THE CHEBYSHEV-LEGENDRE METHOD: IMPLEMENTING LEGENDRE METHODS ON CHEBYSHEV POINTS

Wai Sun Don
David Gottlieb


Unclas

NASA Contract No. NAS1-19480
July 1993

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23681-0001
THE CHEBYSHEV-LEGENDRE METHOD: IMPLEMENTING
LEGENDRE METHODS ON CHEBYSHEV POINTS

Wai Sun Don and David Gottlieb
Division of Applied Mathematics
Brown University
Providence, RI 02912

ABSTRACT

We present a new collocation method for the numerical solution of partial differential equations. This method uses the Chebyshev collocation points, but because of the way the boundary conditions are implemented, has all the advantages of the Legendre methods. In particular $L_2$ estimates can be obtained easily for hyperbolic and parabolic problems.
1 Introduction

Polynomial pseudospectral (or collocation) methods have been extensively used in the numerical solutions of partial differential equations. The underlying idea in those methods is to approximate the unknown function by an interpolation polynomial at some pre-described (collocation) points. The polynomial is then required to satisfy the PDE at the collocation points. This procedure yields a system of ordinary differential equation to be solved.

Historically, (see [10]) the first such points to be used were the Chebyshev collocation points

$$x_j = \cos\left(\frac{\pi j}{N}\right) \quad 0 \leq j \leq N.$$  

Those points were chosen because they allowed the use of Fast-Fourier-Transforms in the computations. It was only later (see [7]) that those points were identified with the nodes of the Gauss-Lobatto-Chebyshev (G-L-C) quadrature formula. This observation is the key in the stability analysis of the pseudospectral Chebyshev methods. The G-L-C quadrature formula led to the weighted $L_2$ norm

$$\int_{-1}^{1} f^2(x) \frac{dx}{\sqrt{1-x^2}}$$

However, it has been noted in [8] that this is not a natural norm for hyperbolic equations. In fact the differential equation is not well posed in this norm. Also it complicated the stability analysis even for parabolic equations. The theory (and therefore the confidence in applying those methods) is not complete.

Once the connection between the collocation points and the Gauss Lobatto points is established, it is natural to use the nodes of the Gauss-Lobatto-Legendre (G-L-L) quadrature formula. We refer the reader to [2] for review of those methods. Recently [1] an $O(N \log N)$ method was proposed for the Legendre points. The main problem with those points are that they are not given explicitly, and their evaluation for large $N$ is not robust due to roundoff errors.

In this paper we present a method (and name it The Chebyshev-Legendre Method) that has the advantages of both the Chebyshev and Legendre methods. The method utilizes the Chebyshev collocation points allowing the use of fast Fourier algorithms and avoiding the roundoff error associated with computing the Legendre grid points. The boundary conditions are imposed via a new penalty technique in such a way that the method is stable in the usual $L_2$ norm (rather than the weighted $L_2$ norm). Hence the Chebyshev-Legendre method enjoys the advantages of the Chebyshev method as well as those of the Legendre method.
The implementation of the boundary conditions is done by a penalty method. A penalty term is added to the PDE at all grid points in such a way that, in the limit of number of grid points tend to infinity, the boundary conditions are satisfied. This procedure seems to be better than the direct imposition of the boundary conditions, and in our case has the extra advantage of yielding the Legendre method at the Chebyshev points.

A similar idea had been tried by Reyna [11]. The difference between his approach and ours is in the imposition of the boundary conditions. Instead of transforming from the Chebyshev basis to the Legendre one as in [11], we impose the boundary conditions via penalty method and through that automatically switch to the Legendre basis without using it in the differentiation procedure.

The paper is organized as follows:

In Section 2 we quote the essential formulas for the use of Chebyshev and Legendre methods.

In Section 3 we present the Chebyshev-Legendre method for hyperbolic equations. In subsection 3.1 we describe the method and prove (in Theorem (3.1.1)) an energy estimate to show stability. In Theorem (3.1.2) we bring another version of this energy estimate.

In subsection 3.2 we consider the relationship of the new method to the Legendre penalty method and show that the differentiation matrices of the two methods are related via a similarity transformation. This fact is proved in Theorem 3.2.2.

In Section 4 we discuss and prove the stability of Chebyshev-Legendre method for the heat equation with Robin boundary conditions.

Section 5 concludes the paper with some numerical experiments with the new method.

In future work we will report on the convergence results of the new method for nonlinear hyperbolic equations.

## 2 Preliminaries

This Section is devoted to the definitions of the pseudospectral methods to be used later. We will discuss Chebyshev and Legendre methods which are based on the Chebyshev polynomials

\[
T_N(x) = \cos(N \cos^{-1} x)
\]  

(2.1)

and the Legendre polynomials

\[
P_N(x) = \frac{1}{2^N N!} \frac{d^N}{dx^N} (x^2 - 1)^N
\]  

(2.2)

respectively. Associated with these two polynomials are several Gauss-type quadrature formulas. We will consider, in this paper, the Gauss Lobatto type formulas.
We start by defining the *Chebyshev Collocation points* \( x_j \) by
\[
x_j = \cos\left(\frac{\pi j}{N}\right) \quad 0 \leq j \leq N
\] (2.3)

These points are the zeroes of the polynomial \((1 - x^2)T'_N(x)\) and associated with it, we have

**The Gauss Lobatto Chebyshev Quadrature Formula:**

Let \( f(x) \) be a polynomial of degree \( 2N - 1 \), then
\[
\sum_{j=0}^{N} f(x_j)c_j = \int_{-1}^{1} f(\xi)(1 - \xi^2)^{-\frac{1}{2}} d\xi
\] (2.4)

where the weight \( c_j \) are given by
\[
c_j = \frac{\pi}{N} \quad 1 \leq j \leq N - 1
\]
\[
c_0 = c_N = \frac{\pi}{2N}
\] (2.5)

