x_0|} P_l(x_0) \tag{5.18}$$

$$= \begin{cases} \frac{4}{\alpha}\left[\frac{\delta_{k,l}}{2k+1} - K_{m+1/2}(\alpha) I_{M+1/2}(\alpha)\right] & (k+l) \text{ even} \\ 0 & (k+l) \text{ odd} \end{cases} \tag{5.19}$$

where $M = \max(k,l)$, $m = \min(k,l)$ and $I_{M+1/2}(\alpha), K_{m+1/2}(\alpha)$ are the half-integer order modified Bessel functions of the first and second kind respectively [AS72]. We can use this identity to express the exponential function in the integrand using a double Legendre series expansion:

$$e^{-\alpha|x-x_0|} = \sum_{k=0}^{\infty} \frac{2k+1}{2} \left(\sum_{l=0}^{\infty} \frac{2l+1}{2} EA_2(\alpha,k,l) P_l(x) \right) P_k(x_0). \tag{5.20}$$

The latter expansion can then be used in the spatial integral (5.17). Figure 5.5 shows the relative error compared to MAPLE results for $EA_k(x,\alpha)$, $x = 0.2$ and different values of k as a function of α. It shows good results for small and very large α.

Third approach: Integration by parts

Let us start from the integral (5.17) transformed to the orthogonality interval $[-1,+1]$ and split it in two parts to get rid of the absolute value notation:

$$EA_k^-(x,\alpha) = \int_{\tilde{c}}^{x} P_k(x_0) e^{-\alpha(x-x_0)} dx_0 \tag{5.21}$$

$$EA_k^+(x,\alpha) = \int_{x}^{\tilde{d}} P_k(x_0) e^{+\alpha(x-x_0)} dx_0 \tag{5.22}$$

where $\tilde{c}$ and $\tilde{d}$ are the transformed integration boundaries. We shall do the analysis using $EA_k^-(x,\alpha)$ and give the result for $EA_k^+(x,\alpha)$ afterwards. Applying integration by parts on Equation (5.21), we find

$$EA_k^-(x,\alpha) = \left[\frac{1}{\alpha} P_k(x_0) e^{-\alpha(x-x_0)}\right]_{\tilde{c}}^{x} - \frac{1}{\alpha} \int_{\tilde{c}}^{x} P_k'(x_0) e^{-\alpha(x-x_0)} dx_0. \tag{5.23}$$

Repeating this process k times, we finally arrive at the following (exact) expression

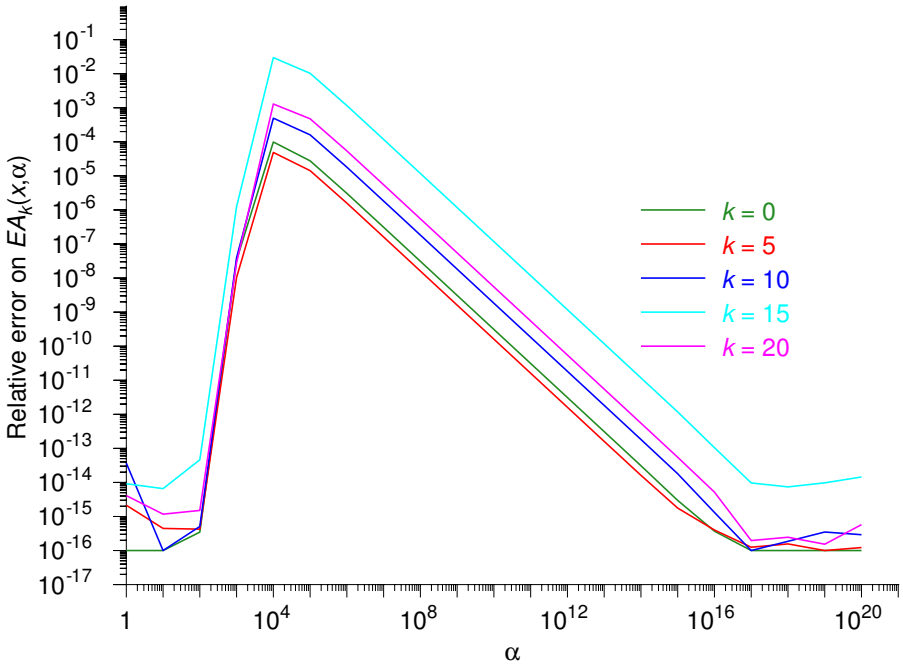


Figure 5.5: *Relative error for the series expansion approach for* $EA_k(x,\alpha)$, $x = 0.2$.

$$EA_k^-(x,\alpha) = \sum_{j=0}^{k}(-)^j \frac{1}{\alpha^{j+1}} \left(\frac{d^j}{dx_0^j} P_k(x_0)\Big|_{x_0=x} \right) - \left[\sum_{j=0}^{k}(-)^j \frac{1}{\alpha^{j+1}} \left(\frac{d^j}{dx_0^j} P_k(x_0)\Big|_{x_0=\tilde{c}} \right) \right] e^{-\alpha(x-\tilde{c})} \tag{5.24}$$

while for $EA_k^+(x,\alpha)$ we find in an analogous fashion

$$EA_k^+(x,\alpha) = \sum_{j=0}^{k} \frac{1}{\alpha^{j+1}} \left(\frac{d^j}{dx_0^j} P_k(x_0)\Big|_{x_0=x} \right) - \left[\sum_{j=0}^{k} \frac{1}{\alpha^{j+1}} \left(\frac{d^j}{dx_0^j} P_k(x_0)\Big|_{x_0=\tilde{d}} \right) \right] e^{-\alpha(\tilde{d}-x)}. \tag{5.25}$$

In both Equations (5.24) and (5.25), we see that the first sum converges quickly for α reasonably large. The second sum has the same convergence

property, but it gets multiplied with an exponential factor. So for x not too close to one of the integration end-points, this factor becomes extremely small for reasonable α and can be safely neglected. For x close to $\tilde{d}$, this term cannot be neglected in $EA_k^+(x,\alpha)$, while for x close to $\tilde{c}$, it cannot be neglected in $EA_k^-(x,\alpha)$. We use the criterion $\alpha \cdot \delta < 100$, where $\delta = (\tilde{d} - x)$ or $\delta = (x - \tilde{c})$ to decide whether to add the term containing the exponential factor as a correction term.

Note also that in the end we are interested in the sum

$$EA_k(x,\alpha) = EA_k^+(x,\alpha) + EA_k^-(x,\alpha).$$

In this full value, the first sum contains only the even derivatives (or the odd negative powers of α) due to the $(-)^j$ factor in $EA_k^-(x,\alpha)$ and its non-appearance in $EA_k^+(x,\alpha)$.

The final problem to be solved is the accurate and fast calculation of the successive derivatives of the Legendre polynomials. From [MOS66, p. 232], we have

$$\frac{d^j}{dx^j} P_k(x) = 2^j \left(\frac{1}{2}\right)_j C_{k-j}^{j+\frac{1}{2}}(x),\ k \geq j \tag{5.26}$$

where

$$\left(\frac{1}{2}\right)_j = \frac{1}{2}(\frac{1}{2}+1)\ldots(\frac{1}{2}+j-1) = \frac{\Gamma(\frac{1}{2}+j)}{\Gamma(\frac{1}{2})}$$

denotes the Pochhammer symbol and the C_n^λ are the Gegenbauer (or ultraspherical) orthogonal polynomials [Sze59; MOS66]. They obey the, for us relevant, recurrence relation for fixed λ

$$(n+1)C_{n+1}^\lambda(x) = 2(n+\lambda)xC_n^\lambda(x) - (n+2\lambda-1)C_{n-1}^\lambda(x) \tag{5.27}$$

with initial values

$$C_0^\lambda(x) = 1, C_1^\lambda(x) = 2\lambda x \text{ if } \lambda \neq 0. \tag{5.28}$$

We have now all elements at hand to implement this third approach to calculate the spatial integral. The final algorithm is shown in Algorithm 5.1. Again, the comparison with MAPLE results are shown in Figure 5.6. We see clearly a very good accuracy for $\alpha > 10^{+2}$. For smaller α, the sums do not converge quickly enough to avoid instability in the summation process (this is comparable to the classical algorithms for the incomplete gamma function – a function closely related to our $EA_k(x,\alpha)$ – which also suffer from the same problem for small arguments).

Input: x, argument; α, argument; k, argument
Input: $\tilde{c}$, left integration end-point
Input: $\tilde{d}$, right integration end-point
Output: $EA_k(x,\alpha)$, where we assume that the integral has already been transformed to the orthogonality interval $[-1,+1]$

Calculate the sum of the first summations in Equations (5.24,5.25), taking only into account the terms containing even derivatives;
if $\alpha(x-\tilde{c}) \leq 100$ **then**
  Add the second summation of Equation (5.24) as a correction;
endif
if $\alpha(\tilde{d}-x) \leq 100$ **then**
  Add the second summation of Equation (5.25) as a correction;
endif

Algorithm 5.1: Algorithm for the spatial integral using integration by parts.

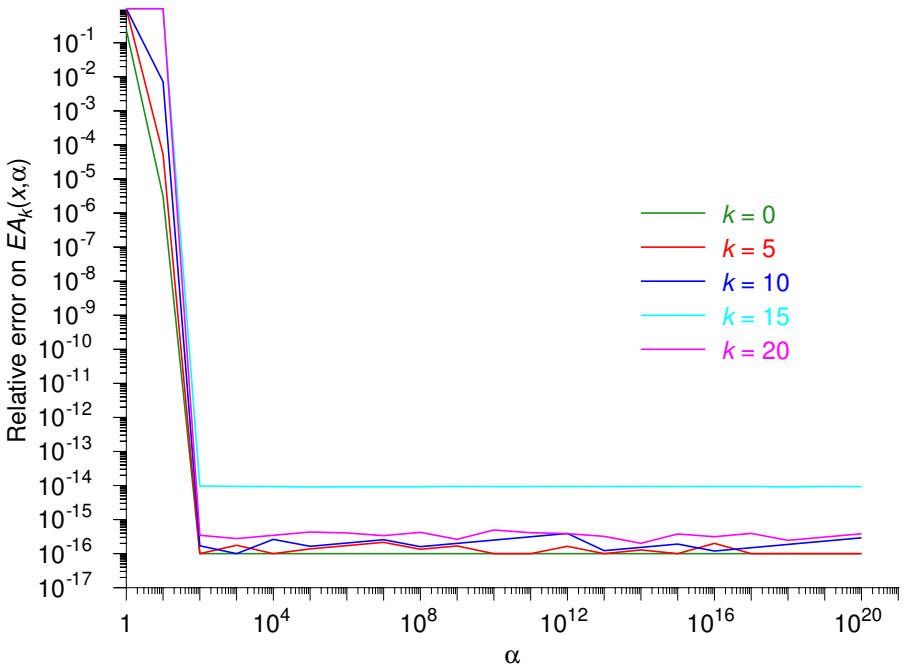


Figure 5.6: *Relative error for the partial integration approach for* $EA_k(x,\alpha)$, $x = 0.2$.

The final algorithm for $EA_k(x, \alpha)$

The final algorithm is a combination of the direct integration method and the integration by parts method, depending on the value of α. It is summarized in Algorithm 5.2. The series expansion method has been discarded since the others cover the interval quite well and are much cheaper (quite large orders of approximation were needed for the series expansion if α was not large enough).

Input: x, argument; α, argument; k, argument
Input: a,b, orthogonality interval
Input: c,d, integration interval
Output: $EA_k(x, \alpha)$, the value of the integral (5.17)

Do a variable substitution so we work on the orthogonality interval $[-1, +1]$ and use the real Legendre polynomials;
if $\alpha < 10^{+04}$ **then**
| Apply the INTDE double-exponential quadrature;
else
| Use the integration by parts method (Algorithm 5.1)
endif

Algorithm 5.2: Combined algorithm for the spatial integral $EA_k(x, \alpha)$.

5.6 Conclusions

We have selected an adequate numerical quadrature scheme for both the angular and angular/spatial moments of the continuum part of the infinite medium Green's function. For the angular part, the double-exponential method was selected as default routine. In case of non-convergence (which is indicated by the routine), the problem is recalculated using the Gaussian quadrature DQAGSE with the IMT-transformation to eliminate the endpoints singularities of the angular moments integrand.

For the spatial moments, the domain of the problem is split in two based on the parameter α. For small values of α, the double-exponential method is used. For larger values of α, we use the expressions derived by integration by parts. Depending on the position of x with respect to the integration boundaries, a correction term is added or not. In both regions for α the problem is split in two to eliminate the absolute value in the exponential function. The numerical results show us that we get more than twelve significant digits on the result, which is quite enough for our purposes.

CHAPTER 6

The Boundary Sources Method

SUMMARY– The Boundary Sources Method is an integral transport method that makes capital use of the knowledge of an exact transport kernel. Indeed, the solution to the single speed neutron transport equation in a finite homogeneous volume possibly containing volume sources is represented as a convolution of the boundary sources, the unknowns, with an exact transport kernel for the infinite homogeneous medium.

The application of the Boundary Sources Method to a one-dimensional slab problem is the topic of this chapter. This has been done already in [Bea68; VdV95] for isotropic scattering. The main equations for anisotropic scattering are in no way different but the fact that the treatment of the transport kernel differs. A discretized version of the Boundary Sources Method is presented together with the classical continuity and boundary condition requirements. Five test cases conclude this chapter.

6.1 Principle and basic equations

The main driving force of the development of the Boundary Sources Method (BSM) is to get the most advantage out of the knowledge of an exact transport kernel, like the Green's function for homogeneous infinite media. This limits the use of the method to planar, cylindrical and spherical applications. However, this aspect fits perfectly in the field of nodal transport methods, in which the multi-dimensional problem is transformed in a coupled set of one-dimensional problems with transfer terms. The resulting one-dimensional problems almost always have a planar geometry, so this work focuses on the treatment of a planar slab problem.

The Boundary Sources Method is an integral method, in which the un-

knowns – the boundary sources – can be treated with arbitrary accuracy. Indeed, the boundary sources are angular dependent and we use a Legendre polynomial expansion to approximate them. The higher the approximation order, the better the final results.

Let us start by writing down the problem we want to solve: the calculation of the solution to the one-speed (non-)homogeneous Boltzmann transport equation in a slab $-\frac{a}{2} \leq x \leq +\frac{a}{2}$ with a possible independent source term $Q(x,\mu)$:

$$\mu\frac{\partial\psi(x,\mu)}{\partial x} + \Sigma_t\psi(x,\mu) = \frac{c\Sigma_t}{2}\sum_{l=0}^{N}(2l+1)f_l P_l(\mu)\int_{-1}^{+1}\psi(x,\mu')\,P_l(\mu')d\mu' + Q(x,\mu). \quad (6.1)$$

According to the general theory behind the Boundary Sources Method [Bea68], one can write the solution to this problem as

$$\begin{aligned}\psi(x,\mu) = &\int_{-1}^{+1} G(-\frac{a}{2},\mu_0|x,\mu)f_-(\mu_0)d\mu_0 \\ &\int_{-1}^{+1} G(+\frac{a}{2},\mu_0|x,\mu)f_+(\mu_0)d\mu_0 + \psi_\infty(x,\mu)\end{aligned} \quad (6.2)$$

where $G(x_0,\mu_0|x,\mu)$ denotes the infinite medium Green's function, $f_\pm(\mu_0)$ are the boundary sources to be determined and $\psi_\infty(x,\mu)$ is a particular solution of the inhomogeneous infinite medium problem.

6.2 The particular solution $\psi_\infty(x,\mu)$

The particular solution of the inhomogeneous transport equation can be derived using the classical convolution integral

$$\psi_\infty(x,\mu) = \int_{-\frac{a}{2}}^{+\frac{a}{2}} dx_0 \int_{-1}^{+1} d\mu_0 G(x_0,\mu_0|x,\mu)Q(x_0,\mu_0). \quad (6.3)$$

In this work, we suppose we can develop the independent source term $Q(x,\mu)$ in a double Legendre approximation:

$$Q(x_0,\mu_0) = \sum_{m=0}^{N_a}\frac{2m+1}{2}q_m(x_0)P_m(\mu_0) \quad (6.4)$$

where the Legendre moments $q_m(x_0)$ are defined as

$$q_m(x_0) = \int_{-1}^{+1} Q(x_0,\mu_0)P_m(\mu_0)d\mu_0 \tag{6.5}$$

and the approximation order N_a is chosen beforehand. Let us define in the same fashion the angular moments of the particular solution as

$$\psi_{\infty,n}(x) = \int_{-1}^{+1} \psi_\infty(x,\mu)P_n(\mu)d\mu. \tag{6.6}$$

Putting the expansion (6.4) and the expression for $\psi_\infty(x,\mu)$ (6.3) in the latter equation we find

$$\psi_{\infty,n}(x,\mu) = \sum_{m=0}^{N_a} \frac{2m+1}{2} \int_{-\frac{a}{2}}^{+\frac{a}{2}} dx_0\, q_m(x_0)\, [G]_{n,m}\,(x_0|x) \tag{6.7}$$

where we use the notation of Chapters 4 and 5 for the full-range angular moments of the Green's transport kernel

$$[G]_{n,m}\,(x_0|x) = \int_{-1}^{+1} d\mu \int_{-1}^{+1} d\mu_0 P_n(\mu)G(x_0,\mu_0|x,\mu)P_m(\mu_0). \tag{6.8}$$

The Legendre angular moments $q_m(x_0)$ can at their turn be developed in a spatial Legendre series on the interval $[-\frac{a}{2},+\frac{a}{2}]$. We use the notation $\tilde{P}_k(x_0)$ for the Legendre polynomials transfered to this interval (they form the set of polynomials orthogonal on this interval with respect to the weight function $w(x) \equiv 1$). The $q_m(x_0)$ can be rewritten as

$$q_m(x_0) = \sum_{k=0}^{N_s} \frac{2k+1}{2} q_{m,k}\tilde{P}_k(x_0) \tag{6.9}$$

where the double moments $q_{m,k}$ are given by

$$q_{m,k} = \int_{-\frac{a}{2}}^{+\frac{a}{2}} q_m(x_0)\tilde{P}_k(x_0)dx_0. \tag{6.10}$$

Putting the latter expansion in Equation (6.7) we arrive at

$$\psi_{\infty,n}(x) = \sum_{m=0}^{N_a} \frac{2m+1}{2} \left(\sum_{k=0}^{N_s} \frac{2k+1}{2} q_{m,k}\,[G]^k_{n,m}\,(x) \right) \tag{6.11}$$

where we use the notation $[G]^k_{n,m}\,(x)$ for the angular/spatial moment of the Green's transport kernel

$$[G]^k_{n,m}(x) = \int_{-\frac{a}{2}}^{+\frac{a}{2}} [G]_{n,m}(x_0|x)\tilde{P}_k(x_0)dx_0. \tag{6.12}$$

6.3 One dimensional slab equations

Suppose we have a slab of R adjacent cells, each having a width a^i, neutronics parameters Σ_t^i and c^i and a given independent neutron source $Q^i(x,\mu)$. If we use local coordinates for each cell (in each cell the spatial variable x goes from $-a^i/2$ to $+a^i/2$), we can write the angular flux in cell i as

$$\begin{aligned}\psi^i(x,\mu) = &\int_{-1}^{+1} G^i(-\frac{a^i}{2},\mu_0|x,\mu)f_-^i(\mu_0)d\mu_0 \\ &\int_{-1}^{+1} G^i(+\frac{a^i}{2},\mu_0|x,\mu)f_+^i(\mu_0)d\mu_0 + \psi_\infty^i(x,\mu)\end{aligned} \tag{6.13}$$

where the $f_\pm^i(\mu_0)$ are the unknown boundary sources. We get a discretized version of the above equations by replacing the boundary sources by a Legendre expansion. This Legendre expansion doesn't need to be a full-range expansion since we are only interested in the half-ranges $\mu_0 \in [0, \pm 1]$. Hence, the expansion is given by

$$f_\pm^i(\mu_0) = \sum_{m=0}^{M} \frac{4m+3}{2} f_{m\pm}^i P_{2m+1}(\mu_0). \tag{6.14}$$

The discretized equations

In order to arrive at a discrete set of equations that one can solve, we put the expansion (6.14) in the angular flux definition (6.13) and we approximate the resulting equation for the angular flux by a number of equations for several angular moments of the former equation. We have $2(M+1)$ unknowns per cell so to have a closed problem, we need $2(M+1)$ equations.

Internal angular flux continuity

For each boundary between two cells, we require continuity of the angular flux over the boundary,

$$\psi^i(+a^i/2,\mu) = \psi^{i+1}(-a^{i+1}/2,\mu),\ i = 1,\ldots,R-1.$$

The corresponding discrete equations are

$$\sum_{m=0}^{M} \frac{4m+3}{2} \Big[f^i_{m-} \left[G\right]^i_{n,2m+1} (a^i|0) + (-)^{n+1} f^i_{m+} \left[G\right]^i_{n,2m+1} (+0|0) - f^{i+1}_{m-} \left[G\right]^{i+1}_{n,2m+1} (+0|0) - (-)^{n+1} f^{i+1}_{m+} \left[G\right]^{i+1}_{n,2m+1} (a^{i+1}|0) \Big] = \psi^{i+1}_{\infty,n} \left(-\frac{a^{i+1}}{2} \right) - \psi^i_{\infty,n} \left(+\frac{a^i}{2} \right) \tag{6.15}$$

for $n = 0, 1, \ldots, 2M+1$. Note that we have used the translational invariance of transport kernels

$$G(x_0, \mu_0 | x, \mu) = G(x_0 - x, \mu_0 | 0, \mu)$$

to arrive at the above equation.

Reflection boundary condition

The reflection boundary condition at the left end is given by

$$\psi^1 \left(-a^1/2, \mu \right) = \psi^1 \left(-a^1/2, -\mu \right)$$

and the discretized form boils down to

$$\sum_{m=0}^{M} \frac{4m+3}{2} \left[f^1_{m-} \left[G\right]^1_{2n+1,2m+1} (+0|0) + f^1_{m+} \left[G\right]^1_{2n+1,2m+1} (a^1|0) \right] = -\psi_{\infty,2n+1} \left(-\frac{a^1}{2} \right) \tag{6.16}$$

for $n = 0, \ldots, M$. A similar equation holds for the right boundary.

