

Matlab package for the Schrödinger equation

Damian Trif

Department of Applied Mathematics, Babes-Bolyai University, Cluj-Napoca, Romania
E-mail: dtrif@math.ubbcluj.ro

Received 1 November 2006; Revised 18 February 2007

The paper presents a Matlab package for the linear time-(in)dependent Schrödinger equation, based on the Hermite spectral method. The matrix form of the discretized problem is suitable for the linear algebra capabilities of Matlab. The high accuracy and efficiency of the algorithm is proved by many examples taken from the literature.

KEY WORDS: Schrödinger equation, Hermite spectral method, Matlab

AMS subject classifications: 65L15, 65M70, 81Q05, 34L40

1. Introduction

Consider the one-dimensional linear Schrödinger operator

$$H = -\frac{d^2}{dx^2} + V(x)$$

defined on the space $L^2(\mathbb{R})$, where the potential V is a real valued, measurable, locally integrable function. Consequently, H is a self-adjoint operator whose domain is an appropriate Sobolev space. Although the unphysical domain \mathbb{R} is taken into consideration, the above operator is of practical interest in view of the quantum mechanical interpretation that H governs the behavior of one electron interacting with an external electrical field.

The evolution of a particle with initial state $u_0(x) \in L^2$ is described by the time-dependent Schrödinger equation

$$\begin{aligned} iu_t(x, t) &= Hu(x, t), \\ u(x, 0) &= u_0(x). \end{aligned}$$

If we can solve the energy eigenvalue equation

$$H\psi_n = E_n\psi_n$$

and if the initial state can be expanded as

$$u_0(x) = \sum_{n=0}^{\infty} a_n \psi_n(x)$$

then, at a later time,

$$u(x, t) = \sum_{n=0}^{\infty} a_n e^{-iE_n t} \psi_n(x).$$

A powerful method for calculating the eigenvalues and eigenfunctions of a Schrödinger operator consists of converting the energy eigenvalue equation into a matrix equation by means of collocation at appropriate grid points x_0, x_1, \dots , or by means of the Galerkin method on a basis set of functions. The matrix representation of the energy eigenvalue problem is

$$A\mathbf{c} = E\mathbf{c},$$

where \mathbf{c} is the column vector which consists of the values of ψ at the grid points or of the coordinates of ψ with respect to the basis under consideration. It is necessary to truncate the matrix A to some finite dimension, say $N \times N$, and then its eigenvalues and eigenvectors can be calculated by the simple Matlab command `eig` for example.

High accuracy calculation of the eigenvalues and eigenfunctions of the Schrödinger operator is an essential part of a good algorithm based on the above ideas. The finite differences or finite element methods do not offer a sufficiently great precision. The CPM method used by MATSLISE [1] is highly accurate but works only for bounded intervals, so that either transformations of $(-\infty, \infty)$ to a bounded interval or artificial boundary conditions are needed. Other methods, such like those of type SLEDGE [2] also perform such transformations and, moreover, like MATSLISE, give the eigenvalues and eigenfunctions only one by one. Spectral methods in unbounded domains are very promising in this field and have received considerable attention, mainly due to their high accuracy and being free from using artificial boundary conditions [3], but we must choose a natural procedure to discretize the continuous problem.

For example, let us consider the spectral problem

$$x \frac{du}{dx} = \lambda u, \quad x \in \mathbb{R}. \quad (1)$$

First, we use the package “*A MATLAB Differentiation Matrix Suite*” of Weideman and Reddy [3] to discretize the problem by the well known pseudospectral method based on Hermite collocation points. The command

$$[x, D] = \text{herdif}(N, 1, b)$$

gives the Hermite points x and the first order differentiation matrix D (of size N and scaling parameter b). The differentiation process is exact for functions of the form $e^{-x^2/2}p(x)$ where $p(x)$ is any polynomial of degree $N - 1$ or less, assuming exact arithmetic. The spectral discretization consists of interpolating at N points x_i with the weight $e^{-x_i^2/2}$, differentiating the interpolant and multiplying by x_i .

