Spectral differentiation matrices for the numerical solution of Schrödinger's equation

2006 J. Phys. A: Math. Gen. 39 10229

(http://iopscience.iop.org/0305-4470/39/32/S21)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **39** (2006) 10229–10237 doi:10.1088/0305-4470/39/32/S21

Spectral differentiation matrices for the numerical solution of Schrödinger's equation

J A C Weideman

Applied Mathematics, University of Stellenbosch, Stellenbosch 7600, South Africa

E-mail: weideman@dip.sun.ac.za

Received 15 February 2006
Published 26 July 2006
Online at stacks.iop.org/JPhysA/39/10229

Abstract
Schrödinger's equation is solved numerically using the spectral collocation (pseudospectral) method based on Hermite weighted polynomial interpolants. Several sets of numerical results from the literature are reproduced using this approach. MATLAB codes that can serve as templates for further exploration are provided both in the paper and online. A new proposal is made for solving Schrödinger's equation in Stokes wedges.

PACS numbers: 02.70.Hm, 02.70.Jn

1. Introduction

As a model problem, we consider Schrödinger's equation in the form

$$-y''(x) + p(x)y = Ey, \qquad -\infty < x < \infty, \tag{1}$$

under boundary conditions

$$\lim_{|x|\to\infty} y(x) = 0. \tag{2}$$

The aim is to compute approximate eigenvalues, E, and eigenfunctions, $y(x)$.

To start consider the quadratic potential $p(x) = x^2$. In this case (1) can be solved in closed form. The eigenvalues and (unnormalized) eigenfunctions are given by

$$E_k = 2k + 1, \qquad y_k(x) = \mathrm{e}^{-x^2/2} H_k(x), \qquad k = 0, 1, 2, \ldots, \tag{3}$$

where $H_k(x)$ is the Hermite polynomial of degree k [4, p 28].

Given the form of the above solution, it seems reasonable to base an approximation method for arbitrary $p(x)$ on the expansion

$$y(x) = \mathrm{e}^{-x^2/2} \sum_{k=0}^{\infty} a_k H_k(x), \qquad -\infty < x < \infty, \tag{4}$$

where the coefficients a_k are to be found. Classic algorithms for computing a_k are associated with the names of Galerkin, Rayleigh, Ritz and others.

What we would like to advertise here, however, is the method of collocation, which has advantages that should become clear shortly. Firstly, consider a grid $\{x_k\}_{k=0}^N$ consisting of $N+1$ distinct nodes on the real line. Define the associated Lagrange cardinal interpolation polynomials by

$$L_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}, \qquad k = 0, 1, \ldots, N.$$

Now suppose that the infinite series in (4) is truncated at $k = N$. Writing the resulting polynomial of degree N in Lagrange form, we obtain the approximation

$$y(x) \approx \mathrm{e}^{-x^2/2} \sum_{k=0}^{N} \mathrm{e}^{x_k^2/2} y_k L_k(x), \qquad -\infty < x < \infty, \tag{5}$$

where we have defined $y_k \equiv y(x_k)$. It is easy to check that the right-hand side of (5) reduces to $y(x_k)$ when $x = x_k$; this verifies that (5) is a weighted polynomial interpolant of $y(x)$.

It should be possible to relate the coefficients a_k in (4) to the function values y_k in (5), but this is not necessary. We prefer the representation (5), since nonconstant coefficients such as those in (1) can be dealt with easily (see section 2). In the numerical analysis literature, methods based on weighted interpolants such as (5) are referred to as spectral collocation or pseudospectral methods [6, 11, 15].

The outline of the paper is as follows. In section 2, we introduce the spectral collocation method and the concept of a differentiation matrix. We also mention the MATLAB package DMSUITE, which can be used to construct such differentiation matrices. To demonstrate the usage of these methods, we compute the spectra of three potentials: a sextic potential (section 3), a PT-symmetric potential (section 4) and a *QES* potential (section 5). (The abbreviations PT and *QES* stand for 'parity and time' and 'quasi-exactly solvable', respectively. But, as we focus primarily on numerical aspects here, these labels are mentioned only in passing.)

