SPECTRAL METHODS IN TIME FOR HYPERBOLIC EQUATIONS

Hillel Tal-Ezer

Contract No. NAS1-17070
February 1984

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association
Spectral methods in time for hyperbolic equations

AUTH: TAL-EZER, H. PAA: A/(Tel-Aviv Univ.)
CORP: National Aeronautics and Space Administration, Langley Research Center, Hampton, Va.
MAJS: ALGORITHMS/HYPERBOLIC FUNCTIONS/SPECTRUM ANALYSIS
MINS: NUMERICAL ANALYSIS/PERIODIC FUNCTIONS
ABA: Author

ABS: A pseudospectral numerical scheme for solving linear, periodic, hyperbolic problems is described. It has infinite accuracy both in time and in space. The high accuracy in time is achieved without increasing the computational work and memory space which is needed for a regular, one step explicit scheme. The algorithm is shown to be optimal in the sense that among all the explicit algorithms of a certain class it requires the least amount of work to achieve a certain given resolution. The class of algorithms referred to consists of all explicit schemes which may be represented as a polynomial in the spatial operator.

ENTER:
SPECTRAL METHODS IN TIME FOR HYPERBOLIC EQUATIONS

Hillel Tal-Ezer
School of Mathematical Sciences
Tel-Aviv University, Tel-Aviv, Israel

Abstract

A pseudospectral numerical scheme for solving linear, periodic, hyperbolic problems is described. It has infinite accuracy both in time and in space. The high accuracy in time is achieved without increasing the computational work and memory space which is needed for a regular, one step explicit scheme. The algorithm is shown to be optimal in the sense that among all the explicit algorithms of a certain class it requires the least amount of work to achieve a certain given resolution. The class of algorithms referred to consists of all explicit schemes which may be represented as a polynomial in the spatial operator.

Research supported in part by the U.S. Army Research and Standardization Group (Europe) under Contract DAJA 38-80-C-0032 and in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-17070 while the author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23665.
1. Introduction

In recent years, it has been shown that spectral methods can provide a very useful tool for the solution of time dependent partial differential equations. A standard scheme uses spectral methods to approximate the space derivatives and a finite difference approach to march the solution in time. This tactic results in an unbalanced scheme; it has infinite accuracy in space and finite accuracy in time. It is obvious that the overall accuracy is influenced strongly by the relatively poor approximation of the time derivative.

In this paper we present an alternative approach that also yields spectral accuracy in time and is optimal in terms of efficiency.

The finite difference approach for the time discretization of the P.D.E. is discussed in Section 2. In Section 3 we present the new approach for marching the solution in time in order to get an overall infinite accuracy.

The method presented in Section 3 is based on expanding the evolution operator by orthogonal polynomials. In Sections 4 and 5 we discuss the resolution and stability properties of the method. The scheme is compared to the leap-frog type schemes in order to clarify its properties. In Section 6 we give a proof of the infinite accuracy of our approach. Section 7 presents the algorithm in detail and in Section 8 we demonstrate its validity for the variable coefficients case. Section 9 concludes the discussion by presenting numerical results which confirm the theoretical results developed in the previous sections.
2. Finite Difference Approach

Consider the differential equation

\[ U_t - GU = 0 \quad 0 \leq x \leq 2\pi \]

(2.1)

\[ U(x,0) = U^0(x) \]

where \( G \) is a linear spatial differential operator. We assume that the coefficients of the derivatives appearing in \( G \) are time independent and \( 2\pi \)-periodic. Suppose further that (2.1) is discretized in space by using the pseudospectral Fourier method, [2], [4]. This involves seeking a trigonometric polynomial \( U_N(x) \) of degree \( N \) that satisfies

\[ \frac{\partial U_N}{\partial t} - R_N G P_N U_N = 0 \]

(2.2)

\[ U_N(x,0) = P_N U^0(x) \]

where for any function \( f(x) \), \( P_N f(x) \) is its trigonometric interpolant at the collocation points

\[ X_j = \frac{j\pi}{N} \quad j = 0,1,\ldots,2N-1; \]

more precisely,

(2.3)

\[ P_N f(x) = \sum_{k=-N}^{N} a_k e^{ikx} \]

where

\[ a_k = \frac{1}{2N} \sum_{j=0}^{2N-1} f(x_j) e^{-ikx_j}. \]
The solution of (2.2) is given by

\[ U_N(x,t) = \exp(tP_NG_P)U^0(x) \]  

Except for very simple operators \( G \), it is impractical to construct the exponential matrix \( \exp(tP_NG_P) \) explicitly. Usually an approximation to the exponent is used. Most frequently an explicit or implicit finite difference scheme is used to march the solution over a time step \( \Delta t \). All these algorithms are based on a Taylor expansion of the evolution operator \( \exp(tP_NG_P) \). Essentially, the scalar function \( e^z \) is expanded either as a Taylor series of the form

\[ e^z = \sum_{k=0}^{m} \frac{z^k}{k!} \]

or by a Padé approximation

\[ e^z = \frac{\sum_{q=0}^{p} b_q z^q}{\sum_{l=0}^{q} c_l z^l} \]

where \( b_q, c_l \), are so chosen that the expansion of the right side of (2.5) agrees with the Taylor expansion and then \( z \) is replaced by the matrix \( \Delta t(P_NG_P) \).