Similarly, the *Legendre Collocation points* \( y_j \) are defined as the roots of the polynomial \((1 - x^2)P'_N(x)\). For these points we have

**The Gauss Lobatto Legendre Quadrature Formula:**

Let \( f(x) \) be a polynomial of degree \( 2N - 1 \), then
\[
\sum_{j=0}^{N} f(y_j)w_j = \int_{-1}^{1} f(\xi) d\xi
\] (2.6)

where the Gauss Lobatto weights \( w_j \) are given by
\[
\omega_j = -\frac{2}{N+1}[P_N(y_j)P_{N-1}(y_j)]^{-1} \quad 1 \leq j \leq N - 1
\]
\[
\omega_0 = \omega_N = \frac{2}{N(N+1)}
\] (2.7)

Unlike the Chebyshev points that are known explicitly, there is no explicit formula for the Legendre points \( y_j \), they have to be computed numerically. It is interesting though that there is a simple formula, easily and robustly computed, for the values of the Legendre polynomials and their derivative at the *Chebyshev points*. In fact we have the following explicit formula for \( P'_N(x_j) \), (taken from [4], page 180)
\[
P'_N(\cos \theta) = \sum_{m=0}^{N-1} \frac{(\frac{3}{2})^m(\frac{3}{2})^{N-1-m}}{m!(N-1-m)!} \cos(N-1-2m)\theta .
\] (2.8)
Pseudospectral (or Collocation) methods are based on interpolations at the points \(x_j\) or \(y_j\). Consider the polynomials

\[
Q_L(x) = (1 - x^2)P'_N(x),
\]
\[
Q_C(x) = (1 - x^2)T'_N(x),
\]

and define the Legendre-Lagrange polynomials by

\[
h_j(x) = \frac{Q_L(x)}{(x - y_j)Q'_L(y_j)},
\]

and the Chebyshev-Lagrange polynomials by

\[
g_j(x) = \frac{Q_C(x)}{(x - x_j)Q'_C(x_j)}.
\]

Then, the Legendre interpolation operator \(I_L\) is defined by

\[
(I_L f)(x) = \sum_{j=0}^{N} f(y_j)h_j(x)
\]

whereas the Chebyshev interpolation operator \(I_C\) is defined by

\[
(I_C f)(x) = \sum_{j=0}^{N} f(x_j)g_j(x).
\]

By definition, we have

\[
(I_L f)(y_j) = f(y_j)
\]
\[
(I_C f)(x_j) = f(x_j).
\]

From the definition of the interpolation operators \(I_L\) and \(I_C\) we get the spectral differentiation matrices \(D_L\) and \(D_C\) as follows:

The Pseudospectral Legendre Differentiation Matrix \(D_L\), is defined by

\[
(D_L)_{i,k} = h'_k(y_j).
\]

The Pseudospectral Chebyshev Differentiation Matrix \(D_C\), is defined by

\[
(D_C)_{i,k} = g'_k(x_j).
\]

(see [3] for explicit expressions for the matrices)
3 Hyperbolic Equations

3.1 Scalar Hyperbolic Equation

In this Section we consider the scalar initial-boundary value hyperbolic equation

\[ U_t = U_x \quad -1 \leq x \leq 1 \quad t \geq 0 \]  

(3.1)

with the initial condition

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

(3.2)

and the boundary condition

\[ U(1,t) = g(t) \]  

(3.3)

The Chebyshev-Collocation (Pseudospectral) method involves seeking an \( N \)th degree \( x \)-polynomial \( u_N(x, t) \) that satisfies

\[ \frac{\partial u_N(x,t)}{\partial t} = \frac{\partial u_N(x,t)}{\partial x} \quad \text{at} \quad x = x_j \quad 1 \leq j \leq N \]  

(3.4)

with the boundary condition

\[ u_N(1,t) = u_N(x_0,t) = g(t) \]  

(3.5)

where \( x, \) are determined in (2.3) \((x_0 = 1)\).

Note that the equation is satisfied at all the grid points except at the boundary point \( x = 1 \) where the boundary condition is satisfied.

In general, the term \( \frac{\partial}{\partial x} u_N(x,t) \) is evaluated at all the grid points, with the use of either FFT or matrix-vector multiplication using the matrix \( D_C \). Equation (3.4) is then advanced at all the grid points. The value of the solution at the boundary is then updated using (3.5).

In [5] a penalty type method was introduced. In that approach we still use equation (3.4) for the inner points \( x_j, 1 \leq j \leq N \), however instead of using (3.5) for the boundary, the following equation is satisfied

\[ \frac{du_N(1,t)}{dt} = \left. \frac{\partial u_N(x,t)}{\partial x} \right|_{x=1} - \tau (u_N(1,t) - g(t)) \]  

(3.6)

where \( \tau \) is determined from stability considerations. In particular it had been found that stability follows if

\[ \tau \geq \frac{N^2}{2} \]
Equations (3.4) and (3.6) can be combined into a single equation by noting that the collocation points $x_j$ defined in (2.3) are the zeroes of the polynomial $(1 - x^2)T_N'(x)$. Thus the penalty method [5] can be written now as

$$\frac{\partial u_N(x_j, t)}{\partial t} = \frac{\partial u_N(x, t)}{\partial x} \bigg|_{x=x_j} - \tau \frac{(1 + x_j)T_N'(x_j)}{2T_N'(1)} (u_N(1, t) - g(t))$$

(3.7)

for $j = 0, \ldots, N$.