Vacuum boundary condition

We treat the vacuum boundary condition as a continuity condition with an extra slab at index $i = 0$ or $i = R+1$ of arbitrary width and vacuum neutronics data: $\Sigma_t = 0, c = 0$. This generates two extra boundary sources per vacuum boundary condition. One of these is arbitrary since the boundary sources furthest away can be set as one wishes, i.e. $f^0_-(\mu)$ and $f^{R+1}_+(\mu)$. It

is custom to set them to zero. The $f_+^0(\mu)$ and $f_-^{R+1}(\mu)$ are the remaining unknowns.

This approach is not the best possible. Indeed, one cannot exactly represent vacuum boundary conditions using full-range angular moments. A better option would be to work with the half-range angular moments and require that the incoming moments at left and right boundary are zero. This approach is implemented in the BSNPL code [VdV95] developed as a predecessor of our code treating only the isotropic scattering case. However, to do so, one needs the half-range moments of the infinite medium Green's function. As seen in Chapter 3, in the case of isotropic scattering, one can derive the half-range functions using the $\mathcal{K}$-transform to obey an homogeneous three-term recursion. But in the case of anisotropic scattering, the same approach leads to a non-homogeneous recursion and this makes the evaluation of the half-range functions a lot more troublesome. We therefore opted to stay with the full-range moments and see how the results of that approach were in the benchmarks.

All these equations lead to a linear system of size $(2M+2) \times (2M+2)$:

$$A f = b \tag{6.17}$$

where A is the BSM matrix, f are the boundary sources to be solved for and b is the right-hand-side. The matrix A is block-diagonal and specialised routines could be used to solve this linear system.

The expression for the angular and scalar flux

Once the boundary sources $f_\pm(\mu)$ are known through their Legendre approximation coefficients $f_{m\pm}$, we can rewrite Equation (6.13) for cell i as

$$\begin{aligned}\psi^i(x,\mu) = \sum_{m=0}^{M} \frac{4m+3}{2} \Bigg\{ & f_{m-}^i \int_{-1}^{+1} G^i(-\frac{a}{2}, \mu_0 | x, \mu) P_{2m+1}(\mu_0) d\mu_0 \\ & f_{m+}^i \int_{-1}^{+1} G^i(+\frac{a}{2}, \mu_0 | x, \mu) P_{2m+1}(\mu_0) d\mu_0 \Bigg\} \\ & + \psi_\infty^i(x,\mu).\end{aligned} \tag{6.18}$$

This expression cannot be evaluated as such but its angular moments can. They are given by

$$\psi_n^i(x) = \sum_{m=0}^{M} \frac{4m+3}{2} \left\{ f_{m-}^i \left[G\right]_{n,2m+1} (-\frac{a}{2}, x) + (-1)^n f_{m+}^i \left[G\right]_{n,2m+1} (+\frac{a}{2}, x) \right\} + \psi_{\infty,n}^i(x) \tag{6.19}$$

and hence the angular flux $\psi^i(x,\mu)$ in cell i can be calculated as a Legendre series

$$\psi^i(x,\mu) = \sum_{n=0}^{N'} \frac{2n+1}{2} \psi_n^i(x) P_n(\mu) \tag{6.20}$$

where the approximation order N' can be as high as one wishes. The above equations serve as the basic evaluation routines for a cell in the slab, once the boundary sources are known. The cell scalar flux is then given by the zeroth angular moment $\psi_0^i(x)$.

6.4 Implementation

We have implemented a multi-cell slab solver using the Boundary Sources Method with the infinite medium Green's function as a transport kernel. The code uses the CASE library created to evaluate the angular and angular/spatial moments of the Green's function described in the previous chapters. We have adopted the "library" approach as opposed to the input file approach. To treat a problem, the user must write a small C++ program and link this with our CASE and CASE-BSM libraries.

The work-flow can be summarized as follows. The user creates a number of cells, assigning them nuclear data c, Σ_t and a scatter function (a vector of size $N+1$ containing the anisotropic scattering coefficients). An external source can also be added to a cell. In the next step, the user creates a slab and adds the previously created cells to the slab in any order she likes. After that, the user can create the boundary conditions that apply at the left and right edge of the slab. Finally, the command to solve the slab for a given BSM order M (the expansion order for the boundary sources) is given.

Following this command, the code loops over all cells and computes the needed angular and angular/spatial moments for each cell. In a second stage, the discretized version of the boundary sources equations are determined, leading to the $(2M+2) \times (2M+2)$ linear system of equations to

be solved. The system is block diagonal, but for the ease of programming, we simply call the LAPACK routine DGESV to solve the system by Gaussian *LU*-factorisation using partial pivoting.

After this step, the problem is solved, i.e. the boundary sources are known. The user then can extract the required data he wants: angular fluxes or total fluxes at certain positions in a specific cell, cell averaged fluxes, etc. For a complete overview of the CASE and CASE-BSM libraries, see Chapter 8.

6.5 Examples and benchmarks

Benchmarking is, of course, a necessary part of code development. In order to benchmark one needs either experimental data or other codes which one believes to be more accurate (or at least as accurate) as the one being under development. Since the code at this stage is only a one-dimensional slab, the choice for code comparison is evident. We have the following codes at our disposal to do so:

BSNPL [VdV95] a Boundary Sources Method code limited to isotropic scattering, maximum 5-th order BSM, FORTRAN 77;

BLUE [Gan05] a Green's function method using numerical Fourier transforms, kindly provided by B. Ganapol, anisotropic scattering, FORTRAN 90;

MCNPX 2.5E [HMW+04] a general-purpose Monte Carlo N-Particle code, we consider this as the "ultimate reference", FORTRAN 95, all MCNPX results are kindly provided by Ir. W. Haeck, SCK•CEN;

CASE-BSM the code developed in this work, a Boundary Sources Method code with arbitrary order anisotropic scattering, C++.

Test Case 1: flux disadvantage factor, isotropic scattering

This test case was already present in [Bea68] and we redid the calculation to test the general correctness of our code (and hence avoid the new part of the anisotropic scattering). The problem is as follows: given a cell calculation with fuel and moderator, calculate the flux disadvantage factor

$$\zeta = \frac{\bar{\psi}_1}{\bar{\psi}_0}$$

	Width a_i	$\Sigma_{t,i}$ cm^{-1}	c_i	Source Q_i
Cell 0	$0.1, 0.2, 0.3, 0.4$	0.717	0.554	0
Cell 1	$3.5 \times a_0$	2.33	0.999957	1

Table 6.1: *Nuclear data for Test Case 1.*

for different sizes of the cells. The cell data are summarized in Table 6.1. The results ζ and relative errors ϵ (where we take the MCNPX calculation as a reference) are shown in Table 6.2. The MCNPX results have a relative error of $\mathcal{O}(10^{-4})$, hence 4 significant digits (at one standard deviation). Results for BLUE are not available since this code does not allow for reflective boundary conditions. To give an idea on the timings: BSNPL requires a few seconds, CASE-BSM between seconds (order 5) and a minute or so (order 30). The MCNPX run took about 12 hours.

Test Case 2: flux disadvantage factor, anisotropic scattering

Just to test our anisotropic part of the code, we retook the example of test case 1, the problem of calculating the disadvantage factor but adding an artificial anisotropic scattering based on the Henyey-Greenstein kernel with $g = \frac{1}{2}$ and an approximation order $N = 4$. For this problem, we could only compare to MCNPX results. The results are in Table 6.3.

Test Case 3: scalar flux, Henyey-Greenstein $g = 0.5$, $c = 0.9$

In our next test case we want to evaluate the scalar neutron flux for a single slab of width $a = 1$ containing a poorly absorbing medium $c = 0.9$ obeying the Henyey-Greenstein scattering law with free parameter $g = 0.5$ which we break off at order $N = 12$. We compare our CASE-BSM code with BLUE and MCNPX. Note that the MCNPX results are not pointwise results. The mesh size used in the MCNPX runs was $h = 0.01$ and hence results for a point x are average fluxes over the interval $[x - h/2, x + h/2]$. In order to be able to compare the results from CASE-BSM and BLUE, we need to have these average values too. Since in CASE-BSM the scalar flux can be evaluated anywhere in the cell quite easily using Equation (6.20), the average over a certain interval can be efficiently calculated using a numerical quadrature. For the BLUE results, we let the code print the scalar flux at more than 4000 equidistant edit points in the cell and use Romberg integration [DR84] to evaluate the average flux over the interval.

Code	$a_0 = 0.1$		$a_0 = 0.2$		$a_0 = 0.3$		$a_0 = 0.4$	
	ζ	ϵ	ζ	ϵ	ζ	ϵ	ζ	ϵ
BSNPL, $M = 5$	1.09476	0.28%	1.23173	0.041%	1.41062	0.028%	1.637053	0.033%
CASE, $M = 5$	1.09473	0.28%	1.23170	0.044%	1.41059	0.026%	1.63702	0.031%
CASE, $M = 10$	1.09746	0.0354%	1.23251	0.022%	1.41078	0.039%	1.63703	0.032%
CASE, $M = 20$	1.097902	0.0057%	1.23253	0.023%	1.41076	0.039%	1.63703	0.032%
CASE, $M = 30$	1.097917	0.0070%	1.23252	0.023%	1.41076	0.039%	1.63703	0.032%
MCNPX	1.09784	ref.	1.23224	ref.	1.41022	ref.	1.63651	ref.

Table 6.2: *Disadvantage factors for Test Case 1.*

Code	$a_0 = 0.1$		$a_0 = 0.2$		$a_0 = 0.3$		$a_0 = 0.4$	
	ζ	ϵ	ζ	ϵ	ζ	ϵ	ζ	ϵ
CASE, $M = 5$	1.08323	0.34%	1.18416	0.14%	1.29987	0.047%	1.43535	0.010%
CASE, $M = 10$	1.08619	0.072%	1.18506	0.059%	1.30008	0.030%	1.43537	0.009%
CASE, $N = 20$	1.08665	0.029%	1.18508	0.058%	1.30007	0.031 %	1.43537	0.009%
CASE, $M = 30$	1.08697	0.028%	1.18508	0.058%	1.30007	0.031 %	1.43537	0.009%
MCNPX	1.08697	ref.	1.18576	ref.	1.30048	ref.	1.43549	ref.

Table 6.3: *Disadvantage factors for Test Case 2.*

Code	Left		Quarter		Mid	
CASE, $M = 5$	0.98015	0.47%	1.37456	0.09%	1.46499	0.10%
CASE, $M = 10$	0.97693	0.14%	1.37497	0.06%	1.46532	0.08%
CASE, $M = 20$	0.97606	0.06%	1.37502	0.05%	1.46532	0.08%
CASE, $M = 30$	0.97586	0.03%	1.37502	0.05%	1.46532	0.08%
BLUE	0.97608	0.06%	1.37511	0.05%	1.46524	0.08%
MCNPX	0.97552	ref.	1.37574	ref.	1.46644	ref.

Table 6.4: *Scalar fluxes for Test Case 3.*

Comparisons of the average scalar flux over a distance $h = 0.01$ at left boundary, a quarter width and midpoint are shown in Table 6.4. A flux plot for the three methods is shown in Figure 6.1. Note that for MCNPX we have plotted the average values, hence the "stair-case" style of the plot. We clearly see that the results of CASE-BSM and BLUE are very good compared to MCNPX results.

Test Case 4: scalar flux, Henyey-Greenstein $g = 0.8$, $c = 0.9$

The next test case investigates the effect of a scattering kernel that becomes negative for certain values of μ. This can happen, for example, when using the Henyey-Greenstein kernel for large values of the free parameter g and a low order approximation (see Chapter 7 for the necessary approximation orders for several criteria amongst which non-negativity). We study the problem of a unit size cell, poorly absorbing medium $c = 0.9$ and the Henyey-Greenstein kernel for $g = 0.8$.

We use three approximation orders: $N = 10$, $N = 20$ and $N = 60$. The real kernel and these three approximations are shown in Figure 6.2. For $N = 10$, the Legendre approximation has four disconnected regions where it is negative. For $N = 20$, there is a very small (both in width as in amplitude) negative dip around $\mu = -0.98$. For $N = 60$, there is no negative region and we even have a relative error between real kernel and Legendre approximation which is less than 10^{-04} (see Chapter 7, Table 7.3).

The fact that the scattering function becomes negative for certain values of μ poses problems for stochastic codes, since they use this scattering function as a probability function and create so-called equiprobability bins based on this scattering function. If there are negative parts, the normalisation of these bins becomes awkward. The calculations using MCNPX use a

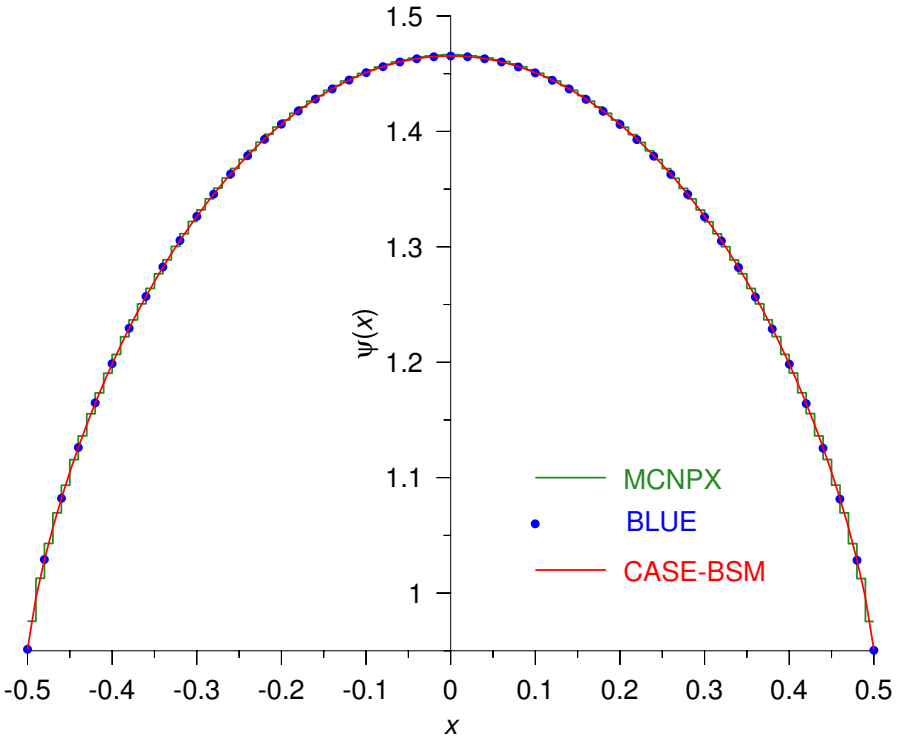


Figure 6.1: *Scalar fluxes for Test Case 3.*

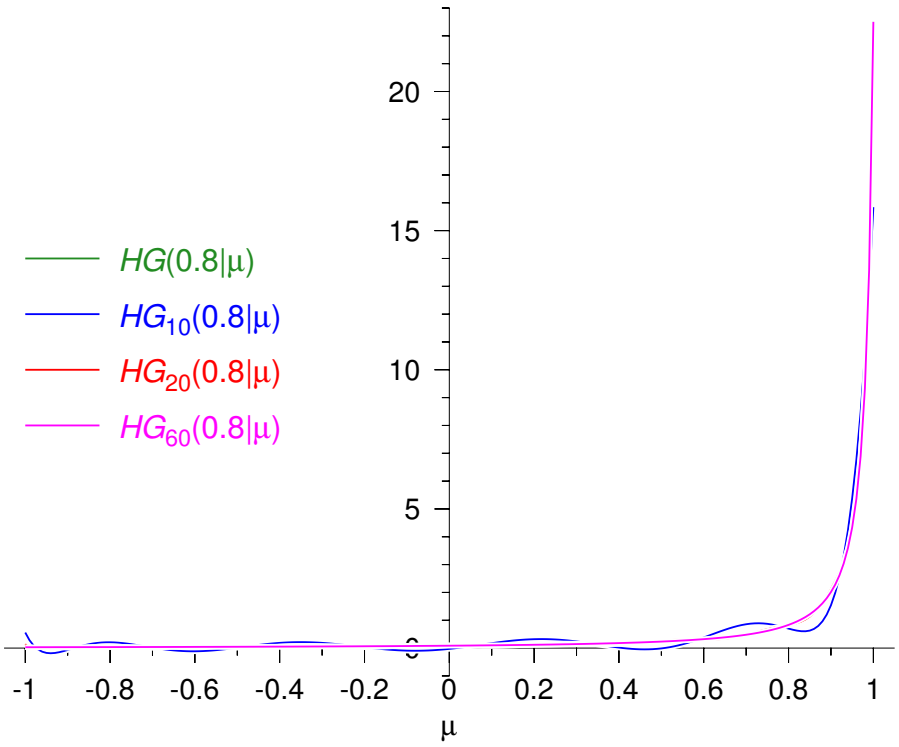


Figure 6.2: *The real Henyey-Greenstein scattering kernel $HG(0.8|\mu)$ and three approximations.*

N	ν_4	ν_3	ν_2	ν_1
10	$\times$	1.106641	1.454518	3.788647
20	1.014262	1.112870	1.454524	3.788647
60	1.014847	1.112870	1.454524	3.788647

Table 6.5: *Discrete eigenvalues for Test Case 4.*

self-made cross section library created by the NJOY code [MM94]. This code indeed flags errors due to the negative regions of the scattering function in case of the $N = 10$ and $N = 20$ approximation orders.

Another observation can be made on the three approximation orders: for $N = 10$, there are only three discrete eigenvalues, while for $N = 20$ and $N = 60$ there are four. Just to give an idea of the change of these eigenvalues as a function of N, we show them in Table 6.5. There are a few things one can notice in this table: (a) the largest eigenvalue doesn't change much (beyond the seventh significant digit), (b) the smallest eigenvalue seems to be "pushed out" of the continuum range $[-1, +1]$ and (c) the eigenvalues in between change relative to their magnitude, i.e. smaller eigenvalues change more than larger eigenvalues.

As in the previous test case, we compare the (mesh averaged) scalar flux at the left boundary, at a quarter width and at midpoint of the single cell. Results and relative errors with respect to MCNPX results are presented in Table 6.6. Note that we have calculated relative errors compared to MCNPX results even when we know they are definitely wrong for $N = 10$ (due to the faulty renormalisation in NJOY). Figure 6.3 shows the scalar flux for the case $N = 10$ and the faulty results of MCNPX are eminent.

Test Case 5: effect of near-singular eigenvalue

In our final test case, we study the effect of the presence of a near-singular discrete eigenvalue. Suppose that our code would not be able to detect, let alone calculate accurately, a near-singular eigenvalue. What would be the effect on the final solution, i.e. the boundary sources?

The problem settings are the following: a single-cell slab of width $a = 1$, $\Sigma_t = 1$, $c = 0.80$ and a fifth order approximation of the Henyey-Greenstein scattering kernel with free parameter $g = 0.485$. For these cell characteristics, we know from Chapter 4 that there are two discrete eigenvalues

$$\nu_1 \approx 1.82492 \quad \nu_2 \approx 1 + 1.645221 \times 10^{-15}$$

$N = 10$						
Code	Left		Quarter		Mid	
CASE, $M = 5$	1.05282	10.8%	1.40216	2.19%	1.47978	4.61%
CASE, $M = 30$	1.04370	9.83%	1.40351	2.10%	1.48096	4.53%
BLUE	1.04395	9.85%	1.40441	2.04%	1.48191	4.47%
MCNPX[a]	0.95032	ref.	1.43361	ref.	1.55125	ref.
$N = 20$						
Code	Left		Quarter		Mid	
CASE, $M = 5$	1.05309	0.87%	1.40200	0.12%	1.47967	0.10%
CASE, $M = 30$	1.04418	0.01%	1.40328	0.03%	1.48078	0.02%
BLUE	1.04426	0.02%	1.40415	0.03%	1.48174	0.04%
MCNPX[b]	1.04404	ref.	1.40368	ref.	1.48109	ref.
$N = 60$						
Code	Left		Quarter		Mid	
CASE, $M = 5$	1.05308	0.85%	1.40200	0.10%	1.47967	0.08%
CASE, $M = 30$	1.04419	0.002%	1.40328	0.01%	1.48078	0.004%
BLUE[c]	1.04428	0.01%	1.40415	0.05%	1.48175	0.06%
MCNPX	1.04417	ref.	1.40345	ref.	1.48084	ref.

[a] Note that these are erroneous results due to negative values in the scattering function.

[b] Note that these should be almost correct (one small negative dip in the scattering function).

[c] BLUE crashed when requiring $N = 60$, the results reported here are for $N = 30$.

