

From Discrete to Continuous Spectra

Exploring Spectral Distribution for Schrödinger Operators on Finite and Infinite Intervals

Christopher J. Winfield

We study the distribution of eigenspectra for operators of the form $-y'' + q(x)y$ with self-adjoint boundary conditions on both bounded and unbounded interval domains. With integrable potentials q , we explore computational methods for calculating spectral density functions involving cases of discrete and continuous spectra where discrete eigenvalue distributions approach a continuous limit as the domain becomes unbounded. We develop methods from classic texts in ODE analysis and spectral theory in a concrete, visually oriented way as a supplement to introductory literature on spectral analysis. As a main result of this study, we develop a routine for computing eigenvalues as an alternative to `NDEigenvalues`, resulting in fast approximations to implement in our demonstrations of spectral distribution.

■ Introduction

We follow methods of the texts by Coddington and Levinson [1] and by Titchmarsh [2] (both publicly available online via archive.org) in our study of the operator $\mathcal{L}[y] \stackrel{\text{def}}{=} -y'' + q(x)y$ and the associated problem

$$\mathcal{L}[y] = \lambda y, \tag{1}$$

where $y' \stackrel{\text{def}}{=} \frac{dy}{dx}$ on the interval $\mathcal{I} = (0, \infty)$ with real parameter λ and boundary condition

$$\sin(\alpha)y(0) - \cos(\alpha)y'(0) = 0 \tag{2}$$

for fixed α , where $0 \leq \alpha < \pi$. For continuous $g \in L^1(\mathcal{I})$ (the set of absolutely integrable functions on \mathcal{I}), we study the spectral function $\rho(\lambda)$ associated with (1) and (2) using two main methods: First, following [1], we approximate ρ by step functions associated with related eigenvalue problems on finite intervals $I_b \stackrel{\text{def}}{=} [0, b]$ for some sufficiently large positive b ; then, we apply asymptotic solution estimates along with an explicit formula for spectral density $\frac{d\rho}{d\lambda}$ [2]. For some motivation and clarification of terms, we recall a major application: For certain solutions $\psi(x, \lambda)$ of (1) and (2) and for any $f \in L^2(\mathcal{I})$ (the set of square-integrable functions on \mathcal{I}), a corresponding solution to (1) may take the form

$$f(x) = \int_0^{+\infty} g(\lambda) \psi(x, \lambda) d\rho(\lambda) = \int_0^{+\infty} g(\lambda) \psi(x, \lambda) \frac{d\rho(\lambda)}{d\lambda} d\lambda$$

where

$$g(\lambda) = \int_{\mathcal{I}} \psi(x, \lambda) f(x) dx$$

(in a sense described in Theorem 3.1 of Chapter 9 [1]); here, g is said to be a spectral transform of f . By way of such spectral transforms, the differential operator \mathcal{L} may be represented alternatively in the integral form

$$\int_{\mathcal{I}} \psi(x, \lambda) \mathcal{L}[f(x)] dx = \lambda g(\lambda),$$

where ρ induces a measure by which $g \in L^2(\mathcal{I}, d\rho)$ (roughly, the set of square-integrable functions when integrated against $d\rho$) and by which Parseval's equality holds. Typical examples are the complete set of orthogonal eigenfunctions $\sin(n\pi x/b)$: $n = 1, 2, \dots$ for $\alpha = \frac{\pi}{2}$ and the corresponding Fourier sine transform in the limiting case $b = +\infty$ (cf. Chapter 9, Section 1 [1]).

For a fixed, large finite interval I_b , we consider the problem (1), (2) along with the boundary condition

$$\cos(\beta) y(b) - \sin(\beta) y'(b) = 0, \quad (3)$$

($0 < \beta \leq \pi$), which together admit an eigensystem with correspondence

$$\lambda_k \iff \psi_k(x), k = 1, 2, \dots,$$

where the eigenvalues λ_k satisfying $\Delta\lambda_k \stackrel{\text{def}}{=} \lambda_{k+1} - \lambda_k > 0$ and where the eigenfunctions $\psi_k(x)$ form a complete basis for $L^2(I_b)$. Since the associated spectral function $\rho_b(\lambda)$ is a step function with jumps at the various λ_k , we first estimate these λ_k by way of a related equation arising from Prüfer (phase-space) variables and compute the corresponding jumps $\Delta\rho_b(\lambda_k) = \|\psi_k\|^{-2}$.

Then, we use interpolation to approximate the continuous spectral function $\rho(\lambda)$ using data from a case of large b at points λ_k and using

$$\left. \frac{d\rho(\lambda)}{d\lambda} \right|_{\lambda=\lambda_k} \approx \frac{\Delta\rho_b(\lambda_k)}{\Delta\lambda_k} = \frac{1}{\|\psi_k\|^2 (\lambda_{k+1} - \lambda_k)}, \quad (4)$$

imposing the condition $\rho(\lambda) \equiv 0$ for all $\lambda \leq 0$.

We compare our results with those of a well-known formula [2] appropriate to our case on \mathcal{I} , which we outline as follows: For fixed $\lambda > 0$, let $\psi(x, \lambda)$ be the solution to (1) with boundary values

$$\psi(0, \lambda) = \cos(\alpha); \quad \psi'(0, \lambda) = \sin(\alpha),$$

for which the asymptotic formula

$$\psi(x, \lambda) = A(\lambda) \cos(\sqrt{\lambda} x) + B(\lambda) \sin(\sqrt{\lambda} x) + o(1) \quad (5)$$

holds as $x \rightarrow \infty$. Then we have

$$\frac{d\rho(\lambda)}{d\lambda} = \frac{1}{\pi \sqrt{\lambda} (A^2(\lambda) + B^2(\lambda))} \quad (6)$$

from Section 3.5 [2].

