NASA Contractor Report 178179

ICASE REPORT NO. 86-60

ICASE

GLOBAL COLLOCATION METHODS FOR APPROXIMATION AND THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Contract Nos. NAS1-17070, NAS1-18107

September 1986

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

Operated by the Universities Space Research Association

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

GLOBAL COLLOCATION METHODS FOR APPROXIMATION AND THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

A. Solomonoff
ICOMP
NASA Lewis Research Center

E. Turkel
Institute for Computer Applications in Science and Engineering,
NASA Lewis Research Center
and
Tel-Aviv University

ABSTRACT

We apply polynomial interpolation methods both to the approximation of functions and to the numerical solutions of hyperbolic and elliptic partial differential equations. We construct the derivative matrix for a general sequence of the collocation points. The approximate derivative is then found by a matrix times vector multiply. We explore the effects of several factors on the performance of these methods including the effect of different collocation points. We also study the resolution of the schemes for both smooth functions and functions with steep gradients or discontinuities in some derivative. We investigate the accuracy when the gradients occur both near the center of the region and in the vicinity of the boundary. The importance of the aliasing limit on the resolution of the approximation is investigated in detail. We also examine the effect of boundary treatment on the stability and accuracy of the scheme.

Research for the second author was supported by NASA Contract Nos. NAS1-17070 and NAS1-18107 while he was in residence at NASA Lewis Research Center, Cleveland, OH 44135 and ICASE, NASA Langley Research Center, Hampton, VA 23665-5225.

1. INTRODUCTION

In this study we consider the accuracy of the pseudospectral approximation both for a function and also for the numerical solution of differential equations. We shall only consider collocation methods, but most of the results shown also apply to Galerkin methods. We approximate the function, $f(x)$, by a polynomial, $p_N(x)$, that interpolates $f(x)$ at $N + 1$ distinct points $x_0,\ldots,x_N$. $f'(x)$ is approximated by $P_N'(x)$ which is calculated analytically. In solving differential equations we use an approach similar to finite differences. Thus, all derivatives that appear are replaced by their pseudospectral approximation. The resultant system is solved in space or advanced in time for time dependent equations. Hence, for an explicit scheme, nonlinearities do not create any special difficulties.

This approach is equivalent to expanding $f(x)$ in a finite series of polynomials related to $x_0,\ldots,x_N$. For a Galerkin method, the coefficients of this series are obtained from the infinite expansion. For a collocation method, the coefficients are obtained by demanding that the approximation interpolate the function at the collocation points. This requires $O(N^2)$ operations. For special sequencies of collocation points, e.g., Chebyshev methods, this can be accomplished by using FFT's and only requires $O(N\log N)$ operations. Every collocation method has two interpretations: one in terms of the collocation points and one in terms of a series expansion. In the past, this has lead to some confusion. As an example we consider the case of a Chebyshev collocation method with $x_j = \cos(\pi j/N)$. From an approximation viewpoint, we know [11, 15 - 18] that the maximum error for interpolation at the zeroes of $T_N(x)$ is within $(4 + 2/_\pi \log N)$ of the minimax error and converges for all functions in C^1. The bound for the error based on the

points x_j, given above, is even smaller than this [13]. There also exist sharp estimates in Sobolev spaces [3]. Since the minimax approximation has an error which is equi-oscillatory we expect the Chebyshev interpolant to be nearly equioscillatory. Indeed, Remez suggests using these x_j as a first guess in finding the zeros of $f - P_N$ in his algorithm for finding the minimax approximation. Thus, we would expect that when used to solve differential equations that the error would be essentially uniform throughout the domain.

On the other hand, viewed as a finite difference type scheme, one expects the scheme to be more accurate near the boundaries where the collocation points are clustered. At the center of the domain the distance between points is approximately $\pi/2N$ while near the boundary the smallest distance between two points is approximately $\pi^2/2N^2$. Hence, the spacing at the center is about $\pi/2$ times coarser than an equivalent equally spaced mesh. Near the boundary the Chebyshev points are about $4N/\pi^2$ times finer than an equally spaced mesh. From this point of view, we expect the accuracy and resolving power of the scheme to be better near the boundaries. However, the bunching of points near the boundaries only serves to counter the tendency of polynomials to oscillate with large amplitude near the boundary. We shall also consider collocation based on uniformly spaced points. Since, we consider polynomial interpolation on the interval $[-1,1]$ we get qualitatively different results than obtained by Fourier or finite difference methods even for the same collocation points. In fact, we shall see that the boundaries exert a strong influence for this case similar to the interpolation based on Chebyshev nodes.

Connected with this, we shall examine the influence of boundary conditions on the accuracy and stability of pseudospectral methods. In

general, global methods are more sensitive to the boundary treatment than local methods. We also consider the effect of the location of the collocation points on both the accuracy and stability of the scheme and its effect on the allowable time step for an explicit time integration algorithm.

2. APPROXIMATION AND DIFFERENTIATION

We assume that we are given $N + 1$ distinct points $x_0 < x_1 < \ldots < x_N$. Given a function $f(x)$ it is well known how to approximate $f(x)$ by a polynomial $P_N(x)$ such that $P_N(x_j) = f(x_j)$, $j = 0,\ldots,N$. We define a function $e_k(x)$ which is a polynomial of degree N and $e_k(x_j) = \delta_{jk}$. Explicitly,

$$e_k(x) = \frac{1}{a_k} \prod_{\substack{\ell=0 \\ \ell \neq k}}^{N} (x - x_\ell) \tag{2.1}$$

$$a_k = \prod_{\substack{\ell=0 \\ \ell \neq k}}^{N} (x_k - x_\ell). \tag{2.1b}$$

Then the approximating polynomial is given by,

$$P_N(x) = \sum_{k=0}^{N} f(x_k) e_k(x). \tag{2.2}$$

We next consider an approximation to the derivative of $f(x)$. We construct this approximation by analytically differentiating (2.2). The value of the approximate derivative at the collocation points is a linear functional of the value of the function itself at the collocation points. Hence, given

the $N + 1$ values $f(x_j)$ we can find the values $P_N'(x)$ by a matrix multiplying the original vector $(f(x_0),\ldots,f(x_N))$. We denote this matrix by $D = (d_{jk})$. By construction, $DP = P_N'$ is exact for all polynomials of degree N or less. In fact, an alternative way of characterizing D is by demanding that it give the analytic derivative for all such polynomials at the $N + 1$ collocation points. In particular, we shall explicitly construct D by demanding that,

$$De_k(x_j) = e_k'(x_j), \qquad j,k = 0,\ldots,N, \tag{2.3}$$

i.e.,

$$D \begin{pmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \text{ - k-th row } = \begin{pmatrix} e_k'(x_0) \\ \cdot \\ \cdot \\ \\ \cdot \\ \\ e_k'(x_N) \end{pmatrix}.$$

Performing the matrix multiply, it is obvious that

$$d_{jk} = \frac{de_k}{dx}(x_j) \tag{2.4}$$

We next explicitly evaluate d_{jk} in terms of the collocation points x_j. Taking the logarithm of (2.1) we have

$$\log(e_k) = \sum_{\substack{\ell=0 \\ \ell\neq k}}^{N} \log(x - x_\ell) - \log(a_k).$$

Differentiating, we have

$$e_k'(x) = e_k(x) \sum_{\substack{\ell=0 \\ \ell\neq k}}^{N} 1/(x - x_\ell). \tag{2.5}$$