2. Differentiation matrices

Consider a weighted polynomial interpolant such as (5), but with arbitrary weight $w(x)$,

$$y_N(x) = \sum_{k=0}^{N} \frac{y_k}{w(x_k)} (w(x) L_k(x)).$$

Using this as an approximation to the eigenfunction in (1) gives

$$-y_N''(x) + p(x) y_N(x) \approx E y_N(x).$$

In the collocation method, one requires that equality holds at each of the nodes $x = x_j$, $j = 0, \ldots, N$. That is,

$$-\sum_{k=0}^{N} \frac{y_k}{w(x_k)} (w(x) L_k(x))''_{x=x_j} + p(x_j) y_j = E y_j,$$

which can be summarized in matrix notation as

$$-D^{(2)} \boldsymbol{y} + \mathrm{diag}(p(x_j)) \boldsymbol{y} = E \boldsymbol{y}. \tag{6}$$

Table 1. MATLAB commands for computing the spectrum of $p(x) = x^2$.

```
>> N = 5; c = 1;
>> [x, D] = herdif(N, 2, c);
>> D2 = D(:,:,2);
>> A = -D2+diag(x.^2);
>> E = sort(eig(A))

E =
   1.00000000000000
   3.00000000000000
   4.99999999999999
   7.00000000000000
   8.99999999999999
```

Here, $\boldsymbol{y}$ is the $(N+1) \times 1$ vector of (unknown) function values $\{y_j\}_{j=0}^N$ at the nodes $\{x_j\}_{j=0}^N$. The $(N+1) \times (N+1)$ matrix $D^{(2)}$, with entries

$$D^{(2)}_{j,k} = (w''(x_j)L_k(x_j) + 2w'(x_j)L'_k(x_j) + w(x_j)L''_k(x_j))/w(x_k), \tag{7}$$

is referred to as a spectral (second) derivative matrix. The diagonal matrix $\mathrm{diag}(p(x_j))$ contains values of the potential sampled at the nodes.

To compute approximate eigenvalues and eigenfunctions, we diagonalize the coefficient matrix in (6), namely

$$A = -D^{(2)} + \mathrm{diag}(p(x_j)). \tag{8}$$

This can be done, for example, with the QR-algorithm. Note that A is a dense and unsymmetric matrix, but otherwise is well behaved. (For example, in the case $p \equiv 0$ and with the Hermite weight all eigenvalues of A are real and positive, with the smallest $O(1)$ and the largest growing only as $O(N)$, $N \to \infty$; see [13].)

An efficient algorithm for constructing differentiation matrices such as (7) has been proposed by Welfert [16]. It has since been implemented in a MATLAB function called `poldif`, which is part of the DMSUITE package [15]. The code takes as input the nodes $\{x_j\}_{j=0}^N$, the weights $\{w(x_j)\}_{j=0}^N$, $\{w'(x_j)\}_{j=0}^N$, $\{w''(x_j)\}_{j=0}^N$, ..., and produces matrices $D^{(\ell)}$ of any specified order ℓ.

For a given $w(x)$ one could, in principle, use an arbitrary set of nodes $\{x_j\}_{j=0}^N$. Such ad hoc choices do not lead to well-conditioned matrices $D^{(\ell)}$, however. One good strategy is to choose, for a particular $w(x)$, the nodes $\{x_j\}_{j=0}^N$ to be the zeros of the $(N+1)$st orthogonal polynomial associated with that weight. In the Hermite case, we thus use the zeros of $H_{N+1}(x)$. DMSUITE contains a function, `herroots`, which computes these zeros, and a function, `herdif`, which assembles the differentiation matrices.

There is one last technicality. The infinite line can be mapped to itself by a scaling $x \mapsto cx$, for any $c > 0$. The parameter c can be exploited to optimize numerical accuracy. To incorporate this, we wrote `herdif` such that it actually implements the weight $w(x) = \mathrm{e}^{-\frac{1}{2}(cx)^2}$ and uses as nodes the zeros of $H_{N+1}(x)$ scaled by the factor c.

In summary, the Hermite differentiation matrix of order $(N+1) \times (N+1)$ produced by `herdif` yields exact derivatives (ignoring round-off error) for functions of the form

$$y(x) = \mathrm{e}^{-\frac{1}{2}(cx)^2} h(x),$$

with $h(x)$ any polynomial of degree N or less.