The main difference between the penalty method (3.7) and the usual Chebyshev method given in (3.4) and (3.5) is that the numerical solution $u_N(x, t)$ does not satisfy the boundary condition exactly, but only in the limit as $N \to \infty$. The boundary condition is now part of the equation.

Another penalty method, based on the Legendre points $y_j$ is presented in [6]. Similar to (2.6) we write this method as

$$\frac{\partial u_N(y_j, t)}{\partial t} = \frac{\partial u_N(x, t)}{\partial x} \bigg|_{x=y_j} - \tau \frac{(1 + y_j)P_N'(y_j)}{2P_N'(1)} (u_N(1, t) - g(t))$$

(3.8)

for $j = 0, \ldots, N$.

The parameter $\tau$ is determined by the stability requirement. Thus the differential equation is satisfied at the points $y_j, j = 1, \ldots, N$. At the boundary $x_0 = 1$ one uses a combination of the boundary condition and the differential equation.

An obvious disadvantage of the method in (3.8) is that it utilizes the Legendre points. However comparing (3.7) and (3.8) shows us how to utilize the Legendre penalty method (3.8) at the Chebyshev points.

The Chebyshev-Legendre (C-L) Method

Let $P_N(x)$ be the Legendre polynomial of degree $N$. In the C-L method we seek a polynomial of degree $N$ in $x$ that satisfies

$$\frac{\partial u_N(x_j, t)}{\partial t} = \frac{\partial u_N(x, t)}{\partial x} \bigg|_{x=x_j} - \tau \frac{(1 + x_j)P_N'(x_j)}{2P_N'(1)} (u_N(1, t) - g(t))$$

(3.9)

for $j = 0, \ldots, N$.

Note that the penalty term $\frac{(1 + x_j)P_N'(x_j)}{2P_N'(1)}$ is different from zero for all the Chebyshev grid points $x_j$. Note also that applying (3.9) entails the use of the differentiation matrix $D_C$ at the Chebyshev points. In fact, given $u_N(x_j, t)$ one finds the derivative based on the Chebyshev points, and then add the penalty term with different weights at every grid point. The term $P_N'(x_j)$ is evaluated using the explicit formula (2.8). This is done once and for all for any grid size $N$. 

6
The surprising fact is that the C-L method, though computed at the Chebyshev points, is stable in the usual $L_2$ norm, rather than the weighted $L_2$ norm. In fact one can state

**Theorem 3.1.1:** *The $L_2$ Stability of the C-L Method.*

Let $u_N(x,t)$ be the solution of (3.9). Let $\omega_j$ be the weights of the Gauss Lobatto Legendre quadrature formula and $y_j$ the nodes of the same quadrature formula. Let $g(t) = 0$ in (3.3) and (3.9), then for

$$\tau \geq \frac{1}{2\omega_0} = N(N + 1)$$

the C-L method is stable in the $L_2$ norm. More specifically,

$$\sum_{j=0}^{N} u_N^2(y_j, t) \omega_j = \sum_{j=0}^{N} u_N^2(y_j, 0) \omega_j - \int_0^t [u_N^2(1, t)(2\omega_0 \tau - 1) + u_N^2(-1, t)] dt.$$  \hspace{1cm} (3.10)

**Proof:**

It follows from (3.9) that

$$\frac{\partial u_N(x,t)}{\partial t} = \frac{\partial u_N(x,t)}{\partial x} - \tau \frac{(1 + x)P_N(x)}{2P'_N(1)}(u_N(1,t) - g(t)).$$  \hspace{1cm} (3.11)

This is because both sides of (3.9) are polynomials of degree $N$ that agree at $N + 1$ points, namely at the Chebyshev collocation points $x_j, 0 \leq j \leq N$.

We now read (3.11) at the Legendre points $y_j$ to get

$$\frac{d}{dt} \sum_{j=0}^{N} u_N^2(y_j, t) \omega_j = 2 \sum_{j=0}^{N} u_N(y_j, t) \frac{\partial u_N(y_j, t)}{\partial x} \omega_j - 2\tau \omega_0 u_N^2(1,t)$$

Since the Gauss-Lobatto-Legendre quadrature formula is exact for polynomials of degree $2N - 1$ it follows that

$$2 \sum_{j=0}^{N} u_N(y_j, t) \frac{\partial u_N(y_j, t)}{\partial x} \omega_j = \int_{-1}^{1} (u_N^2(\xi, t))d\xi$$

$$= u_N^2(1,t) - u_N^2(-1,t)$$

and thus the stability estimate (3.10) follows. The Theorem is thus proven.

Note that unlike the Legendre-Penalty method (3.8), in which one needs to use the Legendre points $y_j$ in the computations, these points do not appear in the computations in
the C-L method. They are just introduced for the sake of the proof. The actual computations are done using the Chebyshev grid points $x_j$.

An energy estimate based on the Chebyshev points $x_j$ can be derived by using (2.11) and (2.12) as follows:

**Theorem 3.1.2:**

Let

$$||u_N(\cdot, t)||^2 = \sum_{j=0}^{N} \sum_{l=0}^{N} H_{j,l} u_N(x_l, t) u_N(x_j, t)$$

(3.12)

with

$$H_{j,l} = \sum_{k=0}^{N} g_j(y_k) g_l(y_k) \omega_k .$$

where the Chebyshev-Lagrange polynomials $g_j(x)$ are defined in (2.10).