Finally, in the last section, we apply the above techniques to extend our study to operators on large domains $[-b, b]$ and on \mathbb{R} , where spectral matrices take the place of spectral functions as a matrix analog of spectral transforms on these types of intervals (cf. equation (5.5) [1]). The techniques are described in detail below, but it is of particular interest that our computations uncover an interesting pattern in a discrete-spectrum case, as we are forced to reformulate our approach according to certain eigen-subspaces involved: our desired spectral approximations are resolved by way of an averaging procedure in forming Riemann sums.

Various sections of Chapters 7–9 [1] (see also [3] and related articles) present useful introductory discussion applied to material presented in this article; yet, with our focus on equations (1)–(6), one may proceed given basic understanding of Riemann–Stieltjes integration along with knowledge of ordinary differential equations and linear algebra, commensurate with (say) the use of `EigenSystem` and `NDSolve`.

■ An Eigenvalue Estimator

We compute eigenvalues by first computing solutions $\theta(x, \lambda)$ on $I_b \times \mathbb{R}$ to the following, arising from Prüfer variables (equation 2.4, Chapter 8 [1]):

$$w' = \cos^2 w + (\lambda - q(x)) \sin^2 w; \quad w(0) = \arctan\left(\frac{y(0)}{y'(0)}\right) = \frac{\pi}{2} - \alpha. \quad (7)$$

Here, $\tan(\theta) = \frac{y}{y'}$, where y is a nontrivial solution to (1), (2) and (3) and θ satisfies

$$\theta(b, l)|_{l=\lambda_k} = \beta + (k - 1)\pi \quad (8)$$

for positive integers k . We interpolate to approximate such solutions as an efficient means to invert (8) in the variable l . And we use the following function on (7) throughout this article.

```
In[1]:= Pruefer[initial_, a_, b_, q_] := ParametricNDSolveValue[
  {
    y'[x] == Cos[y[x]]^2 + (1 - q) Sin[y[x]]^2,
    y[a] == initial
  }, y[b], {x, a, b}, {1}]
```

Consider an example with $a = 0$, $b = 10$, and potential $q(x) = e^{-x}$ for parameter l with $0 \leq l \leq L$, $L = 5$, in the case $\alpha = \frac{\pi}{2}$, $\beta = \pi$.

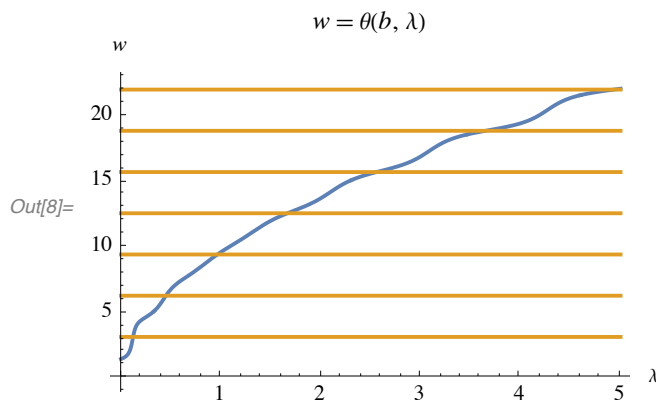
```
In[2]:=  $\alpha 1 = \text{Pi} / 2$ ;
a1 = 0;
b1 = 10;
q1 = Exp[-x];
solution1 = Pruefer[ArcTan[Sin[ $\alpha 1$ ], Cos[ $\alpha 1$ ]], a1, b1, q1];
```

We create an interpolation approximation for eigenvalues λ_k .

```
In[3]:=  $\beta 1 = \text{Pi}$ ;
L1 = 5;
 $\Theta$ Approximation1 =
  Interpolation[Table[{l, solution1[l]}, {l, 0, L1, .001}]];
EigenvaluesApproximation1[k_] :=
  l /. NSolve[ $\Theta$ Approximation1[l] ==  $\beta 1 + (k - 1) \text{Pi}$ , l][[1]]
```

It is instructive to graphically demonstrate the theory behind this method. Here, we consider the eigenvalues as those values of λ where the graph of $w = \theta(b, \lambda)$ intersects the various lines $w = \beta + (k - 1)\pi$ as we use Floor to find k_{\max} (or MaxIndex), our maximum index k , depending on L .

```
In[7]:= MaxIndex1 = Floor[( $\Theta$ Approximation1[L1] -  $\beta 1$ ) / Pi] + 1;
Plot[{ $\Theta$ Approximation1[l],
  Table[ $\beta 1 + (k - 1) \text{Pi}$ , {k, MaxIndex1}]}, {l, 0, L1},
  AxesLabel -> { $\lambda$ , w}, PlotLabel -> w ==  $\theta$ [b,  $\lambda$ ]
```



We choose these boundary conditions so that we may compare our results with those of `NDEigenvalues` applied to the corresponding problem (1) and (2) using `DirichletCondition`.

```
In[9]:= operator[q_] := -y''[x] + q y[x];
WolframEigenvalues =
NDEigenvalues[
  {operator[q1], DirichletCondition[y[x] == 0, True]},
  y[x], {x, a1, b1}, MaxIndex1]
```

```
Out[10]= {0.121624, 0.450057, 0.963838,
  1.66557, 2.56019, 3.65123, 4.9415}
```

```
In[11]:= CoddingtonLevinsonEigenvalues =
Table[Quiet@EigenvaluesApproximation1[k], {k, MaxIndex1}]
```

```
Out[11]= {0.121623, 0.45005, 0.963772,
  1.66522, 2.55891, 3.64748, 4.93221}
```

We now compare and contrast the methods in this case. The percent differences of the corresponding eigenvalues are all less than 0.2%, even within our limits of accuracy.

```
In[12]:= Table[
  Abs[WolframEigenvalues[[n]] -
  CoddingtonLevinsonEigenvalues[[n]]] /
  (Mean[{WolframEigenvalues[[n]],
  CoddingtonLevinsonEigenvalues[[n]]})], {n, MaxIndex1}]
```

```
Out[12]= {2.25034 × 10-6, 0.000015286, 0.0000677834,
  0.000207569, 0.000500677, 0.00102788, 0.0018808}
```

In contrast, our interpolation method allows some direct control of which eigenvalues are to be computed, whereas `NDEigenvalues` (in the default setting) outputs a list up to 39 values, starting from the first. Moreover, our method admits nonhomogeneous boundary conditions, where `NDEigenvalues` admits only homogeneous conditions, Dirichlet or Neumann.