For example, in the case of the modified Euler scheme one advances from the time level \( n \) to \( n + 1 \) by using

\[ I + \Delta t P_NG_P + \frac{\Delta t^2}{2}(P_NG_P)^2 \]

to approximate \( \exp[\Delta t(P_NG_P)] \). Let \( V^n \) be the approximation to \( U_N(n\Delta t) \).

Then
\( v^{n+1} = \left[ \mathbf{I} + \Delta t \left( p_N g_{PN} \right) + \frac{\Delta t^2}{2} \left( p_N g_{PN} \right)^2 \right] v^n \)

or

\[ v^n = \left( \mathbf{I} + \Delta t g_N + \frac{\Delta t^2}{2} g_N^2 \right) u_N(0) \]

where

\[ g_N = p_N g_{PN}. \]

Equation (2.7) can be rewritten as

\[ v^n = \left( \sum_{k=0}^{2n} \beta_k \left( \frac{g_N \Delta t}{k} \right)^k \right) u_N(0) \]

(2.9)

or

\[ v^n = \left( \sum_{k=0}^{2n} \bar{\beta}_k \left( g_N^k \left( \frac{\Delta t}{k} \right) \right)^k \right) u_N(0) \]

where

\[ \bar{\beta}_k = \frac{\beta_k}{k}. \]

These types of approximations result in a limitation on the allowable time step \( \Delta t \) since Taylor expansion possesses high accuracy for small \( \Delta t \) and this accuracy deteriorates rapidly as \( \Delta t \) increases.

For problems in which the solution changes over a time scale which is comparable to the spatial scale, as in hyperbolic problems, the limitation on \( \Delta t \) can make the scheme impractical. We therefore explore other possible expansions that do not suffer from this drawback. A natural candidate is an expansion based on orthogonal polynomials. We thus restrict our discussion to polynomial schemes, i.e., to schemes that employ an algorithm which approximates the numerical solution at time level \( t \) in the following way.
The modified Euler algorithm (2.7) is an example of such a scheme, as well as are most explicit methods.

3. Orthogonal Polynomial Approach

We start by explaining how the new method is constructed in the case of the simple hyperbolic equation

\[ U_t - aU_x = 0 \quad 0 \leq x \leq 2\pi \]
\[ U(x,0) = U_0(x) \]
\[ U(2\pi,t) = U(0,t) \quad t \geq 0 \]

where \( a \) is constant.

The semi-discrete pseudospectral Fourier method can be written in the form

\[ \frac{\partial \overline{V}}{\partial t} = G_N \overline{V} \]

\[ \overline{V}(t=0) = \overline{V}^0 \]

where \( \overline{V}(t) \) is the column vector

\[ (V(x_0), V(x_1), \cdots, V(x_{2N-1}))^T \]

and

\[ V(x,t) = \sum_{k=0}^{m} \alpha_k (G_N t)^k \; V^0(x) \]

The modified Euler algorithm (2.7) is an example of such a scheme, as well as are most explicit methods.
\[ \mathbf{v}^0 = (U_0(x_0), \ldots, U_0(x_{2N-1}))^T. \]

The \(2N \times 2N\) matrix \(G_N\) is given by

\[
(G_N)_{jk} = \begin{cases} 
\frac{a}{2} (-1)^{j+k} \cot \frac{x_j - x_k}{2} & j \neq k \\
0 & j = k.
\end{cases}
\]

(3.3)

(In practice \(G_Nv\) is calculated using two fast Fourier transforms.) \(G_N\) is a skew-symmetric matrix and therefore has a complete set of \(2N\) eigenvectors which will be denoted by \(\bar{\omega}_k\) \(k = 1, 2, \ldots, 2N\). Let

\[
\mathbf{v}^0 = \sum_{k=1}^{2N} b_k \bar{\omega}_k
\]

then

\[
e^{G_N t} \mathbf{v}^0 = \sum_{k=1}^{2N} b_k e^{-\lambda_k t} \bar{\omega}_k = \sum_{k=1}^{2N} b_k \lambda_k t \bar{\omega}_k
\]

(3.4)

where \(\lambda_k\) are the eigenvalues of \(G_N\) corresponding to \(\bar{\omega}_k\). In our case \(\lambda_k\) are purely imaginary. Let \(H_m(G_N t)\) be a polynomial approximation of \(\exp(G_N t)\) of the form

\[
H_m(G_N t) = \sum_{\ell=0}^{m} a_\ell (G_N t)^\ell;
\]

then

\[
H_m(G_N t) \mathbf{v}^0 = \sum_{k=1}^{2N} b_k H_m(\lambda_k t) \bar{\omega}_k
\]

and therefore
\[ \sum_{k=1}^{2N} |b_k|^2 |e^{\lambda_k t} - H_m(\lambda_k t)|^2, \]

if \( b_k \) are arbitrary (see however the remark at the end of this section),

\[ |e^{G_t} - H_m(G_t)|^2 = \max_k |e^{\lambda_k t} - H_m(\lambda_k t)|^2 \leq \max_z |e^z - H_m(z)|^2 \]

where \( z \in \{iat(N-1), iat(N-1)\} \)

We therefore seek a polynomial approximation with real coefficients to the function \( e^z \) that will minimize the expression on the R.H.S. of (3.5).