By setting

$$v(x) = e^{-x^2/2}u(x)$$

the problem (1) becomes

$$x \frac{dv}{dx} + x^2 v = \lambda v, \quad x \in \mathbb{R}. \quad (2)$$

The exact eigenfunctions of the discretized problem are $v_n(x) = e^{-x^2/2}x^n$, $i = 1, \dots, N$ sampled at the interpolation points and the corresponding eigenvalues are $\lambda_n = n$, $n = 0, 1, \dots, N$.

If V is the column vector $(v(x_1), \dots, v(x_n))^T$, the discretized problem (2) is

$$[\text{diag}(x)D + \text{diag}(x^2)]V = \lambda V.$$

The commands

$$\begin{aligned} [x, D] &= \text{herdif}(128, 1, 1); \\ L &= \text{eig}(\text{diag}(x) * D + \text{diag}(x.^2)); \\ &\text{plot}(L, 'l') \end{aligned}$$

give 128 numerical eigenvalues L , see figure 1. The first 16 small eigenvalues approximate very well the exact corresponding eigenvalues of the discretized problem but the others are spurious due to the high sensitivity of the full matrix D to rounding errors. Only for small dimension ($N = 64$ or 32) we obtain better results.

Now we use the Hermite spectral method instead, based on the representation of functions in the transformed space, not in the physical space (i.e., calculating the Hermite–Fourier coefficients of a function, not their values at the Hermite points). We obtain the correct eigenfunctions $u_n(x) = x^n$ and the eigenvalues $\lambda_n = n$, $n = 0, 1, \dots, N - 1$ of the discretized problem

$$[\text{diag}(x)D]V = \lambda V,$$

even for $N = 512$. The commands of our package are

$$X = \text{mult}(512); D = \text{deriv}(512); L = \text{sort}(\text{eig}(X * D)).$$

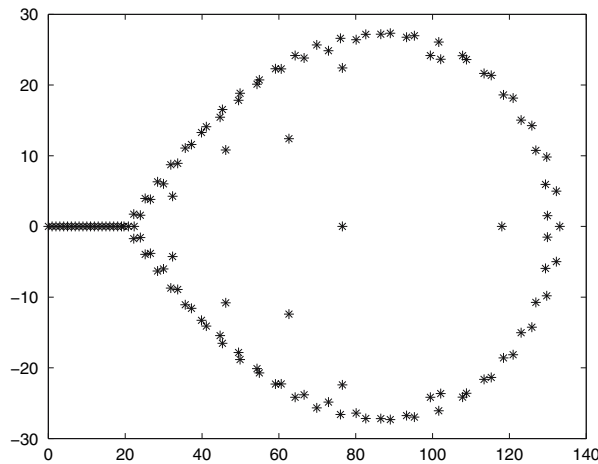


Figure 1. Numerical eigenvalues by collocation at 128 points.

Consequently, we must calculate the numerical eigenvalues and eigenfunctions of the Schrödinger problems using this Hermite spectral method, based on the complete orthonormal sequence of Hermite–Weber functions (eigenfunctions corresponding to the harmonic oscillator with the potential $V(x) = x^2$)

$$\phi_n(x) = N_n e^{-x^2/2} H_n(x), \quad n = 0, 1, 2, \dots,$$

where

$$N_n = \frac{1}{\sqrt{2^n n!} \sqrt{\pi}},$$

and H_n are the usual Hermite polynomials.

The energy eigenfunctions become

$$\psi_n(x) = \sum_{k=0}^{N-1} d_{nk} \phi_k(x),$$

and the initial condition $u_0(x)$ can also be expressed as

$$u_0(x) = \sum_{k=0}^{N-1} c_k \phi_k(x) = \sum_{n=0}^{\infty} a_n \psi_n(x),$$

where

$$a_n = \sum_{k=0}^{N-1} c_k \left(\mathbf{d}^{-1} \right)_{kn}.$$

These coefficients can be used to obtain an approximation of the wave function $u(x, t)$ at any later time t .