Then $u_N(\cdot, t)$ satisfies the energy estimate

$$||u_N(\cdot, t)||^2 = ||u_N(\cdot, 0)||^2 - \int_0^t \{u_N^2(1, t)(2\omega_0 \tau - 1) + u_N^2(-1, t)\} dt$$

(3.13)

**Proof:**

Equation (3.13) is really a restatement of (3.10). Since $u_N(x, t)$ is a polynomial of degree $N$ in $x$, it can be represented exactly by

$$u_N(x, t) = I_C u_N = \sum_{l=0}^{N} u_N(x_l, t) g_l(x) .$$

Thus

$$u_N(y_j, t) = \sum_{l=0}^{N} u_N(x_l, t) g_l(y_j) .$$

(3.14)

The estimate (3.13) follows from (3.10) upon substituting (3.14) for the values of $u_N(y_j, t)$. The theorem is proven.

The C-L method can be viewed from many different points of view. Theorem (3.1.1) shows that this method, in the constant coefficient case, is equivalent to the Legendre-penalty method introduced in [5].

Thus the **C-L method is the realization of the Legendre method at the Chebyshev points.**

We close this section by pointing out that the estimate (3.13) enables one to pass to systems easily. Actually it had been done in [5]. One has to follow the same steps. It shows the C-L method is stable for constant coefficients systems of hyperbolic equations.
3.2 The Differentiation Matrices

Perhaps more insight can be gained if one compares the differentiation matrix induced by the C-L method \(D_{CL}\) with the differentiation matrices induced by the Chebyshev penalty method (3.7) \(D_C\), and the Legendre penalty method (3.8) \(D_L\).

We start by noting that the differentiation matrices \(D_L\) and \(D_C\) defined in (2.13) and (2.14) do not take into account any boundary conditions. The differentiation matrices induced by (3.7), (3.8) and (3.9) are variations of the basic matrices \(D_L\) and \(D_C\), differing only in the method of imposing boundary condition.

Not surprisingly, \(D_L\) and \(D_C\) are similar, after all both differentiate \textit{exactly} polynomials of degree \(N\). Since for these polynomials, the operators \(I_L\) and \(I_C\) are the same, the matrices \(D_L\) and \(D_C\) represent the same operation in a different basis. This implies a similarity relationship. More specifically, this relationship can be written explicitly:

\textbf{Theorem 3.2.1}:

Let \(S\) be the matrix whose elements \(S_{j,k}\) are given by

\[ S_{j,k} = h_j(x_k) \quad (3.15) \]

where \(h_j(x)\) are the Legendre-Lagrange polynomials defined in (2.9). Again \(x_j\) are the Chebyshev points.

Let \(T\) be the matrix whose elements \(T_{j,k}\) are given by

\[ T_{j,k} = g_j(y_k) \quad (3.16) \]

where \(g_j(x)\) the Chebyshev-Lagrange polynomials are defined in (2.10). \(y_k\) are the Legendre collocation points.

Then

\[ S = T^{-1} \quad (3.17) \]

and

\[ D_C = S D_L T \quad (3.18) \]

\textbf{Proof}:

Since \(g_j(x)\) is a polynomial of degree \(N\) it is given by

\[ g_j(x) = \sum_{l=0}^{N} g_j(y_l) h_l(x). \quad (3.19) \]
Substituting \( x_k \) and making use of the fact that \( g_j(x_k) = \delta_{j,k} \), we get
\[
\delta_{j,k} = \sum_{l=0}^{N} g_j(y_l) h_l(x_k),
\]
proving that
\[
I = ST.
\]

Differentiating (3.19) we get
\[
g'_j(x) = \sum_{l=0}^{N} g_j(y_l) h'_l(x)
\]

However \( h'_l(x) \) is itself a polynomial of degree \( N \) and therefore it can be expressed as
\[
h'_l(x) = \sum_{m=0}^{N} h'_l(y_m) h_m(x),
\]
which leads to
\[
g'_j(x_k) = \sum_{l=0}^{N} \sum_{m=0}^{N} g_j(y_l) h'_l(y_m) h_m(x_k)
\]
The theorem is thus proved.

We will show now that the differentiation matrix induced by the C-L method \( D_{CL} \) is similar to the differentiation matrix \( D_{CP} \) induced by the Legendre penalty method (3.8). This will demonstrate the fact that the C-L method is the realization of the Legendre method on the Chebyshev grid.

**Theorem 2.2.2:**

Let \( D_{CL} \) be the differentiation matrix induced by the Chebyshev Legendre method (3.9) and \( D_{LP} \) the differentiation method induced by the Legendre penalty method (3.8) then
\[
D_{CL} = SD_{LP}T
\]
where the matrices \( S, T \) defined in (3.15), (3.16) are the transformation matrices between the Chebyshev points and the Legendre points.

**Proof:**

Note that the differentiation matrix \( D_{LP} \) is essentially the matrix \( D_L \) introduced in (2.13), modified to take into account the boundary conditions, imposed via penalty in (3.8). Thus
\[
(D_{LP})_{j,k} = (D_L)_{j,k} - \tau \delta_{0,k} \delta_{j,0}.
\]
In the same manner we can write explicitly

\[(D_{CL})_{j,k} = (D_C)_{j,k} - \tau \frac{(1 + x_k)P'_N(x_k)}{2P'_N(1)} \delta_{j,0} \tag{3.22}\]

Equation (3.22) is a direct consequence of (3.9). Note that the full first column of \(D_C\) is modified, and not only the first element as in (3.21).