Define

\[ R = |iat(N-1)| \]

\[ \theta = -iz/R \quad (|\theta| \leq 1) \]

Then

\[ |e^z - H_m(z)|^2 = |e^{i\theta R} - H_m(i\theta R)|^2 = \]

\[ |\cos(\theta R) - H^R_m(\theta R)|^2 + |\sin(\theta R) - H^I_m(\theta R)|^2 \]

where

\[ H_m(i\theta R) = H^R_m(\theta R) + iH^I_m(\theta R) \]

\( (H^R_m, H^I_m \) are polynomials with real coefficients.)

The polynomials that minimize (3.8) are the "best approximation" to \( \cos(\theta R) \) and \( \sin(\theta R) \). It is known that Chebyshev polynomials provide an approximation which is "almost" as good as the "best approximation". In fact, we can quote the following result [7].
Theorem: Suppose that $f \in C[-1,1]$ and $S_n(f) = \|f - q_n^*\|_\infty$ where $q_n^*$ is the least-square approximation with respect to the weight function $(1-z^2)^{-1/2}$ then

$$S_n(f) < (4 + \frac{4}{\pi^2} \log n)E_n(f)$$

$(E_n(f) = \|f - q_n\|_\infty, q_n$ is the best polynomial approximation.)

It follows that the improved accuracy of the best polynomial approximation does not make up for the added computational complexity. Taking $H_m^R(\theta R)$, $H_m^I(\theta R)$ as the Chebyshev polynomial approximations to the trigonometric functions we have [8]

$$H_m^R(\theta R) = J_0(R) + 2 \sum_{k=1}^{\infty} (-1)^k J_{2k}(R)T_{2k}(\theta)$$

(3.9)

$$H_m^I(\theta R) = 2 \sum_{k=1}^{\infty} (-1)^k J_{2k+1}(R)T_{2k+1}(\theta)$$

where $J_k(R)$ is Bessel function of order $K$. Hence

$$H_m(i\theta R) = \sum_{k=0}^{m} (i)^k c_k J_k(R)T_k(\theta)$$

(3.10)

$c_0 = 1$, $c_k = 2$ \quad $k \geq 1$ .

Since (3.7) we have

$$T_k(\theta) = T_k(-i\omega) \quad \omega = z/R \quad \omega \in [-i,i]$$

(3.11)

Define

$$Q_k(w) = (i)^k T_k(-i\omega)$$

(3.12)

using the recurrence relation satisfied by Chebyshev polynomials
it is easily verified that $Q_k(w)$ satisfies the following recurrence relation

$$Q_{k+1}(w) = 2wQ_k(w) + Q_{k-1}(w)$$

(3.14)

$$Q_0(w) = 1, \quad Q_1(w) = w.$$  

Thus, $Q_k$'s are polynomials in $z/R$ with real coefficients so that

$$H_m(z) = \sum_{k=0}^{m} c_k J_k(R)Q_k(z/R)$$

(3.15)

which is the desired approximation.

**Remark 1:** The polynomials $Q_k$ are the imaginary analog of Chebyshev polynomials. They are orthogonal on the interval $[-i,i]$ with respect to the following inner product

$$\langle f, g \rangle = -i \int_{-1}^{1} f(w)g(w)(1-|w|^2)^{-1/2}$$

(3.16)

**Remark 2:** It is apparent from (3.5) that in using the maximum norm we did not take into account the fact that the $b_k$'s are decreasing. To do so requires us to consider the larger set of Gegenbauer polynomials. When the degree $m$ is large it can be shown that the improvement thus achieved is negligible so that only for small values of $m$ is the larger set relevant. A detailed analysis of the use of Gegenbauer polynomials will be carried out in a future paper which will deal with nonlinear problems.
4. Resolution in Time

Let us define first the notion of resolution. The accuracy of a polynomial approximation is defined by its asymptotic rate of convergence as \( m \) (the degree of the polynomial) tends to infinity. Denote by \([m_0, \infty)\) the interval of the asymptotic behavior, \( m \) has then to be greater or equal to \( m_0 \) in order to have resolution. This is a necessary condition but not sufficient. For example, if the relative error is of order 1, the results are meaningless and we have no resolution. Therefore, we define the condition of having a meaningful resolution as one in which \( m \geq m_0 \) and the relative error norm is less than 10\%. To be precise, assume that for \( m \in [m_0, \infty) \), the minimal \( m \) which achieves this accuracy is \( \bar{m}_0 \), we then say that a necessary and sufficient condition for resolution is

\[
(4.1) \quad m > \bar{m}_0
\]

Applying the above definition to our case means that one has to apply the spatial operator \( tG_N, \bar{m}_0 \) times in order to resolve \( N \) modes of the exact solution of (3.1) at time level \( t \).

Let us see what is \( \bar{m}_0 \) in the case of Chebyshev polynomials approximation.

