CHAPTER 15

Penalty Method Utilized for Parabolic Problems

Kelly Black[1]

Abstract. We are investigating a method to solve parabolic equations using domain decomposition techniques. The spatial approximations are accomplished using a spectral collocation method. In this application both Legendre and Tschebycheff polynomials are employed and results from both are examined. The time discretization involves two methods. An implicit method is used inside each domain and an explicit method is used on the boundaries of each domain. We are investigating results from a backwards Euler's method on the domains and a Dufort-Frankel method on the boundaries. In order to calculate the boundary values the derivatives are estimated using spectral collocation matrices and a penalty is added for differences between first derivatives on adjacent boundaries.

1 Introduction. Multiple processor computers allow complicated problems to be distributed among many processors in order to increase the speed in which they can be solved. We are examining a method to solve a time-dependent problem using spectral methods and domain decomposition techniques. The goal is to take advantage of parallel architecture to solve large problems and retain the advantage of spectral methods. In this case the machine used is an Intel iPSC i860 with 32 processors. The methods used revolve around a Tschebycheff collocation on the Gauss-Labotto quadrature points on the domain [-1,1] [3].

The PDE solved in this case is a parabolic equation:

$$\begin{aligned} u_t &= u_{xx} - k^2 u \qquad (1)\\ x &\in [0,16] \\ u(x,0) &= \sin\left(\omega\frac{2\pi}{16}x\right) + \exp(kx) - \exp(-kx) \\ u(0,t) &= 0 \\ u(16,t) &= \exp(16k) - \exp(-16k) \end{aligned}$$

Equation (1) is approximated by using polynomial collocation in the spatial dimensions and the time discretization is accomplished by use of an explicit/implicit method. An explicit scheme is used in order to calculate the future time value on the boundaries of the subdomains which is used for the boundary conditions of an implicit scheme on the interior of the subdomains. Numerical results for both Tschebycheff and Legendre collocation are presented.

2 Polynomial Collocation. Because the quadrature points for Tschebycheff polynomials are known in closed form we will examine the collocation matrices generated by Tschebycheff polynomials. The Tschebycheff polynomials can be generated by finding the solutions of the following

[1]Ph.D. Candidate Div. of Applied Math, Brown University, Supported under NASA Grant NAG-1-703 and Air Force Office of Scientific Research Grant 90-0093

ordinary differential equation:

$$\begin{aligned}(\sqrt{1-x^2}T_n'(x))' + \frac{n^2}{\sqrt{1-x^2}}T_n(x) &= 0 \\ \Leftrightarrow (1-x^2)T_n''(x) - xT_n'(x) + n^2T_n(x) &= 0\end{aligned} \tag{2}$$

Solutions to equation (2) are of the form $T_n(x) = \cos(n \arccos x)$. Some properties of the Tschebycheff polynomials are given below [6]:

- $T_0 = 1$ and $T_1 = x$
- $T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x) \Rightarrow T_i(x)$ is a polynomial of degree i.
- $T_{2n}(x)$ is an even polynomial and $T_{2n+1}(x)$ is an odd polynomial.
- The $T_n(x)$'s are orthogonal on [-1,1] with respect to the measure $\frac{dx}{\sqrt{1-x^2}}$.

The last property is shown explicitly:

$$\int_{-1}^{1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}dx = \frac{\pi}{2}\delta_{n,m}c_n \qquad c_0 = 2, c_k = 1$$

This property is used in finding the Tschebycheff coefficients in the expansion of a function, f:

$$P_N f = \sum_{r=0}^{N} \hat{f}_r T_r(x) \tag{3}$$

$$\hat{f}_r = \frac{2}{\pi c_r}\int_{-1}^{1} \frac{f(x)T_r(x)}{\sqrt{1-x^2}}dx \tag{4}$$

$$c_0 = c_N = 2, c_k = 1 \qquad 1 < k < N$$

The next step is to calculate the integral in equation (4). Fortunately the Gauss-Labotto quadrature points can easily be found. The quadrature points are given by $x_j = \cos\frac{\pi j}{N}$. The Tschebycheff coefficients can be easily approximated [1, p. 194]:

$$\begin{aligned}\hat{f}_r &= \frac{2}{c_r N}\sum_{j=0}^{N}\frac{1}{c_j}f(\cos\frac{\pi j}{N})T_r(x_j) \\ &= \frac{2}{c_r N}\sum_{j=0}^{N}\frac{1}{c_j}f(\cos\frac{\pi j}{N})\cos\frac{\pi r j}{N}\end{aligned} \tag{5}$$