In order to evaluate (2.5) at $x = x_j$, $j \neq k$, we need to eliminate the zero divided by zero expression. We therefore, rewrite (2.5) as

$$e_k'(x) = e_k(x)/(x - x_j) + e_k(x) \sum_{\substack{\ell=0 \\ \ell\neq j,k}}^{N} 1/(x - x_\ell).$$

Since, $e_k(x_j) = 0$ for $j \neq k$ we have that

$$e_k'(x_j) = \lim_{x\to x_j} e_k(x)/(x - x_j).$$

Using the definition of $e_k(x)$, (2.1), we have

$$d_{jk} = \frac{1}{a_k} \prod_{\substack{\ell=0 \\ \ell\neq j,k}}^{N} (x_j - x_k) = \frac{a_j}{a_k(x_j - x_k)}, \quad \text{(see (2.1b))} \tag{2.7}$$

While the above formulas (2.5), (2.7) are computationally useable, it is sometimes preferable to express the formulas slightly differently. We, therefore, rederive these formulas using a slightly different notation. Define,

$$\phi_{N+1}(x) = \prod_{\ell=0}^{N} (x - x_\ell). \tag{2.8}$$

Then,

$$\phi'_{N+1}(x) = \sum_{k=0}^{N} \prod_{\substack{\ell=0 \\ \ell \neq k}}^{N} (x - x_\ell) \tag{2.9}$$

and so

$$\phi'_{N+1}(x_j) = a_j. \tag{2.10}$$

It can also be verified that

$$\phi''_{N+1}(x_j) = 2a_j \sum_{\substack{k=0 \\ k \neq j}}^{N} \frac{1}{x_j - x_k}. \tag{2.11}$$

Hence,

$$d_{jk} = \begin{cases} \dfrac{a_j}{a_k(x_j - x_k)} = \dfrac{\phi'_{N+1}(x_j)}{\phi'_{N+1}(x_k)(x_j - x_k)} & j \neq k \quad (2.12a) \\ \displaystyle\sum_{\substack{\ell=0 \\ \ell=k}}^{N} \frac{1}{x_k - x_\ell} = \frac{\phi''_{N+1}(x_k)}{2\phi'_{N+1}(x_k)} & j = k \quad (2.12b) \end{cases}$$

Given a_j it requires another $2N^2$ operations to find the off diagonal elements by (2.12a). It requires N^2 operations to find all the diagonal elements from (2.12b). Hence, it requires about $4N^2$ operations to construct the matrix D. We multiply the matrix D on the left by $\mathrm{diag}(1/a_1,\ldots,1/a_N)$ and on the right by the matrix $\mathrm{diag}(a_1,\ldots,a_N)$. Then D is similar to a matrix D_1 where D_1 is a sum of an antisymmetric matrix and a diagonal matrix. Since ϕ''_{N+1} is a polynomial of degree $N - 1$

ϕ''_{N+1} cannot be zero at all the collocation points. Hence, the diagonal portion of D_1 is nonzero.

In many cases $x_0 = -1$, $x_N = 1$ and the other x_j are zeros of some polynomial $Q_{N-1}(x)$. Hence, $\phi_{N+1}(x) = (x^2 - 1)Q_{N-1}(x)$. One can then rewrite the formula for d_{jk} in terms of $Q_{N-1}(x)$. For $j, k \neq 0, N$ we reproduce the formulas of Tal-Ezer [20]. He further points out that if $Q_{N-1}(x)$ is a Jacobi polynomial associated with the weight function $(1 - x)^{\alpha}$ $(1 + x)^{\beta}$ then

$$\sum_{\substack{\ell \\ \ell \neq k}} \frac{1}{x_k - x_\ell} = \frac{-(\alpha + 1)}{2(x_k - 1)} - \frac{(\beta - 1)}{2(x_k + 1)} \tag{2.13}$$

where the sum is over the roots of $Q_{N-1}(x)$. This can then be used to simplify (2.12b). When the ends points $x = -1$ or $x = 1$ are included in the collocation points then these must be explicitly accounted for to find d_{jk}.

For the standard Chebyshev collocation points, we have

$$x_j = \cos(\pi j/N) \qquad j = 0,\ldots,N. \tag{2.14}$$

Note that this orders the points in reverse order from our usual assumption. In this case one can evaluate the derivative by using FFT's. This requires only NlogN operations rather than the N^2 operations required by a matrix multiply. Computationally it is found that for $N < 100$ that the matrix multiply is faster than the FFT approach, see, e.g., [23]. The exact crossover point depends on the computer and the efficiency of the software for computing FFTs and matrix multiplies. The matrix multiply has the advantage

that it is more flexible and vectorizable. Thus, for example, both the location and the number of the collocation points is arbitrary. In order to use the FFT approach, it is required that the collocation points be related to the Fourier collocation points, e.g., Chebyshev. Furthermore, the total number of collocation points needs to be factorizable into powers of 2 and 3 for efficiency. The efficiency of these factors depends on the memory allocation scheme of the computer. Other collocation nodes than (2.14) are considered in [3, 7, 13]. The matrix D for the Chebyshev points (2.14) is given in [7].

In Appendix A, we consider the problem when we have N collocation nodes but wish the derivative matrix to be exact (in least squares sense) for $M > N$ functions which need not be polynomials.

3. PARTIAL DIFFERENTIAL EQUATION

We consider in this study three applications of collocation methods: (1) approximation theory, (2) hyperbolic equations, and (3) elliptic equations. For approximation theory we need only discuss accuracy. We first need some way to measure the approximation error that can be used on a computer. We cannot use the error at the collocation points since, by construction, this error is zero. Instead, we use

$$\|f - P_N\|^2 = \sum_{\ell=0}^{N'} w_\ell [f(x_\ell) - P_N(x_\ell)]^2 \tag{3.1}$$

for some sequence of points x_j which are not the collocation points. In general, we shall choose N' much larger than N, the number of collocation

points. If the original points are chosen as Chebyshev nodes, then we again choose the x_j, as Chebyshev nodes based on this larger number, N'. Because of this selection of nodes the sum in (3.1) approximates the Chebyshev integral norm, i.e.,

$$\|f - P_N\|^2 \simeq \int_{-1}^{1} \frac{[f(x) - P_N(x)]^2}{\sqrt{1 - x^2}}\, dx \tag{3.2}$$

$$\text{where} \quad w_\ell = \frac{1}{c_\ell}\frac{\pi}{N} \quad \text{and} \quad c_\ell = \begin{cases} 1 & \ell = 0,N \\ 2 & \ell = 0,N \end{cases} .$$

When the collocation points are evenly spaced then we shall choose the nodes of the integration formula to be also uniformly spaced. In this case

$$\|f - P_N\|^2 \simeq \int_{-1}^{1} [f(x) - P_N(x)]^2 dx \tag{3.3}$$

$$\text{where} \quad w_\ell = \frac{1}{c_\ell} \cdot \frac{1}{N} .$$

For general collocation points it is not clear how to choose the weights w_ℓ in the norm. An alternative possibility is to measure the error in some Sobolev norm. In this case, the finite sum can be based on the original collocation points and the norm is the L^2 norm of the derivative. In this study all errors will be given by (3.2) regardless of the distribution of the collocation nodes.