Using the results from the previous section we have

\[
(4.2) \quad e^z = \sum_{k=0}^{\infty} c_k^J J_k(R) Q_k(z/R)
\]

It is known [10] that \( J_k(R) \) converges to zero exponentially fast when \( k \) increases beyond \( R \). It implies that the interval of asymptotic behavior is
While \( m_0 = [R] \). Because of the exponentially rate of convergence, \( \overline{m}_0 \) is close to \( m_0 \), and when \( R \) is large we can consider \( \overline{m}_0 \) as \( m_0 \) for any practical use. Thus, we obtain that in order to resolve \( N \) modes one has to use the spatial operator at least \( |at(N-1)| \) times.

For comparison let us analyze the resolution qualities of the leap-frog scheme. This scheme is a typical explicit scheme which evaluates the numerical solution at the \( n + 1 \) time set, using data from the two previous time levels

\[
\overline{v}^{n+1} = \overline{v}^{n-1} + 2\Delta t G \overline{v}^n
\]

A straightforward eigenvalues analysis implies that there are two solutions for the amplification factor of each mode \( w_k \).

\[
u_1 = \Delta t \lambda_k + \sqrt{\left(\Delta t \lambda_k\right)^2 + 1}
\]

\[
u_2 = \Delta t \lambda_k - \sqrt{\left(\Delta t \lambda_k\right)^2 + 1}
\]

with

\[
\lambda_k = iak
\]

The scheme is stable when \( |ak\Delta t| \leq 1 \) and we get

\[
|\nu_{1,2}| = 1
\]

which means that the error of the scheme is only a phase error.

Let us assume that we choose the initial data at the first two levels in such a way that only \( \nu_1 \) is relevant. Therefore
\[(4.7) \quad \bar{V}^n = \mu^n V^0 \]

or
\[(4.8) \quad \bar{V}^n = e^{i\psi} V^0 = e^{i \frac{t}{\Delta t} \psi} V^0 \]

where
\[(4.9) \quad \psi = \frac{\theta}{\sqrt{1 - \theta^2}} = \theta + \frac{\theta^3}{6} + \mathcal{O}(\theta^5) \]

is the phase shift of the numerical scheme after one time step. The quantity
\[\theta = ak\Delta t\] is the phase shift of the exact solution after \(t = \Delta t\). Hence, the
phase error at time level \(t\) is

\[(4.10) \quad \Delta \theta = \frac{t}{\Delta t} \frac{\theta^3}{6} + \mathcal{O}(\theta^5) = \frac{tk}{6} a^3 \Delta t^2 + \mathcal{O}(\Delta t^4) \]

The largest mode is \(V_N(N = \frac{\pi}{\Delta x})\) so that the maximum phase error is

\[(4.11) \quad \Delta \theta_{\text{max}} = \frac{t}{6} (a\pi)^3 \frac{\Delta t^2}{\Delta x^2} + \mathcal{O}(\Delta t^4) \]

This scheme is obviously second order in time and error \(E\) is

\[(4.12) \quad E = |e^{-i\theta} - e^{i(\theta + \Delta \theta)}| = |1 - e^{i\Delta \theta}| \]

When \(\Delta \theta > \pi\), decreasing \(\Delta t\) would not necessarily decrease the error. Thus
resolution is achieved when \(\Delta \theta_{\text{max}}\) is at least less than \(\pi\). Therefore since
\[(4.11) \) we obtain

\[(4.13) \quad \frac{a^3 \Delta t^2}{\delta x^2} \leq \frac{6}{\pi^2} \frac{1}{\pi^2} + \mathcal{O}(\Delta t^4) \]
or, using the notion of $m_0$ defined at the beginning of the section,

\begin{equation}
(4.14) \quad m_0 = \frac{\pi}{6^{1/2}} \left( \frac{\Delta t}{\Delta x} \right)^{3/2} = \frac{1}{(6\pi)^{1/2}} (\tan N)^{3/2} + O(\Delta t^4)
\end{equation}

and the sufficient condition for 10% error results in

\begin{equation}
(4.15) \quad |\Delta \theta| \leq 10^{-1} \ .
\end{equation}

Hence, using (4.10) we have

\begin{equation}
\bar{m}_0 = \left( \frac{5}{3} \right)^{1/2} (\tan N)^{3/2} + O(\Delta t^4) \ .
\end{equation}

Thus we have obtained the following result: in order to get a resolution of $N$ modes by the leap-frog type scheme one has to operate with $tG_N$ at least

$$\left( \frac{5}{3} \right)^{1/2} |(\tan)|^{3/2} \ \text{times, a requirement much more stringent than the previous result of } |\tan(N-1)| \ \text{operations for the Chebyshev approximation. For example, when } t = 2\pi, a = 1 \ \text{and } N = 32 \ \text{one has to apply the spatial operator, in the leap-frog case, approximately 1300 times compared to 100 times in the Chebyshev approximation case.}

As stated previously, for any practical use one can identify Chebyshev polynomials approximation as the "best polynomial". Thus we conclude that from resolution point of view the scheme based on these polynomials yields the best results.
5. Stability

In the last section we discussed in detail the notion of resolution. It is clear from the above discussion that resolution implies stability. In fact, since

\[ |H_m(G_N t) - e^{G_N t} N_t| \leq \text{const} \]

\( H_m(G_N t) \) must be bounded independently of \( m \) and \( N \). The converse is not true in general. Consider for example the leap-frog time difference scheme. It has been shown in Section 4 that in order to get resolution we need

\[ m \geq \left( \frac{5}{3} \right)^{1/2} (\tan)^{3/2} \]

or, equivalently,

\[ \frac{a \Delta t}{\Delta t} \leq \left( \frac{3}{5m} \right)^{1/3} \frac{1}{\pi} \]

this is in contrast to the stability condition

\[ \frac{a \Delta t}{\Delta t} \leq \frac{N}{N+1} \frac{1}{\pi} \sim \frac{1}{\pi} \]

for the leap-frog scheme [5] which allows a much larger \( \Delta t \).