Table 6.6: *Scalar fluxes for Test Case 4.*

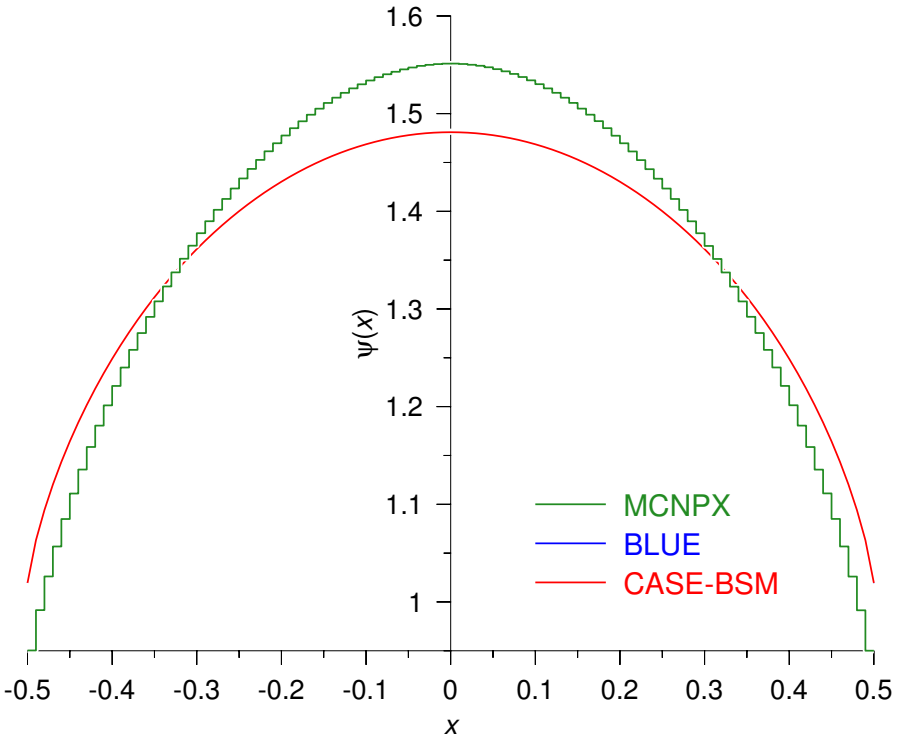


Figure 6.3: *Scalar flux for Test Case 4,* $N = 10$.

with normalisation factors

$$N_1 \approx 0.45826 \quad N_2 \approx 3.07549 \times 10^{+11}.$$

We perform two calculations using our code (for a BSM order of 30): (1) the normal calculation and (2) a crippled calculation where we artificially removed the near-singular root.

We make two types of comparisons: the first one is based on the discretized version of the boundary sources method, i.e. we do perturbation analysis on the linear system of Equation (6.17). The second comparison is based on the final result: the (mesh averaged) scalar flux at the left boundary, quarter width and midpoint as before.

Let us denote by A,f and b respectively the BSM matrix, the boundary sources vector and the right-hand-side for the "normal" problem and by A', f' and b' the same quantities for the crippled version without near-singular eigenvalue. We could consider A' and b' to be a small perturbations of respectively A and b. Let us apply normwise perturbation theory [Hig02, Chapter 7] to this problem. Suppose $\|A - A'\| \leq \epsilon \|A\|$ for a certain ϵ and let $\kappa(A)$ be the matrix condition number. We then have

$$\frac{\|f - f'\|}{\|f\|} \leq \frac{2\epsilon\kappa(A)}{1 - \epsilon\kappa(A)} \tag{6.21}$$

which gives us an upper bound on the normwise perturbation of the solution vector to a perturbation of both the system matrix as well as the right-hand-side. In our test case, we found the following numerical results:

$$\kappa(A) \approx 4337,\ \|A - A'\| \approx 1.97 \times 10^{-09},\ \|f\| \approx 0.716,\ \|f - f'\| \approx 1.51 \times 10^{-10},$$

so the left-hand-side and right-hand-side of the inequality (6.21) are given by

$$\frac{\|f - f'\|}{\|f\|} \approx 4.53 \times 10^{-11}$$

and

$$\frac{2\epsilon\kappa(A)}{1 - \epsilon\kappa(A)} \approx 2.76 \times 10^{-08}$$

and the bound is respected. For completeness, Figure 6.4 shows the (correct) boundary sources vector f for the cell while Figure 6.5 shows the norm of the difference $\|f - f'\|$ between f and f'.

To conclude this example, we show the comparison between MCNPX, BLUE and our code for the scalar flux at three positions in the cell as before. Table 6.7 summarizes these results.

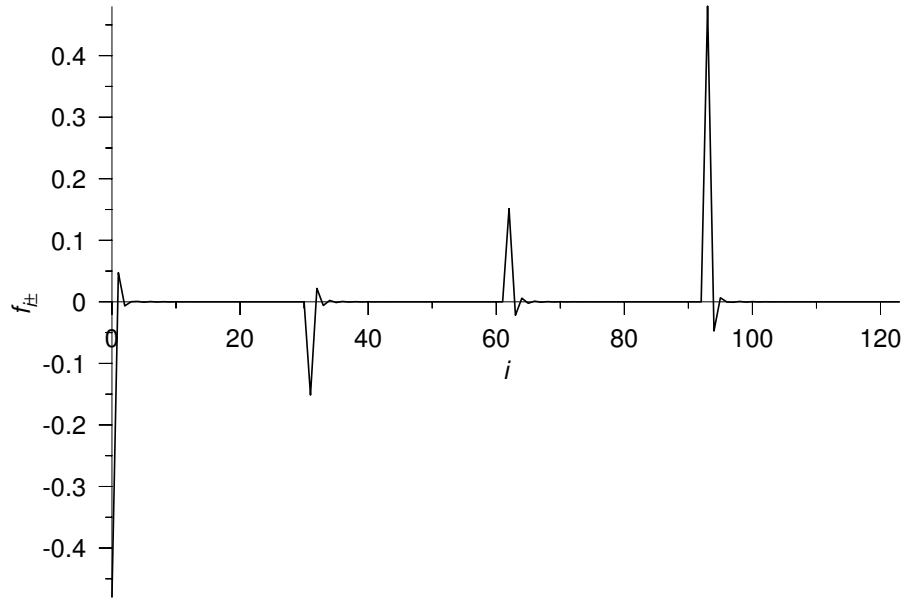


Figure 6.4: *Boundary sources vector for Test Case 5.*

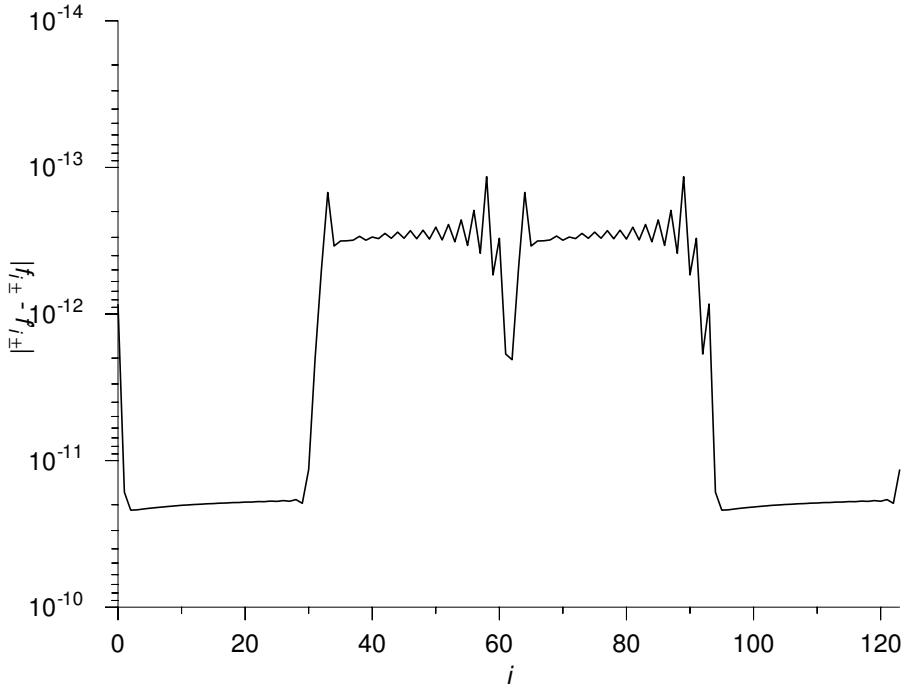


Figure 6.5: *Element-wise distance between f and f' for Test Case 5.*

Code	Left		Quarter		Mid	
CASE, $N = 30$[a]	0.85944	0.003%	1.21720	0.004%	1.29729	0.002%
CASE, $N = 30$[b]	0.85944	0.003%	1.21720	0.004%	1.29729	0.002%
BLUE	0.85909	0.04%	1.21705	0.008%	1.29715	0.01%
MCNPX	0.85941	ref.	1.21715	ref.	1.29731	ref.

[a] normal version, near-singular root included
[b] crippled version, near-singular root ignored

Table 6.7: *Scalar fluxes for Test Case 5.*

6.6 Evaluation of the method

From the results of the above five test cases, it is clear that the code certainly can serve as benchmark code. Its results are very accurate compared to Monte Carlo results from MCNPX. This is both the case for the reflective boundary conditions where no approximation is made with respect to the physics as for the vacuum boundary conditions where we use the full range Legendre polynomials, while from the physical point of view, it would be better to use the half-range (Yvon's) polynomials.

On the complexity of the code, one must admit it is quite complex given the extensive numerical quadrature needed. There is still a lot of room for optimisation of the code by studying specialised quadrature routines using appropriate weight functions. However, this was out of the scope of this work. For small orders of the Boundary Sources Method (say up to order 10), the cost remains acceptable. This order should definitely suffice for the coupling with interface-current nodal codes as the one-dimensional solver. For benchmark purposes, higher orders can be used but in this type of calculations the higher cost is justified by the final result wanted, a benchmark result.

In Chapter 8, a more detailed overview of the code is given, together with a profiling run to analyse the most costly parts of the code and hence the most likely to spend optimisation effort on.

CHAPTER 7

The Henyey-Greenstein scattering kernel

SUMMARY– We study the Henyey-Greenstein scattering kernel, often used in radiative transfer applications. The exact Legendre coefficients are derived and a convergence criterion is obtained on absolute and relative error. Finally, for some combinations of g and c, we calculate the discrete eigenvalues.

7.1 Definition

The Henyey-Greenstein scattering kernel [HG41] is a scattering kernel often used in radiative transfer applications. The kernel is defined as

$$HG(\mu|g) = \frac{1-g^2}{2\left(1-2\mu g+g^2\right)^{3/2}} \tag{7.1}$$

where g is a free parameter used to fit the kernel to experimental data. This parameter usually takes values from $g = -1$ (extremely backward scattering) over $g = 0$ (isotropic scattering) to $g = +1$ (extremely forward peaked scattering). Figure 7.1 shows this function for different values of the parameter g. From Equation (7.1) one can see that the kernel is mirror-symmetric around the y-axis for positive/negative values of g. Therefore, in the remainder of this chapter, we assume $g \in [0, +1[$. All results hold for the negative range, except for the location of the extrema (which should be mirrored around the y-axis for different signs of g).

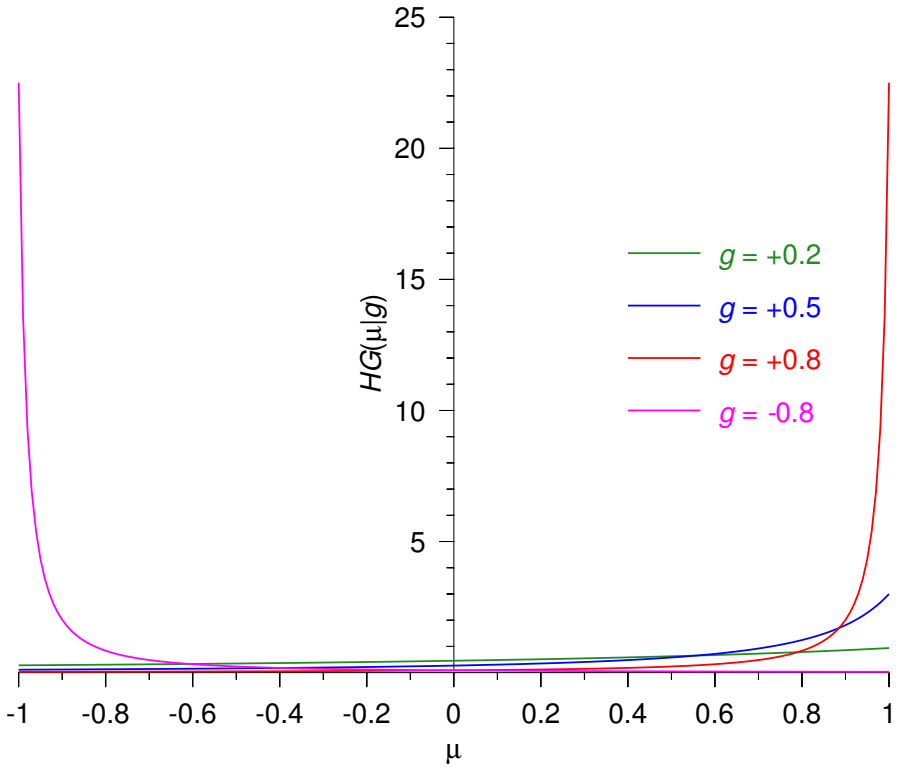


Figure 7.1: *The Henyey-Greenstein scattering kernel for different values of the parameter g.*

7.2 The Legendre approximation

In order to use this scattering kernel, we need to calculate its Legendre approximation. This approximation is of the form

$$HG_N(\mu|g) = \sum_{l=0}^{N} \frac{2l+1}{2} f_l P_l(\mu) \tag{7.2}$$

with

$$f_l = \int_{-1}^{+1} HG(\mu|g) P_l(\mu) d\mu \tag{7.3}$$

where $P_l(\mu)$ is the Legendre polynomial of degree l.

The Legendre moments

Let us first see if we can find a closed expression for the Legendre expansion coefficients f_l.

Theorem 7.1. *The Legendre expansion coefficients f_l of $HG(\mu|g)$ are given by $f_l = g^l$.*

Proof. Given the definition of the generating function for the Legendre polynomials [see for example San59]:

$$\frac{1}{\sqrt{1-2ux+u^2}} = \sum_{k=0}^{\infty} P(k,x) u^k \tag{7.4}$$

we can rewrite the definition of the Henyey-Greenstein scattering kernel (7.1) as

$$HG(\mu, g) = \frac{1}{2\sqrt{1-2g\mu+g^2}} + 2g \frac{\partial}{\partial g} \left(\frac{1}{2\sqrt{1-2g\mu+g^2}} \right) \tag{7.5}$$

and by using Equation (7.4) twice:

$$HG(\mu, g) = \sum_{k=0}^{\infty} \frac{1}{2} g^k P(k,\mu) + g \sum_{k=0}^{\infty} k g^{k-1} P(k,\mu) \tag{7.6}$$

which we can combine to

$$HG(\mu, g) = \sum_{k=0}^{\infty} \frac{2k+1}{2} g^k P(k,\mu) \tag{7.7}$$

and hence we see $f_l = g^l$. ■

This result gives us already some idea on the convergence properties of the approximation. When $g \ll 1$, g^l will decrease very rapidly and thus the contribution of the higher order terms will be very small. However, when g approaches $g = 1$, g^l decreases much slower and higher order terms will be needed to accurately approximate the original kernel.

The approximation error

We first establish a simple minimum property on the kernel itself:

Property 7.1. *For values of $g \in [0, 1[$ and $\mu \in [-1, +1]$, $HG(\mu|g)$ is minimal for $\mu = -1$ and its value there is $HG(-1|g) = \frac{1-g}{2(1+g)^2}$.*

Proof. Given the fact that $HG(\mu|g)$ is always strictly positive (and the numerator and denominator are also positive), this function is minimal when the numerator is minimal and the denominator is maximal. The numerator is no function of μ so it has no influence. The denominator is maximal when $1 - 2\mu g + g^2$ is maximal, hence for $\mu = -1$. The value of $HG(-1|g)$ is easily verified.

■

The function $HG(\mu|g)$ is continuous and hence the series $HG_N(\mu|g)$ converges to $HG(\mu|g)$ for $N \to \infty$. We define the approximation error as

$$\begin{aligned} \epsilon_N(\mu|g) &= HG(\mu|g) - HG_N(\mu|g) \\ &= \sum_{k=0}^{\infty} \frac{2k+1}{2} g^k P(k, \mu) - \sum_{k=0}^{N} \frac{2k+1}{2} g^k P(k, \mu) \\ &= \sum_{k=N+1}^{\infty} \frac{2k+1}{2} g^k P(k, \mu) \end{aligned} \tag{7.8}$$

Property 7.2. *The absolute value of the approximation error $|\epsilon_N(\mu|g)|$ can be bounded by a function of g and N only and is maximal for $\mu = +1$.*

Proof.

$$\begin{aligned} |\epsilon_N(\mu|g)| &= |\sum_{l=N+1}^{\infty} \frac{2l+1}{2} g^l P_l(\mu)| \\ &\leq \sum_{l=N+1}^{\infty} \frac{2l+1}{2} g^l |P_l(\mu)| \\ &\leq \sum_{l=N+1}^{\infty} \frac{2l+1}{2} g^l. \end{aligned}$$

The latter can be written under closed form as

$$\sum_{l=N+1}^{\infty} \frac{2l+1}{2} g^l = \frac{g^{N+1}\left[3-g+2N\left(1-g\right)\right]}{2\left(1-g\right)^2}.$$

So we have

$$|\epsilon_N(\mu|g)| \leq \frac{g^{N+1}\left[3-g+2N\left(1-g\right)\right]}{2\left(1-g\right)^2} \tag{7.9}$$

as an upper bound for $|\epsilon_N(\mu|g)|$. This bound is strict in the sense that it is reached for $\mu = +1$. Hence, $|\epsilon_N(\mu|g)|$ is maximal for $\mu = +1$.

■

7.3 Applying constraints on the approximation

In this section we shall derive minimal approximation orders in function of three constraints put on the approximation. The constraints under investigation are non-negativity of the approximation (important in Monte-Carlo simulations where the kernel is used in combination with probability density functions), a required maximum absolute error of the approximation and a required maximum relative error of the approximation.

Non-negativity of $HG_N(\mu|g)$

Non-negativity of the approximation requires

$$HG_N(\mu|g) \geq 0, \qquad \forall \mu \in [-1,+1], g \in [0,+1[\tag{7.10}$$

Theorem 7.2. *A sufficient (but not necessary condition) for non-negativity of $HG_N(\mu|g)$ is that the maximum absolute error is smaller than the minimal function value of $HG(\mu|g)$.*

Proof. Since $HG(\mu|g)$ itself is non-negative in the domain of interest, the smallest distance between the kernel and the μ-axis is its minimal value. If the maximum error of the approximation is indeed smaller than this distance, the approximation can never dive under the μ-axis and hence the approximation is non-negative in the interval. The condition (or bound) is sufficient but not necessary (hence the bound is not strict) because the minimum value of $HG(\mu|g)$ and the maximum value of $\epsilon_N(\mu|g)$ can occur for different values of μ.

■

Mathematically, the condition of Theorem 7.2 writes

$$g^{N+1}\left[3-g+2N\left(1-g\right)\right] \leq \frac{\left(1-g\right)\left(1-g\right)^2}{\left(1+g\right)^2}. \tag{7.11}$$

An explicit solution of Equation (7.11) for N is given by

$$N \geq \frac{1}{\ln\left(g\right)}\left\{W_{-1}\left(\frac{e^{\frac{-\ln(g)(g+1)}{2(g-1)}}\left(g-1\right)^2\ln\left(g\right)}{2\left(g+1\right)^2}\right)\right\}+\frac{3-g}{2\left(g-1\right)} \tag{7.12}$$

where $W_{-1}(x)$ is a branch of the Lambert W_k-function (see Appendix B for a definition of $W(x)$ and an algorithm to calculate $W_{-1}(x)$) giving positive values for N. Evaluating this equation does not necessarily lead to values for N in the natural numbers. Therefore, we round the result upwards (`ceil` function). Figure 7.2 plots the required order N (already rounded) versus g to have a non-negative approximation $HG_N(\mu|g)$. Table 7.1 gives some approximation orders N for different values of g and a comparison to the real orders, computed by means of MAPLE, needed for non-negativity. The $\times$ symbol denotes that the order was too high to calculate with MAPLE in a reasonable time.

g	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99
N	1	1	2	4	7	11	20	39	107	1877
N exact	1	1	1	2	2	6	10	20	50	$\times$

Table 7.1: *Order of approximation N such that $HG_N(\mu|g) \geq 0$.*

A required absolute error between real kernel and approximation

A second possible constraint is the maximum absolute error between the real scattering kernel and its Legendre approximation. Suppose an absolute error of $\delta > 0$ is allowed, what would the minimum order of approximation be? Property 7.2 gives us the necessary condition:

$$|\epsilon_N(\mu|g)| \leq \delta \quad \forall \mu \in [-1,+1], g \in [0,1[\tag{7.13}$$

and thus

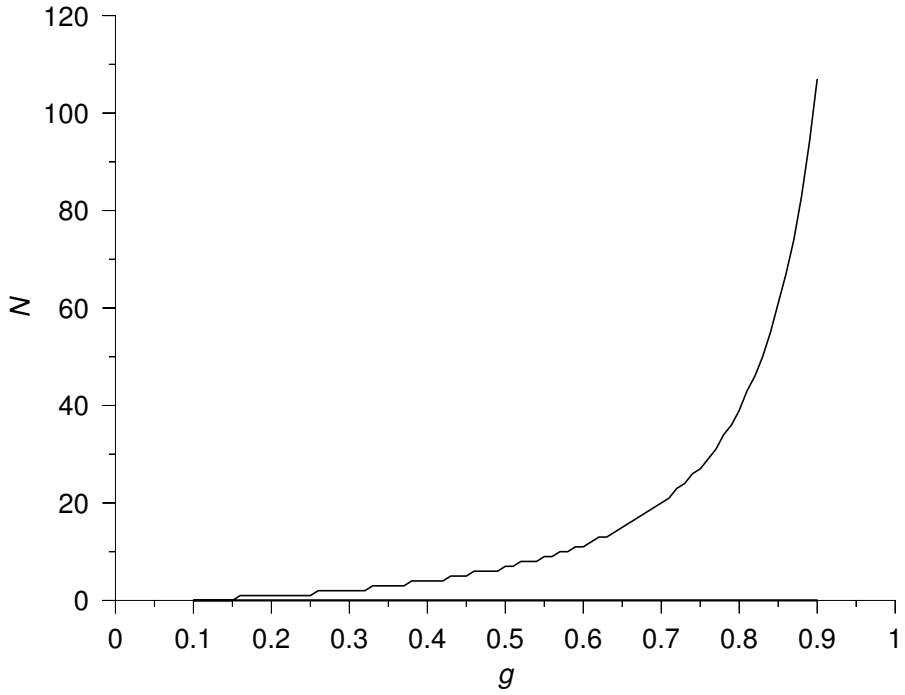


Figure 7.2: *Order of approximation N such that $HG_N(\mu|g) \geq 0$.*

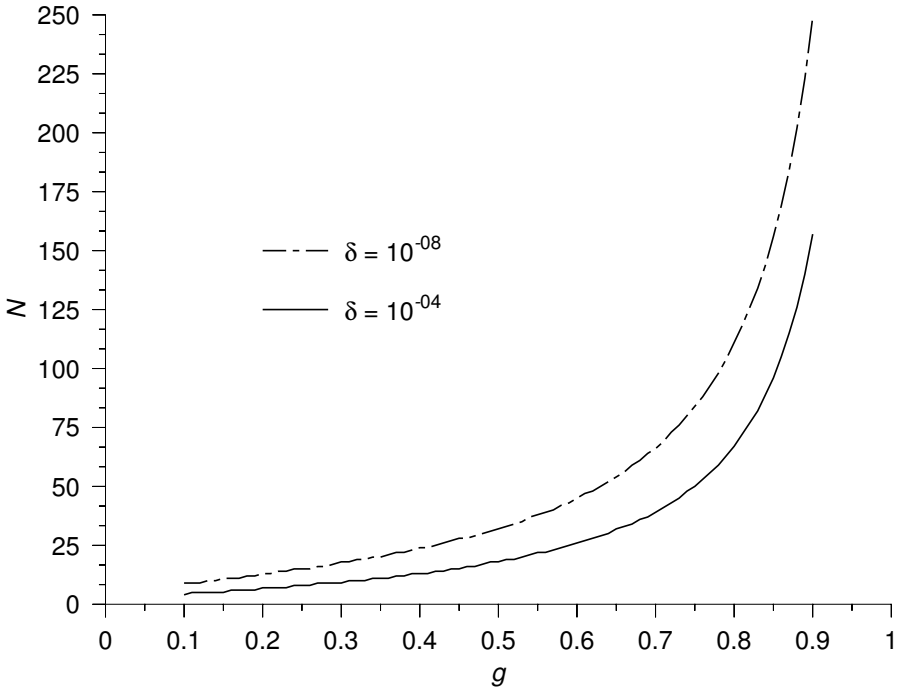


Figure 7.3: *Order of approximation N such that $|\epsilon_N(\mu|g)| \leq \delta(10^{-04}, 10^{-08})$.*

$$\frac{g^{N+1}\left[3-g+2N\left(1-g\right)\right]}{2\left(1-g\right)^2} \leq \delta. \tag{7.14}$$

Solving inequality (7.14) again results in an expression containing Lambert's W_{-1}-function:

$$N \geq \frac{1}{\ln\left(g\right)}\left\{W_{-1}\left(-e^{\frac{\ln(g)(1+g)}{2(1-g)}}\left(g-1\right)\ln\left(g\right)\delta\right)\right\} + \frac{3-g}{2\left(g-1\right)}. \tag{7.15}$$

Figure 7.3 shows the evolution of N as a function of g for a required absolute error of $\delta = 10^{-04}$ and $\delta = 10^{-08}$. Table 7.2 explicitly mentions some results for the same requirements. For a more complete view, Figure 7.4 provides a contour plot where the absolute error is indicated as a function of g and N. Because the bound (7.9) we use on $|\epsilon_N(\mu|g)|$ is strict (the value is reached for $\mu = 1$), these values for N are exact.