Substituting the integral approximation (5) into the projection (3) yields the following approximation:

$$\begin{aligned}P_N f &= \sum_{k=0}^{N}\frac{2T_k(x)}{c_k N}\left(\sum_{j=0}^{N} f(x_j)\frac{T_k(x_j)}{c_j}\right) \\ &= \sum_{j=0}^{N} f(x_j)\left(\frac{2}{c_j N}\sum_{k=0}^{N}\frac{T_k(x)T_k(x_j)}{c_k}\right)\end{aligned}$$

The projection operator is simplified by defining an interpolating polynomial:

$$g_j(x) = \frac{2}{c_j N}\sum_{k=0}^{N}\frac{T_k(x)T_k(x_j)}{c_k}$$

As defined $g_j(x)$ has some important properties:

- $g_j(x)$ is the Lagrange polynomial interpolating the points $\delta_{kj} \qquad 0 \leq k \leq N$
- $g_j(x) = \frac{(-1)^j}{c_j N^2} \frac{(1-x^2)T_N'(x)}{x-x_j}$

The advantage of working with the projection matrix is the ease of finding derivatives:

$$\begin{aligned} P_N u(x,t) = u_N(x,t) &= \sum_{j=0}^{N} u_N(x_j,t) g_j(x) \\ \Rightarrow \frac{du_N}{dx}(x_k,t) &= \sum_{j=0}^{N} u_N(x_j,t) g_j'(x_k) \end{aligned}$$

The derivative of $g_j(x)$ is calculated by taking advantage of the ODE in equation (2):

$$\begin{aligned} g_j'(x) &= \frac{(-1)^{j+1}}{N^2 c_j} \left(\frac{(1-x^2)T_N''(x) - 2xT_N'(x)}{x-x_j} - \frac{(1-x^2)T_N'(x)}{(x-x_j)^2} \right) \\ &= \left(\frac{-N^2 T_N(x) - xT_N'(x)}{x-x_j} - \frac{(1-x^2)T_N'(x)}{(x-x_j)^2} \right) \end{aligned}$$

The second derivative matrix is found by taking another derivative of $g_j'(x)$.

The collocation matrices for Legendre polynomials are found using the same methods. The primary drawback is that a closed form solution for the weights and Gauss-Labotto points is not known and they have to be calculated.

3 Multi-Domain Techniques and the Penalty Method. Spectral Methods can produce accurate results even on a relatively coarse grid. There are some drawbacks, however. The matrices generated are full and can be difficult to invert for fine grids. Spectral methods can also be difficult to apply to irregular geometries; they tend to work best on simple domains such as squares or circles. One way to overcome these disadvantages is to divide the original domain into many subdomains.

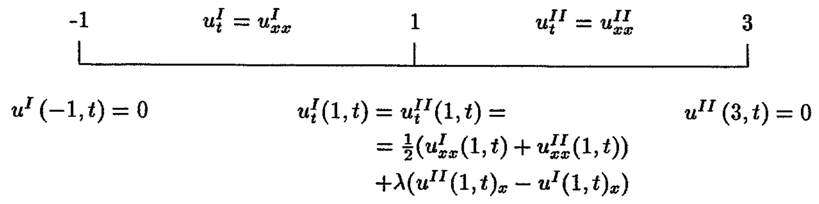
Once the problem is broken up into smaller pieces each smaller problem is easier to solve and they can be done in parallel. These smaller subdomains can also be arranged to fit more complex geometries and each subdomains can be calculated using a course grid. The primary problem with this approach is how to handle the boundaries between the subdomains.

Boundary interface conditions can be handled by examining the continuity conditions at the boundaries. The first thing that is required is piecewise continuity. The approximation has to be continuous at the least. The condition can be taken further by forcing continuity of the first derivative, which can be enforced using an influence matrix technique. If a distributed memory parallel computer is used this technique has a major drawback: the extra communication between processors can become prohibitive. The rate of communication overhead can grow larger as the number of domains increases.