When a time step \( \Delta t \) is chosen based on the condition (5.3) rather than (5.2), one may get meaningless results in spite of the stability of the scheme. To illustrate this we solved the equation

\[ u_t - u_x = 0 \quad 0 \leq x \leq 2\pi \]

(5.4)

\[ u_0(x) = \cos kx \quad 1 \leq k \leq 7 \]

numerically.
The exact solution is

\[ u(x,t) = \cos(k(x+t)). \]

Using a grid of 16 points in space assures us that the error at time level \( t \) comes solely from the time discretization.

The results are:

<table>
<thead>
<tr>
<th>K</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2ERROR</td>
<td>( .9236 \times 10^{-2} )</td>
<td>( .767 \times 10^{-1} )</td>
<td>.2722</td>
<td>.6070</td>
<td>1.291</td>
<td>2.096</td>
<td>.9453</td>
</tr>
</tbody>
</table>

\( t = 3.625; \ \Delta x = .3927; \ \Delta t = \Delta x/\pi = .1250 \)

These computations illustrate the above claim: meaningful results are achieved only for \( 1 \leq k \leq 3 \), while for \( 4 \leq k \leq 7 \) the results are meaningless despite the fact that we have used a spatial approximation that resolves all the modes exactly.

The conclusion is obvious: for nonstiff problems, the important property is resolution rather than stability. It is inefficient to use a scheme which is stable but does not resolve all the modes. A scheme with less modes and the same degree polynomial in time will produce the same results. In the leap-frog case, for example, any results achieved by using the maximum time step allowed by the stability condition could be achieved with less amount of work by using coarser grid and the same time step.
Condition (5.2) can also be written as

\[ \frac{a^3 \Delta t^2}{\Delta x^3} \leq \frac{3}{5} \frac{1}{\pi^2 t} \]

thus, it is obvious that in order to get resolution in the leap-frog case \( t \) has to be proportional to \( (\Delta x)^{3/2} \) and not to \( \Delta x \) as required by stability. A same proportion between \( \Delta t \) and \( \Delta x^{3/2} \) is needed to get resolution for any scheme second order in time, and it does not matter if the scheme is stable or not for \( \Delta t \) proportional to \( \Delta x \).

For any scheme of order \( P \) the truncation error can be written as

\[ E = c \cdot \Delta t^{P+1} \cdot N^{P+1} + O(\Delta t^{P+2} N^{P+2}) \]

thus, the overall error in time \( t \) is

\[ E = \frac{t}{\Delta t} \cdot c \cdot \Delta t^{P+1} \cdot N^{P+1} + O(\Delta t^{P+2} N^{P+2}) \]

or

\[ E = t \cdot c \cdot \pi^{P+1} \frac{\Delta t^P}{\Delta x^{P+1}} + O\left(\frac{\Delta t^{P+1}}{\Delta x^{P+2}}\right) \]

It follows that for a scheme of order \( P \), \( \Delta t \) has to be proportional to \( \frac{\Delta x^P}{\Delta x^P} \) in order to get resolution.

Considering the result of Lemma 2 and using the relations \( \Delta t = t/m \), \( \Delta x = \pi/n \) the requirement for resolution in the Chebyshev polynomial case is equivalent to

\[ (5.6) \quad \frac{a \cdot \Delta t}{\Delta x} \leq \frac{N}{N-1} \frac{1}{\pi} \frac{1}{\pi} \]
which means that $\Delta t$ has to be proportional to $\Delta x$. Thus, this new algorithm can be regarded as a limit case of finite order schemes.

We would like to mention here another important result which follows this stability discussion. The stability condition (5.3) for the leap-frog type schemes is much more stringent than the C.F.L. condition of a similar algorithm based on finite difference approximations despite the fact that spectral approximation uses all the previous mesh points. Thus, it was regarded as an artificial condition which may be overcome by properly designed time algorithm. Observing the fact that this stability restriction is exactly (5.6) which is the resolution condition for the orthogonal polynomials algorithm we conclude that this severe stability condition is an essential one which can not be violated. The reason is due to the fact that the spectral radius of the operator $G_N$ is increasing. For example, in the leap-frog type algorithm the eigenvalues of $G_N$, using finite difference approximation in space, are

$$
\lambda_k = \frac{i \sin(KAx)}{\Delta x}, \quad -N \leq k \leq N
$$

(5.7)

and the spectral radius is

$$
r_{\text{FD}} = \max_{k} \left| \frac{i \sin(KAx)}{\Delta x} \right| < \frac{1}{\Delta x} = \frac{N}{\pi}.
$$