For hyperbolic problems we need to be concerned with stability in addition to accuracy. We also study the influence of the boundary treatment on both the accuracy and stability of the method. For simplicity we shall only consider the model equation

$$u_t = a(x)\, u_x \qquad -1 \leq x \leq 1, \quad t > 0. \tag{3.4}$$

If $a(x)$ is positive at both boundaries then we need to impose a boundary condition at $x = 1$. If $a(-1)$ is negative while $a(1)$ is positive, then we impose boundary conditions at both ends. On the other hand if $a(-1)$ is positive while $a(1)$ is negative then no boundary conditions need be given. For spectral methods, it is important that this distinction be preserved at the approximation level. Thus, whenever analytic boundary conditions are not given the spectral technique will be used to advance the solution at the boundary. The given boundary conditions are always chosen so that we know the analytic solution.

We will solve the differential equation (3.4) by a pseudo-spectral algorithm. Thus, we will consider the solution only at the collocation points. We then replace the derivative in (3.4) by a matrix multiply as described in section 2. We next multiply $a(x)$ at each collocation point by the approximate derivative at that point. We now have a system of ordinary differential equations in time. To advance the solution in time we could use any ODE solver. In particular, we shall use a standard four stage fourth order Runge-Kutta formula. This formula has several advantages. First, since it is fourth order in time (for both linear and nonlinear problems), it is closer to the high spatial accuracy of the spectral method than a second order formula. Also, the region of stability includes a significant portion of the negative real half plane and so is appropriate for Chebyshev methods which have eigenvalues in the negative half plane. Finally, if we look along the imaginary axis it has a comparatively large stability region. An alternative method is to use a spectral method in time. However, it is difficult to

generalize such methods to nonlinear problems while Runge-Kutta methods extend trivially to nonlinear problems.

Since this is an explicit method (even though all the points are connected every time step) it is easy to impose boundary conditions after any stage of the algorithm. Whenever we wish we can let the pseudospectral method advance the solution at the boundary also. Since the method is explicit, there is a limit on the allowable Δt because of stability considerations. Heuristically, one can consider this stability limit as arising from two different considerations. One is based on the minimum spacing between mesh points, which usually occurs near the boundary. As noted above, this is heuristic since the domain of influence of each point is the entire interval. Alternatively, one can derive a stability limit by finding the spectral radius of $a(x)$ times the derivative matrix. This is also heuristic since the derivative matrix is not a normal matrix. For $a(x)$ constant both methods indicate that Δt varies with $1/N^2$. The exact constant varies with the particular Runge-Kutta method used. For a two stage Runge-Kutta method, the stability limit is about three times the minimum spacing. For further details, the reader is referred to [5, 7] and the result section.

In Appendix B, we present the proof of the stability of Chebyshev collocation at points (2.14) for $u_t = u_x$.

For our model elliptic problem, we shall choose a Poisson equation

$$\Delta u = f(x,y), \quad -1 \leq x,y \leq 1 \tag{3.5}$$

with $u(x,y)$ prescribed on all four sides. As before $f(x,y)$ will be chosen so that we know the analytic solution.

We solve a time independent equation since it is easier to distinguish the resolving power of the scheme in different regions of the domain. For a time dependent equation, it can be difficult to distinguish local accuracy since inaccuracies propagate from one part of the domain to another. This is especially true for systems of hyperbolic equations with characteristics travelling in each direction. When the time independent equations is elliptic, then the solution is smooth. In particular, $u(x,y)$ has at least two derivatives even if $f(x,y)$ is only continuous. The smoother $f(x,y)$ is the smoother $u(x,y)$ will be, assuming the boundary conditions are sufficiently smooth.

4. RESULTS

In this section, we describe the computational results that illustrate many of the properties of pseudospectral methods. We begin with the approximaton of functions. Unless otherwise noted, the collocation points will be the Chebyshev nodes, (2.14). As is well known, interpolation at these points yields a maximum error which is not much worse ($O(\log N)$) than the best possible minimax approximation [13 - 18]. Nevertheless, we shall see that the quality of the approximation can vary greatly for different functions. We shall also see the effect of varying the collocation points.

In Figure 1a, we display the pointwise error in approximating the function $u(x) = \sin(20x-m)$ where m varies between 0 and $\pi/2$. Thus, $u(x)$ varies between a sine and a cosine function. The top graph in Figure 1

is the error for an approximation to a sine wave. The phase changes in the following graphs and the bottom graph is the error for a cosine function. In this case we chose 28 Chebyshev collocation points. For $m = 0$, i.e., a sine function, the amplitude of the error is larger. This occurs since $\sin(x)$ is an odd function and hence the coefficient of T_N is zero and so in essence we are only using 27 polynomials. This is verified in Figure 1b by using $N = 29$; for this case the error of the cosine function is larger. Nevertheless, this result is interesting for time dependent problem where the solution varies between a sine and cosine function. In addition, we also notice that for $m = 0$ the largest errors occur in the middle of the domain while for $m = \pi/2$ the larger errors are near the boundaries. Thus, for smooth functions the maximum error can occur anywhere in the domain. There is no need for the error to be smaller near the boundaries where the collocation points are bunched together.

In Figure 2 we show the pointwise error in approximating the function $u(x) = |x - x_0|$ which has a discontinuous derivative at $x = x_0$. We define a point as being half way between two nodes in the Chebyshev sense when

$$x_{j+1/2} \equiv \cos(\pi(j + 1/2)/N).$$

The top of the graph displays the error when the discontinuous derivative is located halfway between nodes while the center of the graph shows the error when the discontinuity in the derivative occurs at a node. The other graphs show progressively other locations of x_j. Thus, we see that when the discontinuous derivative occurs half-way between nodes in the Chebyshev sense, then the error has a sharp peak near the discontinuity but is close to zero

elsewhere. When the discontinuity occurs near a node then the error is more spread out and several peaks may occur but the maximum error is decreased. Gottlieb has observed similar phenomena in other problems. For other values of x the error goes smoothly between these extremes.

In Figure 3a, we examine the effect of the aliasing error in the approximation of a function. Gottlieb and Orszag [5] show that one needs at least π points per wave length when using a Galerkin Chebyshev approximation. In Figure 3, we approximate $\sin(M\pi x)$ with N Chebyshev nodes in a pseudospectral approximation. We plot the L^2 error, (3.2), as a function of the number of points past the aliasing limit. As before the error begins to decrease exponentially when there are π points per wave length. We further see that in order to reach a fixed error the number of collocation points, N, should vary (approximately) as the aliasing limit plus $M^{1/3}$. Computationally, it is hard to find the exact exponent, but it seems to be between 0.3 and 1/3. In Figure 3b, we see that for $f(x) = \tanh(mx)$ there is no sudden aliasing limit. Rather there is a gradual reduction in the error as N increases.

For a Fourier method, it can be shown that one only needs two points per wave length rather than π points per wave length. It might be speculated that this is due to the larger spacing of the Chebyshev method near the middle of the domain. In fact, asymptotically, the largest spacing between Chebyshev node is exactly $\pi/2$ times as large as for Fourier nodes. In Figure 4, we consider the same case as in Figure 3, but where the collocation points are evenly spaced. One sees that one again need about π points per wavelength before exponential accuracy occurs even though the spacing is the same as for the Fourier method. There is a theorem that interpolation based on uniformly

spaced points converges for analytic functions. Nevertheless, we see in Figure 4 that the approximation begins to diverge if N is sufficiently large with respect to M. The calculations for these case were carried out on a CRAY computer which has about 15 significant figures. Using double precision (about 30 significant digits) one stabilizes the procedure until larger N are reached at which point the approximation again diverges. Hence, even though the function is analytic nevertheless roundoff errors eventually contaminate the approximation. Hence, collocation based on uniformly spaced node is risky even for analytic functions because of the great sensitivity of these collocation methods to any noise level.