When working on a distributed memory machine the problem of isolating the domains and minimizing communication between processors is of paramount importance. The method we will examine is a relaxation of the constraints on the first derivative. Ideally continuity of the first derivative is a good thing, but the price paid can become quite large. One way around this is the penalty method of D. Funaro. The idea behind the penalty method is not to force continuity of the first derivative but to keep the first derivatives between adjacent subdomains close. This is achieved by adding a penalty in proportion to the difference between first derivatives on the interface [2].

An example of the penalty method is given for the heat equation with one spatial dimension. The example uses two subdomains but the results can be generalized for more subdomains and a theoretical value for the penalty is calculated. The examples employ Legendre collocation.

Figure 1: The domain $[-1,3]$ divided into two subdomains


-1
$u_t^I = u_{xx}^I$
1
$u_t^{II} = u_{xx}^{II}$
3
$u^I(-1,t) = 0$
$u_t^I(1,t) = u_t^{II}(1,t) =$
$= \frac{1}{2}(u_{xx}^I(1,t) + u_{xx}^{II}(1,t))$
$+\lambda(u^{II}(1,t)_x - u^I(1,t)_x)$
$u^{II}(3,t) = 0$

A formulation of the penalty method is given in figure (1). The domain is the interval $[-1,3]$. This domain is chosen because it is convienently divided into two domains that are easily mapped to the unit circle, $[-1,1]$. Families of orthogonal polynomials are generally examined on the unit circle.

Lemma 3.1 *The penalty method on two domains is stable for the heat equation on a Legendre quadrature.*

First, the Legendre quadrature is defined on the points x_j with weights β_j. The function values $u_N(x_j,t)$ are given as u_j. Under Gaussian quadrature $\beta_0 = \beta_N$. Also note that $u^I(1,t) = u^{II}(1,t)$ as defined in figure (1).

The stability equations for the first domain:

$$\begin{aligned}
\frac{1}{2}\partial_t \int_{-1}^{1} \left(u^I\right)^2 dx &= \int_{-1}^{1} u^I u_t^I dx \\
&= \sum_{i=0}^{N} u_i^I \left(u_i^I\right)_t \beta_i \\
&= \sum_{i=1}^{N} u_i^I \left(u_i^I\right)_{xx} \beta_i \\
&\quad +\beta_0 \frac{1}{2}\left(\left(u_0^I\right)_{xx} + \left(u_N^{II}\right)_{xx}\right) u_0^I \\
&\quad +\beta_0 \lambda \left(\left(u_N^{II}\right)_x - \left(u_0^I\right)_x\right) u_0^I
\end{aligned}$$

Next, the stability equations for the second domain:

$$\begin{aligned}
\frac{1}{2}\partial_t \int_{1}^{3} \left(u^{II}\right)^2 dx &= \int_{1}^{3} u^{II} u_t^{II} dx \\
&= \sum_{i=0}^{N} u_i^{II} \left(u_i^{II}\right)_t \beta_i \\
&= \sum_{i=0}^{N-1} u_i^{II} \left(u_i^{II}\right)_{xx} \beta_i \\
&\quad +\beta_N \frac{1}{2}\left(\left(u_N^{II}\right)_{xx} + \left(u_0^I\right)_{xx}\right) u_N^{II} \\
&\quad +\beta_N \lambda \left(\left(u_N^{II}\right)_x - \left(u_0^I\right)_x\right) u_N^{II}
\end{aligned}$$

Added together, the stability equations yield the following norm:

$$\begin{aligned}
\partial_t \left(\|\cdot\|^2 + \|\cdot\|^2 \right) &= \sum_{i=1}^{N} u_i^I \left(u_i^I\right)_{xx} \beta_i + \sum_{i=0}^{N-1} u_i^{II} \left(u_i^{II}\right)_{xx} \beta_i \\
&\quad + \frac{1}{2} \left(\beta_0 + \beta_N\right) \left(u_0^I\right)_{xx} u_0^I + \frac{1}{2} \left(\beta_0 + \beta_N\right) \left(u_N^{II}\right)_{xx} u_N^{II} \\
&\quad - \lambda \left(\beta_0 + \beta_N\right) \left(u_0^I\right)_x u_0^I + \lambda \left(\beta_0 + \beta_N\right) \left(u_N^{II}\right)_x u_N^{II} \\
&= \sum_{i=0}^{N} u_i^I \left(u_i^I\right)_{xx} \beta_i + \sum_{i=0}^{N} u_i^{II} \left(u_i^{II}\right)_{xx} \beta_i \\
&\quad - 2\lambda\beta_0 \left(u_0^I\right)_x u_0^I + 2\lambda\beta_0 \left(u_N^{II}\right)_x u_N^{II} \\
&= \int_{-1}^{1} u^I u_{xx}^I dx + \int_{1}^{3} u^{II} u_{xx}^{II} dx \\
&\quad - 2\lambda\beta_0 \left(u_0^I\right)_x u_0^I + 2\lambda\beta_0 \left(u_N^{II}\right)_x u_N^{II} \\
&= u_x^I u^I |_{-1}^{1} - \int_{-1}^{1} \left(u_x^I\right)^2 dx \\
&\quad + u_x^{II} u^{II} |_{1}^{3} - \int_{1}^{3} \left(u_x^{II}\right)^2 dx \\
&\quad - 2\lambda\beta_0 \left(u_0^I\right)_x u_0^I + 2\lambda\beta_0 \left(u_N^{II}\right)_x u_N^{II} \\
&\leq \left(u_0^I\right)_x u_0^I - \left(u_N^{II}\right)_x u_n^{II} \\
&\quad - 2\lambda\beta_0 \left(u_0^I\right)_x u_0^I + 2\lambda\beta_0 \left(u_N^{II}\right)_x u_N^{II}
\end{aligned}$$

A sufficient requirement for stability is $\lambda = \frac{1}{2\beta_0}$ □

The formulation presented can also be put into a system. First, map the second domain onto the first domain via $x \rightarrow 2 - x$ and define the vector $\vec{u}_N$:

$$\vec{u}_N = \begin{pmatrix} u_N^I(x) \\ u_N^{II}(2-x) \end{pmatrix}$$

The resulting equations can be used to generalize the lemma to more domains.

4 Time Discretization. A good way to ensure stability is to employ an implicit time discretization. The difficulty with this approach is how to handle the boundary conditions. We are examining an explicit/implicit approach. The boundaries of the subdomains are calculated using an explicit method. Once a future time value on the boundary is calculated an implicit method is employed on the interior of the subdomains. Here we are examining a Dufort-Frankel method to make the explicit calculation and a backwards Euler's method to make the implicit calculation.

The interior of each subdomain is calculated using a backwards Euler's method:

$$\frac{u_N^{n+1} - u_N^n}{\Delta t} = D_N^2 u_N^{n+1} - k^2 u_N^{n+1}$$

u_N^n is the approximation of u at the n^{th} time step and D_N^2 is the spectral collocation matrix for the second derivative.

The Tschebycheff collocation matrix D_N^2 has the following properties[5]:

- D_N^2 is stable: $\|\exp(D_N^2 t)\| \leq 1$
- Eigenvalues of D_N^2 are real, distinct, and negative.
- There is a family of NxN matrices R_N such that
  $R_N^{-1} D_N^2 R_N = \Lambda$ $\quad$ Λ diagonal
  and $\|R_N\|, \|R_N^{-1}\| \leq C$ $\quad$ C independent of N

(Statements and proofs of these properties are found in [5])

The boundary values for the implicit method are calculated by the Dufort-Frankel method [4]:

$$\frac{u_N^{n+1}-u_N^{n-1}}{2\Delta t} = (D_N^2 u_N^n - k^2 u_N^n) - \frac{\gamma}{(\Delta x)^2}(u_N^{n+1} - 2u_N^n + u_N^{n-1})$$

The Dufort-Frankel scheme is an unconditionally stable scheme under Tschebycheff collocation [5].

5 Results. Tests for this method have been made using both Tschebycheff and Legendre collocation. The methods were used on an iPSC i860 Hyper-Cube. The codes were run for different time steps and different size grids on the subdomains until they reached steady state. Comparisons are made for runs with the same size grids on each subdomains and for varying size grids per subdomain such that the total size grid remains constant. Each run varies the number of subdomains from one to thirty-two. There are three tables for each situation. The first table lists the the results from the penalty method, the second lists the results from using the true boundary values between subdomains while retaining the same communication, and the third lists the results with no communication for boundary points. Finally, the results obtained here use $\gamma = 0.2$ in the Dufort-Frankel method.