Although the Hermite spectral methods have some attractive properties, the direct spectral approach may not produce good approximations. In the practical calculations, it is necessary to choose a scaling factor α since we can always use $\phi_n(\alpha x)$ as the basis set for any finite α [4, 5].

Some well-known properties of the spectrum of the Schrödinger operator depending on the properties of the potential $V(x)$ are:

- (1) If $V(x) \rightarrow +\infty$ as $|x| \rightarrow \infty$ then the spectrum of H consists of a discrete sequence of eigenvalues tending to $+\infty$.
- (2) If $V \in L^1(\mathbb{R})$ then H has only point spectrum in \mathbb{R}^- , with 0 as its only possible accumulation point; in \mathbb{R}^+ the spectrum is purely absolutely continuous with an essential support \mathbb{R}^+ itself.
- (3) If $xV(x) \in L^1(\mathbb{R})$ then there are only finitely many bounded states.

2. Problem and implementation

Our MATLAB package `HermiteEig` calculates a finite number of numerical eigenvalues and eigenfunctions of **Schrödinger** problems

$$-\frac{d^2u}{dx^2} + V(x)u = Eu, \quad x \in (-\infty, \infty).$$

Introducing the linear transformation $\xi = \alpha x$ for $\alpha > 0$ the eigenvalue problem becomes

$$-\frac{d^2\Psi}{d\xi^2} + \alpha^{-2}V(\alpha^{-1}\xi)\Psi = \mathcal{E}(\alpha)\Psi, \quad \xi \in (-\infty, \infty), \quad \Psi \in L^2(-\infty, \infty), \quad (3)$$

where the eigenvalues $\mathcal{E}(\alpha)$ depending on the parameter α are connected with the original eigenvalues E by $E = \alpha^2\mathcal{E}(\alpha)$.

A powerful method to solve (3) is to expand Ψ into a series of the form

$$\Psi(\xi) = \sum_{n=0}^{\infty} c_n \phi_n(\xi).$$

Here

$$\phi_n(\xi) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} e^{-\xi^2/2} H_n(\xi), \quad n = 0, 1, 2, \dots$$

are the normalized Hermite–Weber functions, i.e., the complete sequence of eigenfunctions of the harmonic oscillator problem

$$-\frac{d^2\phi_n}{d\xi^2} + \xi^2\phi_n = (2n + 1)\phi_n$$

and $H_n(\xi)$ are the classical Hermite polynomials, see [6] and [7] for recent numerical analysis of this method.

By setting

$$\Psi(\xi) = e^{-\xi^2/2}y(\xi),$$

(3) becomes

$$-y'' + 2\xi y' + \mathcal{V}(\xi)y = \mathcal{E}(\alpha)y,$$

where

$$\mathcal{V}(\xi) = \alpha^{-2}V(\alpha^{-1}\xi) - \xi^2 + 1$$

is the modified potential.

We search for an approximation of the new unknown function y of the form

$$y_N(\xi) = \sum_{n=0}^{N-1} c_n \psi_n(\xi),$$

where

$$\psi_n(\xi) = \frac{1}{\sqrt{2^n n!} \sqrt{\pi}} H_n(\xi), \quad n = 0, 1, 2, \dots$$

are the normalized Hermite polynomials.

For the practical implementation, we define the vectors c and t by

$$\begin{aligned} c^T &= (c_0, c_1, c_2, \dots), \\ t^T &= (\psi_0, \psi_1, \psi_2, \dots), \end{aligned}$$

so that $y(\xi) = c^T t(\xi) = t^T(\xi) c$.