■ Spectral Density: Discrete Approximation

We proceed to build our approximate spectral density function $\frac{d\rho}{d\lambda}$ for the problem (1) and (2) on \mathcal{I} with the same potential q as above. We compute eigenvalues likewise but now on a larger interval $[0, b]$ for $b = 150$ and with nonhomogeneous boundary conditions, say given by $\alpha = \frac{3\pi}{4}, \beta = \frac{\pi}{4}$ (albeit ρ does not depend on β).

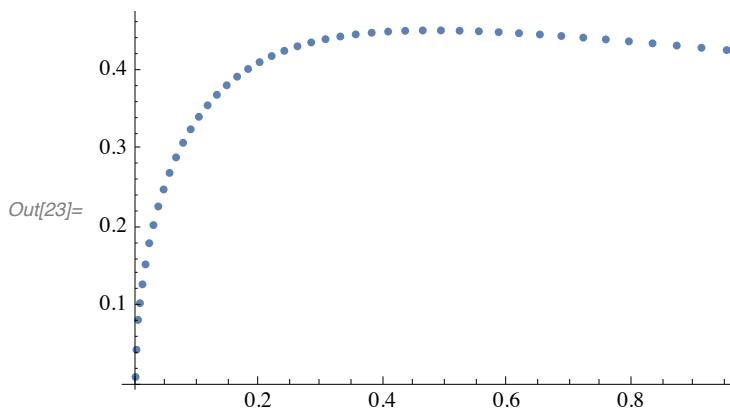
```
In[13]:= a2 = 0; b2 = 150; α2 = 3 Pi / 4; β2 = Pi / 4; L2 = 1; q2 = q1;
```

We compute eigenvalues via our interpolation method and compute a minimum k_{\min} (or `MinIndex`) as well as a maximum index so as to admit only positive eigenvalues; ρ is supported on $[0, \infty)$ and negative eigenvalues result in dubious approximations by $\rho_b(\lambda)$.

```
In[14]:= solution2 = Pruefer[ArcTan[Sin[α2], Cos[α2]], a2, b2, q2];
θApproximation2 =
  Interpolation[Table[{1, solution2[1]}, {1, 0, L2, .0005}]];
λ2[k_] :=
  Quiet[
    1 /. FindRoot[θApproximation2[1] == β2 + (k - 1) Pi,
      {1, 0, -1, 1}][[1]];
  MaxIndex2 = Floor[(θApproximation2[L2] - β2) / Pi] + 1;
  ApproximateEigenvalues2 = Table[λ2[k], {k, MaxIndex2}];
  MinIndex2 = Module[{k}, k = 1;
    While[ApproximateEigenvalues2[[k]] <= 0, k++];
    k];
```

We now compute the values $\frac{r_k^2}{\Delta\lambda_k}$.

```
In[19]:= ψSolution2 = ParametricNDSolve[
  {operator[q2] == 1 y[x], y[0] == Cos[α2], y'[0] == Sin[α2]},
  y, {x, 0, b2}, {1}];
Δeigenvalues2[k_] := ApproximateEigenvalues2[[k + 1]] -
  ApproximateEigenvalues2[[k]];
rSquared2[k_] :=
  1 /
  (NIntegrate[
    (y[ApproximateEigenvalues2[[k]]][x] /. ψSolution2)^2,
    {x, 0, b2}]);
DensityFunction2[k_] := rSquared2[k] / Δeigenvalues2[k];
CoddingtonLevinsonPointPlot2 =
  ListPlot[
    Table[{ApproximateEigenvalues2[[k]], DensityFunction2[k]},
      {k, MinIndex2, MaxIndex2 - 1}]]
```



■ Fitting Method

We now apply the method of [2] as outlined in equation (6). We use `FindFit` to include data from an interval near the endpoint $x = b$ that includes at least one half-period of the period of the fitting functions $\sin(\sqrt{\lambda} x)$ and $\cos(\sqrt{\lambda} x)$.

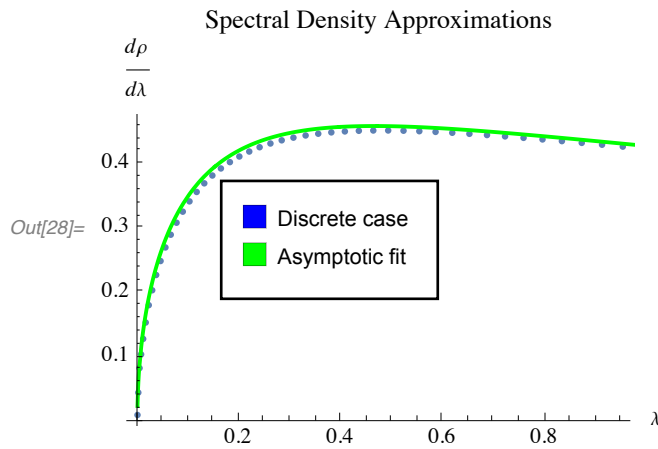
```
In[24]:= FitTable2[k_] :=
  Table[{x, y[ApproximateEigenvalues2[[k]]][x] /.
    ψSolution2},
  {x, b2 - Min[2 Pi / (Sqrt[ApproximateEigenvalues2[[k]]]),
    b2 / 2], b2,
  Min[2 Pi / (Sqrt[ApproximateEigenvalues2[[k]]]), b2 / 2] /
  10}];
coefficient[k_] := FindFit[FitTable2[k],
  AA Cos[Sqrt[ApproximateEigenvalues2[[k]]] x] +
  BB Sin[Sqrt[ApproximateEigenvalues2[[k]]] x],
  {AA, BB}, x];
TInterpolation =
  Interpolation[
  Table[{ApproximateEigenvalues2[[k]],
    1 / (Pi Sqrt[ApproximateEigenvalues2[[k]]]
      ((AA / . coefficient[k]) ^ 2 +
      (BB / . coefficient[k]) ^ 2))},
  {k, MinIndex2, MaxIndex2}}];
TPlot = Plot[TInterpolation[x],
  {x, ApproximateEigenvalues2[[MinIndex2]],
  ApproximateEigenvalues2[[MaxIndex2]]}, PlotRange → Full,
  PlotStyle → Green];
```

The function `ApproximateEigenvalues` may return non-numerical results among the first few, in which case we recommend that either b or L be readjusted or that `MinIndex` be set large enough to disregard such results.