In Figure 5a, we study the resolving power of Chebyshev methods when there are sharp gradients. It is often stated, that Chebyshev methods are ideal for boundary layer flows since they naturally bunch points in the boundary layer. In Figure 5a, we plot the L^2 error when we are approximating the function $u(x) = \tanh(M(x - x_0))$, for $M = 8$, 32, 128, 512, and 2048 and $N = 31$. As M increases the gradient becomes steeper and in the limit approaches a Heaviside function. Furthermore, $\tanh(x)$ is also a solution to Burger's equation and so appropriate to model boundary layers. We see that, indeed, for moderate values of M the accuracy is greater when the gradient occurs closer to the boundary. Thus, given a moderate slope a Chebyshev collocation method "sees" the gradient better if it is near the edge of the domain. Thus it may be advantageous to consider multidomain approaches [9, 10]. However, when the slope becomes too large so that it is not resolved by the collocation points, then the error is equally large everywhere. In particular, a true discontinuity, e.g., a shock, is not resolved any better near the boundary then it is in the middle of the domain.

In the previous case it was implied that the Chebyshev method resolves gradients better near the boundary because the nodes are closer together near the edge. To check this hypothesis, we plot the same case in Figure 5b where now the collocation is based on uniformly spaced points. We consider the same case in Figure 5a but now choose $M = 2, 4, 8, 16, 32$. We choose lower values of M then before since the approximation based on evenly spaced points does not converge when the gradient is too large. For the same M the errors are much larger for the uniformly spaced nodes than Chebyshev spaced nodes. Nevertheless, the errors are much smaller when the gradients occur near the boundary. Thus, gradients in the "boundary layer" are better resolved than in the center of the domain even though we are using interpolation based on uniformly spaced points. In fact, the ratio of the L^2 error when the gradient is at the center to the L^2 error when the gradient is near the edge is even larger for uniformly spaced nodes than for a Chebyshev distribution of nodes. In both cases, we used the Chebyshev norm (3.2). However, the results do not depend on the details of the norm.

In order to explain this phenomenon we examine the singularity of the function in the complex plane. To simplify the discussion we shall consider the easier case of an expansion of a function in Chebyshev polynomials. In this case it is known [14] that the approximation converges in the largest ellipse with foci at $+1$ and -1 that does not contain any singularities. The equation of an ellipse with foci at ± 1 is

$$\frac{x^2}{\ell^2} + \frac{y^2}{\ell^2 - 4} = 4$$

where $\ell > 2$ measures the size of the ellipse. Let, $r = \ell + \sqrt{\ell^2 - 4}$.

Then r is the sum of the semi-major and semi-minor axes. It is known [12, 16] that the convergence rate of the scheme is bounded by r^{-N}. Hence, as ℓ increases the approximation converges faster. ℓ is determined by the closest singularity. Suppose that this singularity occurs at $\bar{x}$, $\bar{y}$, then

$$\ell^2 = 2(\bar{x}^2 \bar{y}^2 + 1 + \sqrt{(\bar{x}^2 + \bar{y}^2 - 1)^2 + 4\bar{y}^2}).$$

For $f(x) = \tanh(M(x - x_0))$ we have that $\bar{x} = x_0$ and $\bar{y} = \frac{\pi i}{2M}$. Thus, as x_0 varies, $\bar{y}$ is fixed while $\bar{x}$ changes. It is easily shown that $\frac{\partial}{\partial \bar{x}^2}(\ell^2) > 0$. Thus, for fixed $\bar{y}$, ℓ^2 is a minimum at $\bar{x} = 0$ and ℓ increases as $|\bar{x}| = |x_0|$ increases. Hence, as x_0 approaches the boundaries, ± 1, the rate of convergence increases. Also, as M increases, i.e., the function has a larger gradient, then $\bar{y}$ decreases and ℓ decreases and so the rate of convergence decreases. For interpolation approximations, both at Chebyshev and uniformly spaced points, a similar phenomenon occurs but the quantitative analysis is more complex [12].

For uniformly spaced collocation points in [0,1], the ellipses are replaced by the curves $u(x,y) = \text{constant}$ where

$$u(x,y) = 1 - x\ln\sqrt{x^2 + y^2} - (1 - x)\ln\sqrt{(1 - x)^2 + y^2} + y \arctan \frac{y}{x - x^2 - y^2}.$$

By examining graphs of this curve [11, p. 249] one sees that having the first singularity at $x_0 + i\bar{y}$ increases the size of the region as x_0 moves toward the boundaries. As before, this increases the rate of convergence.

In Figure 6, we consider the same case as Figure 5a for the function $f(x) = \tanh[Q(x - x_0)]$. Here, Q is a function of M and x_0, specifically

$$Q = \frac{\pi M}{2} \sqrt{\frac{M^2 + 1}{1 + M^2(1 - x_0)}}, \qquad M = 2, 4, 8, 15, 32.$$

At the center, $x_0 = 0$, $Q = \frac{M\pi}{2}$ while near the edge $x_0 \simeq 1$ and $Q \simeq \frac{M^2\pi}{2}$. With this scaling the L^2 error is essentially independent of x_0. This indicates that an adaptive collocation method could be useful [2, 8].

In order to further investigate the resolving power of the schemes, we repeat the experiment of Figure 5 but for a function that is not analytic. In this case, our previous analysis is no longer valid. We choose

$$f(x) = \begin{cases} \operatorname{sgn}(\eta) & |\eta| > 1 \\ \frac{1}{8}(3\eta^5 - 10\eta^3 + 15\eta) & |\eta| \leq 1 \end{cases} \tag{4.1}$$

where $\eta = M(x - x_0)$. Hence, $f(x) = -1$ when $x < x_0 - \frac{1}{M}$, $f(x) = +1$ when $x > x_0 + \frac{1}{M}$ and is a quintic polynomial in between. Furthermore, $f(x)$ has two continuous derivatives, but the third derivative is dicontinuous at $x = x_0 \pm \frac{1}{M}$. Thus, as before, x_0 denotes the center of the "jump" and the gradient becomes larger as M increases. In Figure 7a, we plot the L^2 error for Chebyshev collocation with 31 nodes. As x_0 goes toward the boundary, there is a small decrease in the error, but not as pronounced in Figure 5a. Even more surprising is the fact that the decrease in error is greater for $M = 32$ than for $M = 2$. Thus, in contrast to Figure 5a, there

is no longer a sharp decrease for smoother functions as x_0 approaches 1. When using uniformly spaced points the absolute error is larger than when using Chebyshev points. However, now there is a large decrease in the error as x_0 approaches the boundary. We compare the case $M = 16$; for Chebyshev collocation the error decrease by about 2 orders of magnitude as x_0 varies from the center to the edge. For uniformly spaced points the error decreases by about 6 orders of magnitude. This is despite the fact that the Chebyshev collocation method bunches the points near the edge. We also note that nothing special happens when x_0 is sufficiently close to the boundary that the discontinuous third derivative at $x_0 + \frac{1}{M}$ is no longer in the domain.