We proceed by writing explicitly the elements of the matrix \(S D_{LP} T\). In fact

\[(S D_{LP} T)_{j,k} = (S D_L T)_{j,k} - \tau \sum_{l=0}^{N} \sum_{m=0}^{N} g_j(g_l) \delta_{l,0} \delta_{m,0} h_m(x_k) \cdot \]

Thus using (3.18) we get

\[(S D_{LP} T)_{j,k} = (D_C)_{j,k} - \tau g_j(y_0) h_0(x_k) \tag{3.23}\]

From (2.9)

\[h_0(x_k) = \frac{(1 + x_k)P'_N(x_k)}{2P'_N(1)}\]

and since \(y_0 = x_0 = 1\) \(g_j(y_0) = \delta_{j,0}\) so the right hand side of (3.23) is exactly the same as this of (3.22).

Thus (3.20) is established. The proof is completed.

4 Parabolic Equations

In this Section we present the Chebyshev-Legendre method for the parabolic equation

\[\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad -1 \leq x \leq 1, t > 0 \tag{4.1}\]

with Robin boundary condition

\[\alpha u(1,t) + \beta u_x(1,t) = g^+(t) \]
\[\gamma u(-1,t) + \delta u_x(-1,t) = g^-(t) \tag{4.2}\]

We will assume that \(\alpha, \beta, \gamma\) are non-negative and \(\delta\) is non-positive. This assures the time decay (or non-growth) of \(u(x,t)\).

We note that by now there is a very limited stability theory for the Chebyshev method. In fact stability had been proved first for the Dirichlet case \(\beta = 0, \delta = 0\), (see [7]) and then for Neumann case \(\alpha = \gamma = 0\). [9]. Here we present the C-L method and prove stability for the approximation to (4.1), (4.2) for the general Robin case.
Denote by $\mathcal{P}_N$ the finite dimensional space of polynomial of degree at most $N$. We define the operator $\mathcal{A}$

$$\mathcal{A} : \mathcal{P}_N \rightarrow \mathcal{P}_N$$

by

$$\mathcal{A}v(x,t) = -\frac{\partial^2 v(x,t)}{\partial x^2} + R(x,t) \quad (4.3)$$

where

$$R(x,t) = \tau_0 Q^+(x)[B^+(t) - g^+(t)] + \tau_N Q^-(x)[B^-(t) - g^-(t)] \quad (4.4)$$

with

$$Q^+(x) = \frac{(1 + x)P'_N(x)}{2P_N(1)} \quad , \quad B^+(t) = \alpha v(1,t) + \beta v_x(1,t)$$
$$Q^-(x) = \frac{(1 - x)P'_N(x)}{2P_N(1)} \quad , \quad B^-(t) = \gamma v(-1,t) + \delta v_x(-1,t).$$

The numbers $\tau_0, \tau_N$ will be determined later to assure stability. We define also the following scalar product

$$(v, w)_N = \sum_{j=0}^{N} v(y_j)w(y_j)\omega_j \quad (4.5)$$

where $y_j, \omega_j$ are the Legendre points and weights respectively.

**The Chebyshev-Legendre Method for Parabolic Equations**

We seek the polynomial of degree $N$ in $x$, $v(x,t)$ that satisfies

$$\frac{\partial v(x_j,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} \bigg|_{x=x_j} - R(x_j,t) \quad 0 \leq j \leq N \quad (4.6)$$

where $x_j$ are Chebyshev collocation points.

Note that again, the work is done on the Chebyshev points $x_j$, the penalty values $Q^+(x_j), Q^-(x_j)$ are computed by (2.8) and are nonzero for any $x_j$.

To prove the stability of (4.6), we set $g^+(t) = g^-(t) = 0$. In the following lemma, we will find conditions on $\tau_0$ and $\tau_N$ such that the operator $\mathcal{A}$ to be semi bounded.

**Lemma 4.1:**
Let \( v \in \mathcal{P}_N \) and
\[
\tau^+_{\alpha, \beta} = \frac{1}{\omega_0 b} [(1 + 2\kappa) + 2\sqrt{\kappa + \kappa^2}]
\]
and
\[
\tau^-_{\alpha, \beta} = \frac{1}{\omega_0 b} [(1 + 2\kappa) - 2\sqrt{\kappa + \kappa^2}]
\]
with \( \kappa = \omega_0 a/b \).

Let the operator \( \mathcal{A} \) be defined in (4.3). Then
\[
(\mathcal{A}v, v)_N \geq \sum_{j=1}^{N-1} v_x^2(y_j)\omega_j
\]
provided
\[
\tau^-_{\alpha, \beta} \leq \tau_0 \leq \tau^+_{\alpha, \beta}
\]
and
\[
\tau^-_{\gamma, |\delta|} \leq \tau_N \leq \tau^+_{\gamma, |\delta|}.
\]

**Proof:**

Since the Gauss Lobatto Quadrature formula (2.6) is exact for polynomials of degree \( 2N - 1 \) and since \( v(x, t) \) is a polynomial of degree \( N \), we have
\[
\sum_{j=0}^{N} v(y_j)v_{xx}(y_j)\omega_j = \int_{-1}^{1} v(x)v_{xx}(x)dx
\]
\[
= v(1)v_x(1) - v(-1)v_x(-1) - \int_{-1}^{1} v_x(x)v_x(x)dx
\]
using the standard integration by parts technique.