Let us now find a matrix X for which

$$\xi \cdot y(\xi) = \xi \cdot c^T t(\xi) = (Xc)^T t(\xi),$$

i.e., the coefficients of $\xi \cdot y(\xi)$ are Xc . By using the properties of Hermite polynomials we have

$$H_{n+1}(\xi) = 2\xi H_n(\xi) - 2n H_{n-1}(\xi)$$

so that we have for the functions ψ_n the three-term recursion

$$\xi \psi_n(\xi) = \sqrt{\frac{n}{2}} \psi_{n-1}(\xi) + \sqrt{\frac{n+1}{2}} \psi_{n+1}(\xi),$$

which $\psi_{-1}(\xi) = 0$ and $\psi_0(\xi) = \pi^{-1/4}$. Consequently,

$$\begin{aligned} \xi \cdot \psi_0 &= \frac{\psi_1}{\sqrt{2}}, \\ \xi \cdot \psi_1 &= \frac{\psi_0 + \sqrt{2}\psi_2}{\sqrt{2}}, \\ &\dots\dots\dots \\ \xi \cdot \psi_n &= \frac{\sqrt{n}\psi_{n-1} + \sqrt{n+1}\psi_{n+1}}{\sqrt{2}}, \quad n \geq 2 \end{aligned}$$

and the non-zero elements of X are

$$X_{i,i+1} = X_{i+1,i} = \sqrt{\frac{i}{2}}, \quad i = 1, 2, \dots$$

Of course, in practical calculation we must truncate this infinite matrix to have a finite order N , so that we obtain only an approximation of the exact result.

We also have

$$\xi^m y(\xi) = (X^m c)^T t(\xi)$$

and

$$f(\xi)y(\xi) = (f(X)c)^T t(\xi)$$

for analytical functions f , i.e.,

$$f(\xi) = \sum_{k=0}^{\infty} f_k \frac{\xi^k}{k!}.$$

Moreover,

$$\frac{y(\xi)}{\xi^m} = (X^{-m}c)^T t(\xi)$$

if the l.h.s. has no singularity at the origin. Of course, X^m is a banded matrix for small m but, generally, the matrix version $\text{funm}(X)$ of the scalar function $f(x)$ or $X^{-m} = [\text{inv}(X)]^m$ are no longer sparse matrices.

Similarly, let us find the differentiation matrix D giving

$$\frac{dy}{d\xi} = (Dc)^T t(\xi).$$

The derivatives of H_n satisfy $H'_n = 2nH_{n-1}$ so that $\psi'_n = \sqrt{2n}\psi_{n-1}$ and the non-zero elements of D are

$$D_{i,i+1} = \sqrt{2i}, \quad i = 1, 2, \dots$$

We truncate this matrix to the order N but this time, if y is described by N coefficients c_n , the above differentiation is exact. More, we have

$$\frac{d^2y}{d\xi^2} = (D^2c)^T t(\xi).$$

Applying these formulae to equation (3), we get

$$\left[-D^2 + 2XD - X^2 + \mathcal{V}(X) + I_N\right]c = \mathcal{E}c.$$

If we define the matrix

$$A = -D^2 + 2XD - X^2 + \mathcal{V}(X) + I_N,$$

the vector c satisfies the eigenvalues/eigenvectors problem

$$Ac = \mathcal{E}c. \tag{4}$$

Remark. We could obtain similar matrices for the basis ϕ_n . The matrix X is the same while the differentiation matrix D must be replaced by $D-X$ ($\psi_n = e^{\xi^2/2}\phi_n$ and $\psi'_n = \sqrt{2n}\psi_{n-1}$ becomes $\xi e^{\xi^2/2}\phi_n + e^{\xi^2/2}\phi'_n = \sqrt{2n}e^{\xi^2/2}\phi_{n-1}$ from where $\phi'_n = \sqrt{2n}\phi_{n-1} - \xi\phi_n$). The equation (3) becomes

$$\left[-D^2 + DX + XD - X^2 + \mathcal{V}(X)\right]c = \mathcal{E}c$$

and actually we have $DX + XD = 2XD + \mathcal{I}_N$ where \mathcal{I}_N is a slightly modified unit matrix I_N .

The package `HermiteEig` contains the function files `pd.m`, `x2t.m`, `mult.m`, `deriv.m`, and `sph.m`.