In Figure 8a, we study a similar phenomena. In this case, we study the L^2 error as we vary the strength of the singularity. We consider the function $u(x) = H(x - x_0) * (x - x_0)^M$, where $H(x)$ is the Heaviside function. Thus $u(x)$ has a discontinuous M-th derivative. As expected, based on previous cases, we see that when the high order derivatives are discontinuous that the Chebyshev collocation method resolves the functions best when the discontinuity is near the boundary. However, when low order derivatives are discontinuous than the differential between the edge and the center decreases. For a step function, $M = 0$, the error oscillates with equal amplitude throughout the domain. As x approaches the boundary only the frequency of the oscillation changes. In Figure 8b, we again see that the same qualitative picture occurs when the collocation is based on uniformly spaced points. We also see that global collocation based on uniformly spaced points is not convergent when the function is not smooth. This divergence is amplified if the discontinuity occurs near the center of the domain. In this case, the divergence is no longer caused by roundoff error. Rather it already

occurs at moderate values of N and begins at larger error levels. For $f(x) = |x|$ it can be proved [16] that collocation based on uniformly spaced nodes converges only for the points $x = 0, +1, -1$.

In order to further study the resolving power of the global schemes near the boundary, we consider the function

$$f(x) = \begin{cases} +1 & x < 1 \\ -1 & x \geq 1 \end{cases} .$$

We plot the pointwise error in Figure 9a for both Chebyshev nodes and for uniformly spaced notes. For uniformly spaced nodes, the error is very small in the interior, (see Figure 9b for a logarithmically scaled plot) but is very large near, i.e., within $O(\frac{1}{N})$, $x = 1$. From Figure 9b we see that the error is larger near $x = -1$ than in the center. For the Chebyshev nodes, the error is more global, but the large error near the boundary is confined to an interval of size $O(\frac{1}{N^2})$.

We next consider the partial differential equation

$$\begin{aligned} u_t &= u_x \qquad -1 \leq x \leq 1, \; t > 0 \\ u(x,0) &= f(x) \quad u(1,t) = g(t). \end{aligned} \tag{4.2}$$

We first discretize (4.2) in space using

$$v_t = D\, v \tag{4.3}$$

where D is the matrix derivative based on the collocation points $x_0,...x_N$ and v is the vector of the dependent function evaluated at the collocation nodes. We shall further assume that the point $x = 1$ is a collocation point. As before D is explicitly given by (2.12). To advance (4.3) in time we use a four-stage fourth order Runge-Kutta formula.

In studying (4.2) we shall be interested in both accuracy and stability properties of the algorithm. For stability we need to distinguish between space stability and time stability [6]. By space stability, we mean the behavior of the approximation v as the number of modes N increases when $0 \leq t \leq T$. By time stability we mean the behavior of v as time increases, for fixed N. Since, D can be diagonalized the scheme is time stable whenever all the eigenvalues of $\Delta t \cdot D$ lie in the stability region of the Runge-Kutta scheme. This does not necessarily prove space stability since the norm of the matrix that diagonalizes D depends itself on N. Obviously, the spectral radius of D and also the maximum allowable time step depends on the implementation of the boundary conditions.

Since the temporal accuracy is lower than the spatial accuracy the maximum Δt allowed by stability considerations will not yield very accurate approximations. However, by decreasing the time step we can increase the accuracy of the solution. This general technique works equally well for nonlinear problems. When the model equation (4.1) is replaced by a more realistic system with several wave speeds then the stability limit will also give approximations that are accurate [1]. Also, when one is only interested in the steady state then frequently the time step can be chosen by stability considerations alone. An alternative, which will not be persued in this study, is to use spectral methods also in the time domain, e.g., [4, 21].

In order to measure the accuracy of the approximation, we shall choose $u(x,t) = f(x - t)$ for some $f(x)$. Hence, the approximation can be compared pointwise with the analytic solution. The boundary data is then given by $g(t) = f(1 - t)$. We shall measure the error either pointwise or else in a weighted L^2 norm given by (3.2).

We first study the effect of the boundary treatment on the stability and accuracy of (4.3). One property of global methods is that the approximation is automatically updated at all collocation points including the boundaries. Thus, if one wished, the scheme could be advanced without ever imposing the given boundary data; but this would be an unstable scheme. For a multistage time scheme, one can impose the boundary conditions at any stage one wishes. We now consider (4.1) with $f(x) = \sin(\pi x)$. In Figure 10a, we impose the given boundary condition after each stage while in Figure 10b we impose the boundary condition only after the fourth stage. We define the Courant number, CFL, by

$$CFL = N^2 \Delta t.$$

In both plots, 10a and 10b, we display the error for several values of the Courant number. We see that imposing boundary conditions after each stage allows a larger maximum stable CFL number. For the four stage scheme, the maximum CFL is about 35. However, for smaller time steps the error is slightly larger than when one imposes the boundary condition only at the end of all the stages. One also sees that for a given error level that the approximate solution is essentially independent of the time step below some critical time step. As one demands more accuracy the necessary CFL number decreases. For a smooth solution, the necessary time step depends

exponentially on N. The largest stable Δt do not give accurate solution at any error level. We also found that the error grows in time if the solution is not sufficiently resolved in either space or time. There was no growth when N was large enough and Δt was sufficiently small.

In Figures 11a and 11b, we again plot the L^2 error for approximating (4.2) with $f(x) = \sin(x)$ as N increases and for a selection of CFL numbers. In this plot, we choose a different sequence of collocation points given by

$$x_j = -(1 - \alpha) \cos \frac{\pi j}{N} + \alpha(-1 + \frac{2j}{N}) \qquad j = 0,\ldots,N \tag{4.4}$$

so $x_0 = -1$ and $x_N = 1$. These points are a linear combination of Chebyshev nodes and uniformly spaced nodes. Letting

$$\alpha = \frac{(\beta - 1)(1 - \cos\frac{\pi}{N})}{\frac{2}{N} - (1 - \cos\frac{\pi}{N})} .$$

Then

$$\begin{aligned} \alpha &\simeq \frac{\pi^2(\beta - 1)}{4N} \cdot \frac{1}{1 - \frac{\pi^2}{4N}} \\ &= O(\tfrac{1}{N}) \qquad \text{when } \beta = O(1), \end{aligned} \tag{4.5}$$

and we find that

$$\beta = \frac{\text{new spacing at edge}}{\text{Chebyshev spacing at edge}} \tag{4.6}$$

We solve (4.1) by using the derivative matrix (2.12). We do not use a mapping to Chebyshev collocation nodes. In Figure 11a, we choose $\beta = 2$, i.e., a spacing at the edge twice as coarse as the usual Chebyshev spacing. We see that in this case we cannot increase the allowable time step beyond the stability condition for the Chebyshev nodes. Hence, the stability condition is not directly related to the minimum spacing. In Figure 11b, we display the error for $\beta = 1/2$, i.e., a spacing twice as small as the Chebyshev spacing near the wall. In this case the largest stable time step is reduced compared with the Chebyshev nodes. In this example, we have considered constant coefficients. For a problem with variable coefficients it is possible that coarsening the mesh near the boundary will allow a larger time step. This is because the coarser mesh near the boundary may just counteract the behavior of the variable coefficients near the boundary.

In Figure 12, we consider uniformly spaced nodes, i.e., $\alpha = 0$. From Figure 12, we see that even for small CFL levels that the error first decreases but then increases as N gets larger. These calculations were carried out in double precision on the CRAY. Nevertheless, it is difficult to distinguish between a mathematical instability and an instability caused by rounding errors on the computer.