On the other hand, for spectral (in space) approximation, the eigenvalues are

$$
\lambda_k = ik, \quad -N \leq k \leq N
$$

(5.8)

hence, the spectral radius is
(5.9) \[ r_{SP} = \max_{k} |ik| = N - 1 \]
so that
\[ \frac{r_{SP}}{r_{FD}} = \frac{\pi N - 1}{N} \sim \pi \]

6. Accuracy

According to (4.4) we have the following expression for the coefficients

\[ b_k = (i)^k c_k J_k(R) \quad c_0 = 2, \quad c_k = 1 \quad k \geq 1 \]

Bessel functions satisfy the following inequality [10]

(6.1) \[ |J_m(m\phi)| \leq \left| \frac{\phi \exp(\sqrt{1-\phi^2})}{1+\sqrt{1-\phi^2}} \right| m \quad |\phi| < 1 \]

Define

(6.2) \[ \alpha \equiv \left| \frac{\phi \exp(\sqrt{1-\phi^2})}{1+\sqrt{1-\phi^2}} \right| \quad |\phi| < 1 \]

so that

\[ \frac{d\alpha}{d\phi} = \begin{cases} \frac{\beta e^\beta}{1+\beta} & \phi \geq 0 \\ -\frac{\beta e^\beta}{1+\beta} & \phi < 0 \end{cases} \]

where

\[ \beta = \sqrt{1-\phi^2} \]
Thus $\alpha$ is monotone decreasing for $-1 < \phi \leq 0$ and increases monotonically for $0 \leq \phi < 1$; and also

\[ \alpha(0) = 0, \quad \alpha(1) = \alpha(-1) = 1. \]

Hence it follows that

\[ 0 \leq \alpha < 1. \]

In our case $m = R$ which implies

\[ \phi = \frac{R}{m} = \frac{taN - 1}{m} \sim \frac{taN}{m}. \]

Thus, refinement of the approximation in space and time keeping the same proportion of $N/m$ reproduces a scheme whose error in time converges to zero as

\[ \alpha^m \to 0 \quad 0 \leq \alpha < 1 \]

Hence, we have produced with a scheme which has spectral accuracy both in time and space.

We would like to point out an interesting result which can be concluded from this analysis. When $T$ is large $m$ is large as well since it has to be greater than $R$ in order to have resolution. According to (6.8), once we obey this requirement of resolution ($\phi < 1$) the time error is negligible, and the error of the numerical solution of the (2.1) comes only from the spatial approximation.
7. The New Algorithm

In this section we describe the actual construction of the algorithm.

In order to obtain our scheme we use the expansion (4.2)

\[ e^z = \sum_{k=0}^{m} c_k J_k(R) Q_k(z/R) \]

\[ c_0 = 1, \quad c_k = 2 \quad k \geq 1. \]

Substituting \( G_N t \) for \( z \) results in

\[ V(t) = e^{-G_N t} V^0 = \sum_{k=0}^{m} c_k J_k(R) Q_k(G_N) V^0 \]

where \( G_N = \frac{t}{R} N \). Using the recurrence relation (3.14) and (3.11) we get

\[ Q_k(G_N) V^0 = 2 G_N Q_k-1(G_N) V^0 + Q_k-2(G_N) V^0 \]

\[ Q_0(G_N) V^0 = V_0, \quad Q_1(G_N) V = G_N V^0. \]

Using (7.2) in (7.1) enables us to compute \( \overline{V}(t) \) by operating with \( G_N \) \( m \) times. It is obvious that because of the use of the recurrence relation this scheme may be regarded as a two level scheme. Therefore, it has the disadvantage of requiring extra memory. This disadvantage cannot be overcome by converting (7.1) to a power series in \( G_N \) and using Horner scheme to compute \( \overline{V}(t) \) because huge roundoff errors result.

A useful way to compute \( \overline{V}(t) \) by a one level scheme is to calculate the roots of the polynomial
(7.3) \[ p(z) = \sum_{k=0}^{m} c_k J_k(R) Q_k(z). \]

Let us assume that the roots are

(7.4) \[ \lambda_1, \lambda_2, \ldots, \lambda_m. \]

Since \( c_k J_k(R) \) are real, every complex root appears with its conjugate. Rearranging (7.4) in such a way that the first \( 2p \) roots are \( p \) couples of a complex number and its conjugate we get

(7.5) \[ \mu_1, \overline{\mu}_1, \ldots, \mu_p, \overline{\mu}_p, \mu_{2p+1}, \ldots, \mu_m. \]

Thus, (7.3) can be written as

(7.6) \[ p(z) = \prod_{i=1}^{p} \left( 1 - \alpha_i z + \beta_i z^2 \right) \prod_{i=2p+1}^{m} \left( 1 - \gamma_i z \right) \]

while

\[ \alpha_0 = \frac{m}{2} \sum_{k=0}^{m/2} c_{2k} J_{2k}(R), \quad \beta_i = \frac{1}{|\mu_i|^2} \]

(7.7) \[ \alpha_i = \frac{2R \mu_i}{|\mu_i|^2}, \quad \gamma_i = \frac{1}{\mu_i}. \]

Hence we get

(7.8) \[ p(z) = \prod_{i=1}^{p} \left[ I - \alpha_i \left( \overline{\gamma}_N \right) + \beta_i \left( \overline{\gamma}_N \right)^2 \right] \prod_{i=2p+1}^{m} \left[ I - \gamma_i \left( \overline{\gamma}_N \right) \right] \psi^o. \]
This is obviously a scheme that uses the minimal memory required for an explicit scheme.