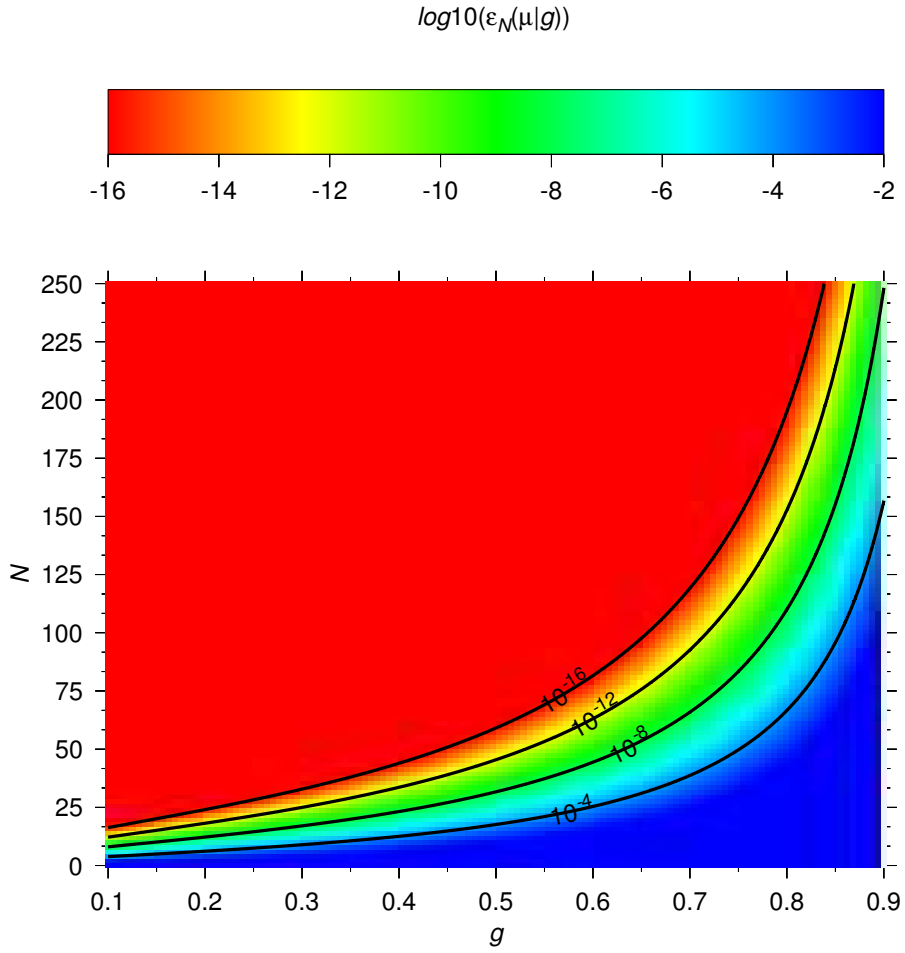


Figure 7.4: *Order of approximation N such that $|\epsilon_N(\mu|g)| \leq \delta$.*

	g	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99
$\delta = 10^{-04}$	N	4	7	9	13	18	26	39	67	157	2142
$\delta = 10^{-08}$	N	9	13	18	24	32	45	66	111	249	3093

Table 7.2: *Order of approximation N such that* $|\epsilon_N(\mu|g)| \leq \delta$.

Relative error between real kernel and approximation

A third constraint under investigation is the relative error between the real Henyey-Greenstein scattering kernel and its Legendre approximation. The relative error is defined as

$$\tilde{\epsilon}_N(\mu|g) = \frac{|\epsilon_N(\mu|g)|}{HG(\mu|g)}. \tag{7.16}$$

Since $HG(\mu|g)$ is strictly positive everywhere on $\mu \in [-1, +1], g \in [0, 1[$, the relative error (7.16) is defined in the complete interval of interest.

Property 7.3. *The minimum of the relative error* $\tilde{\epsilon}_N(\mu|g)$ *is zero in the interval of interest:*

$$\min_{-1 \leq \mu \leq +1} \tilde{\epsilon}_N(\mu|g) = 0.$$

Proof. The error $\epsilon_N(\mu|g)$ is the error between a continuous function and its Legendre least squares approximation. For a least-squares approximation using a complete system of orthogonal polynomials, it is known that the N-th order least-squares approximation of a continuous function interpolates this function in at least $N + 1$ points [Buy92, p.72]. The absolute error $\epsilon_N(\mu|g)$ has thus at least $N + 1$ zeros and since the Henyey-Greenstein kernel is continuous and strictly positive, the relative error $\tilde{\epsilon}_N(\mu|g)$ also has at least $N + 1$ zeros. Since the latter is also non-negative (absolute value), the minimum value is 0 and this value is reached in at least $N + 1$ points. ■

Property 7.4. *The maximum relative error is bound by a function of g and N only:*

$$\max_{-1 \leq \mu \leq +1} \tilde{\epsilon}_N(\mu|g) \leq \frac{g^{N+1} \left[3 - g + 2N\left(1 - g\right)\right] \left(1 + g\right)^2}{\left(g - 1\right)^3}.$$

Proof. The relative error is defined as

$$\tilde{\epsilon}_N(\mu|g) = \frac{|\epsilon_N(\mu|g)|}{HG(\mu|g)}$$

where the denominator and numerator are both positive. An upper bound for this fraction can be thus given by taking the maximum of the numerator and the minimum of the denominator leading to

$$\tilde{\epsilon}_N(\mu|g) \leq \frac{\max_{-1\leq\mu\leq+1} |\epsilon_N(\mu|g)|}{\min_{-1\leq\mu\leq+1} HG(\mu|g)}.$$

The maximum of the absolute error is given by Property (7.2) while the minimum of the $HG(\mu|g)$ value is given by Property (7.1) which allows us to write

$$\tilde{\epsilon}_N(\mu|g) \leq \frac{g^{N+1}\left[3-g+2N\left(1-g\right)\right]\left(1+g\right)^2}{\left(g-1\right)^3}$$

what was to be proved.

■

The previous bound is not strict. Indeed, the μ value corresponding to the minimum of $HG(\mu|g)$ ($\mu = -1$) is not necessarily a maximum of $|\epsilon_N(\mu|g)|$. Solving the inequality leads to the following inequality (7.17) for the degree N depending on g and the relative error δ requested. Figure 7.5 shows the evolution of N as a function of g for $\delta = 10^{-04}$ and $\delta = 10^{-04}$. Table 7.3 shows some explicit values for N. The orders indicated with $\hat{N}$ are exact results calculated using MAPLE. And again, for a complete view, a contour plot in Figure 7.6 is shown.

$$N \geq \frac{1}{\ln\left(g\right)}\left\{W_{-1}\left(e^{\frac{-\ln(g)(1+g)}{2(1-g)}}\frac{\left(g-1\right)^2\ln\left(g\right)\delta}{2\left(g+1\right)^2}\right)\right\}+\frac{3-g}{2\left(g-1\right)} \tag{7.17}$$

	g	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99
$\delta = 10^{-04}$	N	5	8	11	15	22	31	48	84	200	2833
	$\hat{N}$	4	6	9	12	17	24	34	57	136	×
$\delta = 10^{-08}$	N	9	14	19	26	35	50	75	127	291	3777
	$\hat{N}$	7	13	18	24	32	42	62	102	228	×

Table 7.3: *Order of approximation N such that $\tilde{\epsilon}_N(\mu|g) \leq \delta$.*

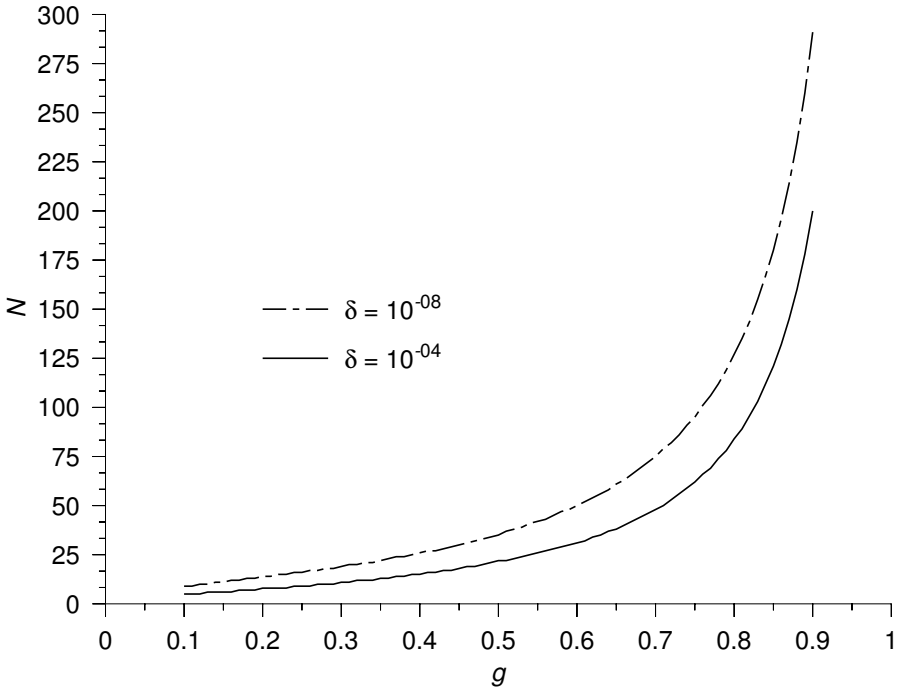


Figure 7.5: *Order of approximation N such that $\tilde{\epsilon}_N(\mu|g) \leq \delta(10^{-04}, 10^{-08})$.*

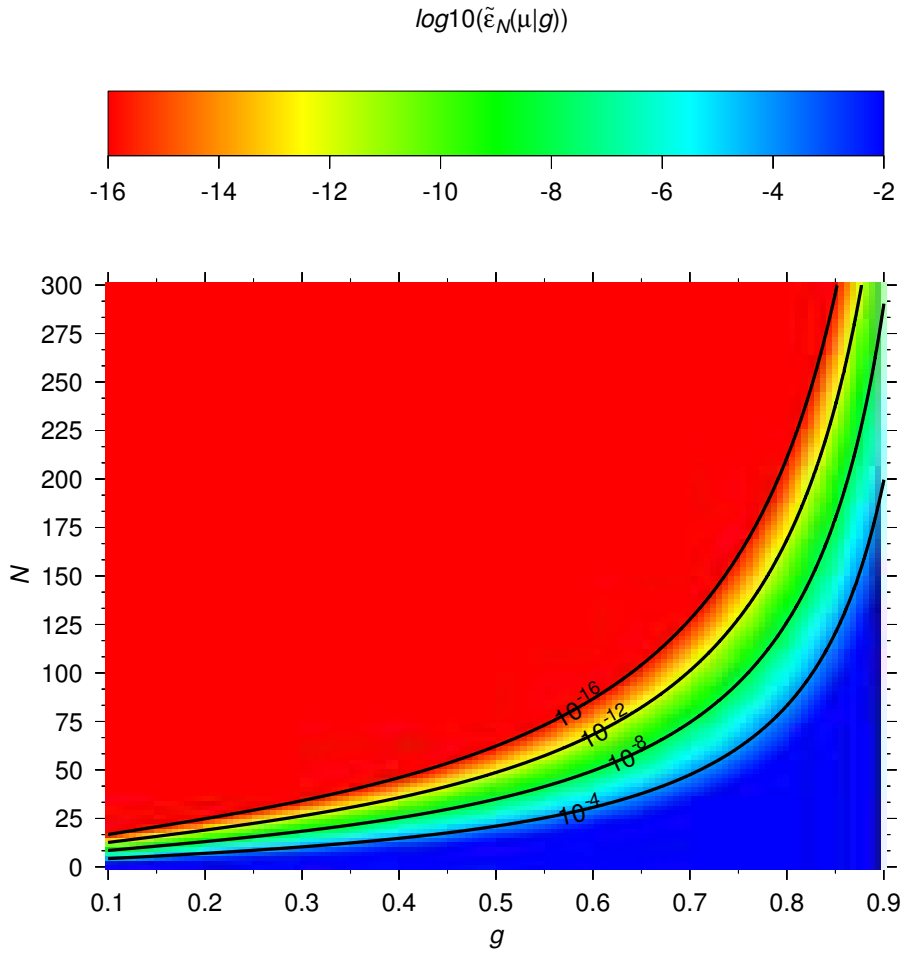


Figure 7.6: *Order of approximation N such that $\tilde{\epsilon}_N(\mu|g) \leq \delta$.*

7.4 The discrete eigenvalues

In this section, we study the discrete part of the eigenfunction expansion. First, we use our code to evaluate the number of discrete eigenvalues on a grid for the free parameter $g \in [0, 0.95]$ and the value of $c \in [0, 0.95]$. For both variables a grid spacing of 0.01 was used. We show graphically the number of positive discrete eigenvalues for three expansion orders: one based on the non-negativity requirement (Figure 7.7), the other two on the requirement of a relative error lower than 10^{-04} (Figure 7.8) and 10^{-08} (Figure 7.9). Taking the case for the relative error requirement of 10^{-08} as reference, we can produced two "diff" plots, showing the difference between the non-negativity requirement (Figure 7.10) and the reference case and another one for the difference between the relative error 10^{-04} requirement (Figure 7.11) and the reference case.

One can observe that there is a difference at certain points between the results for the non-negativity requirement and the reference case, but that there is no difference between the less restrictive relative error case and the reference case.

7.5 An example: $c = 0.90$, $g = 0.85$

Let us take as an example the case where the Henyey-Greenstein free parameter $g = 0.85$ and $c = 0.90$. As approximation criterion, we choose to have a relative error smaller than 10^{-04}, which leads to an approximation order of $N = 121$ according to our bounds.

The characteristic polynomial

The required order of approximation, $N = 121$, is too high to allow for building the characteristic polynomial by interpolation. The code uses the Newton-Raphson with Maehly routine as explained in Chapter 4 to locate the roots in the interval $\nu \in [0, 1]$. Figure 7.12 shows the characteristic polynomial. It has 5 roots in the interval, so the number of positive discrete eigenvalues is equal to 6, following Conjecture 2.2. The roots of the characteristic polynomial are listed in Table 7.4.

The discrete eigenvalues

From the previous section, we note that the characteristic equation $\Lambda(\nu)$ has 6 positive roots. Figure 7.13 shows the behaviour of $\Lambda(\nu)$. The inset is a

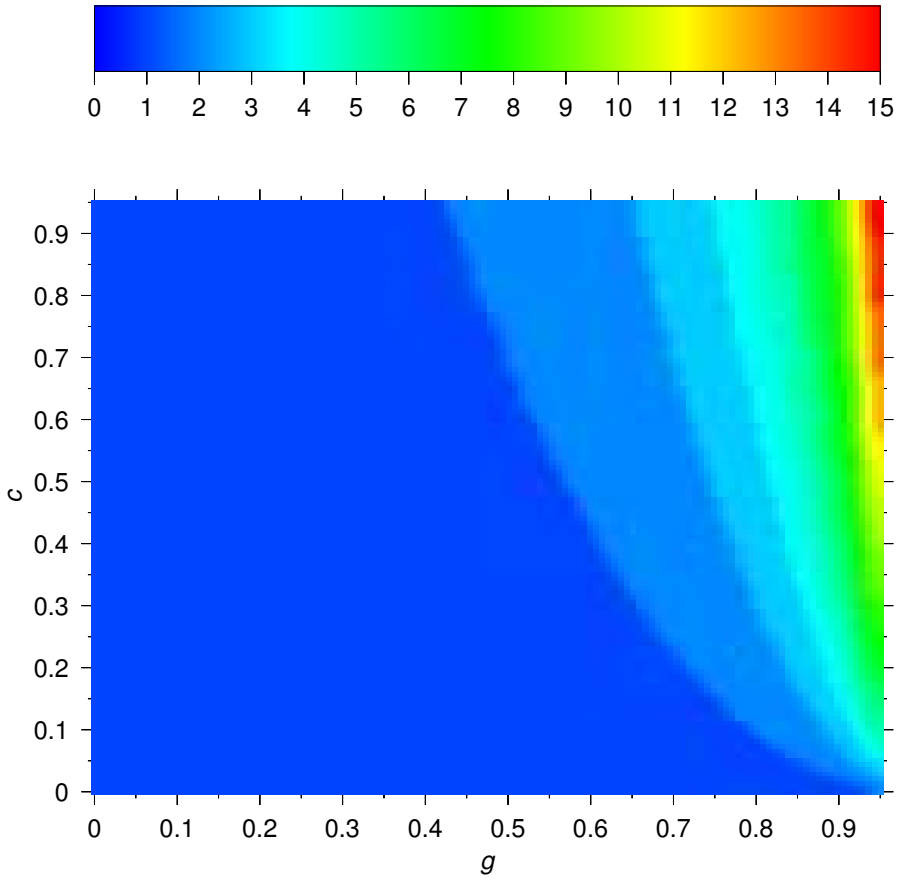


Figure 7.7: *Number of positive discrete eigenvalues for $HG(\mu|g)$, non-negativity requirement.*

index	root	index	root
1	4.789502×10^{-01}	4	9.858245×10^{-01}
2	7.891690×10^{-01}	5	9.988659×10^{-01}
3	9.336479×10^{-01}		

Table 7.4: *Roots of the characteristic polynomial $P_c(\nu)$ for $g = 0.85, c = 0.9, N = 121$ for $\nu \in [0, 1]$.*

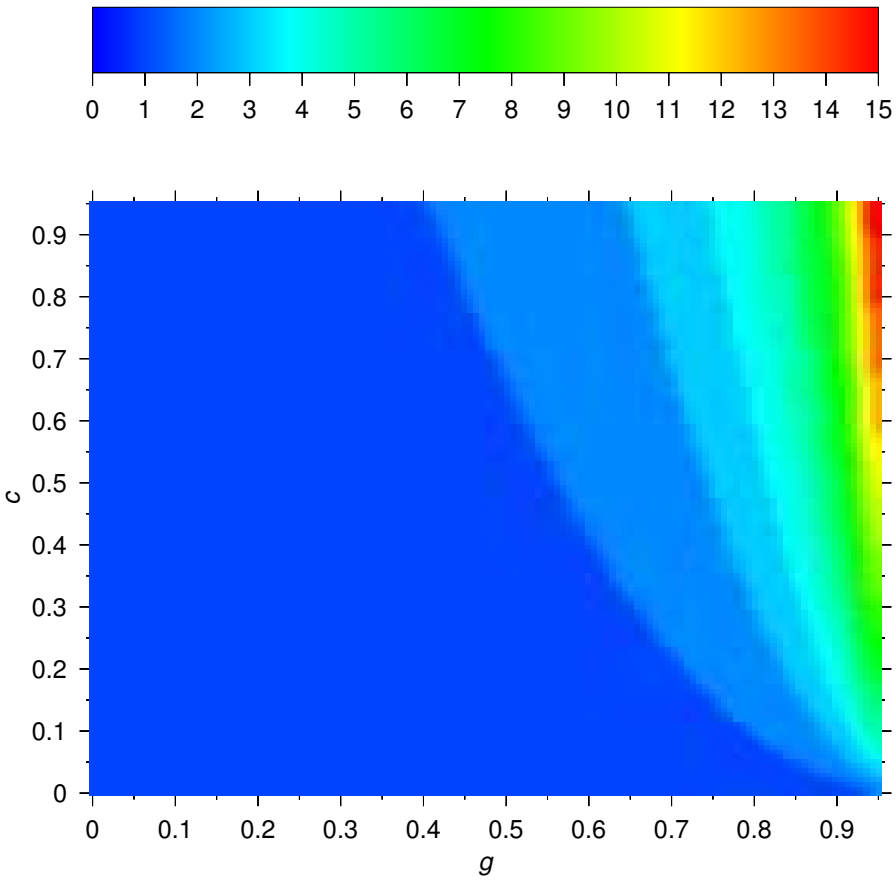


Figure 7.8: *Number of positive discrete eigenvalues for* $HG(\mu|g)$*, relative error* $< 10^{-04}$.