Using again the Gauss Lobatto formula, one would get
\[
- \sum_{j=0}^{N} v(y_j)v_{xx}(y_j)\omega_j = \sum_{j=0}^{N} v_x^2(y_j)\omega_j - v(1)v_x(1) + v(-1)v_x(-1)
\]
\[
= \sum_{j=1}^{N-1} v_x^2(y_j)\omega_j + v_x^2(1)\omega_0 + v_x^2(-1)\omega_N - v(1)v_x(1) + v(-1)v_x(-1)
\]
(4.10)

Thus making use of (4.10) we can write
\[
(\mathcal{A}v, v)_N = F(1, \alpha, \beta, 0) + F(-1, \gamma, |\delta|, N) + \sum_{j=1}^{N-1} v_x^2(y_j)\omega_j
\]
(4.11)
where

\[ F(x,a,b,k) = x(\tau_k b \omega_k - 1)v(x)v_x(x) + \tau_k a \omega_k v^2(x) + \omega_k v_x^2(x) \]  \hspace{1cm} (4.12)

In order for \( A \) to be positive we need to choose \( \tau_0 \) and \( \tau_N \) such that \( F(1,\alpha,\beta,0) \) and \( F(-1,\gamma,|\delta|,N) \) are non-negative. For \( F(1,\alpha,\beta,0) \) to be positive, we need

\[ (\tau_0 \beta \omega_0 - 1)^2 \leq 4\alpha \tau_0 \omega_0^2 \]

or

\[ \tau_0^2 \beta^2 \omega_0^2 - 2\tau_0 \omega_0 (\beta + 2\alpha \omega_0) + 1 \leq 0 \]

Thus \( \tau_0 \) has to lie between the roots of the parabola described in the left hand side, namely \( \tau_{a,\beta}^0 \) and \( \tau_{a,\beta}^+ \).

The same kind of consideration holds for \( \tau_N \). Thus \( F(1,\alpha,\beta,0) \) and \( F(-1,\gamma,|\delta|,N) \) are non-negative for the range of \( \tau_0 \) and \( \tau_N \) given in (4.8) and (4.9), respectively. (4.7) follows from (4.11).

**Remarks**

1. The Dirichlet boundary condition for \( x = 1 \) is obtained from (4.2) by setting \( \alpha = 1, \beta = 0 \). In this case

\[ \tau_{1,0}^+ = \infty \]
\[ \tau_{1,0}^- = \frac{1}{4\omega_0^2} \]

which yields the condition for the penalty amplitude

\[ \tau_0 \geq \frac{1}{16} N^4 (N + 1)^4 \]

2. The Neumann boundary condition for \( x = 1 \) corresponds to the case \( \alpha = 0, \beta = 1 \). In this case \( \tau_{0,1}^+ = \tau_{0,1}^- \) yielding the condition

\[ \tau_0 = \frac{1}{\omega_0} = \frac{N(N + 1)}{2} \]

We are now ready to state the stability theorem for the C-L method when applied to parabolic equations with Robin boundary conditions:

**Theorem 4.1:**
Let $\tau_0$ and $\tau_N$ satisfy (4.8) and (4.9) respectively. Let $v(x, t) \in \mathcal{P}_N$ be the C-L approximation to $u(x, t)$, obtained by (4.6). Assuming that $g^+(t) = g^-(t) = 0$, $v(x, t)$ satisfies the energy estimate

\[(v(x, t), v(x, t))_N \leq (v(x, 0), v(x, 0))_N - 2 \int_0^T \sum_{j=1}^{N-1} v_x^2(y_j, t)dt \tag{4.13}\]

where the scalar product $(f, g)_N$ is defined in (3.5).

**Proof:**

Since (4.6) holds for $j = 0, \ldots, N$ and since $v, v_x$ and $R$ are polynomials of degree at most $N$, we conclude that both sides of (4.6) are equal not only at the grid points but also for every $x$.

\[\frac{\partial v(x, t)}{\partial t} = \frac{\partial^2 v(x, t)}{\partial x^2} - R(x, t) \quad -1 \leq x \leq 1\]

where $R(x, t)$ is defined in (4.4).

Noting the definition of $A$ in (4.3), we get

\[\frac{\partial v}{\partial t} = Av.\]

Thus

\[(v, \frac{\partial v}{\partial t})_N = (v, Av)_N\]

Using Lemma 4.1 yields

\[\frac{1}{2} \frac{d}{dt}(v, v)_N \leq - \sum_{j=1}^{N-1} v_x^2(y_j, t)\omega_j\]

and integration yields the stability result (4.13).

We stress again that the Legendre collocation points $y_j$ are "ghost points", which are never used in the computations but only in the proof of the stability. Actually we could restate the proof in terms of the Chebyshev collocation points $x_j$ as in Theorem 3.1.2.

5 Numerical Results

Case 1: Linear scalar PDE
In this Section, we will consider some numerical examples that verify our claims stated in previous Sections. Consider the scalar linear initial-boundary value hyperbolic PDE

\[ U_t = U_x \quad -1 \leq x \leq 1, t > 0 \]  

(5.1)

with initial condition

\[ U(x, 0) = \sin(2\pi k x) \]

and boundary condition at \( x = 1 \)

\[ U(1, t) = g(t) = \sin(2\pi k(1 + t)) \]

We seek an \( N \) degree \( x \)-polynomial \( v(x, t) \) that satisfies

\[ \frac{d}{dt} v(x_j, t) = Dv(x_j, t) - \tau Q(x_j)(v(1, t) - g(t)) \]  

(5.2)

at Chebyshev collocation point \( x_j = \cos(\pi j/N) \), \( j = 0, \ldots, N \) and \( D \) is the differentiation operator (matrix).

For different construction of the \( N \)th degree polynomial \( Q(x) \), one could have different type of boundary treatments. For examples,

1. if \( Q(x) = \frac{(1+x)P_N''(x)}{2P_N'(1)} \), \( D = D_C \) and \( x = x_j \) are the Gauss-Lobatto-Chebyshev points, then we have the Chebyshev-Legendre method (C-L).

2. if \( Q(x) = \frac{(1+x)T_N''(x)}{2T_N'(1)} \), \( D = D_C \) and \( x = x_j \) are the Gauss-Lobatto-Chebyshev points, then we have the Chebyshev penalty method (C-P).