(a) The function `pd.m` (physical domain),

$$[x, w] = \text{pd}(n);$$

calculates the nodes and weights for the GaussHermite quadrature, i.e.,

$$I \equiv \int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{k=1}^n f(x_k) w_k = w * f,$$

where f contains the values of the function f at the grid points x .

(b) The function `x2t.m`

$$t = \text{x2t}(n, x)$$

calculates the values of the normalized Hermite polynomials ψ_k , $k=0, 1, \dots, n$ at the points x . Consequently, if

$$f(x) = c_0\psi_0(x) + c_1\psi_1(x) + \dots + c_n\psi_n(x)$$

and $c^T = (c_0, \dots, c_n)$, we have $f(x) = t^T * c$, for any $x \in (-\infty, \infty)$.

(c) The functions `X=mult(n)` and `D=deriv(n)` calculate the matrices X , respectively, D .

(d) The main function `[L,psi,x,w,t]=sph(fun,n,a)` gives the eigenvalues L , the eigenfunctions psi , the nodes x , the weights w , and the matrix t . The input parameters are the potential $V(x)$ in the function `fun`, the dimension of the discretized problem n and the scaling factor a . Of course, only the first eigenvalues are highly accurate while the last ones are spurious.

The folder `examples` contains 21 test problems from the literature [1, 2, 6–9].

3. Examples

We give here more details of some interesting examples.

3.1. Morse problem

Let us consider the potential

$$V(x) = (e^{-\gamma x} - 1)^2, \quad \gamma = 0.005\pi$$

with $N = 100$ and the scaling parameter $\alpha = 0.1$. Many of the calculated eigenvalues λ_n (file `morse.m`) coincide with the exact eigenvalues

$$E_n = \left(n + \frac{1}{2}\right) \gamma \left[2 - \left(n + \frac{1}{2}\right) \gamma\right], \quad n = 0, 1, \dots, \left[\frac{1}{\gamma} - \frac{1}{2}\right],$$

n	λ_n	E_n
0	0.01564627824044	0.01564627824044,
5	0.16532370761911	0.16532370761911,
10	0.30266413149642	0.30266413149643,
25	0.64066337012045	0.64066337012019.

3.2. Symmetric double well potential

For the potential

$$V(x) = v_4 \left(x^2 - \frac{1}{2} v_4^{-1} \right)^2$$

with $v_4 = 0.01$, $\alpha = 1.1$, and $N = 100$, some of the calculated eigenvalues (file `sdwp.m`) are, in ascending order,

```
1.40404860529767    9.49857838718786,
1.40404860529776    9.49857838719109,
4.17019360599931   12.04930948633407,
4.17019360599934   12.04930948667311,
6.87008883371401   14.51420502298122,
6.87008883371409   14.51420504812104
```

in very good coincidence with those from [6].

3.3. Anharmonic potential

For the potential

$$V(x) = x^2 + \frac{\lambda x^2}{1 + g x^2}, \quad \lambda = 0.1, \quad g = 0.1,$$

the numerical computation (`anharmonic`) gives, with $N = 100$ and $a = 1$ the first eigenvalue $L(1) = 1.04317371304445$, which coincides with the value found by `matslise` [1].

3.4. Bender–Orszag potential

For the potential

$$V(x) = -\frac{m(m+1)}{\cosh^2(x)}$$

the exact eigenvalues are $E_k = -(m-k)^2$, $0 \leq k < m$. The calculated values for $m = 3$, $a = 1$, and $N = 100$ are

```
-8.99999999998992,
-4.00000000003926,
-0.99999999993041.
```

3.5. Non-analytical potential

Let us consider the potential [2]

$$V(x) = \begin{cases} 1 - e^{-x^2}, & x < 0, \\ x \log(x + 1), & x > 0 \end{cases}$$

with the unique eigenvalue $\lambda_0 = 0.7125276014$.