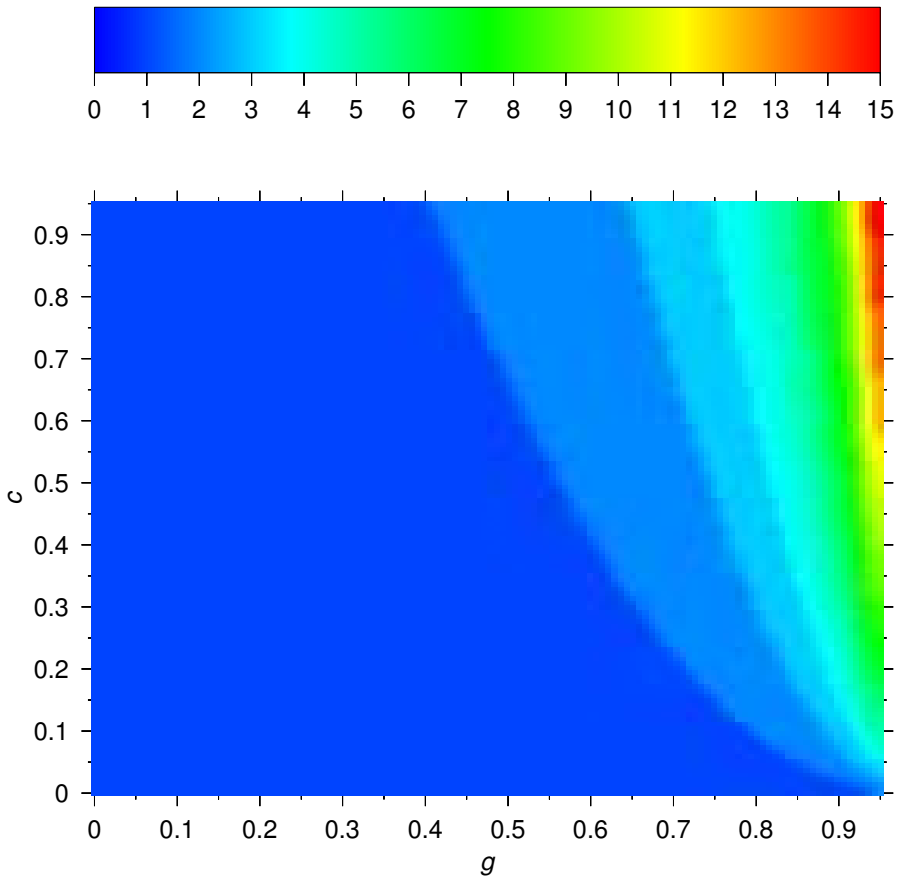


Figure 7.9: *Number of positive discrete eigenvalues for* $HG(\mu|g)$*, relative error* $< 10^{-08}$.

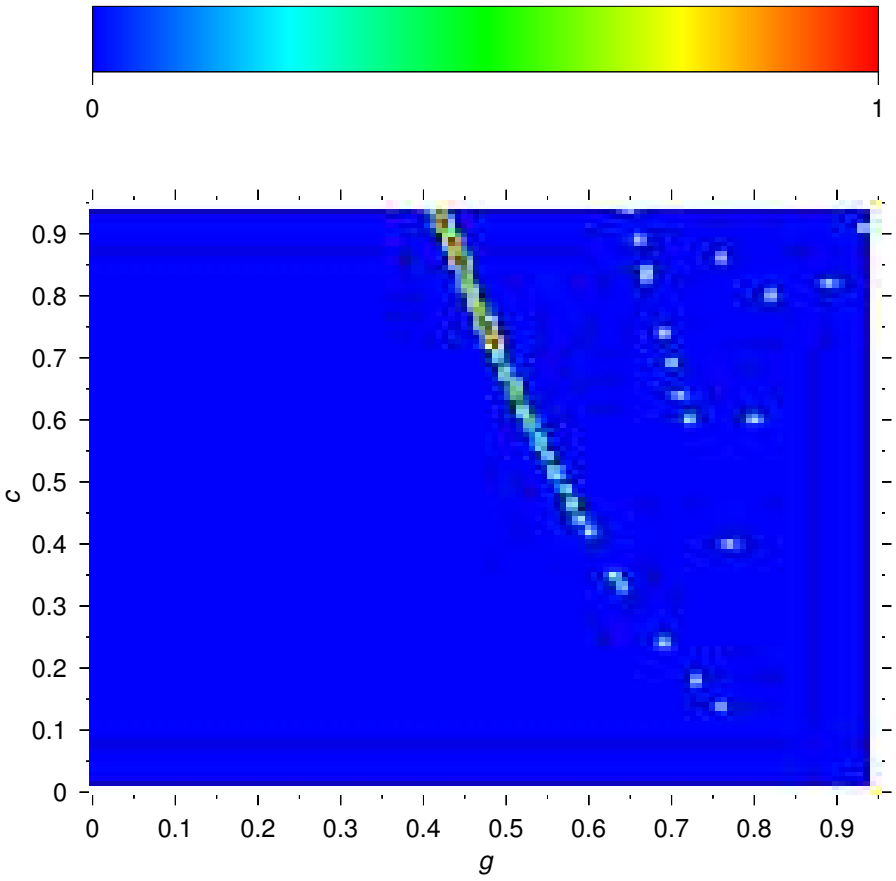


Figure 7.10: *Difference of the number of discrete eigenvalues for $HG(\mu|g)$, non-negativity vs relative error $< 10^{-08}$.*

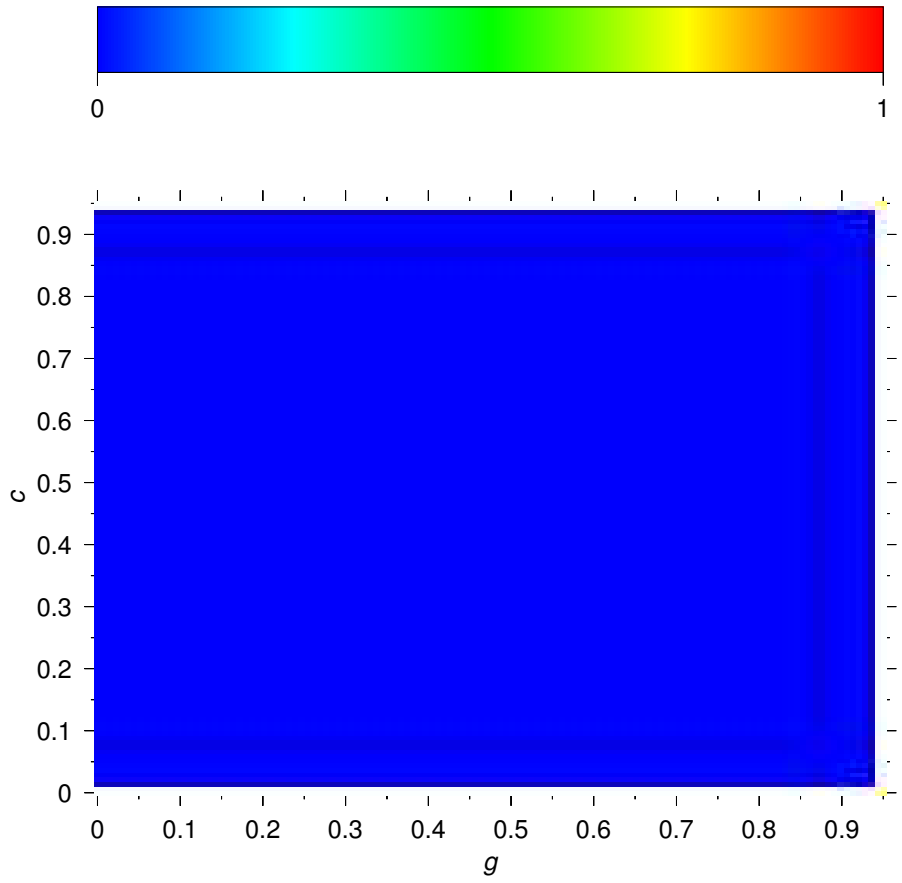


Figure 7.11: *Difference of the number of discrete eigenvalues for* $HG(\mu|g)$*, relative error* $< 10^{-04}$ *vs relative error* $< 10^{-08}$.

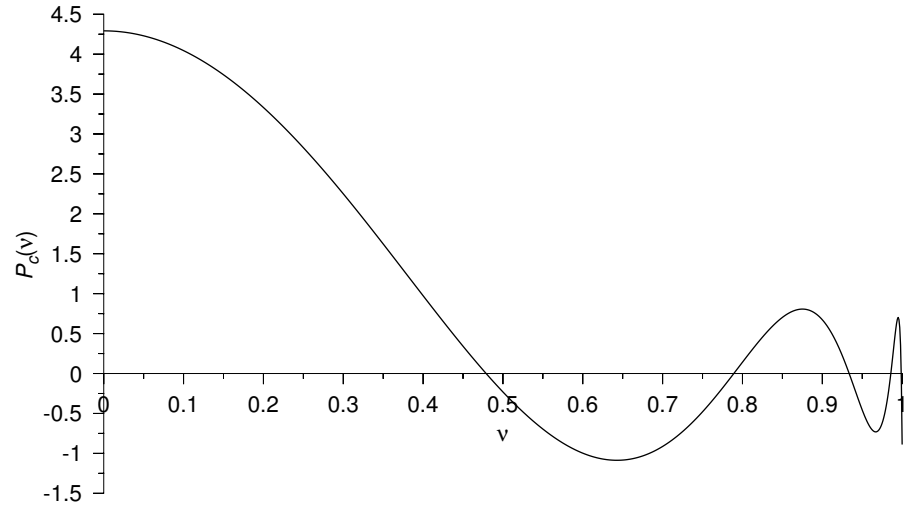


Figure 7.12: *Characteristic polynomial $P_c(\nu)$ for $g = 0.85, c = 0.9, N = 121$.*

index j	root ν_j	index j	root ν_j
1	4.2171063832	4	1.0920232305
2	1.7232646877	5	1.0218569159
3	1.2626821399	6	1.0004649869

Table 7.5: *Roots of the characteristic equation $\Lambda(\nu)$ for $g = 0.85, c = 0.9, N = 121$.*

zoom (y-axis) for $\nu = 1.1 \ldots 4.5$. The limit for $\nu \to +1$ goes to plus infinity. There are thus 3 roots between $\nu = 1$ and $\nu = 1.1$. Three other roots are more clearly visible. Again, this example confirms our conjecture that the maximum number of roots (obtained by means of the characteristic polynomial) is indeed reached. Table 7.5 summarizes the roots of the characteristic equation, hence the discrete eigenvalues. Following classical notation, the roots are counted from index $j = 1$ up to $j = 6$, where $j = 1$ denotes the largest root.

Relative contribution to the Green's function

It is of interest to see how each discrete eigenvalue contributes to the discrete part of the Green's function. Since in the Boundary Sources Method we work with angular moments of the Green's function, we shall compare the relative contributions of each discrete eigenvalue for two angular moments

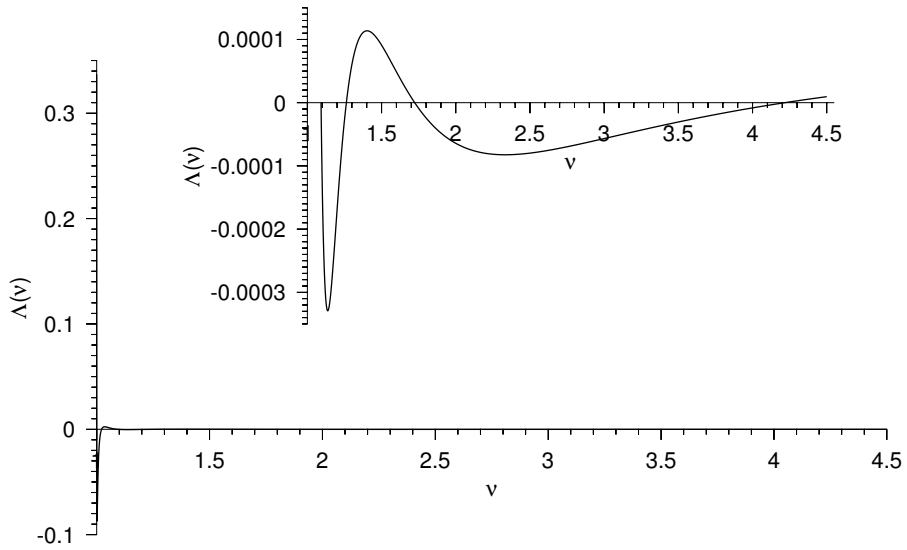


Figure 7.13: *Characteristic equation $\Lambda(\nu)$ for $g = 0.85, c = 0.9, N = 121$.*

index j	N_j	index j	N_j
1	$+5.4127154868 \times 10^{-01}$	4	$+1.0209793708 \times 10^{+01}$
2	$+2.1640618049 \times 10^{+00}$	5	$+2.4908404483 \times 10^{+01}$
3	$+4.8731363732 \times 10^{+00}$	6	$+2.0478584890 \times 10^{+02}$

Table 7.6: *Normalisation factors for the different discrete eigenvalues.*

$G_{0,0}(0|x)$ and $G_{10,04}(0|x)$.

The total discrete Green's function angular moments for this example are defined as (see Eq. (4.50))

$$G_{n,m}(x_0|x) = \sum_{j=0}^{5} \frac{\phi_n(\pm\nu_j)\phi_m(\pm\nu_j)}{N_j} e^{-\frac{|x-x_0|}{\pm\nu_j}} \tag{7.18}$$

where the sign of ν_j depends on the position of x compared to x_0. The discrete eigenvalues ν_j have been determined, the polynomials $\phi_k(\nu)$ are known. What remains are the normalisation factors. They are summarised in Table 7.6.

Figures 7.14 and 7.15 show the different contributions from the six discrete eigenvalues for the angular moments $G_{0,0}(0|x)$ and $G_{10,04}(0|x)$.

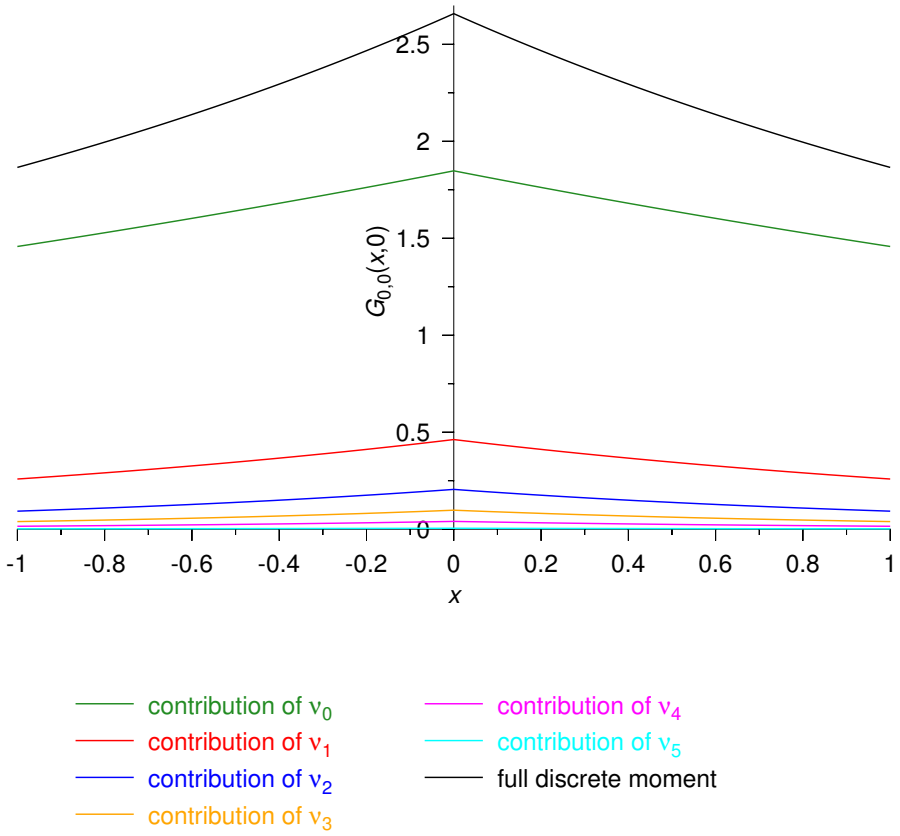


Figure 7.14: *Contribution of the six discrete eigenvalues to the Green's angular moment* $G_{0,0}(0|x)$.

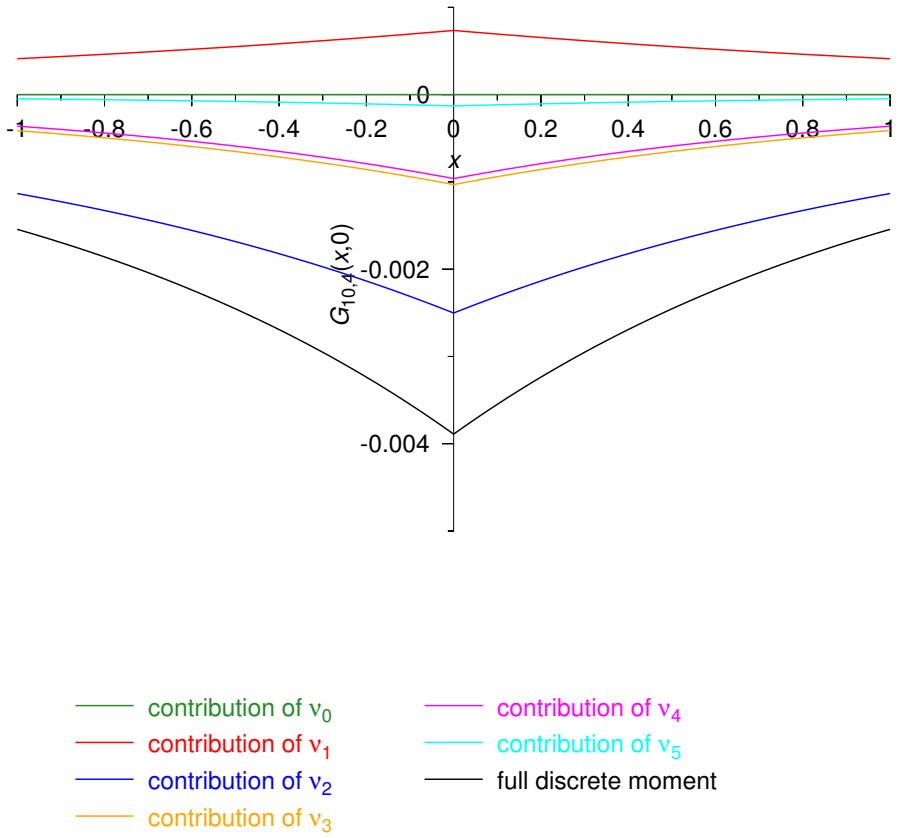


Figure 7.15: *Contribution of the six discrete eigenvalues to the Green's angular moment* $G_{10,04}(0|x)$.

CHAPTER 8

The CASE and CASE-BSM libraries

SUMMARY– In this chapter, we provide a high-level overview of all components (C++ classes) that are part of the CASE and CASE-BSM libraries. They are grouped in six different parts: special function classes, auxiliary function classes, nuclear data related classes, Green's function related classes, particular solution related classes and finally BSM related classes. For each, we shall describe its components and the main member functions.

8.1 Overview

The code is written in C++ extensively using its object-oriented approach and the data structures available in the Standard Library [Str97]. Two parts of the code are written by others. The first being the C++ linear algebra package LinAl [KD04] providing matrix/vector classes and an interface to the LAPACK library [ABB+99]. This code is freely available under the Lesser Gnu Public License (LGPL). The second part that is not self-written are the numerical quadrature packages QUADPACK [PdDKÜK83] and the double exponential [OM91] and sinc [SS84] quadrature methods. The code contains more than fifteen classes, reaching a total of more than 10000 lines of code. In the following, we shall discuss briefly the six different parts and the classes in each part.

8.2 Detailed structure

In this section we shall discuss in more detail the different parts of the CASE and CASE-BSM libraries. Classes are grouped in different functional groups

as can be seen in the overview graph in Figure 8.1 in which the ordinary arrows denote a "has"-relationship and the dotted line heads denote a "uses"-relationship.

The special functions

The first group are the special function routines. Both the Legendre polynomials and the Legendre functions of the second kind are implemented. Their implementation is based on different references [MOS66; OS83; ZJ96; Tho97; GS97; SG99]. For the spatial integral we needed the Gegenbauer polynomials as well. Their implementation uses the forward recurrence relation for fixed λ, see Chapter 5. In the development of the series expansion for the Legendre function of the second kind for arguments of the form $1 + \delta$, we needed the $\psi(n)$ function. Finally, in our analysis of the Henyey-Greenstein scattering kernel the Lambert W-function needs to be evaluated. As explained in Appendix B, its evaluation is based on the algorithm presented in [CGH$^+$96]. Finally, the PhiPoly routines contain forward and backward evaluation routines for the transport polynomials described in Chapter 3. The backward evaluation scheme is described in more detail in Appendix C. All these are implemented as functions, not as classes since they are not really needed as objects.