Our algorithm can be used as one step method by getting the solution at the final time $t$ directly from the initial data. It can also be used as a marching scheme if one is interested in intermediate results. The size of the time step $\Delta t$ depends only on the information one wants to get out of the numerical procedure. $\Delta t$ enters instead of $t$ in the expressions above and the parameters $R,m$ are determined accordingly. In any case, the refinement of the algorithm is done by increasing the degree of the polynomial and not by decreasing the size of the time step.

8. Variable Coefficients

In the variable coefficient case, the operator $G$ is

$$(8.1) \quad G = a(x) \frac{\partial^3}{\partial x^3}.$$ 

It is approximated in the numerical procedure by the matrix $G_N$ which is a multiplication of two matrices

$$(8.2) \quad G_N = A_N \cdot D_N$$

where $A_N$ is a diagonal matrix whose elements are

$$(8.3) \quad (A_N)_{ij} = a(x_j)\delta_{ij}$$
and $D_N$ is a skew-hermitian matrix which approximates the derivative spectrally. It is clear from this representation that $G_N$ is no longer normal. Nevertheless the main results of our approach are still valid. In the non-normal case, the set of eigenvectors is not always complete and one cannot use (3.4) - (3.5).

In order to justify our approach in the non-normal case we use the following definition for a function fo a matrix $A$ [3]:

$$f(A) = \sum_{k=1}^{n} \left[ f(\lambda_k)z_{k1} + f^{(1)}(\lambda_k)z_{k2} + \cdots + f^{(m_k-1)}(\lambda_k)z_{km_k} \right]$$

where $\lambda_k$ are the eigenvalues of the matrix $A$ and $m_k$ is the multiplicity of $\lambda_k$ in the minimal polynomial of $A$. The matrices $z_{kj}$ are completely determined when $A$ is given and do not depend on the choice of the function $f$. Expression (8.4) has a meaning when $f$ and its required derivatives are defined on the spectrum of $A$. In our case $f$ is the exponent function which is a well defined analytic function. Hence, using definition (8.4) it is also obvious that in the general case approximating the exponent matrix is equivalent to approximating the scalar exponent and its derivatives on a domain which includes all the eigenvalues of the matrix.

In the constant coefficients case the domain is

$$I = [-iaN,iaN].$$

The following theorem implies that this result is valid also in the variable coefficients case when $a(x)$ doesn't change sign in the interval and
(8.6) \[ a = \max_x |a(x)| \quad x \in [0, 2\pi] \]

**Theorem:** If \( a(x) > 0 \) and \( \lambda_k \) is an eigenvalue of \( A_ND_N \) then \( \lambda_k \in I \).

\( I \) is defined by (8.5), (8.6).

**Proof:** Define the following inner product

(8.7) \[ [u, v]_{A^{-1}} = (u, A^{-1}v) \]

Then we have

(8.8) \[ [u_n, (u_N)^*]_{A^{-1}} = [u_N, A_N D_N u_N]_{A^{-1}} = (u_N, D_N u_N) \]

where \((u_N, D_N u_N)\) is real; thus

(8.9) \[ (u_N, D_N u_N) = (D_N u_N, u_N) = (u_N, D_N^* u_N) = -(u_N, D_N u_N). \]

Hence

(8.10) \[ (u_N, D_N u_N) = 0 \]

Using this result in (8.8) we get

(8.11) \[ \frac{d}{dt} \left\{ \frac{1}{2} [u_N, u_N]_{A_N^{-1}} \right\} = 0 \]

thus

(8.12) \[ \| u_N \|_{A_N^{-1}} = \text{const}, \]

Assuming that \( w_k \) is the eigenvector of \( A_ND_N \) corresponding to \( \lambda_k \) we can use it as the initial data so that
(8.13) \[ u_N = e^{A_N D_N t} w_k = e^{\lambda_k t} w_k ; \]

hence

(8.14) \[ [u_N^*, u_N]_{A_N}^{-1} = [e^{\lambda_k t} w_k^*, e^{\lambda_k t} w_k]_{A_N}^{-1} = e^{2(Re \lambda_k)t} [w_k^*, w_k]_{A_N}^{-1} \]

and according to (8.12) this is constant. This implies that

(8.15) \[ Re \lambda_k = 0 \]

hence \( \lambda_k \) is pure imaginary. In addition

(8.16) \[ \max_k |\lambda_k| \leq \|A_N D_N\| \leq \|A_N\| \|D_N\| = a^* N \]

and the proof is concluded.

The proof is essentially the same when \( a(x) < 0 \) in the interval.