The first table in each list is the table for the runs using calculated subdomain boundaries. The boundaries are calculated using the Dufort-Frankel method and the interior is calculated using the backwards Euler's method. The entries in the table include the number of domains, the number of grid points used on each domain, and the total resulting number of grid points. The number of time steps needed until reaching the steady state are included. Also included is the error generated. The errors include the L_1 error which is calculated using the Clenshaw-Curtiss quadrature on the difference between the calculated and the true steady state solution. The L_∞ error is calculated by checking for the largest error on the grid. Finally, each table includes the time required. The time is calculated by the zero node on the machine by polling the system clock.

The second table in each list is the table for runs using the true solution on the subdomain boundaries. The communication is retained between adjacent processors but the true solution is used for interface conditions. This run is used to compare the number steps required with the first table, and it is used to compare the communication times with the third table. The third table in each list is the table for runs using the true solution on the subdomain boundaries. In this situation no communication is employed for transferring boundary conditions. The only communication is the cost of calculating the errors across the full domain which is done every ten time steps.

Table 1:

Tschebycheff: Calculated Boundary
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	16	16	250	9.64e-06	9.47e-07	0.17
2	16	32	40	9.12e-07	1.17e-07	0.16
4	16	64	650	9.93e-06	9.69e-07	0.67
8	16	128	140	9.64e-07	1.25e-07	0.18
16	16	256	210	6.17e-06	7.46e-07	0.25
32	16	512	260	4.18e-07	6.72e-08	0.33

5.1 Tschebycheff Results: Grid Size Constant on Each Subdomain. The runs in tables (1) through (3) display the time cost of increasing numbers of domains. In each run the size of the grid on each subdomain is kept constant.

Table 2:

Tschebycheff: True Boundaries with Communication
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	16	16	250	9.64e-06	9.47e-07	0.17
2	16	32	40	9.12e-07	1.17e-07	0.08
4	16	64	140	5.53e-06	5.43e-07	0.22
8	16	128	50	2.25e-06	3.12e-07	0.11
16	16	256	30	3.78e-06	3.98e-07	0.13
32	16	512	30	3.34e-06	3.32e-07	0.15

Table 3:

Tschebycheff: True Boundaries with no communication
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	16	16	250	9.64e-06	9.47e-07	0.17
2	16	32	40	9.12e-07	1.17e-07	0.09
4	16	64	140	5.53e-06	5.43e-07	0.14
8	16	128	50	2.25e-06	3.12e-07	0.11
16	16	256	30	3.78e-06	3.98e-07	0.15
32	16	512	30	3.34e-06	3.32e-07	0.14

Table (1) demonstrates that cost of thirty-two domains is roughly twice that of one one domain. The loss is balanced by the gain of a grid that is thirty-two times as dense. The table also shows that the time required increases nearly linearly with the number of iterations required. Unfortunately the number of steps increases with the number of domains. The situation is a problem of resolving the subdomain interfaces in order to reduce the extra number of time steps required.

Table (2) demonstrates how much is lost in calculating the boundaries. For thirty-two domains it takes more than eight times as many steps for the calculated boundaries.

Table (3) demonstrates that the communication between adjacent processors may not be significant. It appears that the large portion of communication is spent calculating the residual, the difference between u_N^{n+1} and u_N^n. This may not be a good indication due to the low number of iterations performed.

5.2 Tschebycheff Results: Total Grid Size Constant. The runs in tables (4) through (6) display the time costs of increasing numbers of domains but with finer grids on each domain. In each run the size of the total grid size remains the same except for the case of thirty-two domains where less than eight grid points would not retain the advantage of spectral approximation.

Comparison of table (4) and table (5) shows that they took roughly the same amount of steps. Table (4) shows a time of 0.07 sec. for the sixteen domain case while table (5) shows a time of 0.20 sec. Because both runs required an equal amount of communication the difference must be due to the specific run and they should be equal.

Table 4:

Tschebycheff: Calculated Boundary
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	128	128	40	9.11e-07	8.95e-08	5.83
2	64	128	40	9.08e-07	8.95e-08	0.93
4	32	128	640	1.00e-05	9.29e-07	0.96
8	16	128	140	9.64e-07	1.25e-07	0.18
16	8	128	30	1.52e-06	1.98e-07	0.07
32	8	256	20	1.91e-06	3.03e-07	0.12

Table 5:

Tschebycheff: True Boundaries with Communication
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	128	128	40	9.11e-07	8.95e-08	5.83
2	64	128	40	9.08e-07	8.95e-08	0.94
4	32	128	140	5.53e-06	5.43e-07	0.34
8	16	128	50	2.25e-06	3.12e-07	0.10
16	8	128	30	3.78e-06	3.95e-07	0.20
32	8	256	30	3.35e-06	3.31e-07	0.09