3. if \( Q(x) = \frac{(1+x)P_N''(x)}{2P_N'(1)} \), \( D = D_L \) and \( x = y_j \) are the Gauss-Lobatto-Legendre points, then we have the Legendre penalty method (L-P).

Let denote \( v_j^{(n)} = v(x_j, t_n) \) and \( \Delta t \) be the time step increment, then for \( j = 0, \ldots, N \), we would advance the system of ODE (5.2) in time by the third order Heun Runge Kutta scheme that has the following form:

For \( j = 0,1, \ldots, N \), and \( v_j^{(0)} = g(0) \)

\[
\begin{align*}
v_j^{(1)} &= v_j^{(n)} + \frac{\Delta t}{3} (Dv_j^{(n)} - \tau Q(x_j)(v_j^{(n)} - g(t_n))) \\
v_j^{(2)} &= v_j^{(n)} + \frac{2\Delta t}{3} (Dv_j^{(1)} - \tau Q(x_j)(v_j^{(1)} - g(t_n) - \frac{\Delta t}{3} g'(t_n))) \\
v_j^{(n+1)} &= \frac{1}{4} v_j^{(n)} + \frac{3}{4} v_j^{(1)} + \frac{3\Delta t}{4} (Dv_j^{(2)} - \tau Q(x_j)(v_j^{(2)} - g(t_n) - \frac{\Delta t}{3} g'(t_n) - \frac{2\Delta t^2}{9} g''(t_n)))
\end{align*}
\]  

(5.3)
where \( g'(t_n) \) and \( g''(t_n) \) are the derivatives of the time-dependent boundary conditions in time at \( t = t_n \).

It has been observed before that if one imposes boundary condition at each intermediate stages of the Runge Kutta scheme, a larger time-step (CFL number) can be used. Otherwise, CFL number has to be reduced by as much as four time for stability. In this study, we define \( \Delta t = CFL/N^2 \).

The traditional way of the imposing exact boundary condition at \( x = 1 \) can be described as following:

For \( j = 0, \ldots, N \), and \( v_j^{(0)} = u(x,0) \)

\[
\begin{align*}
 v_j^{(1)} &= v_j^{(n)} + \frac{\Delta t}{3} Dv_j^{(n)} \\
 v_0^{(1)} &= g(t_n + \frac{\Delta t}{3}) \\
 v_j^{(2)} &= v_j^{(n)} + \frac{2\Delta t}{3} Dv_j^{(1)} \\
 v_0^{(2)} &= g(t_n + \frac{2\Delta t}{3}) \\
 v_j^{(n+1)} &= \frac{1}{4} v_j^{(n)} + \frac{3}{4} v_j^{(1)} + \frac{3\Delta t}{4} Dv_j^{(2)} \\
 v_0^{(n+1)} &= g(t_n + \Delta t)
\end{align*}
\]

(5.4)

However, as shown in Table I that this procedure would lead to reduction of accuracy in time as \( N \) increases.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( L_2 ) Error</th>
<th>Rate</th>
<th>( L_2 ) Error</th>
<th>Rate</th>
<th>( L_2 ) Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.82E-03</td>
<td></td>
<td>0.10E-03</td>
<td></td>
<td>0.29E-05</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0.15E-04</td>
<td>2.89</td>
<td>0.18E-05</td>
<td>2.91</td>
<td>0.28E-07</td>
<td>3.35</td>
</tr>
<tr>
<td>64</td>
<td>0.42E-06</td>
<td>2.57</td>
<td>0.49E-07</td>
<td>2.61</td>
<td>0.72E-09</td>
<td>2.64</td>
</tr>
<tr>
<td>128</td>
<td>0.17E-07</td>
<td>2.31</td>
<td>0.19E-08</td>
<td>2.33</td>
<td>0.28E-10</td>
<td>2.34</td>
</tr>
<tr>
<td>CFL</td>
<td>8</td>
<td></td>
<td>4</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Hence, the above procedure is modified as following (Detailed discussion and analysis will appear in a future paper):

For \( j = 0,1, \ldots, N \), and \( v_j^{(0)} = u(x,0) \)

\[
v_j^{(1)} = v_j^{(n)} + \frac{\Delta t}{3} Dv_j^{(n)}
\]
\[ v_0^{(1)} = g(t_n) + \frac{\Delta t}{3} g'(t_n) \]
\[ v_j^{(2)} = v_j^{(n)} + \frac{2\Delta t}{3} Dv_j^{(1)} \]
\[ v_0^{(2)} = g(t_n) + \frac{2\Delta t}{3} g'(t_n) + \frac{2\Delta t^2}{9} g''(t_n) \]
\[ v_j^{(n+1)} = \frac{1}{4} v_j^{(n)} + \frac{3}{4} v_j^{(1)} + \frac{3\Delta t}{4} Dv_j^{(2)} \]
\[ v_0^{(n+1)} = g(t_{n+1}) \]

(5.5)

We shall denote this procedure as (XBC). Table I indicated that the order of time accuracy for this procedure is third order for all \( N \).

**Table II**

\( L_2 \) Error and order of accuracy for (5.5) with \( k = 1 \)

<table>
<thead>
<tr>
<th>( N )</th>
<th>Error</th>
<th>Rate</th>
<th>Error</th>
<th>Rate</th>
<th>Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.77E-03</td>
<td>2.99</td>
<td>0.98E-04</td>
<td>3.00</td>
<td>0.28E-05</td>
<td>3.00</td>
</tr>
<tr>
<td>32</td>
<td>0.12E-04</td>
<td>3.00</td>
<td>0.15E-05</td>
<td>3.00</td>
<td>0.24E-07</td>
<td>3.44</td>
</tr>
<tr>
<td>64</td>
<td>0.19E-06</td>
<td>3.00</td>
<td>0.24E-07</td>
<td>3.00</td>
<td>0.37E-09</td>
<td>3.00</td>
</tr>
<tr>
<td>128</td>
<td>0.30E-08</td>
<td>3.00</td>
<td>0.37E-09</td>
<td>3.00</td>
<td>0.58E-11</td>
<td>2.99</td>
</tr>
<tr>
<td>CFL</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Next, using the C-L method, one get the \( L_2 \) error and the order of accuracy as listed in Table II.