We now compare our results of the discrete and continuous (asymptotic fit) spectral density approximations.

```

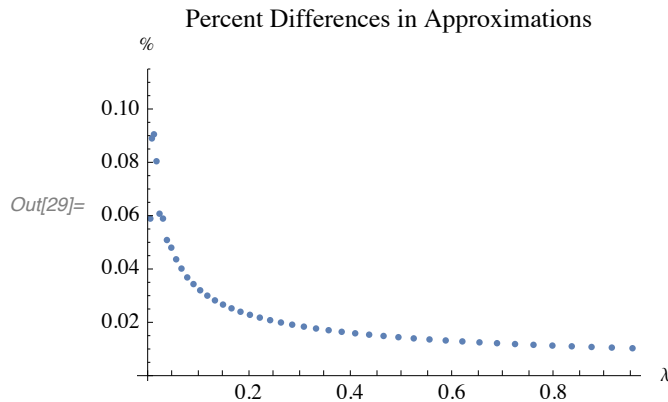
In[28]:= Show[CoddingtonLevinsonPointPlot2, TPlot,
  AxesLabel → {λ,  $\frac{\text{Row}[\{d, \rho\}]}{\text{Row}[\{d, \lambda\}]}$ },
  PlotLabel → "Spectral Density Approximations",
  Epilog →
    Inset[Framed[SwatchLegend[{Blue, Green},
      {"Discrete case", "Asymptotic fit"}]],
    Scaled[ {.4, .6} ]]
```



We compare the results by plotting percent differences, all being less than 0.1%.

```

In[29]:= ListPlot[
  Table[{ApproximateEigenvalues2[[k]],
    Abs[TInterpolation[ApproximateEigenvalues2[[k]]] -
      DensityFunction2[k]] /
    Mean[{TInterpolation[ApproximateEigenvalues2[[k]]],
      DensityFunction2[k]}]}, {k, MinIndex2, MaxIndex2 - 1}],
  AxesLabel → {λ, "%"},
  PlotLabel → "Percent Differences in Approximations"]
```



■ Check with Exact Calculation

We chose q as above because, in part, the solutions can be computed in terms of well-known `BesselI` (modified Bessel) functions. Replacing λ by $l = \lambda + i\epsilon$, for $\lambda, \epsilon > 0$, the solutions are linear combinations of

$$\begin{aligned} & \text{BesselI}\left[2i\sqrt{l}, 2\sqrt{e^{-x}}\right], \\ & \text{BesselI}\left[-2i\sqrt{l}, 2\sqrt{e^{-x}}\right]. \end{aligned} \tag{9}$$

From asymptotic estimates (cf. equation 9.6.7 [4]), we see that the former is dominant and the latter is recessive as $x \rightarrow +\infty$ when $\text{Im } l > 0$. Then, from Chapter 9 [1], equation 2.13 and Theorem 3.1, we obtain the density function by computing

$$\frac{d\rho(\lambda)}{d\lambda} = \frac{-1}{\pi} \lim_{\epsilon \rightarrow 0^+} \left[\lim_{b \rightarrow +\infty} \varphi(b, \lambda + i\epsilon) / \psi(b, \lambda + i\epsilon) \right] \stackrel{\text{def}}{=} \frac{-1}{\pi} \lim_{\epsilon \rightarrow 0^+} m(\lambda + i\epsilon), \tag{10}$$

where $\psi(x, l)$ is a solution as above and $\varphi(x, l)$ is a solution with boundary values $\varphi(0, l) = \sin(\alpha)$, $\varphi'(0, l) = -\cos(\alpha)$. (Here, m is commonly known as the Titchmarsh–Weyl m -function.) In the following code, we produce the density function in exact form by replacing functions from (9), the dominant by 1 and the recessive by 0, to compute the inside limit and thereafter simply allowing $l = \lambda$ to be real.

