Spectral Differentiation Matrices for the Numerical Solution of Schrödinger’s Equation

J.A.C. Weideman
Department of Applied Mathematics
University of Stellenbosch
South Africa
http://dip.sun.ac.za/ weideman

Thanks to:

- NRF-FA2005032300018 “I have no satisfaction in formulas
- NSF DMS-9404399 unless I feel their numerical magnitude”

Lord Kelvin
Overview of talk:

1. Quick Introduction
2. Software (DMSUITE)
3. Solving Schrödinger’s Equation
1. Quick Introduction

Consider toy problem

\[ y''(x) = \lambda y(x), \quad -1 \leq x \leq 1, \]

subject to boundary conditions

\[ y(-1) = y(+1) = 0. \]

Discretize by Method of Finite Differences

\[ a \quad h = (b-a)/N \quad b \]

\[ x_0 \quad x_1 \quad x_2 \quad \ldots \quad x_N \]

Denote

\[ y_j \approx y(x_j) \]
Approximate second derivative by central differences

\[ y''(x) = \lambda y(x) \]

\[ \Rightarrow \quad \frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} = \lambda y_j, \quad j = 1, \ldots, N - 1 \]

\[ \Rightarrow \quad D_2 y = \lambda y \]

Here

\[ D_2 = \frac{1}{h^2} \begin{pmatrix} -2 & 1 \\ 1 & -2 \\ \vdots & \vdots \\ 1 & -2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{pmatrix} \]

Differentiation matrix: \( D_2 = \) matrix representation of \( \frac{d^2}{dx^2} + \text{BCs} \)
Basic idea of **Spectral Collocation** a.k.a. **Pseudospectral Method**: Do not use local interpolants, use a single global interpolant instead.
Lagrange form of polynomial interpolant

\[ p_N(x) = L_0(x) u_0 + L_1(x) u_1 + \ldots + L_N(x) u_N \]

where

\[ L_k(x) = \text{pol. degree } N, \quad L_k(x_j) = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \]

Differentiate twice and evaluate at \( x = x_j \)

\[ p''_N(x_j) = L''_0(x_j) u_0 + L''_1(x_j) u_1 + \ldots + L''_N(x_j) u_N \]

Substitute into

\[ p''_N(x_j) = \lambda y(x_j) \]

\[ \implies \begin{pmatrix} L''_1(x_1) & L''_2(x_1) & \ldots & L''_{N-1}(x_1) \\ L''_1(x_2) & L''_2(x_2) & \ldots & L''_{N-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ L''_1(x_{N-1}) & L''_2(x_{N-1}) & \ldots & L''_{N-1}(x_{N-1}) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{pmatrix} \]

Defines spectral differentiation matrix \( D_2 \).
Important: Do not use equidistant points: they lead to ill-conditioning & Runge phenomenon

Rather use Chebyshev points of the second kind, defined by

\[ x_k = \cos \left( \frac{k\pi}{N} \right), \quad k = 0, \ldots, N \]

References:


Solve **Toy Example** as test problem

\[ y''(x) = \lambda y(x) \quad \implies \quad D_2 y = \lambda y \]

On \([-1,1]\) exact eigenvalues/functions are given by

\[ \lambda_k = -\frac{k^2\pi^2}{4}, \quad y_k(x) = \begin{cases} \cos\left(\frac{1}{2}k\pi x\right) & k \text{ odd} \\ \sin\left(\frac{1}{2}k\pi x\right) & k \text{ even}. \end{cases} \]

Solve algebraic eigenvalue problem \( D_2 y = \lambda y \) numerically and plot absolute errors in \( k = 1 \) eigenvalues as functions of \( N \):
Variations on the Theme

Weighted Polynomial Methods:

\[ p_N(x) = w(x) \left( L_0(x) u_0 + L_1(x) u_1 + \ldots + L_N(x) u_N \right) \]

- Hermite weight:
  \[ w(x) = e^{-x^2/2}, \quad x \in (-\infty, \infty), \quad x_k = \text{zeros of } H_{N+1}(x) \]

- Laguerre weight:
  \[ w(x) = e^{-x}, \quad x \in [0, \infty), \quad x_k = \text{zeros of } L_{N+1}(x) \]

Non-Polynomial Methods:

- Trigonometric (Fourier)

- Sinc (cardinal)
2. SOFTWARE

DMSUITE = MATLAB package developed by JACW and SC Reddy

- It computes pseudospectral D matrices corresponding to
  - Chebyshev
  - Hermite
  - Laguerre
  - Fourier
  - Sinc
- plus utilities for BCs.