In Figure 13, we consider the differential equation

$$u_t = -xu_x, \qquad -1 \leq x \leq 1 \tag{4.7}$$

$$u(x,0) = f(x).$$

For this differential equation, we do not specify boundary conditions at either end of the domain. The solution is given by $u(x,t) = f(xe^{-t})$ and

so $u(x,t)$ decays to a constant value. It is easy to verify that the eigenfunctions of the spatial operator are $v_j(x) = x^j$ with corresponding eigenvalue $\lambda_j = -j$, $j = 0,\ldots,N$ (see also [5]). Hence, the stability condition is $\Delta t \leq \frac{C}{N}$ where C depends on the details of the explicit time integration scheme. Since $u(x,t)$ is almost constant for large time the error levels become very small. In figure 13, we plot the L_2 error at time $t = 10$, with $f(x) = \sin\pi x$. For the fourth order Runge-Kutta scheme $C \sim 2.8$ is the stability limit. In Figure 13a, we use double precision on the cray (about 30 significant figures) while in Figure 13b we only use five significant figures. We define the CFL number for this problem as

$$CFL = N\Delta t.$$

Comparing 13a with 13b, we note that $CFL = 2$ is stable using double precision but is unstable using only five significant digits. The effect of roundoff on stability is studied in [24]. For this case the effects of roundoff are important only for time steps very close to the stability limit. The effect of roundoff is more pronounced when $\Delta t \sim 1/N$ than when $\Delta t \sim 1/N^2$.

We further see from this case that the maximum allowable time step is not necessarily related to the minimum spacing in the grid. In this case, the fact that no boundary conditions were specified allowed Δt to vary with $1/N$ rather than the usual $1/N^2$. We also saw a similar phenomenon where coarsening the mesh near the boundary did not allow a larger maximum time step. A similar conclusion was found by Tal-Ezer [22] for the Legendre-Tau method which has a time step limitation that depends on $1/N$ even though the

minimum grid spacing is $1/N^2$. Thus, to find the stability limit, one must analyze the derivative matrix appropriate for each case rather than using a heuristic approach based on the spacing between collocation nodes.

We finally discuss the solution to the Poisson equation (3.5). The right hand side and boundary conditions are chosen by deciding a priori on the exact solution. We solve (3.5) by a Chebyshev collocation method in each direction. The matrix equation that results is solved by a multigrid technique [25].

In Figure 14 we consider the case where the exact solution is $u(x,y) = \sin\pi y \tanh(M(x - x_0))$. Thus, the solution is smooth in y and has a gradient in the x direction. The sharpness of the gradient and its location are given by M and x_0 respectively. Hence, this models boundary layer type behavior. As before (see Figure 5a) when M is not too small then the approximation is more accurate when the gradient occurs near the boundary. For sharp gradients, i.e., M very large, the gradient is not resolved by the mesh and the Chebyshev L^2 error is approximately independent of the position of the gradient. As shown by Figure 5b, this increased accuracy in the boundary layer is not only due to the increased number of collocation points in the "boundary layer". Rather it is due to properties of global approximation techniques. It is of interest to note that for $M = 1024$, i.e., a discontinuity, that the error is almost constant. However, for $M = 64$ and 256, i.e., a sharp gradient, there are peaks in the error as x_0 approaches a collocation node.

5. CONCLUSIONS

We consider the properties of global collocation methods to problems in approximation theory and also partial differential equations. In particular, we study concepts that have been used by many authors without verification.

In order to be able to study differential equations for a general sequence of collocation nodes we calculate the approximate derivative by a matrix times vector multiply. For Fourier or Chebyshev methods one could also use a FFT [5]. For a small number of nodes, $N \lesssim 64$, the matrix multiply is faster than the FFT. For sufficiently large N the FFT is always faster since it grows as NlogN rather than N^2. The exact cross-over point between the two techniques is very machine dependent as well as software dependent. There obviously are differences between scalar, vector, and parallel computers. Nevertheless, for practical N used in most partial differential equation solvers the matrix multiply is not much slower than the FFT. Hence, we only use the matrix multiply technique due to its greater generality and flexibility.

It follows from the results presented in Section 4 that a global collocation method must be distinguished from a local finite difference or finite element approximation. In particular, the greater density of points, for a Chebyshev collocation method, near the boundary does not give increased accuracy, for a smooth function, near the boundary. The extra density near the boundary is needed to counteract the tendency of polynomials to have large errors near the edges of the domain.

Chebyshev collocation methods do have lower errors when sharp gradients are near the boundary than when they are in the center of the domain. Similar results occur when there is a discontinuity in some derivative. However,

qualitatively similar results are obtained using uniformly spaced nodes. Thus, the increased resolution near the boundary is due to the global nature of the approximation and not the bunching of collocation nodes. Of course, in terms of absolute error, it is preferable to use Chebyshev collocation rather than uniform collocation. This indicates that domain decomposition methods should be advantageous [9, 10] but not for shocks. In fact, even in cases where collocation based on a uniform mesh should converge the actual interpolation process on a computer eventually diverges due to roundoff errors. These roundoff errors contaminate the results for relatively small N.

As a further distinction between global and local techniques we consider the aliasing limit. For a Fourier (periodic) method we need 2 points per wave length to resolve a sine wave. For a Chebyshev method we need π points per wavelength. The difference between 2 and π is not due to the different distribution of points in these techniques. Polynomial collocation based on uniformly spaced points again needs π points per wavelength. Furthermore, for other functions, e.g., tanh x, one does not observe any sharp aliasing limit. Thus, one can not speak of number of points per wave length for general functions on nonperiodic domains.

An alternative to improving the accuracy of an approximation is to map the x domain $[-1,1]$ onto another computational domain s, for simplicity again $[-1,1]$. The above conclusions do not extend to such mappings. First, a polynomial in s is no longer a polynomial in x. Hence, in the physical space x we are not considering polynomial collocation methods. In addition, the L^2 norm in s-space corresponds to a weighted L^2 norm in x-space. Hence, it is difficult to measure the effectiveness of such mappings. In practice [2] has shown that in some cases adaptive mesh mappings can be effective for spectral methods.

The results obtained for approximating solutions to elliptic partial differential equations seem to correspond to the results for the approximation problem. Again one can not interpret the properties of a Chebyshev collocation method in terms of finite difference properties. Such concepts as number of points in a local region are not meaningful. If one chooses another set of collocation points then there are two ways of implementing the method. One can map one set of points to the other and then use a Chebyshev method in the computational space. This introduces metrics into the equation. Alternatively, one can solve the equation in physical space using the general derivative matrix (2.12). We have not investigated the differences between these two approaches.

For a time dependent partial differential equation, the study is more complicated. First, there is an accumulation of errors as time progresses. Thus, for example, for a stationary problem one can distinguish between the discontinuity being at a node or in between nodes. For a time dependent problem the discontinuity is moving and so all effects are combined. This is especially true for systems with variable coefficients where there is coupling between all the components.

Also, there is the question of stability in addition to accuracy. Thus, we have found that the implementation of boundary conditions influences both the maximum time step allowed and the accuracy. At times an implementation which increases the stability will decrease the accuracy.

We also found that there is no direct correlation between the smallest distance in the mesh and the maximum allowable time step. Coarsening the mesh near the boundary does not allow a larger time step. This again demonstrates the fallacy of describing a global method in terms of local behavior. As is

well known, for wave equation type problems one should not choose the maximum allowable time step allowed by stability. Since we use a fourth order accurate method in time but a spectrally accurate method in space one should choose a smaller time step to compensate. Thus to achieve time accuracy there is no need to increase the $0(1/N^2)$ time step restriction for hyperbolic equations. For stiff problems or if one is not interested in time accuracy then one may wish to exceed the stability restriction. Furthermore, for parabolic equations $\Delta t \simeq 0(1/N^4)$ which is much too restrictive. As before, one can consider other sets of collocation points. Again using mappings or the derivative matrix based on these nodes give rise to different schemes.