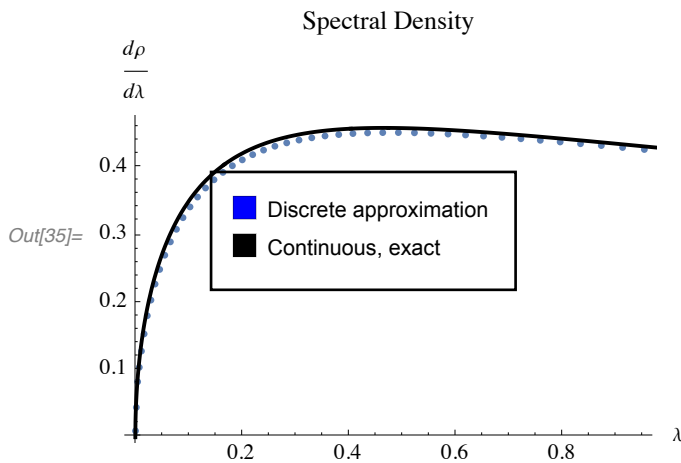
```
In[30]:=  $\psi$ Exact2 =
  y[x] /.
  DSolve[{operator[q2] == 1 y[x], y[0] == Cos[α2],
    y'[0] == Sin[α2]}, y, x][[1]];
 $\phi$ Exact2 =
  y[x] /.
  DSolve[{operator[q2] == 1 y[x], y[0] == Sin[α2],
    y'[0] == -Cos[α2]}, y, x][[1]];
ratio2 =  $\phi$ Exact2 /  $\psi$ Exact2

Out[32]:= - ( ( BesselI[-1 + 2 i √1, 2] BesselI[-2 i √1, 2 √e-x] +
  BesselI[1 + 2 i √1, 2] BesselI[-2 i √1, 2 √e-x] +
  2 BesselI[-2 i √1, 2 √e-x] BesselI[2 i √1, 2] -
  BesselI[-1 - 2 i √1, 2] BesselI[2 i √1, 2 √e-x] -
  BesselI[1 - 2 i √1, 2] BesselI[2 i √1, 2 √e-x] -
  2 BesselI[-2 i √1, 2] BesselI[2 i √1, 2 √e-x] ) /
  ( BesselI[-1 + 2 i √1, 2] BesselI[-2 i √1, 2 √e-x] +
  BesselI[1 + 2 i √1, 2] BesselI[-2 i √1, 2 √e-x] -
  2 BesselI[-2 i √1, 2 √e-x] BesselI[2 i √1, 2] -
  BesselI[-1 - 2 i √1, 2] BesselI[2 i √1, 2 √e-x] -
  BesselI[1 - 2 i √1, 2] BesselI[2 i √1, 2 √e-x] +
  2 BesselI[-2 i √1, 2] BesselI[2 i √1, 2 √e-x] ) )
```

```
In[33]:= Density2 =
  -1
  Pi
  Im[ratio2 /. {BesselI[2 i sqrt[1], 2 sqrt[e^-x]] -> 1,
    BesselI[-2 i sqrt[1], 2 sqrt[e^-x]] -> 0}]
  Im[
    -BesselI[-1-2 i sqrt[1], 2] - BesselI[1-2 i sqrt[1], 2] - 2 BesselI[-2 i sqrt[1], 2]
    -BesselI[-1-2 i sqrt[1], 2] - BesselI[1-2 i sqrt[1], 2] + 2 BesselI[-2 i sqrt[1], 2]
  ]
Out[33]= -----
                                     pi
```

We likewise compare the exact formula for the continuous spectrum with the discrete results, noting that the exact graph appears to essentially be the same as that obtained by our asymptotic fitting method (not generally expecting the fits to be accurate for small λ !).

```
In[34]:= ExactPlot2 = Plot[Density2, {1, 0, L2}, PlotRange -> Full,
  PlotStyle -> Black];
Show[CoddingtonLevinsonPointPlot2, ExactPlot2,
  AxesLabel -> {lambda, Row[{d, rho] / Row[{d, lambda}]}, PlotLabel -> "Spectral Density",
  Epilog ->
  Inset[Framed[SwatchLegend[{Blue, Black},
    {"Discrete approximation", "Continuous, exact"}]],
  Scaled[ {.45, .65} ]]
```



■ Extension to Unbounded Domains: A Proof of Concept

For the operator \mathcal{L} we now extend our study to large domains $\mathcal{I}_b \stackrel{\text{def}}{=} [-b, b]$ in the discrete-spectrum case and to the domain \mathbb{R} in the continuous-limit case. We choose an odd function potential of the form $q(x) = c x e^{-r|x|}$ for positive constants c, r . We focus on the spectral density associated with specific boundary values at $x = 0$ and an associated pair of solutions to (1): namely, we consider expansions in the pair $\phi_1(x, \lambda)$ and $\phi_2(x, \lambda)$ such that

$$\begin{aligned}\phi_1(0, \lambda) &= 1, \\ \phi_1'(0, \lambda) &= 0, \\ \phi_2(0, \lambda) &= 0, \\ \phi_2'(0, \lambda) &= 1.\end{aligned}\tag{11}$$

We apply the above computational methods to the analytical constructs from Chapter 5 [1] in both the discrete and continuous cases. First, for the discrete case, we compute spectral matrices associated with self-adjoint boundary-value problems and the pair as in (11): We estimate eigenvalues $\lambda_k : k = 1, 2, \dots$ for an alternative two-point boundary-value problem on \mathcal{I}_b for (moderately) large $b > 0$ to compute the familiar jumps of the various components $\rho_{ij;b}(\lambda)$. These components induce measures that appear in the following form of Parseval's equality for square-integrable functions f on \mathcal{I}_b (taken in a certain limiting sense):

$$\begin{aligned}g_j(\lambda) &= \int_{-b}^b f(t) \phi_j(t, \lambda) dt, \\ \int_{-b}^b |f(t)|^2 dt &= \int_{-\infty}^{\infty} \left(\sum_{i,j=1}^2 g_i(\lambda) g_j(\lambda) d\rho_{ij;b}(\lambda) \right) d\lambda\end{aligned}$$

(real-valued case). Second, we compute the various densities as limits as $b \rightarrow +\infty$ by the formulas

$$\begin{aligned}\frac{d\rho_{jk}(\lambda)}{d\lambda} &= -\frac{1}{\pi} \text{Im } M_{jk}(\lambda), \\ M_{11} &= \frac{1}{m^- - m^+}, \\ M_{12} = M_{21} &= \frac{1}{2} (m^- + m^+) M_{11}, \\ M_{22} &= m^- m^+ M_{11},\end{aligned}\tag{12}$$

where $m^+(\lambda)$ and $m^-(\lambda)$ are certain limits of m -functions, related to equation (10), but for our ODE problem on domains $[0, +\infty)$ and $(-\infty, 0]$, respectively. The densities are computed by procedures more elaborate than (6), as discussed later. Then, we compare results of the discrete case like in (4), approximating

$$\left. \frac{d\rho_{ij}(\lambda)}{d\lambda} \right|_{\lambda=\lambda_k} \approx \frac{\Delta\rho_{ij;b}(\lambda_k)}{\Delta\lambda_k}.\tag{13}$$