The auxiliary classes

The second set of routines are the auxiliary routines. These are the classes Error, Poly, Functor, NLESolver and Integrator. The Error class is a simple error handling class in which there is a distinction made between critical errors (the code stops), warnings, notices and information messages. During compilation, the user can select the "noisiness" of the code, i.e. whether warnings should be issued and notices ignored and so on.

A Functor is a "function with a state". In C++ terms, this refers to a pointer to a member function of an object of a certain class. We need functors to have the non-C++ quadrature routines and the non-linear equation solver routines working on a member function of our class (for example the `operator()` member function of the characteristic equation class).

The Poly class handles ordinary polynomials. They can be constructed, evaluated, differentiated and their roots can be calculated using the companion matrix method as described in Chapter 4.

Non-linear equations can be solved using the NLESolver class. The class constructor takes a pointer to a non-linear equation (a simple function or a member function of another object by using a Functor) together with a

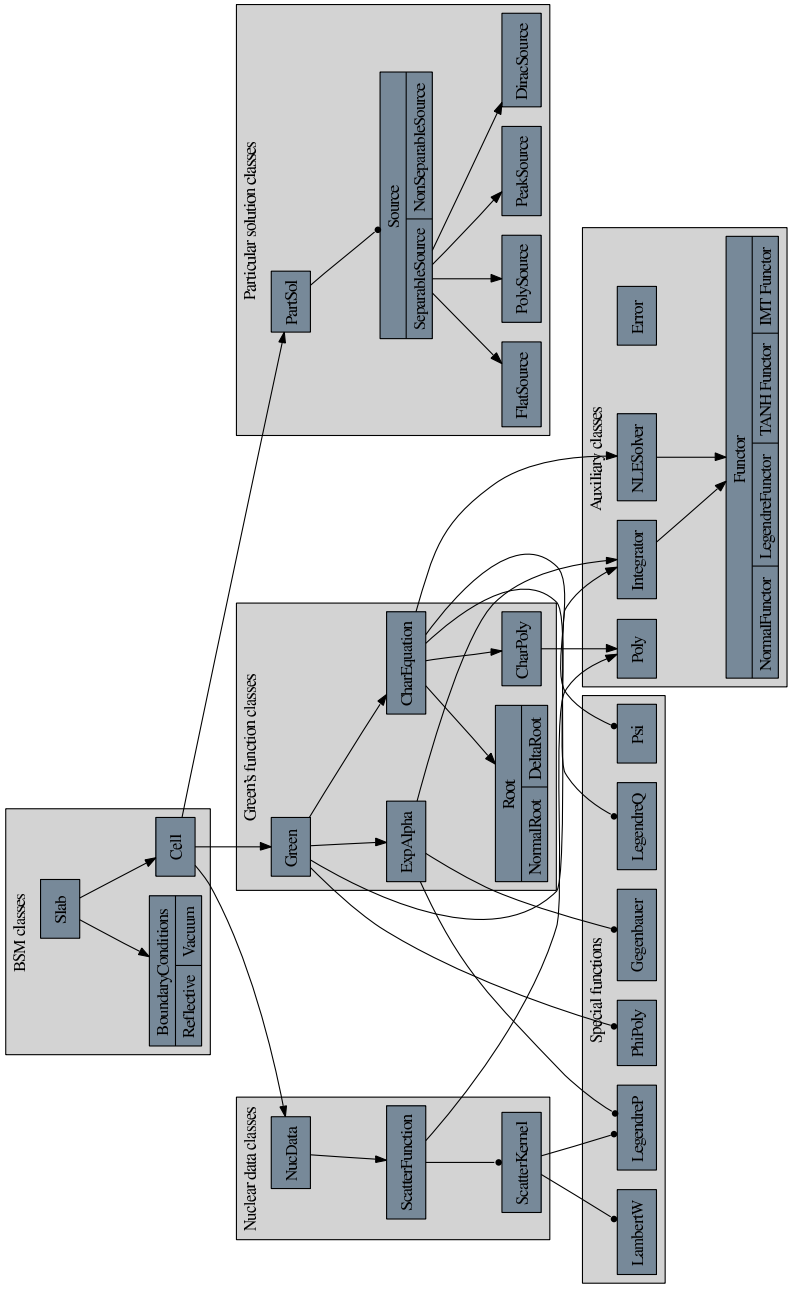


Figure 8.1: *Class overview for the* CASE *and* CASE-BSM *libraries.*

requested tolerance on the relative error. Also the left and right boundary of the search interval must be given. Three solver algorithms are implemented: the classical bisection method, the logarithmic bisection method and the Van Wijngaarden-Dekker-Brent method based on [PTVF96; Hig02].

The numerical quadrature is done using the Integrator class. Its constructor takes a pointer to the function to be integrated, together with left and right integration boundary and an absolute and relative tolerance. By using the functor approach, different variable transforms can be applied to the user-supplied function during the integration process: the TANH-transform and the IMT-transform. The quadrature routines implemented are the classic trapezoidal rule and Romberg (extrapolation to the limit) integration [DR84]. Interfaces are provided to the DQAGSE and DQAGE routines from the QUADPACK [PdDKÜK83] library. Also interfaces for the double-exponential integrator INTDE [OM91] as well as for the sinc quadrature routine INTHP [SS84] are implemented.

The nuclear data classes

The complete set of nuclear data needed in this code is the total cross section Σ_t, the number of secondaries per collision c and the anisotropic scattering function. The latter can be generated automatically (by numerical quadrature) from a scattering kernel. The ScatterKernel class is a parent class with at this moment two child classes: the Henyey-Greenstein scattering kernel and the binomial scattering law. Feeding such a scattering kernel to the ScatterFunction class together with the requested approximation order N, the vector of scattering coefficients f_l is generated. This scattering function is combined with Σ_t and c in the NucData class. The ScatterKernel class can easily be extended with other child classes for different scattering kernels.

The Green's kernel classes

This set of classes form the main matter of the code. It consists of CharPoly, CharEquation, Root, ExpAlpha and finally Green. The CharPoly class contains routines to handle the characteristic polynomial $P_c(\nu)$ which is needed in the calculation of the number of discrete eigenvalues, see Chapter 4. When the order of anisotropy order is less than $N = 50$, the characteristic polynomial is constructed using the Chebyshev approximation method. Its roots are calculated using the companion matrix method by using the degree-doubling idea. If the degree of anisotropy exceeds $N = 50$, the polynomial is not constructed but its roots are calculated using a boxed Newton-Raphson algorithm.

The CharEquation class implements the evaluation routines for the characteristic equation $\Lambda(\nu)$ both for "normal" arguments as for the $\nu = 1 + \delta$ type of arguments. It also contains the routines for the bracketing procedure and an NLESolver object to either use the Van Wijngaarden-Dekker-Brent method or the (logarithmic) bisection method.

Once a root has been located, an object of the NormalRoot or DeltaRoot class – both derived classes of the parent class Root– is created and added to a list containing all roots. The normalisation integrals are calculated for each root.

The main work horse of this group is the Green class. Based on the results of the CharEquation class, i.e. the discrete eigenvalues, and numerical quadrature, this class allows for the evaluation of angular and angular/spatial moments of the infinite medium Green's function. For the spatial moments, it relies on the ExpAlpha class that evaluates the integral $\int_c^d \tilde{P}_k(x_0) e^{-\alpha|x-x_0} dx_0$.

The particular solution classes

The infinite medium particular solution $\psi_\infty(x, \mu)$ is calculated using the convolution integral of the infinite medium Green's function and the external source $Q(x, \mu)$. The latter is developed in a double Legendre series (one development for the angular variable, one for the spatial variable). The external source can be defined using the Source class. This class is an abstract base class for either a separable source (the spatial and angular variable are independent of one another) or a non-separable source. In the former case, the Legendre coefficients can be stored in two vectors of sizes N_a and N_s, respectively the angular and spatial approximation order while in the latter case a full matrix is needed (the angular coefficients are functions of space or vice versa). A non-separable source can only be constructed by giving the coefficients by hand. A separable source can be constructed as the product of an angular and a spatial function. Some basic functions have already been implemented: a flat function (useful for isotropic sources), a delta Dirac function, a peak function (exponential peak) and a polynomial function.

The Boundary Sources classes

Three classes make up the Boundary Sources Method: BoundaryCondition, Cell and Slab. A Slab is built by adding different Cells to it and assigning a BoundaryCondition at left and right boundary. A cell is constructed based on its nuclear data (NucData object). An infinite medium Green's function

is created and assigned to the cell. Once the slab is constructed, the Boundary Sources matrix is constructed together with the right-hand-side. This linear system is solved and the boundary sources coefficients are returned to their respective cells. The Cell class has different evaluation routines like AngularFlux, ScalarFlux, AverageFlux to extract information from the cell once the calculations are done.

8.3 Profiling of CASE-BSM

In order to give an extensive overview of the contributions of the different parts of the CASE library to the total running time for a certain problem, we have performed some profiling analysis using the VALGRIND framework [SN04]. This framework allows us to run the code and it generates a complete profile: a table of all routines called in the run and their respective running times needed. A call-graph is also created in which one sees immediately the interdependencies between the different routines in the code.

In this kind of profiling analysis, one must distinguish between four types of routines based on two characteristics and the combinations thereof: the number of times the routine is called and the time spent in this routine (the "self" time cost). It is clear that a routine that is called often and has a huge self time cost is a very likely candidate to be optimised. At the other extreme, a routine that isn't called much and has a low self time cost shouldn't need optimisation.

We have done this kind of analysis for the last Test Case in the previous Chapter: $\Sigma_t = 1$, $c = 0.8$, $g = 0.485$, $N = 5$ and a BSM order of $M = 30$. The result of such a profiling run is shown in Figures 8.2–8.4 which were created using the KCACHEGRIND [Wei] profiling visualisation tool.

Figure 8.2 clearly shows that about 98% of the running time is spent in `Integrate::Integrate`, the basic numerical quadrature routine that wraps the low-level methods DQAGSE, INTDE and INTHP. About 11% of the time is spent in DQAGSE while the main quadrature routine used is INTDE taking a 87% share of the total running time, as can be seen in Figure 8.3. Also from this figure, one can see that the overhead of the two numerical quadrature routines is quite low: $97.90\% - (91.32\% + 4.62\%) \approx 2\%$. The majority of the running time, 91.32%, is indeed spent in the evaluation of the integrand (`Integrator::IntegratorEvalFunctor`).

From Figure 8.4, we see that three quarters of the running time is spent in the evaluation of the integrand needed for the continuum part of the angular moments of the infinite medium Green's function (see equation (5.4)),

while about 8% is spent in the evaluation of the integrand for the angular/spatial moment of the continuum part of the Green's function (see equation (5.16)).

Finally, from Figure 8.4 it is clear for what function optimisation effort should be spent as much as possible: `PhiPoy_fw`, the routine that evaluates the transport polynomials $\phi_k(\nu)$ in the forward direction (which is stable for the range of interest $\nu \in [0, +1]$). Figure 8.4 also learns us that in the routine `PhiPoly_fw` about 17% of the time is spent in routines that manipulate the result vector, leaving about 30% to the routine itself. This implies two options: either one tries to optimise the current linear algebra routines from LINAL or one goes for another linear algebra library.

This kind of profiling runs gives a massive amount of useful information to the programmer. In the blink of an eye, he can see what routines need optimisation effort and what routines can safely be ignored in the optimisation process. We presented here only one run of the code as it is now but during the development, several of these runs were done and their results analysed. It is a safe guess to say that this approach has halved the running time for a typical problem comparing our initial code and the code as it is now.

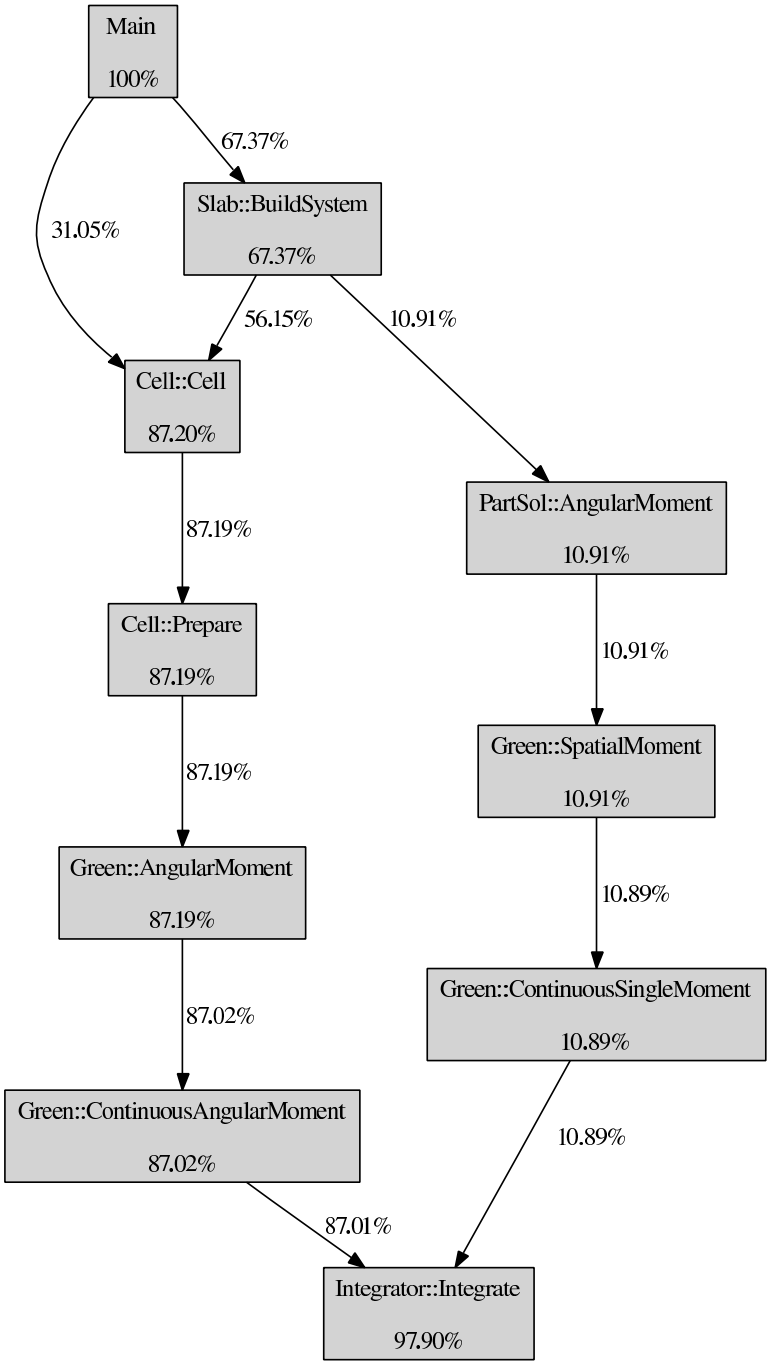


Figure 8.2: *Profiling run, first part.*

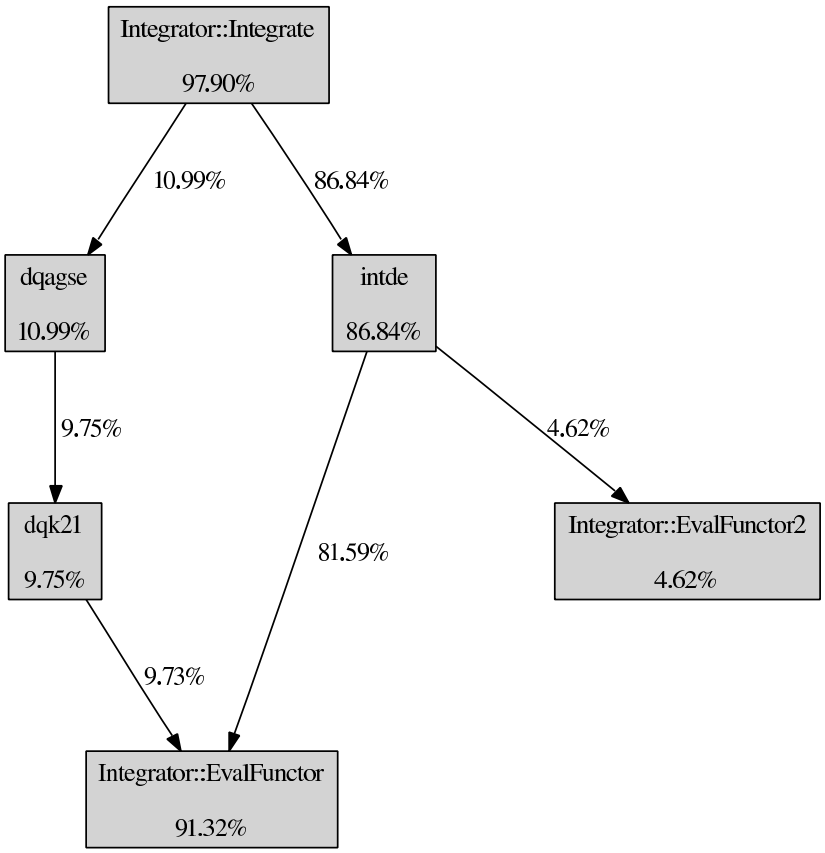


Figure 8.3: *Profiling run, second part.*

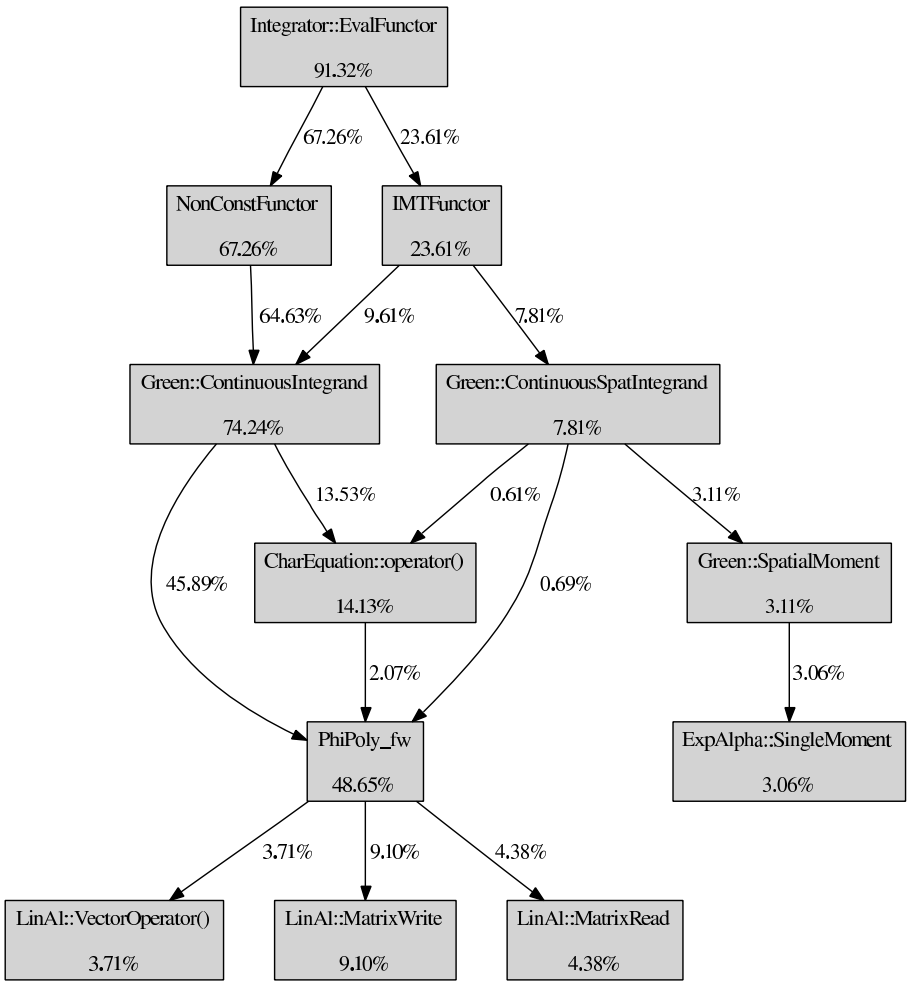


Figure 8.4: *Profiling run, third part.*

CHAPTER 9

Conclusions and future work

9.1 Conclusions

The work can be divided in three main parts: an overview of the general theory of neutron transport in the presence of an anisotropic scattering medium where we try to give the result of an extensive literature study, a robust and stable method for the computation of the Mika/Case eigenexpansion and finally two applications.

The general theory, summarized in Chapter 2, retakes the classical eigenexpansion first developed by Mika and later on studied extensively by Case. This eigenexpansion provides an orthogonal and complete basis to represent solutions to the one-speed homogeneous Boltzmann equation. In the development of this eigenexpansion, a set of orthogonal polynomials – the transport orthogonal polynomials – appeared. It has been shown that there exists a limiting relation between the roots of these polynomials and the discrete eigenvalues of the eigenexpansion. These polynomials are also a fundamental part of both the discrete eigenfunction and the continuum eigendistribution. It is of crucial importance that they are calculated in a stable way. This was the subject of Chapter 3.

From the general theory we have seen that for a subcritical medium– and under mild conditions on the scattering coefficients – there exists only a finite set of discrete eigenvalues all lying on the real axis. By means of the argument principle, one is able to bound the number of discrete eigenvalues by means of the number of roots of a polynomial – the characteristic polynomial – in the real interval $[0, +1]$. We have conjectured that under the same mild conditions this bound is reached and hence gives us the exact number of discrete eigenvalues. Using the exact number of discrete eigenvalues

and an upper bound on the largest positive discrete eigenvalue, we start a bracketing procedure returning segments of the real axis containing exactly one discrete eigenvalue. On each of these segments an iterative non-linear equation solving routine is applied to locate the discrete eigenvalues. The corresponding normalization integrals are also calculated. As a final result, the discrete part of the infinite medium Green's function is expressed using the discrete eigenfunctions. The complete procedure and numerical results were described in Chapter 4.