Table 1 demonstrates the usage of `herdif`. It shows the actual command-line solution of the quadratic oscillator problem; cf (3). The code computes the 5×5 Hermite second

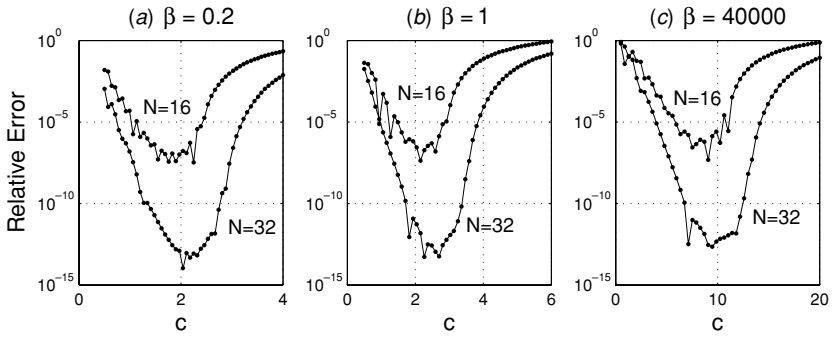

Figure 1. Relative errors in the Hermite collocation method, with scaling parameter c, when computing the ground-state eigenvalue of (9).

derivative matrix, assembles the matrix A and computes the eigenvalues using MATLAB's built-in function `eig` (which implements the QR-algorithm).

All the computed eigenvalues in table 1 are exact to within a machine round-off error, but here the solution method was tailored to the problem. In the next section, we investigate the accuracy when more general potentials are considered.

In conclusion, other quantum-mechanical computations may require different weights. On $[0, \infty)$ the Laguerre weight $w(x) = \mathrm{e}^{-x}$ is appropriate; an example involving the Woods–Saxon potential was presented in [15]. A method based on the logistic weight, $w(x) = \mathrm{sech}^2 x$ on $(-\infty, \infty)$, was applied to the Morse potential in [14].

3. A sextic potential

For our next example, we consider the sextic oscillator

$$p(x) = x^2 + \beta x^6, \qquad \beta > 0. \tag{9}$$

To demonstrate the dependence of the accuracy on the scaling parameter c, we present figure 1. It shows relative errors in the computed ground-state eigenvalues E_0 (the values of which we obtained from [9]). For all values of β considered, it is possible to reach an accuracy of at least six significant digits when $N = 16$ and at least twelve when $N = 32$, provided c is chosen well.

There exists some theory for choosing c; see [5]. Alternatively, one can proceed empirically by solving a problem with the same essential characteristics, but with a known explicit solution. For example, when one solves a Schrödinger equation, the WKBJ theory provides approximate solutions that have the same decay as $|x| \to \infty$ as the eigenfunctions [4, ch 10]. In the case of (9), we therefore take as test function

$$y(x) = \mathrm{e}^{-\sqrt{\beta}x^4/4}.$$

By numerically differentiating this function with a first derivative matrix, one can compare the result with the exact derivatives. When the maximum error at the gridpoints is plotted as a function of c, error curves are obtained that reach their minima at roughly the same values of c as for the curves in figure 1. This is a practical way of estimating near-optimal values of c for any given β and N.

Table 2. MATLAB commands for computing the real spectrum of (10).

```
N = 32; c = 1.8;              % Construct second derivative
[x, DM] = herdif(N, 2, c);    % Hermite matrix of size NxN and
D2 = DM(:,:,2);               % scaling parameter c.

for a = linspace(0,22,1000);      % Start loop over a.

    A = -D2+diag(x.^4+i*a*x);     % Assemble A and compute
    E = sort(eig(A));             % and sort its eigenvalues.
    E = E(1:10);                  % Retain only ten.

ell = find(abs(imag(E)) < sqrt(eps));     % Retain only e-values
    E = E(ell);                           % that are real (approx).

plot(a*ones(size(E)),E,'.','MarkerSize',12); hold on;   % Plot them.

end
```

4. A PT-symmetric potential

A potential that has attracted attention in the last few years is

$$p(x) = x^4 + \mathrm{i}ax, \tag{10}$$

with a real. Although $p(x)$ is complex for $a \neq 0$, the spectrum is real, discrete and positive for $|a| < a_*$, where $a_* \approx 3.17$; see [1, 2].

We list the MATLAB commands for solving this in table 2. The code uses a differentiation matrix of order 32×32 and parameter $c = 1.8$ (an empirical choice). For 1000 equidistant values of a in $[0, 22]$, the matrix A in (8) is assembled and its eigenvalues computed. Only the ten eigenvalues near the origin are retained and only if they are real. To allow for a round-off error, we have used as test for reality

$$|\mathrm{Im}\, E_k| < \sqrt{\epsilon}, \tag{11}$$

where ϵ is MATLAB's machine round-off unit (approximately 2.2×10^{-16} and a built-in constant `eps` in MATLAB). These real eigenvalues are then plotted as dots in the (a, E) plane. The result, shown in figure 2, reproduces figure 1 in [1]. (There a different numerical method was used.)