The case when \( a(x) \) changes sign is more complicated and not much is yet known. [6] gives a proof of stability for the simple case when \( a(x) \) is a trigonometric polynomial of order 1. It implies that in this simple situation

(8.17) \[ -\alpha \leq Re \lambda_k \leq \alpha \quad a > 0 \]

while \( \alpha \) doesn't depend on \( N \). Because of (8.16) there exists an ellipse whose larger axis is \([-iaN, iaN]\) and the small one is \([-\alpha, \alpha]\) which contains the eigenvalues of \( A_N D_N \). The theory of Chebyshev polynomial approximations guaranties convergence in this domain [9].
Remark Numerical experiments lead us to the assumption that (8.17) is also valid in the general case that \( a(x) \) is any periodic function.

9. Numerical Results

In order to illustrate the spectral convergence of our scheme we shall consider the following scalar problem

\[
(9.1) \quad u_t - a(x)u_x = 0 \quad 0 \leq x \leq 2\pi .
\]

In Table 1 we take

\[
(9.2) \quad a(x) = \frac{1}{2 + \cos x}, \quad u_0(x) = \sin(2x + \sin x) .
\]

The exact solution to this problem is

\[
(9.3) \quad u(x,t) = \sin(2x + \sin x + t) .
\]

The numerical solution is computed at time level \( T = 6.283 \).

\( N = \) Number of mesh points in space.

\( M = \) The degree of the polynomial approximation.

The ratio of the \( L_2 \) errors illustrates very clearly the spectral convergence of the scheme.
In Table 2 we take

\[(9.4) \quad a(x) = \sin(x), \quad u_0(x) = \sin(x)\]

\((a(x) \) changes sign in the interval). The exact solution is

\[(9.5) \quad u(x,t) = \sin(2\tan^{-1}(e^{t\tan\frac{x}{2}}))\]

The solution is computed at \( T = 1.571 \).
Table 3 illustrates the resolution properties of the scheme. In this case we
\[
a(x) = \frac{1}{2 + \cos x}, \quad T = 50.27, \quad N = 16
\]

Because of the high resolution in space only time errors occur. According to the theory developed previously (Lemma 2) the degree of the polynomial approximation has to be at least \( R \). We have
\[
R = at(N-1) = 50.27 \times 15 = 754.05
\]

<table>
<thead>
<tr>
<th>( M )</th>
<th>( L2 ) Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>740</td>
<td>1.120</td>
</tr>
<tr>
<td>750</td>
<td>5.981 \times 10^{-1}</td>
</tr>
<tr>
<td>760</td>
<td>1.354 \times 10^{-1}</td>
</tr>
<tr>
<td>770</td>
<td>1.476 \times 10^{-2}</td>
</tr>
<tr>
<td>780</td>
<td>1.048 \times 10^{-3}</td>
</tr>
<tr>
<td>840</td>
<td>5.391 \times 10^{-13}</td>
</tr>
</tbody>
</table>

The result for \( M = 840 \) shows that, while the minimal \( M \) required for resolution is 755, increasing \( M \) by only 11% gives machine accuracy.

In Tables 4 and 5 we compare our scheme to the leap-frog scheme. The model problem is (9.1), (9.2) with \( N = 32 \). In Table 4 we compute the numerical solution at different time levels. The results clearly shows the high accuracy of the Chebyshev polynomials approach. Another interesting
phenomenon which is illustrated by the table is the following: while the
leap-frog case the error increased due to accumulation this is not so in our
case. In contrast it decreased geometrically. The explanation is obvious
using (6.7), (6.8). Increasing \( t \) and \( m \), keeping the same proportion
between them, results in \( \phi = \text{constant} \) and therefore (6.8) is valid.

In the following table

C.P. = Chebyshev polynomials.

L.F. = leap-frog.

\[
\begin{array}{cccc}
\text{Time} & \text{M} & \text{L2 Error (L.F.)} & \text{L2 Error (C.P.)} \\
1.571 & 35 & 5.247 \times 10^{-4} & 8.726 \times 10^{-5} \\
3.142 & 70 & 1.108 \times 10^{-3} & 2.084 \times 10^{-7} \\
6.283 & 140 & 2.184 \times 10^{-2} & 1.429 \times 10^{-13} \\
\end{array}
\]

In Table 5 we compare our scheme to the leap-frog scheme from the point of
view of the amount of work needed to achieve a certain degree of accuracy.
The L2 Error is computed at time level \( T = 6.283 \).

\[
\begin{array}{ccc}
\text{L2 Error} & \text{u(L.F.)} & \text{u(C.P.)} \\
1.0 \times 10^{-4} & 580 & 110 \\
1.0 \times 10^{-6} & 3480 & 117 \\
1.0 \times 10^{-8} & 56000 & 122 \\
\end{array}
\]
Acknowledgement

I would like to thank my advisor, Professor David Gottlieb, for his helpful guidance.
References


A pseudospectral numerical scheme for solving linear, periodic, hyperbolic problems is described. It has infinite accuracy both in time and in space. The high accuracy in time is achieved without increasing the computational work and memory space which is needed for a regular, one step explicit scheme. The algorithm is shown to be optimal in the sense that among all the explicit algorithms of a certain class it requires the least amount of work to achieve a certain given resolution. The class of algorithms referred to consists of all explicit schemes which may be represented as a polynomial in the spatial operator.