During our numerical experiments we came across some cases where our conjecture seemed to fail: we found by means of the characteristic polynomial that there should be two positive discrete eigenvalues, however we were able to locate only one. Thanks to the symbolic algebra package MAPLE, we were able to locate the second discrete eigenvalue and "save" our conjecture. It appeared that the second positive discrete eigenvalue was located extremely close to the continuum. So close even, that we needed to represent it as a difference with respect to one: $\nu = 1 + \delta$. We have adapted our solution method so that the occurrence of such near-singular eigenvalues is automatically detected and a modified non-linear solver is applied. In order to do so, we needed also to modify the evaluation routine for the characteristic equation in order to evaluate it using δ and not ν. If not, we would loose accuracy quite rapidly. We have encountered δ's ranging from 10^{-15} up to 10^{-44}. All were calculated with ten or more significant digits on δ. Two challenging problems, posed in a paper by Garcia and Siewert, were successfully solved by our code.

The second part, the continuous spectrum, has no corresponding eigenfunctions, but eigendistributions. These eigendistributions can only be characterized by their angular and angular/spatial moments. The accurate and fast calculation of these moments was the subject of Chapter 5. Using a test function that has the same characteristics as the continuum integrand, several numerical quadrature routines were evaluated on accuracy and computational cost. The angular/spatial integral needed special attention since the integrand can become very much peaked and round-off is eminent. The angular and angular/spatial moments of the infinite medium Green's function are expressed using the continuum part of the eigenexpansion.

As a first application of the Mika/Case eigenexpansion, a Boundary Sources Method has been implemented for a multi-cell slab problem. It can be shown that the solution of the finite medium neutron transport equation can be written as the convolution of a transport kernel (the infinite medium Green's function in our case) with two sources located at the boundaries of the cell complemented with a particular solution for the infinite problem (if

an external source is present) which can also be expressed using a convolution integral with the transport kernel with the external source. This method is exact upon the point where the boundary sources, continuous functions of the cosine of the angle, are approximated by a finite Legendre approximation. It is the order of the approximation that determines the order of the method and hence its accuracy. We have compared our code to other codes available and have shown that we can safely consider our code as a benchmark code for others. The full development and numerical results are shown in Chapter 6.

The second application, see Chapter 7, is the study of the discrete spectrum of a well-known scattering kernel from radiative transfer studies: the Henyey-Greenstein kernel. This kernel is a continuous function with a free parameter which can be used to adapt the model to fit experimental data more properly. Depending on this parameter, the kernel describes forward or backward peaked scattering. Since the Mika/Case eigenexpansion works only for a finite Legendre approximation of a real scattering kernel, we need to represent the Henyey-Greenstein kernel using a finite series. The choice of the approximation order has, of course, a large influence on the quality of the approximation and this has indirectly an influence on the discrete spectrum. Minimal orders for three different criteria were derived: non-negativity and absolute or relative error requirements. Finally, we calculated the discrete spectrum for different combinations of the free parameter and the approximation order.

The final chapter gives a global overview of the code we have implemented: CASE and CASE-BSM. All classes are grouped in six parts and described. We showed the relations between the different classes. Finally, a profiling run has been done on the final code, showing where the code spends most of its time. These kind of profiling runs have been run during the whole development cycle and it can be safely said that the code has become twice as fast as a result of the profiling analysis.

9.2 Future work

As can be seen from the profiling session in Chapter 8, the code spends 99% of its time in numerical quadrature calculating the angular and angular/spatial moments of the infinite medium Green's function. Every possible optimisation in the quadrature itself is a gain in computing time. In order to speed up quadrature two paths are open:

- speed up the evaluation of the integrand or

- try to minimize the number of integrand evaluations by improving the quadrature method.

In order to speed up the evaluation of the integrand, the profiling session gives clear indication what to improve: the evaluation of the transport polynomials $\phi_k(\nu)$ takes about 50% of the total computation time. Of this 50% about 17% is spent in the linear algebra routines manipulating the result vector and about 30% is spent in the evaluation of the recurrence itself. Improving the linear algebra routines is not a trivial task and it should be an option to switch to different existing linear algebra classes (GMM++, BOOST/UBLAS, BLITZ++,... How to speed up the recurrence is a totally different question. The implementation of the forward recurrence is as trivial as can be. Probably the only way to improve this routine is to do an extensive profiling analysis and see how different compiler options (and even different compilers) behave.

The second option to improve the speed of the code is to work with specialised quadrature routines. In our code, we work mainly with DQAGSE from QUADPACK and INTDE. Both are general purpose adaptive quadrature routines. Both handle possible end-point singularities: DQAGSE by extrapolation using Wynn's ϵ-algorithm and INTDE by applying the double exponential transformation. However, one could try to "capture" the difficulties in the integrand in a weight function and develop a Gaussian quadrature rule using that weight function.

Of course, the final goal is to implement this one-dimensional solver inside the nodal code HEXNODYN. By doing this, HEXNODYN will be able to handle arbitrary order anisotropic scattering media. However, since this requires a big modification in the code, mainly in the data handling routines, it was decided to leave this out of this work since the academic surplus didn't justify the time needed to do the implementation.

APPENDIX A

The Legendre orthogonal polynomials and functions of the second kind

A.1 Definition

The ordinary Legendre functions of the first and second kind are the solutions of the differential equation [San59; MOS66; EMOT53]

$$(1-z^2)\frac{d^2W}{dz^2} - 2z\frac{dW}{dz} + \nu(\nu+1)W(z) = 0 \tag{A.1}$$

and they are denoted by $P_\nu(z)$ and $Q_\nu(z)$ respectively. The general solution of Equation (A.1) can be written as

$$W(z) = A_\nu P_\nu(z) + B_\nu Q_\nu(z) \tag{A.2}$$

with A_ν and B_ν arbitrary constants. The functions $P_\nu(z)$ and $Q_\nu(z)$ are single-valued and regular in the z-plane cut along the real axis from $-\infty$ to $+1$. If z is real and lies in the cut $[-1,+1]$, we have also two independent solutions $P_\nu(x)$ and $Q_\nu(x)$. We use the variable x here to indicate that these are the solutions on the cut. They are given by

$$P_\nu(x) = \lim_{\epsilon\to 0}\frac{1}{2}\Big(P_\nu(x+i\epsilon) + P_\nu(x-i\epsilon)\Big) \tag{A.3}$$

$$P_\nu(x) = \lim_{\epsilon\to 0}\frac{i}{\pi}\Big(Q_\nu(x+i\epsilon) - Q_\nu(x-i\epsilon)\Big) \tag{A.4}$$

$$Q_\nu(x) = \lim_{\epsilon\to 0}\frac{1}{2}\Big(Q_\nu(x+i\epsilon) - Q_\nu(x-i\epsilon)\Big) \tag{A.5}$$

When $\nu = n$, an integer, the $P_n(z)$ are polynomials of degree n in z and are known as the Legendre orthogonal polynomials [Sze59; Ask75].

A.2 The Legendre polynomials $P_n(x)$

The Legendre polynomials are well known and therefore we summarize shortly the main properties and some relations we need in this work. For more information, one can consult classical textbooks like [Sze59; Ask75; MOS66; San59]. In the following, we assume the argument z to be on the real axis and hence use the notation x.

Explicit formulae

The Legendre polynomials are given by

$$P_n(x) = \sum_{k=0}^{M} (-1)^k \frac{(2n-2k)!}{2^n k!(n-k)!(n-2k)!} x^{n-2k} \tag{A.6}$$

where $M = n/2$ when n is even and $M = (n-1)/2$ when n is odd. The first few are

$$P_0 = 1, \qquad P_1 = x, \tag{A.7}$$

$$P_2 = \frac{1}{2}(3x^2 - 1), \qquad P_3 = \frac{1}{2}(5x^3 - 3x). \tag{A.8}$$

Special values and symmetries

$$P_n(1) = 1, \qquad P_n(-1) = (-1)^n. \tag{A.9}$$

$$P_n(0) = \begin{cases} \frac{(-1)^n}{\sqrt{\pi}} \frac{\Gamma\left(\frac{1}{2}+\frac{n}{2}\right)}{\Gamma\left(1+\frac{n}{2}\right)} & \text{if } n \text{ is even,} \\ 0 & \text{if } n \text{ is odd.} \end{cases} \tag{A.10}$$

$$P_n(-x) = (-1)^n P_n(x). \tag{A.11}$$

Recurrence relation

The Legendre polynomials obey the following three-term recurrence relation:

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1} \tag{A.12}$$

which is valid from $n = 0$ when using the starting values $P_{-1} = 0$ and $P_0 = 1$. The recurrence is forward stable in the interval of interest $[-1, +1]$ (and also outside this interval, for that matter) so it can be used to evaluate these functions accurately. The algorithm used, Smiths algorithm [Buy92] or sometimes called Clenshaws recurrence [PTVF96], is based on the recurrence equation. The cost of this evaluation is $2n - 1$ adds and multiplies, which is about twice the cost of the classical Horner scheme for polynomial evaluation. Therefore, in our code, we use the Horner scheme for the first 10 degrees allowing for a lower cost in these cases.

Orthogonality and Legendre series

The Legendre polynomials are orthogonal on the real interval $x \in [-1, +1]$ with respect to the weight function $w(x) \equiv 1$. The orthogonality is expressed as

$$\int_{-1}^{+1} P_n(x) P_m(x) dx = \frac{2}{2n+1} \delta_{n,m} \tag{A.13}$$

where $\delta_{n,m}$ is the Kronecker delta

$$\delta_{n,m} = \begin{cases} 0 & n \neq m, \\ 1 & n = m. \end{cases} \tag{A.14}$$

The Legendre series is a way of approximating a function on the orthogonality interval $[-1, +1]$ where all points are equally weighed. The series is usually written as

$$f(x) = \sum_{n=0}^{\infty} \frac{2n+1}{2} f_n P_n(x) \tag{A.15}$$

where the Legendre coefficients are given by

$$f_l = \int_{-1}^{+1} f(x) P_n(x) dx. \tag{A.16}$$

When the function $f(x)$ is continuous, the series (A.15) converges to the exact function. In general, continuity is a sufficient, but not necessary condition.

A.3 The Legendre functions of the second kind $Q_n(z)$

The Legendre functions of the second kind, $Q_n(z)$, sometimes also called spherical Legendre functions, are the second set of solutions of the Legendre differential Equation (A.1). As said before, we shall use the notation $Q_n(z)$ for the Legendre functions in the complex z-plane cut on the real axis from $-\infty$ to $+1$. We use the notation $Q_n(x)$ for values on the cut $[-1, +1]$.

Explicit formulae

The $Q_n(z)$ are defined by

$$Q_n(z) = \frac{1}{2} P_n(z) \log\left(\frac{z+1}{z-1}\right) - W_{n-1}(z) \tag{A.17}$$

where $W_{n-1}(z)$ is a polynomial of degree $n-1$ given by

$$W_{n-1}(z) = \sum_{m=0}^{[(n-1)/2]} \frac{(2n-4m-1)}{(n-m)(2m+1)} P_{n-2m-1}(z) \tag{A.18}$$

$$= \sum_{m=1}^{n} \frac{1}{m} P_{m-1}(z) P_{n-m}(z). \tag{A.19}$$

For values x on the cut $[-1, +1]$, the functions $Q_n(x)$ are defined as (from Eq.A.5).

$$Q_n(x) = \lim_{\epsilon \to 0} \frac{1}{2} \Big(Q_n(x + i\epsilon) + Q_n(x - i\epsilon) \Big) \tag{A.20}$$

Special values and symmetries

$$\lim_{x \to +1} Q_n(x) = \infty, \qquad \lim_{x \to -1} Q_n(x) = (-1)^{n+1} \infty. \tag{A.21}$$

$$Q_n(0) = \begin{cases} 0 & \text{if } n \text{ is even,} \\ (-1)^{\left(\frac{n}{2} - \frac{1}{2}\right)} \sqrt{\pi} \frac{\Gamma\left(\frac{1}{2} + \frac{n}{2}\right)}{\Gamma\left(1 + \frac{n}{2}\right)} & \text{if } n \text{ is odd.} \end{cases} \tag{A.22}$$

$$Q_n(-z) = (-1)^{n+1} Q_n(z). \tag{A.23}$$

Recurrence relation

The Legendre functions of the second kind obey the same recurrence relation as the Legendre polynomials

$$Q_{n+1}(z) = \frac{2n+1}{n+1} z Q_n(z) - \frac{n}{n+1} Q_{z-1} \tag{A.24}$$

which is valid from $n = 1$ when using the starting values

$$Q_0(z) = \frac{1}{2} \log\left(\frac{z+1}{z-1}\right), \qquad Q_0(x) = \frac{1}{2} \log\left(\frac{1+x}{1-x}\right), \tag{A.25}$$

$$Q_1(z) = \frac{1}{2} z \log\left(\frac{z+1}{z-1}\right) - 1, \qquad Q_1(x) = \frac{1}{2} x \log\left(\frac{1+x}{1-x}\right) - 1. \tag{A.26}$$

On the cut $[-1, +1]$, the recurrence is forward stable and can be used as such to compute the needed values of $Q_n(x)$. However, for $|z| > 1$, the forward direction is unstable and is useless to calculate $Q_n(z)$. We therefore need another way to do so. We have chosen for the improved Miller algorithm, see Appendix C.

Relations between $P_n(z)$ and $Q_n(z)$

The Wronskian determinant

$$W\left[P_n(z), Q_n(z)\right] = P_n(z) \frac{d}{dz} Q_n(z) - Q_n(z) \frac{d}{dz} P_n(z) \tag{A.27}$$

is equal to

$$W\left[P_n(z), Q_n(z)\right] = \frac{1}{1-z^2}. \tag{A.28}$$

One also has

$$P_n(z) Q_{n-1}(z) - P_{n-1}(z) Q_n(z) = \frac{1}{n} \tag{A.29}$$

and

$$P_{n+1}(z) Q_{n-1}(z) - P_{n-1}(z) Q_{n+1}(z) = \frac{2n+1}{n(n+1)} z^2. \tag{A.30}$$

An integral relation between $P_n(z)$ and $Q_n(z)$ is sometimes used as the definition of the latter:

$$Q_n(z) = \frac{1}{2}\int_{-1}^{+1} \frac{P_n(t)}{z-t}dt, \quad z \text{ not on the cut } [-1,+1]. \tag{A.31}$$

Expansions for $Q_n(z)$ for $z \to 1$

From [WFS04a], we have

$$\begin{aligned} Q_n(z) = &\sum_{k=0}^{n} \frac{(k+n)!\psi(k+1)}{k!^2(n-k)!}\left(\frac{z-1}{2}\right)^k \\ &+ \left[\frac{1}{2}\left(\log\left(\frac{1+z}{1-z}\right)\right) - \psi(n+1)\right]\sum_{k=0}^{\infty}\frac{(-n)_k(n+1)_k}{k!^2}\left(\frac{1-z}{2}\right)^k \end{aligned} \tag{A.32}$$

where $n \in \mathbb{N}$. The logarithmic derivative of the Gamma function [MOS66, p.13] $\psi(z)$ is defined for natural arguments as

$$\psi(n+1) = \sum_{m=1}^{n}\frac{1}{m} - \gamma \tag{A.33}$$

with $\psi(1) = -\gamma$ where γ is the Euler constant

$$\gamma = \lim_{m\to\infty}\left(\sum_{l=1}^{m}\frac{1}{l} - \log m\right) = 0.577215\ldots. \tag{A.34}$$

The Pochhammer symbol $(\alpha)_n$ is defined as

$$(\alpha)_n = \alpha(\alpha+1)\ldots(\alpha+n-1) = \frac{\Gamma(\alpha+n)}{\Gamma(\alpha)}. \tag{A.35}$$

APPENDIX B

The Lambert W-function

B.1 Definition and properties

The Lambert W-function originated in a work of Lambert dating from 1758 where he solved the equation $x = q + x^m$. He developed a series for x in powers of q. The series for $\log x$ defines a function $T(v)$ called the *tree function*. It is related to the W-function by $T(v) = -W(-v)$. The rigorous definition of the W-function is:

$$W(z)e^{W(z)} = z. \tag{B.1}$$

If z is real and $-1/e \leq z < 0$, there are two possible values for $W(z)$. Formally said, there are two branches for this range of argument that result in a real function value. The branch satisfying $-1 \leq W(z)$ is the $W_0(z)$ branch, and the other branch is the $W_{-1}(z)$ branch. Figure B.1 shows these two branches. These two branches are the only real-valued branches of the W-function. In general, due to the multiple branches of the complex logarithmic function, there are multiple branches of the W-function. All the other branches return a complex number as function value.

For more historical information on the Lambert W and analytical properties the overview paper by Corless et al. [CGH$^+$96] is recommended. For a summary of the main properties of function, including some series, the Wolfram Function site has an entry on this function [WFS04b].

B.2 Calculation scheme for $W_{-1}(x)$

Given the fact that the Lambert W-function is given only implicitly, one way to evaluate the function is by solving the equation defining the func-

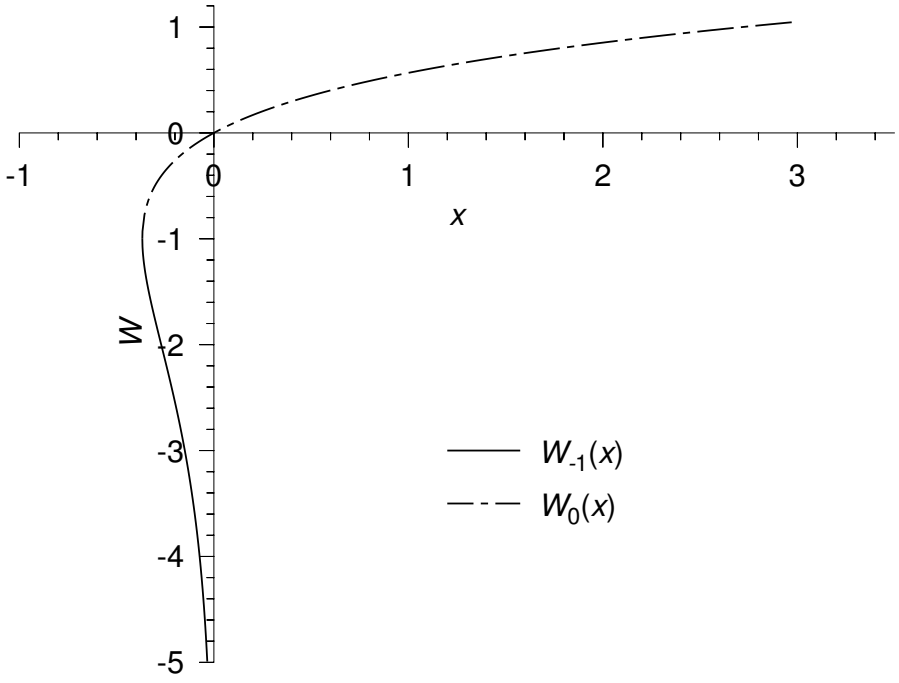


Figure B.1: *The two real-valued branches of the Lambert W-function.*

z	$W(-1,z)$	Relative error	Iterations
$-1.0/e \approx -0.367879$	-1.000000	10^{-14}	1
$-0.8/e \approx -0.294304$	-1.824388	10^{-15}	4
$-0.6/e \approx -0.220728$	-2.376421	10^{-15}	3
$-0.4/e \approx -0.147152$	-3.022313	10^{-15}	3
$-0.2/e \approx -0.073576$	-3.994308	10^{-15}	3

Table B.1: *Some calculation statistics for Halley's iteration applied to the calculation of $W_{-1}(z)$.*

tion, Equation (B.1). According to Corless et al. [CGH$^+$96], a third order method is preferred, namely Halley's method. From a good initial guess, this method converges quite rapidly to the correct value.

The initial guess is taken from the series development for $W(z)$ in logarithms and iterated logarithms, see [CGH$^+$96, Eq. 4.19]:

$$W(z) = L_1 - L_2 + \frac{L_2}{L_1} + \frac{L_2(-2+L_2)}{2L_1^2} + \frac{L_2(6-9L_2+2L_2^2)}{6L_1^3} + \frac{L_2(-12+36L_2-22L_2^2+3L_2^3)}{12L_1^4} + \mathcal{O}\left(\left\{\frac{L_2}{L_1}\right\}^5\right), \quad \text{(B.2)}$$

where $L_1 = \log z$ and $L_2 = \log(\log z)$. For the initial guess, we take only the first four terms:

$$W(z) \approx L_1 - L_2 + \frac{L_2}{L_1} + \frac{L_2(-2+L_2)}{2L_1^2} \quad \text{(B.3)}$$

Halley's iteration for the Lambert W-function is given by the formula:

$$w_{j+1} = w_j - \frac{w_j e^{w_j} - z}{e^{w_j}(w_j+1) - \frac{(w_j+2)(w_j e^{w_j} - z)}{2w_j+2}} \quad \text{(B.4)}$$

The iteration needs a few iterations to achieve convergence up to IEEE double machine precision. Table B.1 gives an overview of the accuracy and cost for some values of the parameter z in $[-1/e, 0[$. The relative error is computed with respect to values from MAPLE.

APPENDIX C

Calculating minimal solutions of a second order recurrence relation

C.1 Minimal versus maximal solutions of a second order recurrence relation

A three-term (or second order) recurrence relation

$$\phi_{k+1} + a_k \phi_k + b_k \phi_{k-1} = 0, \quad k = 0, 1, \ldots \tag{C.1}$$

has two independent solutions, f_k and g_k. We say that if

$$\lim_{n \to \infty} \frac{f_k}{g_k} = 0 \tag{C.2}$$

the solution f_k is the minimal solution, while g_k is the maximal (or dominant) solution. The maximal solution can easily be calculated starting from two initial values ϕ_0 and ϕ_1 using the forward direction (increasing k) of the recurrence (C.1).