**Table III**

\( L_2 \) Error and order of accuracy for C-L method with \( k = 1, \tau = 4\omega_0 \)

<table>
<thead>
<tr>
<th>( N )</th>
<th>Error</th>
<th>Rate</th>
<th>Error</th>
<th>Rate</th>
<th>Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.47E-03</td>
<td>2.99</td>
<td>0.60E-04</td>
<td>3.01</td>
<td>0.28E-05</td>
<td>3.00</td>
</tr>
<tr>
<td>32</td>
<td>0.74E-05</td>
<td>2.99</td>
<td>0.93E-06</td>
<td>3.00</td>
<td>0.15E-07</td>
<td>3.81</td>
</tr>
<tr>
<td>64</td>
<td>0.12E-06</td>
<td>3.00</td>
<td>0.15E-07</td>
<td>3.00</td>
<td>0.23E-09</td>
<td>3.00</td>
</tr>
<tr>
<td>128</td>
<td>0.18E-08</td>
<td>3.00</td>
<td>0.23E-09</td>
<td>3.00</td>
<td>0.36E-11</td>
<td>2.99</td>
</tr>
<tr>
<td>CFL</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table IV**

\( L_2 \) Error of C-L method for different choices of \( \tau = 2\omega_0\alpha \) with \( k = 1, CFL = 1 \)

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \alpha = 8 )</th>
<th>( \alpha = 2 )</th>
<th>( \alpha = 1 )</th>
<th>( \alpha = 0.9 )</th>
<th>( \alpha = 0.5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.31E-05</td>
<td>0.28E-05</td>
<td>0.65E-05</td>
<td>0.83E-05</td>
<td>0.32E-03</td>
</tr>
<tr>
<td>32</td>
<td>0.15E-07</td>
<td>0.15E-07</td>
<td>0.15E-07</td>
<td>0.15E-07</td>
<td>0.18E-01</td>
</tr>
<tr>
<td>64</td>
<td>0.23E-09</td>
<td>0.23E-09</td>
<td>0.23E-09</td>
<td>0.23E-06</td>
<td>unstable</td>
</tr>
<tr>
<td>128</td>
<td>0.36E-11</td>
<td>0.36E-11</td>
<td>0.36E-11</td>
<td>unstable</td>
<td>unstable</td>
</tr>
</tbody>
</table>
From Table IV, we can see that for \( \tau < 2\omega_0 \), C-L becomes unstable while for \( \tau \geq 2\omega_0 \), the convergent of the scheme confirms the theoretical prediction.

**Case 2: Nonlinear scalar PDE**

Consider the scalar nonlinear initial boundary value hyperbolic equation

\[
U_t = U_x^2 - 2\pi k \cos(2\pi k(x + t))(1 + \sin(2\pi k(x + t)))
\]

\[-1 \leq x \leq 1, \quad t > 0\]

with initial condition

\[U(x, 0) = 2 + \sin(2\pi kx)\]

and boundary condition at \( x = 1 \)

\[U(1, t) = g(t) = 2 + \sin(2\pi k(1 + t)).\]

This PDE has an exact solution given as \( U(x, t) = 2 + \sin(2\pi k(x + t)) \).

**Table V**

\(L_2\) Error of C-L method for different choices of \( \tau = 2\omega_0\alpha \) with \( k = 1, \, \text{cfl} = 1\)

<table>
<thead>
<tr>
<th>(N)</th>
<th>(\alpha = 8)</th>
<th>(\alpha = 4)</th>
<th>(\alpha = 3)</th>
<th>(\alpha = 2.5)</th>
<th>Exact BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.86E-02</td>
<td>0.10E-01</td>
<td>0.18E-01</td>
<td>0.27E-01</td>
<td>0.72E-02</td>
</tr>
<tr>
<td>32</td>
<td>0.40E-07</td>
<td>0.40E-07</td>
<td>0.40E-07</td>
<td>0.11E-04</td>
<td>0.39E-07</td>
</tr>
<tr>
<td>64</td>
<td>0.68E-09</td>
<td>0.68E-09</td>
<td>0.68E-09</td>
<td>unstable</td>
<td>0.67E-09</td>
</tr>
<tr>
<td>128</td>
<td>0.11E-10</td>
<td>0.11E-10</td>
<td>0.11E-10</td>
<td>unstable</td>
<td>0.11E-10</td>
</tr>
</tbody>
</table>

Different values of \( k \) are also tested, similar results are obtained.

**References**


THE CHEBYSHEV-LEGENDRE METHOD: IMPLEMENTING LEGENDRE METHODS ON CHEBYSHEV POINTS

Wai Sun Don
David Gottlieb

Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23681-0001

National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

Langley Technical Monitor: Michael F. Card
Final Report Submitted to SIAM Journal on Numerical Analysis

Unclassified - Unlimited
Subject Category 64

We present a new collocation method for the numerical solution of partial differential equations. This method uses the Chebyshev collocation points, but because of the way the boundary conditions are implemented, has all the advantages of the Legendre methods. In particular $L_2$ estimates can be obtained easily for hyperbolic and parabolic problems.