ACKNOWLEDGEMENT

The authors would like to thank A. Bayliss of Northwestern University for the use of his code to find adaptive points and critically reading the paper. We also thank C. Street of NASA Langley for the use of his spectral multigrid package to solve the Poisson equation. The second author also thanks D. Gottlieb of Tel-Aviv University for many discussions on the paper.

APPENDIX A

In Section 2, we saw that given the collocation points $x_0,\ldots,x_N$ and $N+1$ functions $u_j(x)$ then the derivative matrix, D, is determined by demanding that D times $(\phi_j(x_0),\ldots,\phi_j(x_N))^t$ give the exact derivative at the collocation points. Let $D = (d_{jk})$ and define the matrix U by $U_{jk} = u_j(x_k)$ $j,k = 0,\ldots,N$. Given the matrices D and U we denote the j-th column of these matrices as d_j and u_j. Then each column of D is determined by the equation

$$Ud_j = u_j' \qquad \text{at all collocation points } x_k,\ k = 0,\ldots,N. \tag{A1}$$

If we wish D to be exact for $M > N+1$ functions, then in general there is no solution. Instead we can demand that D give the smallest L^2 error over these M functions. Intuitively if D is almost exact for many functions, then it should be a good approximation to the derivative. In particular, one may choose functions that are more appropriate to a given problem than polynomials. Choosing D to give the least sqares minimization is equivalent to demanding that

$$U^tUd_j = U^Tu_j' \tag{A2}$$

instead of (A1). It is easily to verify that

$$(U^T U)_{jk} = \sum_{i=0}^{M} u_i(x_j) u_i(x_k)$$

and

$$(U^t u_j')_k = \sum_{i=0}^{M} u_i(x_k) u_i'(x_j), \qquad j,k = 0,\ldots,N. \tag{A3}$$

We now define

$$v_k(x) = \sum_{i=0}^{M} u_i(x_k) u_i(x), \qquad k = 0,\ldots,N \tag{A4}$$

then

$$v_k'(x_j) = (U^t u_j')_k.$$

Assuming $\det U \neq 0$ then the $v_k(t)$ are linearly independent. It also follows that D is exact for the $N + 1$ functions $v_k(x)$ at the collocation points. Hence, demanding least square minimization for $u_i(x)$, $i = 0,\ldots,M$ is equivalent to demanding exactness for $v_i(x)$, $i = 0,\ldots,N$ given in (A4).

We next extend this by letting M become infinite and replacing the sums by integrals. Thus, given the continuum of function $u_i(x)$ and demanding that D be the best least squares approximation to the derivative at the collocation points is equivalent to demanding that D be exact for the $N + 1$ functions

$$v_k(x) = \int_0^m u_i(x_k) u_i(x) di. \tag{A5}$$

To demonstrate this, we consider a specific example. Let $u_k(x)$ be the functions $\sin(kx)$ and $\cos(kx)$ for $0 \leq k \leq \frac{N\pi}{2}$ and choose $N + 1$ uniformly spaced collocation points, x_j. Since $k \leq \frac{N\pi}{2}$ we are always below the aliasing limit. It follows from (A5) that

$$v_k(x_j) = \int_0^{\frac{N\pi}{2}} [\sin(ix_j)\sin(ix_k) + \cos(ix_j)\cos(ix_k)]di$$

$$= \begin{cases} \dfrac{\sin \frac{N\pi}{2}(x_j - x_k)}{x_j - x_k} & j \neq k \\ \dfrac{N\pi}{2} & j = k \end{cases} \qquad \text{(A6)}$$

These functions, $v_j(x)$ are known as SINC functions and have been used for interpolation formulae [19]. Demanding that D be exact for $v_j(x)$, $j = 0,\ldots,N$ yields the derivative matrix

$$d_{jk} = \begin{cases} \dfrac{(-1)^{j+k}}{x_j - x_k} & j \neq k \\ 0 & j = k \end{cases}$$

which is an antisymmetric matrix. We also note that this matrix resembles the derivative matrix for the Chebyshev nodes [7].

APPENDIX B

In this section, we present the proof that Chebyshev collocation at the standard Gauss-Lobatto points is stable for solving scalar hyperbolic equations. This result was given in [7] without proof.

Consider the collocation points

$$x_j = \cos \frac{\pi j}{N}, \qquad j = 0,\ldots,N. \tag{B1}$$

Let u be the solution to

$$u_t = u_x \qquad u(1,t) = 0 \qquad u(x,0) = f(x). \tag{B2}$$

If v is a N-th order polynomial which is found by collocation at x_j (see [5], [7]), then v exactly solves the modified equation

$$v_t = v_x + \frac{(1 + x)T_N' \dot{a}_N(\tau)}{N}, \qquad v(1,t) = 0 \tag{B3}$$

where

$$\dot{a}_N = \frac{d}{dt} a_N$$

and

$$v(x,t) = \sum_{k=0}^{N} a_k(t)T_k(x). \tag{B4}$$

We need the following fact:

Lemma:

$$\frac{d}{dt}\left[(2a_N - a_{N-1})^2\right] = -4N(2a_N^2 - a_N a_{N-1}). \tag{B5}$$

Proof:

$$\frac{d}{dt}\left[(2a_N - a_{N-1})^2\right] = 2(2a_N - a_{N-1})(2\frac{da_N}{dt} - \frac{da_{N-1}}{dt}) \tag{B6}$$

comparing the coefficient of T_{N-1} in (B3) we find that

$$\frac{da_{N-1}}{dt} = 2Na_N + 2\frac{da_N}{dt}$$

or

$$2\frac{da_N}{dt} - \frac{da_{N+1}}{dt} = -2Na_N.$$

Inserting this into (B6) gives the lemma. With this lemma we prove the following theorem. Let

$$c_j = \begin{cases} 1 & j = 0,N \\ 2 & j \neq 0,N \end{cases},$$

then

Theorem: Let v solve (A2.3), then if $\frac{N}{3N-1} \leq \beta \leq \frac{4}{5}$ then

$$\frac{d}{dt}\left\{\frac{\pi}{2N}\sum_j \frac{1}{c_j}(1+x_j)(1-\beta x_j)v^2(x_j,t) + \frac{\pi}{8N}(2a_N - a_{N-1})^2\right\} \le 0 \qquad \text{(B7)}$$

and so the solution v is stable.