Attention to:
- efficient MATLAB coding (i.e., vectorization)
- numerical stability


Software is free and may be downloaded from

Tutorial-style Applications:

- **Schrödinger** \(-y''(x) + y(x) = \lambda q(x) y(x), \quad x \in [0, \infty)\)

- **Error Function** \(y(x) = e^{x^2} \text{erfc}(x), \quad x \in [0, \infty)\)

- **Mathieu Equation**
  \[
y''(x) + (a - 2q \cos 2x)y = 0
  \]
  \[
x \in [0, 2\pi)
  \]

- **Sine-Gordon**
  \[
  u_{tt} = u_{xx} - \sin u,
  \]
  \[
x \in (-\infty, \infty)
  \]

- **Orr-Sommerfeld**
  \[
  R^{-1}(y''' - 2y'' + y) - 2i y - i (1-x^2) (y'' - y) = c (y - y'')
  \]
3. Application

Schrödinger’s equation on $-\infty < x < \infty$

$$-y''(x) + p(x) y(x) = \lambda y(x)$$

Approximate by

$$-D_2 y + \text{diag}(p(x_k)) y = \lambda y$$

Numerically compute the eigenvalues of

$$\Lambda = -D_2 + \text{diag}(p(x_k))$$

For $D_2$, we use Hermite differentiation matrix. I.e., $D_2 y$ provides exact second derivatives whenever $y$ is sampled from a function of the form

$$y(x) = e^{-x^2/2} \times \left(\text{Polyn. degree}< N\right)$$
Quadratic oscillator

\[-y''(x) + x^2 y(x) = \lambda y(x)\]

Eigenvalues/functions \( k = 0, 1, 2, \ldots, \)

\[\lambda_k = 2k + 1, \quad y_k(x) = e^{-x^2/2}H_k(x)\]

Expect numerical method based on \( N \times N \) Hermite differentiation matrix to produce exact eigenvalues/functions, for \( k = 0, \ldots, N - 1 \).

\[
[x, DM] = \text{herdif}(N, 2, b) \\
D2 = DM(:,:,2) \\
A = -D2 + \text{diag}(x.^2) \\
\text{lambda} = \text{sort(}\text{eig(A)}\text{)}
\]
Test quartic oscillator against WKB formula (Bender & Orszag, p. 523)

\[ \lambda_k \sim \left[ \frac{3\sqrt{\pi} \Gamma\left(\frac{3}{4}\right)(k + \frac{1}{2})}{\Gamma\left(\frac{1}{4}\right)} \right]^{4/3}, \quad k \to \infty. \]

<table>
<thead>
<tr>
<th>Numerical</th>
<th>WKB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0604</td>
<td>0.8671</td>
</tr>
<tr>
<td>3.7997</td>
<td>3.7519</td>
</tr>
<tr>
<td>7.4557</td>
<td>7.4140</td>
</tr>
<tr>
<td>11.6447</td>
<td>11.6115</td>
</tr>
<tr>
<td>16.2618</td>
<td>16.2336</td>
</tr>
<tr>
<td>21.2384</td>
<td>21.2137</td>
</tr>
<tr>
<td>26.5285</td>
<td>26.5063</td>
</tr>
<tr>
<td>32.0986</td>
<td>32.0785</td>
</tr>
<tr>
<td>37.9230</td>
<td>37.9045</td>
</tr>
<tr>
<td>43.9812</td>
<td>43.9639</td>
</tr>
</tbody>
</table>
Now change problem to

\[-y''(x) + (x^4 + iax) y(x) = \lambda y(x)\]

With $D_2$ defined as above the complete code is

```matlab
figure(1); axis([0 15 0 25]); hold on;

for a = linspace(0,15,300);
    A = -D2+diag(x.^4+i*a*x);
lambda = sort(eig(A));
lambda = lambda(1:6);
    ell = find(abs(imag(lambda)) < sqrt(eps));
lambda = lambda(ell);
plot(a*ones(size(lambda)),lambda,'*','MarkerSize',4);
drawnow
end;
```
DMSUITE combines well with EigTool (Trefethen, Wright)

For example, compute pseudospectrum of matrix

\[
A = -D_2 + \text{diag}(x_k^4 + ia x_k), \quad a = 3.169
\]

Plots show values of \( F(z) = \|(A - zI)^{-1}\| \)