□ Discrete Case

After choosing (self-adjoint) boundary conditions (of which the limits ρ_{ij} happen to be independent)

$$\begin{aligned} y(-b) &= \sin(\alpha), \\ y'(-b) &= \cos(\alpha), \\ y(b) &= \sin(\beta), \\ y'(b) &= \cos(\beta), \end{aligned} \tag{14}$$

on an interval \mathcal{I}_b , we estimate eigenvalues and compute coefficients $r_{1;k}, r_{2;k}$ from the linear combinations

$$h_k(x) = r_{1;k} \phi_1(x, \lambda_k) + r_{2;k} \phi_2(x, \lambda_k)$$

for the associated orthonormal (complete) set of eigenfunctions $h_k; k = 1, 2, 3, \dots$, whereby

$$\Delta\rho_{ij;b}(\lambda_k) = r_{i;k} \cdot r_{j;k}$$

(real-valued case). Here, the functions $h_k(x)$ result by normalizing eigenfunctions $\psi_k(x)$ satisfying (14) so that we obtain

$$\begin{aligned} r_{1;k} &= \frac{\psi_k(0)}{\|\psi_k\|}, \\ r_{2;k} &= \frac{\psi_k'(0)}{\|\psi_k\|}. \end{aligned}$$

We are ready to demonstrate. Let us choose $q(x) = x e^{-5|x|}$, $b = 50\pi$ and $\alpha = \frac{\pi}{2}$, $\beta = 0$ (arbitrary). Much of the procedure follows as above, with minor modification, as we include `ParametricNDSolveValue` to obtain the values $\psi_k(0)$ and $\psi_k'(0)$ (the next result may take around three minutes on a laptop).

```
In[36]:= L3 = 2; b3 = 50 Pi; a3 = -b3; alpha3 = Pi / 2; beta3 = 0;
q3 = x Exp[-5 Abs[x]];
solution3 = Pruefer[ArcTan[Cos[alpha3], Sin[alpha3]], a3, b3, q3];
psiSolution3 = ParametricNDSolve[
  {operator[q3] == 1 y[x], y[a3] == Sin[alpha3], y'[a3] == Cos[alpha3]},
  y, {x, a3, b3}, {1}];
psi3[l_, x_] := y[1][x] /. psiSolution3;
ThetaApproximation3 =
  Quiet[Interpolation[Table[{1, solution3[1]},
    {1, 0, L3, .001}]]];
MaxIndex3 = Floor[(ThetaApproximation3[L3] - beta3) / Pi];
lambda3[n_] :=
  Quiet[1 /. NSolve[ThetaApproximation3[1] == beta3 + (n - 1) Pi, 1]][[
    1]]
eigenvalues3 = Table[lambda3[n], {n, MaxIndex3}];
```

```

MinIndex3 = Module[{k},
  k = 1;
  While[eigenvalues3[[k]] <= 0, k++]; k
];
NormSquared3[n_] := NIntegrate[ψ3[eigenvalues3[[n]], x]^2,
  {x, a3, b3}];
NormFactor3 = Table[NormSquared3[n],
  {n, MinIndex3, MaxIndex3}];
ψAtZero3 := ParametricNDSolveValue[
  {operator[q3] == 1 y[x], y[a3] == Sin[α3], y'[a3] == Cos[α3]},
  y[0], {x, a3, b3}, {1}];
ψPrimeAtZero3 := ParametricNDSolveValue[
  {operator[q3] == 1 y[x], y[a3] == Sin[α3], y'[a3] == Cos[α3]},
  y'[0], {x, a3, b3}, {1}];
r1[n_] :=
  Quiet[Evaluate[ψAtZero3[eigenvalues3[[n]]]] /
  Sqrt[NormFactor3[[n]]]];
r2[n_] :=
  Quiet[Evaluate[ψPrimeAtZero3[eigenvalues3[[n]]]] /
  Sqrt[NormFactor3[[n]]]];

```

We now approximate the density functions by plotting $(\tilde{\lambda}_k, \mathcal{D}_{i,j;k})$ where

$$\mathcal{D}_{i,j;k} \stackrel{\text{def}}{=} \Delta\rho_{i,j;b}(\lambda_k) / (\lambda_{k+2} - \lambda_k) + \Delta\rho_{i,j;b}(\lambda_{k+1}) / (\lambda_{k+3} - \lambda_{k+1}) \quad (15)$$

(for certain $K \leq k_{\max} - 3$) as we compute the difference quotients at the various jumps, over even and odd indices separately, and assign the corresponding sums $\mathcal{D}_{i,j;k}$ to the midpoints $\tilde{\lambda}_k$ of corresponding intervals $[\lambda_k, \lambda_{k+1}]$.

```

In[52]:= DifferenceQuotients11 =
  Table[{Mean[{eigenvalues3[[n]], eigenvalues3[[n+1]]}],
    r1[n]^2 / (eigenvalues3[[n+2]] - eigenvalues3[[n]]) +
    r1[n+1]^2 /
    (eigenvalues3[[n+3]] - eigenvalues3[[n+1]])},
  {n, MinIndex3, MaxIndex3 - 3, 2}];
DifferenceQuotients22 =
  Table[{Mean[{eigenvalues3[[n]], eigenvalues3[[n+1]]}],
    r2[n]^2 / (eigenvalues3[[n+2]] - eigenvalues3[[n]]) +
    r2[n+1]^2 /
    (eigenvalues3[[n+3]] - eigenvalues3[[n+1]])},
  {n, MinIndex3, MaxIndex3 - 3, 2}];
DifferenceQuotients12 =
  Table[{Mean[{eigenvalues3[[n]], eigenvalues3[[n+1]]}],
    r1[n] × r2[n] / (eigenvalues3[[n+2]] - eigenvalues3[[n]]) +
    r1[n+1] ×
    r2[n+1] /
    (eigenvalues3[[n+3]] - eigenvalues3[[n+1]])},
  {n, MinIndex3, MaxIndex3 - 3, 2}];
DiscreteCasePlot =

```

```
ListPlot[{DifferenceQuotients11, DifferenceQuotients22,
          DifferenceQuotients12},
          PlotStyle -> {Blue, Orange, Green}];
```

We give the plots below, in comparison with those of the continuous spectra, and give a heuristic argument in the Appendix as to why this approach works.