The code of table 2 can serve as a basis, for example, for computing the critical value of a. We have combined it with one of MATLAB's optimization routines to minimize the distance between E_0 and E_1. This yielded $a_* = 3.16904$, which agrees to six significant digits with the value obtained in [7]. Our code can be found in [12].

5. A QES potential

Another complex potential of interest is, for b real,

$$p(z) = -z^4 - 2bz^2 - 6\mathrm{i}z, \tag{12}$$

which is a special case of the one studied in [3]. We have switched notation from x to z to indicate that the independent variable in (1) is now complex. The limit (2) is taken inside the two wedges bounded by the Stokes lines at 0 and $-\pi/3$ on the right and $-2\pi/3$ and $-\pi$ on the left; see figure 3.

As a computational domain, we propose the use of the contour

$$z = \mathrm{i}\mu(\sin\alpha + \sin(\mathrm{i}w - \alpha)), \qquad -\infty < w < \infty, \tag{13}$$

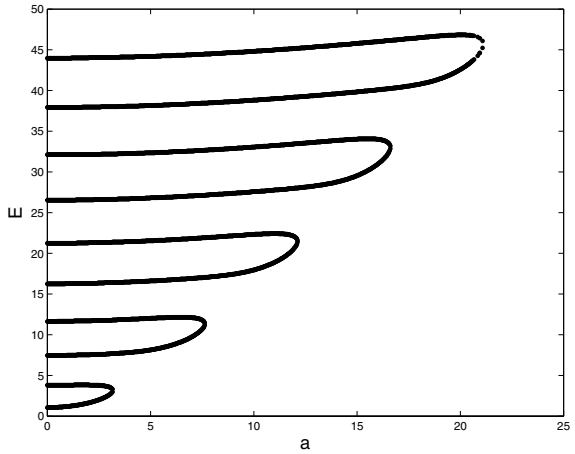

Figure 2. Real spectrum of (10) as computed by the code in table 2.

which is a parametrization of the lower branch of the hyperbola

$$\left(\frac{y-\mu\sin\alpha}{\mu\sin\alpha}\right)^2-\left(\frac{x}{\mu\cos\alpha}\right)^2=1, \qquad z=x+\mathrm{i}y.$$

Here, $\mu>0$ and α is a real parameter that can be used to adjust the hyperbola's asymptotic angle. When $0<\alpha<\pi/3$, this contour lies inside the prescribed Stokes wedges. Figure 3 shows one such contour, corresponding to $\mu=1$ and $\alpha=\pi/6$. The dots on the contour are the images of the Hermite points, with $N=64$ and $c=5$.

Equation (1) is now transplanted to the real w domain via the chain rule

$$-\left(\left(\frac{\mathrm{d}w}{\mathrm{d}z}\right)^2\frac{\mathrm{d}^2y}{\mathrm{d}w^2}+\frac{\mathrm{d}^2w}{\mathrm{d}z^2}\frac{\mathrm{d}y}{\mathrm{d}w}\right)+p(z(w))y=Ey, \qquad -\infty<w<\infty, \tag{14}$$

where

$$\frac{\mathrm{d}w}{\mathrm{d}z}=-\frac{1}{\mu}\sec(\mathrm{i}w-\alpha), \qquad \frac{\mathrm{d}^2w}{\mathrm{d}z^2}=\frac{\mathrm{i}}{\mu^2}\sec^2(\mathrm{i}w-\alpha)\tan(\mathrm{i}w-\alpha).$$

The transformed problem (14) is solved by the Hermite method, that is, we introduce a Hermite grid $\{w_j\}_{j=0}^N$ and construct differentiation matrices $D^{(1)}$ and $D^{(2)}$ to approximate $\mathrm{d}y/\mathrm{d}w$ and $\mathrm{d}^2y/\mathrm{d}w^2$. The analogue of the matrix in (8) is therefore

$$A=-\mathrm{diag}\left(\left(\frac{\mathrm{d}w}{\mathrm{d}z}(w_j)\right)^2\right)D^{(2)}-\mathrm{diag}\left(\left(\frac{\mathrm{d}^2w}{\mathrm{d}z^2}(w_j)\right)\right)D^{(1)}+\mathrm{diag}(p(z(w_j))).$$

By computing the eigenvalues of this matrix for various b, we generated figure 4, which is virtually identical to figure 1 in [3]. (There explicit formulae were used to plot the curves.)