In this case, the formula $V(\xi)y(\xi) = (V(X)c)^T t(\xi)$ does not work. We must use the transformation matrix

$$F = [t * \text{diag}(w)] * [\text{diag}(V(\xi)) * t^T]$$

instead of $V(X)$.

The commands `[L, psi, x, w, t]=tab(250, 1.35)`; (i.e., $N = 250$, $a = 1.35$) give the numerical eigenvalues

$$0.71252779084313,$$

.....

and the first of them approximates the true eigenvalue with an error less than 2×10^{-7} .

3.6. Evolution problem

In [5] an instability was observed numerically when a kind of Hermite spectral method was applied for the linear diffusion equation in a unbounded domain. Precisely, let us consider the problem

$$\begin{aligned} u_t - u_{xx} &= f(x, t), & x \in \mathbb{R}, & \quad t > 0, \\ u(x, 0) &= u_0(x), & x \in \mathbb{R}, \end{aligned}$$

which

$$f(x, t) = \left[5(1 + x) \cos 5(x + t) + (25 + 0.5 - 0.25x^2) \sin 5(x + t) \right] e^{-\frac{x^2}{4}},$$

and

$$u_0(x) = \sin 5x e^{-\frac{x^2}{4}}.$$

The exact solution of the problem is $U(x, t) = \sin 5(x + t) e^{-\frac{x^2}{4}}$.

With the transformation $u(x, t) = y(x, t)e^{-\frac{x^2}{2}}$ the above problem becomes

$$y_t = y_{xx} - 2xy_x + (x^2 - 1)y + f(x, t)e^{\frac{x^2}{2}}, \quad x \in \mathbb{R}, \quad t > 0,$$

$$y(x, 0) = u_0(x)e^{\frac{x^2}{2}}, \quad x \in \mathbb{R}.$$

If we expand

$$y(x, t) = \sum_{k=0}^N c_k(t)\psi_k(x)$$

we obtain for the coefficients c the differential system

$$c' = \mathcal{D}c + F,$$

$$c(0) = c^0,$$

where F and c^0 are the coefficients of the expansions of $f(x, t)e^{\frac{x^2}{2}}$ and $u_0(x)e^{\frac{x^2}{2}}$ with respect to ψ and \mathcal{D} is the matrix $D^2 - 2XD + X^2 - I_N$.

Now, by using the Crank–Nicolson method, we obtain for $c^j = c(t_j)$ the equations

$$\frac{c^{j+1} - c^j}{dt} = \mathcal{D} \frac{c^{j+1} + c^j}{2} + \frac{F^{j+1} + F^j}{2},$$

i.e.,

$$\left(I - dt \frac{\mathcal{D}}{2}\right) c^{j+1} = c^j + \frac{dt}{2} (\mathcal{D}c^j + F^{j+1} + F^j).$$

Using $N = 64$ and $dt = 0.01$ (`diffusion.m`), we obtain after 100 time steps the coefficients c of the numerical solution at $t = 1$ and the corresponding values $(t^T c) e^{-x^2/2}$, with a maximal error 4×10^{-5} at the grid points x , without any numerical instability.

3.7. Fokker–Planck equation

We solve the Fokker–Planck equation [8],

$$\frac{\partial f(x, t)}{\partial t} = \frac{\partial^2 (B(x)f(x, t))}{\partial x^2} + \frac{\partial (A(x)f(x, t))}{\partial x}$$

via eigenfunction expansion. It is more convenient to work with a self-adjoint operator. Let us consider the equilibrium solution

$$f_0(x) = \frac{1}{B(x)} \exp\left(-\int^x \frac{A(y)}{B(y)} dy\right).$$

If $\Phi(x, t)$ is defined by $f(x, t) = f_0(x)\Phi(x, t)$, we obtain the equation for Φ

$$\begin{aligned} \frac{\partial \Phi(x, t)}{\partial t} &= B(x) \frac{\partial^2 \Phi(x, t)}{\partial x^2} - A(x) \frac{\partial \Phi(x, t)}{\partial x} \\ &= -L\Phi(x, t). \end{aligned}$$

The operator L satisfies the eigenvalue equation $L\Phi_n(x) = \epsilon_n\Phi_n(x)$ and it is a self-adjoint operator on the space spanned by the eigenfunctions Φ_n , with the inner product with the weight function $f_0(x)$. The function $\Phi(x, t)$ can be expanded by the complete set of eigenfunctions according to

$$\Phi(x, t) = \sum_n b_n e^{-\epsilon_n t} \Phi_n(x),$$

where the expansion coefficients b_n are determined by the initial condition $f(x, 0)$.