However, the minimal solution can never be calculated in this forward direction using finite precision arithmetic. The smallest error, for example due to rounding in the multiplication and addition operations performed while evaluating the recurrence relation or a small error on the initial values, will result in the poisoning of the minimal solution by the dominant solution and after a few steps in the recurrence, the calculated solution will have few digits or none at all correct.

Let us show the result of following the forward direction for the Legendre functions of the first kind $P_k(z)$ and second kind $Q_k(z)$, which are respectively the maximal and minimal solution of the recurrence relation (C.3)

k	$P_k(z)$	Relative error	$Q_k(z)$	Relative error
0	$1.000000 \times 10^{+00}$	10^{-15}	5.180460×10^{-01}	10^{-15}
1	$2.100000 \times 10^{+00}$	10^{-15}	8.789653×10^{-02}	10^{-15}
2	$6.115000 \times 10^{+00}$	10^{-15}	1.785108×10^{-02}	10^{-14}
3	$2.000250 \times 10^{+01}$	10^{-15}	3.881098×10^{-03}	10^{-14}
4	$6.892294 \times 10^{+01}$	10^{-15}	8.747260×10^{-04}	10^{-13}
5	$2.445267 \times 10^{+02}$	10^{-15}	2.015853×10^{-04}	10^{-11}
6	$8.839920 \times 10^{+02}$	10^{-15}	4.716525×10^{-05}	10^{-10}
7	$3.237975 \times 10^{+03}$	10^{-15}	1.115704×10^{-05}	10^{-09}
8	$1.197603 \times 10^{+04}$	10^{-15}	2.661252×10^{-06}	10^{-08}
9	$4.462673 \times 10^{+04}$	10^{-15}	6.389308×10^{-07}	10^{-07}
10	$1.672822 \times 10^{+05}$	10^{-15}	1.542071×10^{-07}	10^{-05}

Table C.1: *Evaluating $P_k(z)$ and $Q_k(z)$ using the forward direction for $z = 2.1$ using starting values correct up to machine precision.*

$$(k+1)\phi_{k+1}(z) - (2k+1)z\phi_k(z) + k\phi_{k-1}(z) = 0 \tag{C.3}$$

when the argument z is greater than one. Table C.1 clearly shows that the number of correct digits decreases rapidly for the $Q_k(z)$ when using the forward direction.

C.2 Miller's algorithm

Luckily, the minimal solution becomes dominant when evaluating the recurrence in the backwards (decreasing k) direction and vice versa. And this idea forms the basis of Miller's algorithm [PTVF96]. The idea is to choose a large N, and impose that in finite precision arithmetic $\phi_{N+2} = 0$ and $\phi_{N+1} = macheps$ where *macheps* is the machine precision. Starting from these two values, the recurrence is followed in the backwards direction until ϕ_0 is computed. Finally, a normalisation is required to get the final result. This normalisation can be based on the (known) value of ϕ_0 or some other relation involving the different ϕ_k. Algorithm C.1 summarizes Miller's method.

Input: x, argument; N, order required; $M \gg N$, start order
Output: ϕ_k, $k = 0 \ldots N$

$\phi_{M+1} \leftarrow 0$;
$\phi_M \leftarrow macheps$;
for $k \leftarrow M$ **to** 1 **do**
  $\phi_{k-1} \leftarrow -\frac{1}{b_k}\phi_{k+1} - \frac{a_k}{b_k}\phi_k$;
endfor
Normalize the series based on a known value (ϕ_0) or other normalisation relation;

Algorithm C.1: Miller's algorithm.

M	$Q_{10}(2.1)$	Relative error
10	$1.4428185853 \times 10^{-07}$	10^{-02}
12	$1.5416454183 \times 10^{-07}$	10^{-04}
14	$1.5420542741 \times 10^{-07}$	10^{-06}
16	$1.5420559640 \times 10^{-07}$	10^{-09}
18	$1.5420559710 \times 10^{-07}$	10^{-11}
20	$1.5420559710 \times 10^{-07}$	10^{-14}

Table C.2: *Applying Miller's algorithm to $Q_{10}(2.1)$ for different starting orders M.*

As an example we apply this algorithm for different starting orders M for the Legendre $Q_k(x)$ function for $x = 2.1$ and look at the value for $k = 10$. Table C.2 clearly shows that we a need a reasonably higher order to start with than the order we are finally interested in.

C.3 Improved Miller's algorithm

The choice of the starting order in the classical Miller algorithm is somewhat based on heuristic and experience. It is made even more difficult due to the fact that this order can depend on the argument. Usually one wants to be on the safe side and make sure that the order is high enough to avoid loss of correct digits in the final result. This leads, of course, to a higher evaluation cost than necessary.

The main problem in Miller's algorithm is the fact that f_{M+1} is set to

zero and f_M to *macheps*. These two values are not really related to each other by the recurrence relation. One simply assumes that when f_M is equal to *macheps*, f_{M+1} is so small that due to the finite precision of floating point arithmetic we can set it to zero.

Continued fraction form

Fortunately, one can rewrite recurrence (C.1) in the form [PTVF96; GS97]

$$\frac{\phi_k}{\phi_{k-1}} = -\frac{b_k}{a_k + \phi_{k+1}/\phi_k} \tag{C.4}$$

and iterating this leads to the following continued fraction representation of this fraction:

$$CF_k \equiv \frac{\phi_k}{\phi_{k-1}} = -\frac{b_k}{a_k -} \frac{b_{k+1}}{a_{k+1} -} \dots \tag{C.5}$$

Pincherlé's theorem [PTVF96; GS97] tells us that this continued fraction converges to the fraction of the minimal solution f_k/f_{k-1}. When we are able to evaluate this fraction CF_k by means of the continued fraction, we could apply Miller's algorithm, but instead of starting with

$$f_{M+1} = 0, f_M = macheps,$$

we start with

$$f_{M+1} = macheps, f_M = \frac{f_{k+1}}{CF_{k+1}}.$$

With this approach, we could –in theory– start with a value M just above the required order N. For "safety" reasons, we take $M = N + 5$.

The evaluation of the continued fraction CF_k can be done either by Steed's method or by the Modified Lentz method [Len76; PTVF96]. The former can have trouble converging [PTVF96], therefore we opted for the latter. Pseudo-code for the Modified Lentz algorithm can be found in Algorithm C.2 [PTVF96, p. 171]. Our evaluation algorithm for the minimal solution of a three-term recurrence is indicated in Algorithm C.3. Applying this algorithm for the calculation of $Q_k(2.1)$ gives results all within the precision of IEEE double as can be seen in Table C.3.

Input: a_k, b_k, the continued fraction coefficients;
Input: N, order requested $CF_N = f_{N+1}/f_N$;
Input: *tiny*, a very small number, say 10^{-30};
Input: *tolerance*, the tolerance requested on CF_N;
Output: CF_N, the continued fraction;

$f_0 \leftarrow b_0$, if $b_0 = 0$, set $f_0 \leftarrow tiny$;
$C_0 \leftarrow f_0$, $D_0 \leftarrow 0$; **for** $j \leftarrow 1, 2, \ldots$ **do**
  $D_j \leftarrow b_j + a_j D_{j-1}$;
  if $D_j = 0$ **then**
    $D_j \leftarrow tiny$;
  endif
  $C_j \leftarrow b_j + a_j / C_{j-1}$;
  if $C_j = 0$ **then**
    $C_j \leftarrow tiny$;
  endif
  $D_j \leftarrow 1/D_j$;
  $\Delta_j = C_j D_j$;
  $f_j = f_{j-1}\Delta_j$;
  if $|\Delta_j - 1| < tolerance$ **then**
    break
  endif
endfor

Algorithm C.2: The Modified Lentz's method.

Input: x, argument; N, order required;
Output: ϕ_k, $k = 0 \ldots N$

$M \leftarrow N + 5$;
Calculate $frac = CF_{M+1}$ using the Modified Lentz method;
$\phi_{M+1} \leftarrow macheps$;
$\phi_M \leftarrow \frac{macheps}{\phi_{M+1}}$;
for $k \leftarrow M$ **to** 1 **do**
  $\phi_{k-1} \leftarrow -\frac{1}{b_k}\phi_{k+1} - \frac{a_k}{b_k}\phi_k$;
endfor
Normalize the series based on a known value (ϕ_0) or other normalisation relation;

Algorithm C.3: Improved Miller's algorithm.

k	$Q_k(z)$	Relative error
0	5.180460×10^{-01}	10^{-15}
1	8.789653×10^{-02}	10^{-15}
2	1.785108×10^{-02}	10^{-15}
3	3.881098×10^{-03}	10^{-15}
4	8.747260×10^{-04}	10^{-15}
5	2.015853×10^{-04}	10^{-15}
6	4.716525×10^{-05}	10^{-15}
7	1.115704×10^{-05}	10^{-15}
8	2.661252×10^{-06}	10^{-15}
9	6.389304×10^{-07}	10^{-15}
10	1.542056×10^{-07}	10^{-15}

Table C.3: *Evaluating $Q_k(z)$ using the improved Miller algorithm for $z = 2.1$.*

Bibliography

[ABB+99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. `http://www.netlib.org/lapack`.

[Act90] F. Acton. *Numerical Methods That Work*. American Mathematical Society, 2nd edition, 1990.

[AD86] B. Arien and J. Daniels. Cassandre, code description and user's guide. Technical Report BLG-591, SCK-CEN, Mol, Belgium, 1986.

[AN04] P. Abbott and J. Noble. Private communication, 2004.

[AS72] M. Abramowitz and I.A. Stegun, editors. *Handbook of mathematical functions*. Dover publications, 1972.

[ASDM84] B. Arien, A. Siebertz, J. Devooght, and E. Mund. Cassandre, a two-dimensional multigroup diffusion code for reactor transient analysis. Technical Report BLG-571, SCK-CEN, Mol, Belgium, 1984.

[Ask75] Richard Askey. *Orthogonal Polynomials and Special Functions*. SIAM, 1975.

[Bad90] A. Badruzzaman. Nodal methods in transport theory. In *Advances in Nuclear Science and Technology*, volume 21. Plenum Press, New York, 1990.

[BDM+90] R. Beauwens, J. Devooght, E. Mund, R. Rydin, and R. Wagner. A 3D multigroup transport kinetics code in hexagonal geometry for fast reactor transient analysis. In *Proc. of the Int. Conf.*

on the Physics of Reactors: Operation, Design and Computation, 1990.

[Bea68] R. Beauwens. *Etude du transport des neutrons en milieu hétérogène, à une vitesse par la méthode des sources superficielles*. PhD thesis, 1968.

[Bea94] R. Beauwens. Final report on the development of HEXNODYN-2. Technical Report contract number 4137-90 11 ED ISP B, May 1994.

[Boy02] J. Boyd. Computing zeros on a real interval through Chebyshev expansion and polynomial rootfinding. *SIAM Journal of Numerical Analysis*, 40(5):1666–1682, 2002.

[BP70] A. Björk and V. Pereyra. Solution of Vandermonde systems of equations. *Mathematics of Computation*, 24(112):893–903, 10 1970.

[Bre78] C. Brezinski. *Algorithmes d'accélération de la convergence, étude numérique*. Editions technip, Paris, 1978.

[BS62] F.L. Bauer and J. Stoer. Algorithm 105: Newton Maehly. *Communications of the ACM*, 5:387–388, 1962.

[Buy92] L. Buyst. *Numerieke methodes in de benaderingstheorie*. Departement Computerwetenschappen, 1992. Katholieke Universiteit Leuven.

[Cac82] D. Cacuci. On the finiteness of the number of discrete eigenvalues in neutron transport theory. *Journal of Mathematical Physics*, 23(11):2205–2214, 1982.

[Cas74] K. Case. Scattering theory, orthogonal polynomials, and the transport equation. *Journal of Mathematical Physics*, 15(7):974–983, 1974.

[CGH+96] R.M. Corless, G.H. Gonnet, D.E. Hare, D.J. Jeffrey, and D.E. Knuth. On the Lambert *W* function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[Chi90] T. Chihara. The three term recurrence relation and spectral properties of orthogonal polynomials. In P. Nevai, editor, *Orthogonal Polynomials*, pages 99–114. Kluwer Academic Publishing, 1990.

[CN82] T. Chihara and P. Nevai. Orthogonal polynomials and measures with finitely many point masses. *Journal of Approximation Theory*, 35:370–380, 1982.

[CZ67] K. Case and P. Zweifel. *Linear transport theory*. Addison-Wesley, 1967.

[DP76] E. De Doncker and R. Piessens. Algorithm 32: Automatic computation of integrals with singular integrand over a finite or an infinite range. *Computing*, 17:265–279, 1976.

[DR84] Ph. Davis and Ph. Rabinowitz. *Methods of numerical integration*. Academic press, 2nd edition, 1984.

[EM95] A. Edelman and H. Murakami. Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation*, 64(210):763–776, 4 1995.

[EMOT53] A. Erdélyi, W. Magnus, F. Oberhettinger, and F. Tricomi. *Higher transcendental functions*, volume 1. McGraw-Hill Book Company, 1953.

[Fre86] Géza Freud. On the greatest zero of an orthogonal polynomial. *Journal of Approximation Theory*, 46:16–24, 1986.

[Gan05] B. Ganapol. BLUE, 2005. private communication.

[GC62] I.M. Guelfand and G.E. Chilov. *Les distributions*. Collection universitaire de mathémathiques. Dunod, Paris, 1962. Translated by G. Rideau.

[GC80] J. Geronimo and K. Case. Scattering theory and polynomials orthogonal on the real line. *Transactions of the American Mathematical Society*, 258(2):467–494, 4 1980.

[Gle97] B. Gleyse. Sturm sequences and the number of zeros of a real polynomial in the unit disk: numerical computation. *Applied Mathematics Letters*, 10(2):123–127, 1997.

[Goe94] S. Goedecker. Remark on algorithms to find roots of polynomials. *SIAM Journal of Scientific Computing*, 15(5):1059–1063, 1994.

[GS85] R. Garcia and C. Siewert. Benchmark results in radiative transfer. *Transport Theory and Statistical physics*, 14(4):437–483, 1985.

[GS89] R. Garcia and C. Siewert. On discrete spectrum calculations in radiative transfer. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 42(5):385–394, 1989.

[GS97] A. Gil and J. Segura. Evaluation of Legendre functions of argument greater than one. *Computer Physics Communications*, 105:273–283, 1997.

[GVL96] G.H. Golub and C.F. Van Loan. *Matrix computations*. The John Hopkins University Press, 3rd edition, 1996.

[Han73] R. Hangelbroek. *A functional analytic approach to the linear transport equation*. PhD thesis, 1973.

[Hen88] P. Henrici. *Power Series, Integration, Conformal Mapping, Location of Zeros*, volume 1 of *Applied and Computational Complex Analysis*. Wiley, 1988.

[Het93] D. Hetrick. *Dynamics of nuclear reactors*. American Nuclear Society, 1993.

[HG41] L.G. Henyey and J.L. Greenstein. Diffuse radiation in the galaxy. *Astrophysics Journal*, 93:70–83, 1941.

[Hig02] N.J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2nd edition, 2002.

[HMW+04] J. Hendricks, G. McKinney, L. Waters, T. Roberts, H. Egdorf, and J. Finch et al. MCNPX *2.5e*, 2004.

[IMT87] M. Iri, S. Moriguti, and Y. Takasawa. On a certain quadrature formula. *Journal of Computational and Applied Mathematics*, 17:3–20, 1987. Translation of the original Japanese paper published in 1970.

[Inö70] E. Inönü. Orthogonality of a set of polynomials encountered in neutron-transport and radiative-transfer theories. *Journal of Mathematical Physics*, 11(2):568–577, 1970.

[Jen75] M. Jenkins. Algorithm 493: Zeros of a real polynomial. *ACM Transactions on Mathematical Software*, 1(2):178–189, 1975.

[JT70] M. Jenkins and J. Traub. A three-stage algorithm for real polynomials using quadratic iteration. *SIAM Journal of Numerical Analysis*, 7:545–566, 1970.

[KD04] R. Krause and C. Drexler. *LinAl*, 2004. `http://linal.sourceforge.net/LinAl/Doc/linal.html`.

[KL99] K. Kwon and D. Lee. On the extreme zeros of orthogonal polynomials. *J. Korean Math. Soc.*, 36(3):489–507, 1999.

[KSV70] H. Kaper, J. Shultis, and J. Veninga. Numerical evaluation of the slab albedo solution in one-speed anisotropic transport theory. *Journal of computational physics*, 6:288–313, 1970.

[Lej57] F. Leja. Sur certaines suites liées aux ensembles plans et leur application à la représentation conforme. *Ann. Soc. Pol. Math.*, 4:9–13, 1957.

[Len76] W.J. Lentz. Generating Bessel functions in Mie scattering calculations using continued fractions. *Applied Optics*, 15(3):668–671, 1976.

[Map04] MapleSoft. MAPLE, 2004. Version 9.51.

[MI82] K. Murota and M. Iri. Parameter tuning and repeated application of the IMT-type transformation in numerical quadrature. *Numerische Mathematik*, 38:347–363, 1982.

[Mik61] J.R. Mika. Neutron transport with anistropic scattering. *Nuclear Science and Engineering*, 11:415–427, 1961.

[MM94] R. MacFarlane and D. Muir. The NJOY nuclear data processing system version 91. Technical Report LA-12470-M, Los Alamos National Laboratory, 1994.

[MOS66] W. Magnus, F. Oberhettinger, and R.P. Soni. *Formulas and theorems for the special functions of mathematical physics*. Springer-Verlag, 1966.

[Mos86] R. Mosier. Root neighborhoods of a polynomial. *Mathematics of Computation*, 47(175):265–273, 7 1986.

[Nev79] P. Nevai. Orthogonal polynomials. volume 18 of *Memoirs of the American Mathematical Society*. American Mathematical Society, 1979.

[OM91] T. Ooura and M. Mori. The double exponential formula for oscillatory functions over the half infinite interval. *Journal of Computational and Applied Mathematics*, 38:353–360, 1991.

[ON85] K.O. Ott and R.J. Neuhold. *Introductory Nuclear Reactor Dynamics*. American Nuclear Society, 1985.

[OS83] F. Olver and J. Smith. Associated Legendre functions on the cut. *Journal of Computational Physics*, 51:502–518, 1983.

[PdDKÜK83] R. Piessens, E. de Doncker-Kapenga, C. Überhuber, and D. Kahaner. *Quadpack, a subroutine package for automatic integration*. Springer Series in computational mathematics. Springer-Verlag, 1983.

[PTVF96] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C*. Cambridge University Press, 2nd edition, 1996.

[San59] G. Sansone. *Orthogonal Functions*, volume 9 of *Pure and Applied Mathematics*. Interscience Publishers, inc, New York, 1959.

[SG99] J. Segura and A. Gil. Evaluation of associated Legendre functions off the cut and parabolic cylinder functions. *Electronic Transactions on Numerical Analysis*, 9:137–146, 1999.

[Sie80] C. Siewert. On computing eigenvalues in radiative transfer. *Journal of Mathematical Physics*, 21(9):2468–2470, 1980.

[Sik82] K. Sikorski. Quadrature algorithms in H_p spaces. *Numerische Mathematik*, 39:405–410, 1982.

[SN04] J. Seward and N. Nethercote. *Valgrind, version 2.2.0*, 2004. `http://valgrind.org`.

[SS64] T. Sag and G. Szekeres. Numerical evaluation of high-dimensional integrals. *Mathematics of Computation*, 18:245–253, 1964.

[SS84] K. Sikorski and F. Stenger. Optimal quadratures in H_p spaces. *ACM Transactions on Mathematical Software*, 10(2):140–151, 1984.

[Ste81] F. Stenger. Numerical methods based on Whittaker cardinal, or sinc functions. *SIAM Review*, 23(2):165–224, 1981.

[Str97] Bjarne Stroustrup. *The C++ programming language*. Addison-Wesley, 1997.

[Sze59] Gabor Szegö. *Orthogonal polynomials*, volume 23 of *Colloquium publications*. American Mathematical Society, 2nd edition, 1959.

[Tho97] W. Thompson. *Atlas for computing mathematical functions*. John Wiley and sons, New York, 1997.

[TM74] H. Takahasi and M. Mori. Double exponential formulas for numerical integration. *Publ. RIMS Kyoto University*, 9:721–741, 1974.

[TT94] K-C. Toh and L.N. Trefethen. Pseudozeros of polynomials and pseudospectra of companion matrices. *Numerische Mathematik*, 68:403–425, 1994.

[Uhl92] F. Uhlig. Are the coefficients of a polynomial well-conditioned functions of its roots? *Numerische Mathematik*, 61:383–393, 1992.

[VA90] W. Van Assche. Asymptotics for orthogonal polynomials and three-term recurrences. In P. Nevai, editor, *Orthogonal Polynomials*. Kluwer Academic Publishing, 1990.

[VdE01] G. Van den Eynde. HEXNODYN, a time-dependent three-dimensional hexagonal-z transport code. Master's thesis, 2001.

[VdE03] G. Van den Eynde. Calculating the MYRRHA core using DIF3D 8.0. Technical Report gvde.34.B043200/85-60, 2003.

[VdV95] A. Van de Velde. *A multidimensional boundary sources method*. PhD thesis, 1995.

[Wag89] R. Wagner. Three-dimensional nodal diffusion and transport theory methods for hexagonal-z geometry. *Nuclear Science and Engineering*, 103:377–391, 1989.

[Wei] J. Weidendorfer. KCachegrind. `http://kcachegrind.sf.net`.

[Wen89] E. Weniger. Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. *Comput. Phys. Rep.*, 10:189–371, 1989.

[WFS04a] The Wolfram Functions Site. Legendre function of the second kind. `http://functions.wolfram.com/HypergeometricFunctions/LegendreQGeneral`, 2004.

[WFS04b] The Wolfram Functions Site. Productlog. `http://functions.wolfram.com/ElementaryFunctions/ProductLog/`, 2004.

[Wyn66] P. Wynn. On the convergence and stability of the epsilon algorithm. *J. SIAM Numerical Analysis*, 3(1):91–122, 1966.

[ZJ96] S. Zhang and J. Jin. *Computation of special functions*. John Wiley and sons, New York, 1996.