The MATLAB code that produces figure 4 is only slightly more complicated than the code of table 2. It is therefore not reproduced here, but can be found online at [12]. A few words on implementation aspects are, however, in order.

Since this problem is more complicated and also more sensitive to the round-off error than the one of the previous section, we found it necessary to increase N (from 32 to 64) and also to relax the test for reality to $|\mathrm{Im}\,E_k|<10^{-5}$; cf (11). Parameter selection in this problem is

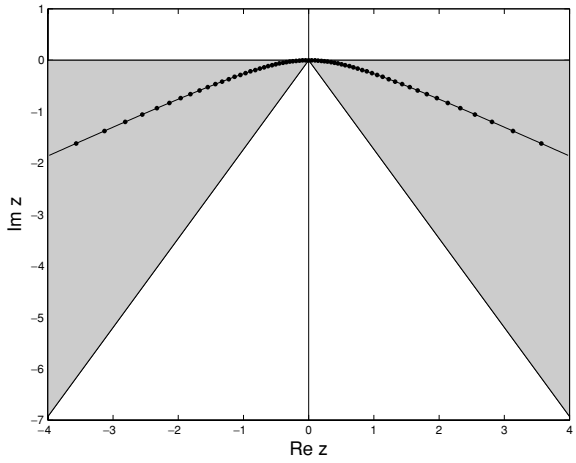

Figure 3. Contour and grid used for computing the spectrum of (12).

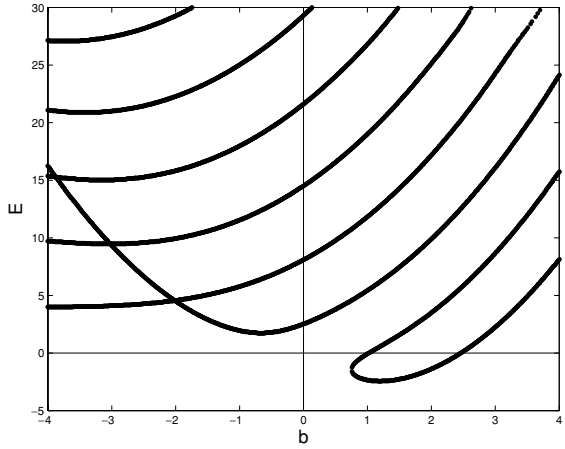

Figure 4. Real spectrum of the potential (12).

also more of a challenge. In addition to the scaling parameter, c, the contour (13) introduces two additional free parameters, μ and α.

Because of the restriction $0 < \alpha < \pi/3$, we thought $\alpha = \pi/6$ to be a reasonable choice. We determined c and μ empirically by plotting errors in numerical differentiation of $f(z) = \mathrm{e}^{-\mathrm{i}z^3/3}$ in the (μ, c) parameter plane and then selecting a good choice manually. With $N = 64$, the choice $(\mu, c) = (1, 5)$ produces errors on the order of 10^{-5}.

The function $f(z) = \mathrm{e}^{-\mathrm{i}z^3/3}$ was selected as a test function based on the ansatz in [3]. Since this corresponds to $b = 0$, we can expect the accuracy to deteriorate when $|b|$ increases. Indeed, in figure 4 one can see slight irregularities in the northeast section of the plot, which can be improved by better parameter selection.

We believe this proposal for dealing with Stokes wedges to be new. The idea of using the hyperbola (13) was taken from [8], where a similar contour was used for the numerical inversion of sectorial Laplace transforms.

6. Conclusions

We have discussed the Hermite spectral collocation method for solving Schrödinger's equation and demonstrated its use through a few examples. In addition to the two templates given in tables 1 and 2, we provide further templates online at [12].

An alternative to the Hermite method is the Fourier spectral method, which should work just as well if not better. It is based on trigonometric interpolants on a truncated domain $[-L, L]$. Codes for generating Fourier differentiation matrices are part of DMSUITE [15] and can also be found in [6, 11].

We remark that there exist numerous other computational strategies for solving (1). These include shooting methods, finite differences, finite elements and variational methods. In addition, there are the well-tested general purpose software codes for Sturm–Liouville problems SL02F, SLEDGE, SLEIGN and SLEIGN2, all four of which have been combined in the package SLDRIVER [10]. A comparison of all these methods falls outside the scope of the present paper. Instead, let us merely summarize what we consider to be the most attractive features of the spectral collocation method.