□ Continuous Case

First, we apply the asymptotic fitting method using the solutions ϕ_1 and ϕ_2 . Here, we have to compute full complex-valued formulas for the corresponding m -functions (cf. Section 5.7 [2]) where a slight modification of the derivation of m^+ , via a change of variables and a complex conjugation, results in m^- (See Appendix).

```
In[56]:=  $\phi$ 1solution = ParametricNDSolve[
          {Evaluate[operator[q3] == 1 y[x]], y[0] == 1, y'[0] == 0},
          y, {x, a3, b3}, {1}];
 $\phi$ 1[l_, x_] := y[1][x] /.  $\phi$ 1solution;
 $\phi$ 2solution = ParametricNDSolve[
          {Evaluate[operator[q3] == 1 y[x]], y[0] == 0, y'[0] == 1},
          y, {x, a3, b3}, {1}];
 $\phi$ 2[l_, x_] := y[1][x] /.  $\phi$ 2solution;
Fit $\psi$ 2Plus[l_] :=
  FindFit[Table[{x,  $\phi$ 2[l, x]},
               {x, b3 - 2 Pi / Sqrt[1 + 1], b3, 2 Pi / (15 Sqrt[1 + 1])}],
          aa Cos[x Sqrt[1]] + bb Sin[x Sqrt[1]], {aa, bb}, x]
Fit $\psi$ 1Plus[l_] :=
  FindFit[Table[{x,  $\phi$ 1[l, x]},
               {x, b3 - 2 Pi / Sqrt[1 + 1], b3, 2 Pi / (15 Sqrt[1 + 1])}],
          cc Cos[x Sqrt[1]] + dd Sin[x Sqrt[1]], {cc, dd}, x]
Fit $\psi$ 2Minus[l_] :=
  FindFit[Table[{x,  $\phi$ 2[l, x]},
               {x, a3, a3 + 2 Pi / Sqrt[1 + 1], 2 Pi / (15 Sqrt[1 + 1])}],
          aa Cos[x Sqrt[1]] + bb Sin[x Sqrt[1]], {aa, bb}, x]
Fit $\psi$ 1Minus[l_] :=
  FindFit[Table[{x,  $\phi$ 1[l, x]},
               {x, a3, a3 + 2 Pi / Sqrt[1 + 1], 2 Pi / (15 Sqrt[1 + 1])}],
          cc Cos[x Sqrt[1]] + dd Sin[x Sqrt[1]], {cc, dd}, x]

APlus[l_] := aa /. Fit $\psi$ 2Plus[l];
BPlus[l_] := bb /. Fit $\psi$ 2Plus[l];
CPlus[l_] := cc /. Fit $\psi$ 1Plus[l];
DPlus[l_] := dd /. Fit $\psi$ 1Plus[l];
```

```

Aminus[l_] := aa /. Fitψ2Minus[l];
Bminus[l_] := bb /. Fitψ2Minus[l];
Cminus[l_] := cc /. Fitψ1Minus[l];
Dminus[l_] := dd /. Fitψ1Minus[l];

mPlus[l_] := -(CPlus[l] + I DPlus[l]) / (APlus[l] + I BPlus[l]);
mMinus[l_] := -(CMinus[l] - I DMinus[l]) /
  (Aminus[l] - I Bminus[l]);

dρ11[l_] := 1 / (Pi (mMinus[l] - mPlus[l]));
dρ12[l_] := .5 (mPlus[l] + mMinus[l]) dρ11[l];
dρ22[l_] := mPlus[l] × mMinus[l] × dρ11[l];

AsymptoticFit = Plot[{Im[dρ11[l]], Im[dρ22[l]], Im[dρ12[l]]},
  {l, eigenvalues3[[MinIndex3]], L3},
  PlotStyle → {Blue, Orange, Green}];

```

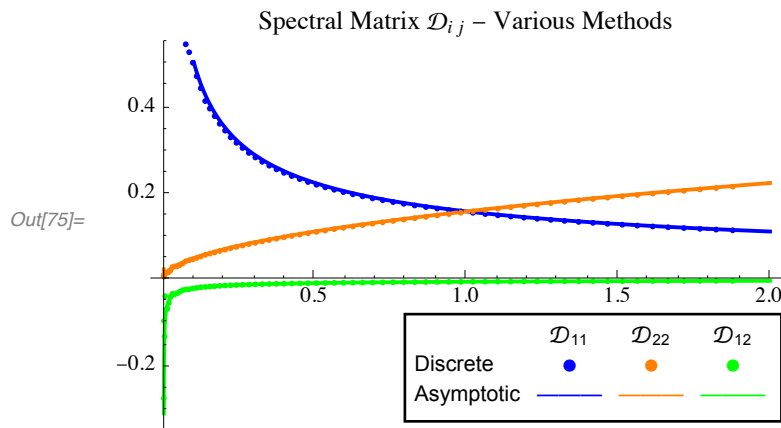
We now compare the result of the discrete and asymptotic fitting methods for the elements

$$\mathcal{D}_{ij} \stackrel{\text{def}}{=} \frac{d\rho_{ij}}{d\lambda}.$$

```

In[75]:= Show[AsymptoticFit, DiscreteCasePlot, Epilog → Inset[
  Text@Framed@Grid[{
    {"",  $\mathcal{D}_{11}$ ,  $\mathcal{D}_{22}$ ,  $\mathcal{D}_{12}$ },
    {"Discrete", Style["●", Blue, 8],
      Style["●", Orange, 8], Style["●", Green, 8]},
    {"Asymptotic", Style["—", Blue],
      Style["—", Orange], Style["—", Green]}
  }, Alignment → {{Left, Center, Center}, Automatic}],
  Scaled[{0.7, 0.167}]],
  PlotLabel → "Spectral Matrix  $\mathcal{D}_{ij}$  - Various Methods",
  ImageSize → {400, 300}]