Computationally it is more convenient to solve the above eigenvalue problem using the Schrödinger equation

$$-\frac{d^2 \Psi_n(z)}{dz^2} + V(z)\Psi_n(z) = \epsilon_n \Psi_n(z),$$

where

$$\begin{aligned} \Psi_n(z) &= (f_0(x(z)) \sqrt{B(x(z))})^{1/2} \Phi_n(x(z)), \\ z(x) &= \int^x (B(y))^{-1/2} dy. \end{aligned}$$

The effective potential is given by

$$V(z) = \frac{1}{4} \left[W^2(z) - 2W'_z(z) \right],$$

where

$$\begin{aligned} W(z) &= \frac{1}{\sqrt{B}} \left(A + \frac{B'}{2} \right), \\ B' &= \frac{dB(x(z))}{dz}. \end{aligned}$$

Let us consider, for example, the case of the bistable system

$$\frac{\partial f(x, t)}{\partial t} = \epsilon \frac{\partial^2 f(x, t)}{\partial x^2} - \frac{\partial ((\gamma x - gx^3) f(x, t))}{\partial x}$$

with $\gamma = g = 1$. The corresponding effective potential is

$$V(z) = \frac{(z^3 - z)^2}{4\epsilon^2} - \frac{1}{2\epsilon} (3z^2 - 1).$$

For $\epsilon = 0.0125$, the package (`evol.m` with $N = 101$, $\epsilon = 0.0125$, the scaling factor $a = 7$, the time step $dt = 0.1$ and as the initial distribution $f(x, 0)$ – a Gauss function) gives the eigenvalues

```
L(1:10)*49*0.0125,
-0.0000000000000000  1.83463714345201,
0.000000000090974   2.56075256468075,
0.95897283387818    3.00731102631005,
1.81602047864552    3.16316264504337,
1.82507262410861    3.42456872119340
```

and the time evolution in good concordance with those from [9].

4. Conclusion

The package `HermiteEig` is useful for steady or evolution Schrödinger equations (and not only). It gives the important eigenvalues and eigenfunctions very fast and accurately. The Matlab code could be freely obtained from [10].

References

- [1] V. Ledoux, MATSLISE package, <http://users.ugent.be/~vledoux/MATSLISE/>.
- [2] S. Pruess, C.T. Fulton and Y. Xie, Performance of the Sturm-Liouville Software Package Sledge, T. R. MCS-91-19, September 1994.
- [3] J.A.C. Weideman and S.C. Reddy, ACM TOMS 26 (2000) 465.
- [4] T. Tang, SIAM J. Sci. Comp. 14(3) (1993) 594.
- [5] H. Ma, W. Sun and T. Tang, SIAM J. Numer. Anal. 43(1) (2005) 58.
- [6] H. Taşeli and H. Alici, J. Math. Chem. 38(3) (2005) 367.
- [7] H. Taşeli and M.B. Erseçen, J. Math. Chem. 34(3–4) (2003) 177.
- [8] G.W. Wei, J. Chem. Phys. 110(18) (1999) 8930.
- [9] G.W. Wei, D.S. Zhang, D.J. Kouri and D.K. Hoffman, J. Chem. Phys. 107(8) (1997) 3239.
- [10] D. Trif, HermiteEig Tutorial 2006, MATLAB Central>File Exchange>Chemistry and Physics>HermiteEig, <http://www.mathworks.com/matlabcentral/fileexchange>.