The main reason for using spectral methods has always been rapid convergence. It is not easy to achieve accuracies on the order of 10^{-12} (cf figure 1) using methods based on low-order approximants such as finite differences or finite elements. This rapid convergence translates into low computational costs. Figures 2 and 4 were produced in less than 5 s and 25 s, respectively, on a machine with a Pentium 4 processor running at 3 GHz. (The bulk of the work is computing the eigenvalues of 1000 full matrices of order 33×33 and 65×65, respectively.)

A second attractive feature of the collocation method is the preservation of the 'visual' appearance of the problem; compare (6) with (1). This makes coding as well as interactive experimentation easy. With only a basic familiarity of MATLAB programming, a user can develop codes such as those in tables 1 and 2 in a matter of minutes, not hours.

A third feature worth noting is that the spectral collocation method provides one with a matrix approximation to the differential operator $-\frac{\mathrm{d}^2}{\mathrm{d}x^2} + p(x)$; cf (8). By passing the matrix A to the MATLAB package EigTool [17], for example, it becomes possible to compute norms of resolvents and plot pseudospectra.

What the spectral method cannot do is compute only selected eigenvalues, something that is possible with the methods described in [10]. Complicated boundary conditions, singularities and continuous spectra will also pose a challenge.

The spectral collocation method is well known in scientific computing but perhaps its use for solving Schrödinger's equation has been under-appreciated by computational physicists. We hope that this paper can contribute towards changing that situation.

Acknowledgments

The author acknowledges discussions with practically all participants of the *Workshop on the Physics of Non-Hermitian Operators*, Stellenbosch, South Africa, November 2005. This work was supported by the NRF in South Africa under grant FA2005032300018.

References

[1] Bender C M, Berry M, Meisinger P N, Savage V M and Simsek M 2001 Complex WKB analysis of energy-level degeneracies of non-Hermitian Hamiltonians *J. Phys. A: Math. Gen.* **34** L31–6

[2] Bender C M, Berry M V and Mandilara A 2002 Generalized PT symmetry and real spectra *J. Phys. A: Math. Gen.* **35** L467–71
[3] Bender C M and Boettcher S 1998 Quasi-exactly solvable quartic potential *J. Phys. A: Math. Gen.* **31** L273–7
[4] Bender C M and Orszag S A 1999 *Advanced Mathematical Methods for Scientists and Engineers* (New York: Springer)
[5] Boyd J P 1984 Asymptotic coefficients of Hermite function series *J. Comput. Phys.* **54** 382–410
[6] Fornberg B 1996 *A Practical Guide to Pseudospectral Methods* (Cambridge: Cambridge University Press)
[7] Handy C R and Wang X 2003 Spectral bounds for the PT-breaking Hamiltonian $p^2 + x^4 + iax$ *J. Phys. A: Math. Gen.* **36** 11513–32
[8] López-Fernández M and Palencia C 2004 On the numerical inversion of the Laplace transform of certain holomorphic mappings *Appl. Numer. Math.* **51** 289–303
[9] Meissner H and Steinborn E O Quartic, sextic, and octic anharmonic oscillators: precise energies of ground state and excited states by an iterative method based on the generalized Bloch equation *Phys. Rev.* A **56** 1189–200
[10] Pryce J D 1997 SLDRIVER: A tool for exploring SL solvers and SL problems *Spectral Theory and Computational Methods of Sturm-Liouville Problems (Knoxville, TN, 1996) (Lecture Notes in Pure and Appl. Math.* vol 191*)* (New York: Dekker) pp 349–76
[11] Trefethen L N 2000 *Spectral methods in MATLAB* (Philadelphia, PA: SIAM)
[12] Weideman J A C 2006 http://dip.sun.ac.za/weideman/schrodinger/
[13] Weideman J A C 1992 The eigenvalues of Hermite and rational spectral differentiation matrices *Numer. Math.* **61** 409–32
[14] Weideman J A C 1999 Spectral methods based on nonclassical orthogonal polynomials *Applications and Computation of Orthogonal Polynomials (Oberwolfach, 1998) (Int. Ser. Numer. Math.* vol 131*)* (Basel: Birkhäuser) pp 239–51
[15] Weideman J A C and Reddy S C 2000 A MATLAB differentiation matrix suite *ACM Trans. Math. Software* **26** 465–519 Codes available at http://www.mathworks.com/matlabcentral/fileexchange/ and at http://dip.sun.ac.za/weideman/research/differ.html
[16] Welfert B D 1997 Generation of pseudospectral differentiation matrices: I *SIAM J. Numer. Anal.* **34** 1640–57
[17] Wright T 2002 *EigTool* http://www.comlab.ox.ac.uk/pseudospectra/eigtool/