```



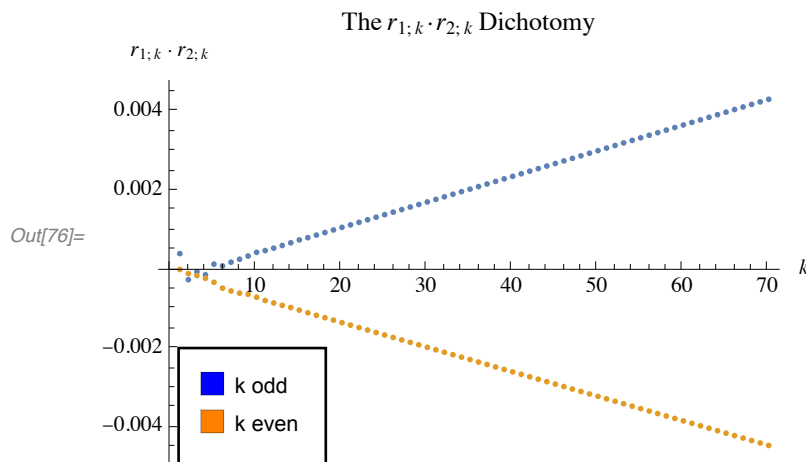
■ Appendix

We have deferred some discussion on our use of `Quiet`, comparison of eigenvalue computations, discrete eigenspace decomposition and Weyl m -functions to this section.

First, we have used `Quiet` to suppress messages warning that some solutions may not be found. From Chapter 8 [1], we expect unique solutions since the functions $\theta(b, \cdot)$ are strictly increasing. We have also used `Quiet` to suppress various messages from `ParametricNDSolve` and other related functions regarding small values of $q(x)$ to be expected with short-range potentials and large domains.

Second, our formulation of \mathcal{D}_{ij} and the midpoints $\tilde{\lambda}_k$ as in (15) arises from a decomposition of the eigenspace by even and odd indices. We motivate this decomposition by an example plot of the values $r_{1k} \cdot r_{2k}$, where the dichotomous behavior is quite pronounced, certainly for large k .

```
In[76]:= ListPlot[{Table[r1[n] × r2[n], {n, 1, MaxIndex3, 2}],
  Table[r1[n] × r2[n], {n, 2, MaxIndex3, 2}]],
  Epilog →
  Inset[Framed[SwatchLegend[{Blue, Orange},
    {"k odd", "k even"}]], Scaled[ {.15, .15} ]],
  PlotLabel → "The r1;k · r2;k Dichotomy",
  AxesLabel → {k, "r1;k · r2;k"}]
```



We are thus inspired to compute the quotients over even and odd indices separately. Then, we consider, say, a relevant expression from Parseval's equality: for appropriate Fourier coefficients $g_{i;k}$, $i = 1, 2$, associated with respective solutions ϕ_i , we write

$$\sum_{k=1}^{\infty} g_{i;k} g_{j;k} \Delta \rho_{i;j;b}(\lambda_k) = \sum_{k \text{ odd}} (g_{i;k} g_{j;k} \Delta \rho_{i;j;b}(\lambda_k) + g_{i;k+1} g_{j;k+1} \Delta \rho_{i;j;b}(\lambda_{k+1})) \approx \sum_{k \text{ odd}} g_i(\tilde{\lambda}_k) g_j(\tilde{\lambda}_k) \mathcal{D}_{i;j;k} \Delta \lambda_k.$$

We suppose that $\lambda_{k+2} - \lambda_k \approx 2 \Delta \lambda_k$ and $g_{i,k} \approx g_{i,k+1} \approx g_i(\tilde{\lambda}_k)$ for the corresponding transforms $g_i(\lambda)$ in the limit $b \rightarrow \infty$. Of course, a rigorous argument is beyond the scope of this article.

Finally, we elaborate on the calculations of the m -functions m^+ and m^- : Given the asymptotic expressions

$$\begin{aligned}\phi_1^\pm(x, \lambda) &= a^\pm(\lambda) \cos(\sqrt{\lambda} x) + b^\pm(\lambda) \sin(\sqrt{\lambda} x) + o(1), \\ \phi_2^\pm(x, \lambda) &= c^\pm(\lambda) \cos(\sqrt{\lambda} x) + d^\pm(\lambda) \sin(\sqrt{\lambda} x) + o(1),\end{aligned}$$

as $x \rightarrow \pm \infty$ (resp.), we follow Section 5.7 of [2], making changes as needed, with a modification via complex conjugation ($l \rightarrow \lambda - i \epsilon$, say) for m^- to arrive at

$$m^\pm(\lambda) = -\frac{c^\pm(\lambda) \pm i d^\pm(\lambda)}{a^\pm(\lambda) \pm i b^\pm(\lambda)} \text{ (resp.)}.$$

■ Acknowledgments

The author would like to thank the members of MAST for helpful and motivating discussions concerning preliminary results of this work in particular and Mathematica computing in general.

■ References

- [1] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*, New York: McGraw-Hill, 1955. archive.org/details/theoryofordinary00codd.
- [2] E. C. Titchmarsh, *Eigenfunction Expansions Associated with Second-Order Differential Equations*, 2nd ed., London: Oxford University Press, 1962. archive.org/details/eigenfunctionexp0000titc.
- [3] E. W. Weisstein. "Operator Spectrum" from MathWorld—A Wolfram Web Resource. mathworld.wolfram.com/OperatorSpectrum.html.
- [4] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Wiley, 1972.

C. Winfield, "From Discrete to Continuous Spectra," *The Mathematica Journal*, 2019. <https://doi.org/10.3888/tmj.21-3>.

About the Author

C. Winfield holds an MS in physics and a PhD in mathematics and is a member of the Madison Area Science and Technology amateur science organization, based in Madison, WI.

Christopher J. Winfield

Madison Area Science and Technology
3783 US Hwy. 45
Conover, WI 54519
cjwinfield2005@yahoo.com