Table 6:

Tschebycheff: True Boundaries with no communication
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	128	128	40	9.11e-07	8.95e-08	5.83
2	64	128	40	9.08e-07	8.95e-08	0.91
4	32	128	140	5.53e-06	5.43e-07	0.27
8	16	128	50	2.25e-06	3.12e-07	0.08
16	8	128	30	3.78e-06	3.95e-07	0.09
32	8	256	30	3.35e-06	3.31e-07	0.11

In these specific runs there appears not to be a large penalty for increasing the number of domains and the method is able to resolve the subdomain interfaces All of the tables display a dramatic decrease in time required. The exception to this is the case of four domains. Initial investigations indicate that the initial conditions may be to blame for a disadvantageous arrangement around the subdomain interface. Again, these test used $\gamma = 0.2$ for the Dufort-Frankel method. This value seems to provide the best trade off between accuracy and stability. More research needs to be done on the relationship between stability and accuracy, and γ. Also more tests for different initial values and time steps need to be examined.

5.3 Legendre Results. In the following examples all of the tables will compare a constant grid size per domain. When calculating the collocation points and the weights for the Legendre polynomials there is a great deal of error accrued for fine grids.

Table 7:
Legendre: Calculated Boundary
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	16	16	90	6.46e-06	7.56e-07	0.19
2	16	32	40	8.81e-07	8.95e-08	0.13
4	16	64	560	8.61e-06	8.45e-07	0.65
8	16	128	80	2.16e-06	2.14e-07	0.17
16	16	256	30	4.99e-06	5.69e-07	0.14

Table 8:
Legendre: True Boundaries with Communication
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	16	16	90	6.46e-06	7.56e-07	0.17
2	16	32	40	8.81e-07	8.95e-08	0.13
4	16	64	140	5.53e-06	5.43e-07	0.28
8	16	128	50	2.25e-06	3.12e-07	0.15
16	16	256	30	3.78e-06	3.98e-07	0.17

Table 9:
Legendre: True Boundaries with no communication
Frequency = 5.000000 k = 0.062500 dt = 0.125000

Number Domains	Grid Points Domain	Total Grid Points	Number Steps	L1 Error	L Infinity Error	Elapsed Time (sec.)
1	16	16	90	6.46e-06	7.56e-07	0.23
2	16	32	40	8.81e-07	8.95e-08	0.11
4	16	64	140	5.53e-06	5.43e-07	0.16
8	16	128	50	2.25e-06	3.12e-07	0.12
16	16	256	30	3.78e-06	3.98e-07	0.12

Tables (7) through (9) display the results obtained from runs using Legendre collocation. The results are a little different than the Tschebycheff results. Here the method appears to have better resolved the boundaries and the number of iterations required is less. The results of the times required offer the same lesson as that of the Tschebycheff results.

References

[1] DAVIS, PHILIP J. AND RABINOWITZ, PHILIP, *Methods of Numerical Integration*, Academic Press, inc. (Harcourt Brace Jovanovich), New York, second ed., 1984.

[2] FUNARO, DANIELE, *Domains Decomposition Methods for Pseudo Spectral Approximations*, tech. report, Univerità degli studi di Pavia, Strada Nuova 65, 27100 Pavia, Italy. Research supported in part by AFOSR Grant 85-0303.

[3] ——, *A Multidomain Spectral Approximation of Elliptic Equations*, Numerical Methods for P.D.E., (1986), pp. 187–205.

[4] GOTTLIEB, DAVID AND GUSTAFSSON, BERTIL, *Generalized Dufort-Frankel Methods for Parabolic Initial-Boundary-Value Problems*, ICASE Report no. 75-5, ICASE, NASA Langley Research Center, Feb. 1975.

[5] GOTTLIEB, DAVID AND LUSTMAN, LIVIU, *The Dufort-Frankel Chebyshev Method for Parabolic Initial Boundary Value Problems*, ICASE Report No. 81-42, ICASE, NASA Langley Research Center, Dec. 1981.

[6] GOTTLIEB, DAVID AND ORSZAG, STEVEN A., *Numerical Analysis of Spectral Methods: Theory and Applications*, Society of Industrial and Applied Mathematics, Philadelphia,PA USA, 1977.