Proof: We multiply (B3) by $\frac{\pi}{NC_j}(1+x_j)(1-\beta x_j)v(x_j,t)$ and sum over the collocation nodes. We then have

$$\frac{\pi}{2N}\frac{d}{dt}\sum_j \frac{1}{c_j}(1+x_j)(1-\beta x_j)v_j^2 = \frac{\pi}{N}\sum_j \frac{1}{c_j}[(1+x_j)(1-\beta x_j)]vv_x + \frac{\dot{a}_N}{N}\frac{\pi}{N}\sum_j \frac{1}{c_j}(1+x_j)^2(1-\beta x_j)T_N'(x_j)v(x_j). \qquad \text{(B8)}$$

However, the last term is zero since $T_N'(x_j) = 0$ at interior points, $1 + x_j = 0$ at $x_j = -1$ and $v(x_j) = 0$ at $x_j = +1$. Furthermore, if

$$f(x) = \sum b_j T_j(x) \qquad f \in P_{2N-3}$$

then

$$\frac{\pi}{N}\sum \frac{1}{c_j} f(x_j) = \int \frac{f(x)}{\sqrt{1-x^2}}\,dx + \pi b_{2N}.$$

By algebra, it can be verified that the 2N-th Chebyshev coefficient of $(1+x)(1-\beta x)vv_x$ is $b_{2N} = \frac{(1-\beta)Na_N^2}{2} - \beta(\frac{2N-1}{4})a_N a_{N-1}$ where as before a_j are the Chebyshev coefficients of v. Therefore, (B8) can be rewritten as

$$\frac{\pi}{2N}\frac{d}{dt}\sum_j \frac{1}{c_j}(1+x_j)(1-\beta x_j)v_j^2 = \int_{-1}^{1}\frac{(1+x)(1-\beta x)vv_x}{\sqrt{1-x^2}}\,dx + \frac{\pi}{2}\left[(1-\beta)Na_N^2 - \beta\left(\frac{2N-1}{2}\right)a_N a_{N-1}\right]. \tag{B9}$$

Integrating by parts and using the fact that $v(1,t) = 0$ we find that

$$\frac{\pi}{2N}\frac{d}{dt}\sum_j \frac{1}{c_j}(1+x_j)(1-\beta x_j)v^2(x_j) = -\frac{1}{2}\int_{-1}^{1}\frac{(1-\beta-\beta x+\beta x^2)v^2(x,t)dx}{(1-x)\sqrt{1-x^2}} + \frac{\pi}{2}\left[(1-\beta)Na_N^2 - \beta\left(\frac{2N-1}{2}\right)a_N a_{N-1}\right]. \tag{B10}$$

Using the lemma this is equivalent to

$$\frac{d}{dt}\left\{\frac{\pi}{2N}\sum_j \frac{1}{c_j}(1+x_j)(1-\beta x_j)v^2(x_j,t) + \frac{\pi}{8N}(2a_N - a_{N-1})^2\right\} = -\frac{1}{2}\int_{-1}^{1}\frac{1-\beta-\beta x+\beta x^2}{(1-x)\sqrt{1-x^2}}v^2(x,t)dx - \frac{\pi}{2}\left[(3\beta-1)N-\beta\right]a_N^2. \tag{B11}$$

If $\beta \le \frac{4}{5}$, then the integral term is negative while if $\beta \ge \frac{N}{3N-1} \sim \frac{1}{3}$, then the second term on the right hand side is also negative. Hence, when

$\frac{N}{3N-1} \leq \beta \leq \frac{4}{5}$ then the right hand side of (B11) is negative and the theorem is proven.

If we choose the special case $\beta = \frac{4}{5}$ then (B11) becomes

$$\frac{d}{dt}\left\{\frac{\pi}{2N}\sum_j \frac{1}{c_j}(1 + x_j)(1 - \frac{4}{5}x_j)v^2(x_j,t) + \frac{\pi}{8N}(2a_N - a_{N-1})^2\right\}$$

$$= -\frac{1}{10}\int_{-1}^{1} \frac{(1 - 2x)^2}{(1 - x)\sqrt{1 - x^2}} v^2(x,t)dx - \frac{\pi}{10}(7N - 4)a_N^2. \qquad \text{(B12)}$$

As a corollary, this theorem implies that all the eigenvalues of D lie in the left half of the complex plane.

REFERENCES

1. A. Bayliss, K. E. Jordan, B. J. LeMesurier, E. Turkel, "A fourth order accurate finite difference scheme for the computation of elastic waves," to appear in Bull. Seismological Soc. America.

2. A. Bayliss and B. J. Matkowsky, "Fronts, relaxation oscillations and period doubling in solid fuel combustion," submitted to J. Comput. Phys.

3. C. Canuto and A. Quarteroni, "Approximation results for orthogonal polynomials in Sobolev spaces," Math. Comput., 1982, Vol. 38, pp. 67-86.

4. M. Deville, P. Haldenwang, G. Labrosse, "Comparison of time integration (finite difference and spectral) for the nonlinear Burger's equation," Proc. 4th GAMM-Conference Numerical Methods in Fluid Dynamics, Friedr. Viewey, 1982.

5. D. Gottlieb and S. Z. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM, Philadelphia, 1977.

6. D. Gottlieb, S. Z. Orszag, and E. Turkel, "Stability of pseudospectral and finite difference methods for variable coefficient problems," Math. Comput., 1981, Vol. 37, pp. 293-305.

7. D. Gottlieb and E. Turkel, "Spectral methods for time dependent partial differential equations," Lecture Notes in Mathematics, 1985, Vol. 1127, pp. 115-155, Springer-Verlag.

8. H. Guillard, R. Peyret, "On the use of spectral methods for the numerical solution of stiff problems," University of Nice, Dept. of Math., Report 108, 1986.

9. D. Kopriva, "A spectral multidomain method for the solution of hyperbolic systems," submitted to Appl. Numer. Math.

10. K. Z. Korczak, A. T. Patera, "An isoparametric spectral element method for solution of the Navier-Stokes equations in complex geometry," J. Comput. Phys., 1986, Vol. 62, pp. 361-382.

11. P. P. Korovkin, "Linear operators and approximation theory," trans. from Russian, Hindustan Publ. Corp., Delhi, 1960.

12. V. I. Krylov, Approximate Calculation of Integrals, trans. by A. H. Stroud, McMillan Co., New York, 1962.

13. J. H. McCabe and G. M. Phillips, "On a certain class of Lebesque constants," BIT, 1973, Vol. 13, pp. 434-442.

14. A. I. Markushevich, Theory of Functions of a Complex Variable, Vol. III, trans. by R. A. Silverman, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

15. G. Meinardus, Approximation of Functions: Theory and Numerical Methods, Springer, Berlin, 1967.

16. I. P. Natanson, Constructive Function Theory, Vol. III, Fredrick Ungar Publishing Company, New York, 1965.

17. M. J. D. Powell, Approximation Theory and Methods, Cambridge University Press, Cambridge, 1975.

18. T. J. Rivlin, An Introduction to the Approximation of Functions, Blaisdell Publishing Company, Waltham, Massachusetts, 1979.

19. F. Stenger, "Numerical methods based on Whittaker Cardinal, or sine functions," SIAM Review, 1981, Vol. 23, pp. 165-224.

20. E. Tadmor, "The exponential accuracy of Fourier and Chebyshev differencing methods," SIAM J. Numer. Analy., 1986, Vol. 23, pp. 1-10.

21. H. Tal-Ezer, "Spectral methods in time for hyperbolic problems," SIAM J. Numer. Analys., 1986, Vol. 23, pp. 11-26.

22. H. Tal-Ezer, "A pseudospectral Legendre method for hyperbolic equations with an improved stability condition," to appear J. Comput. Phys.

23. T. D. Taylor, R. S. Hirsh, M. M. Nadworny, "Comparison of FFT, direct inversion, and conjugate gradient methods for use in pseudo-spectral methods," Computers and Fluids, 1984, Vol. 12, pp. 1-9.

24. L. N. Trefethen and M. R. Trummer, "An instability phenomenon in spectral methods," to appear in J. SIAM Numer. Analy.

25. T. A. Zang, Y. S. Wong, M. Y. Hussaini, "Spectral multigrid methods for elliptic equations, II," J. Comput. Phys., 1984, Vol. 54, pp. 489-507.

26. A. Zygmund, Trigonmetric Series, Cambridge University Press, Cambridge, 1968, Vol. 1.

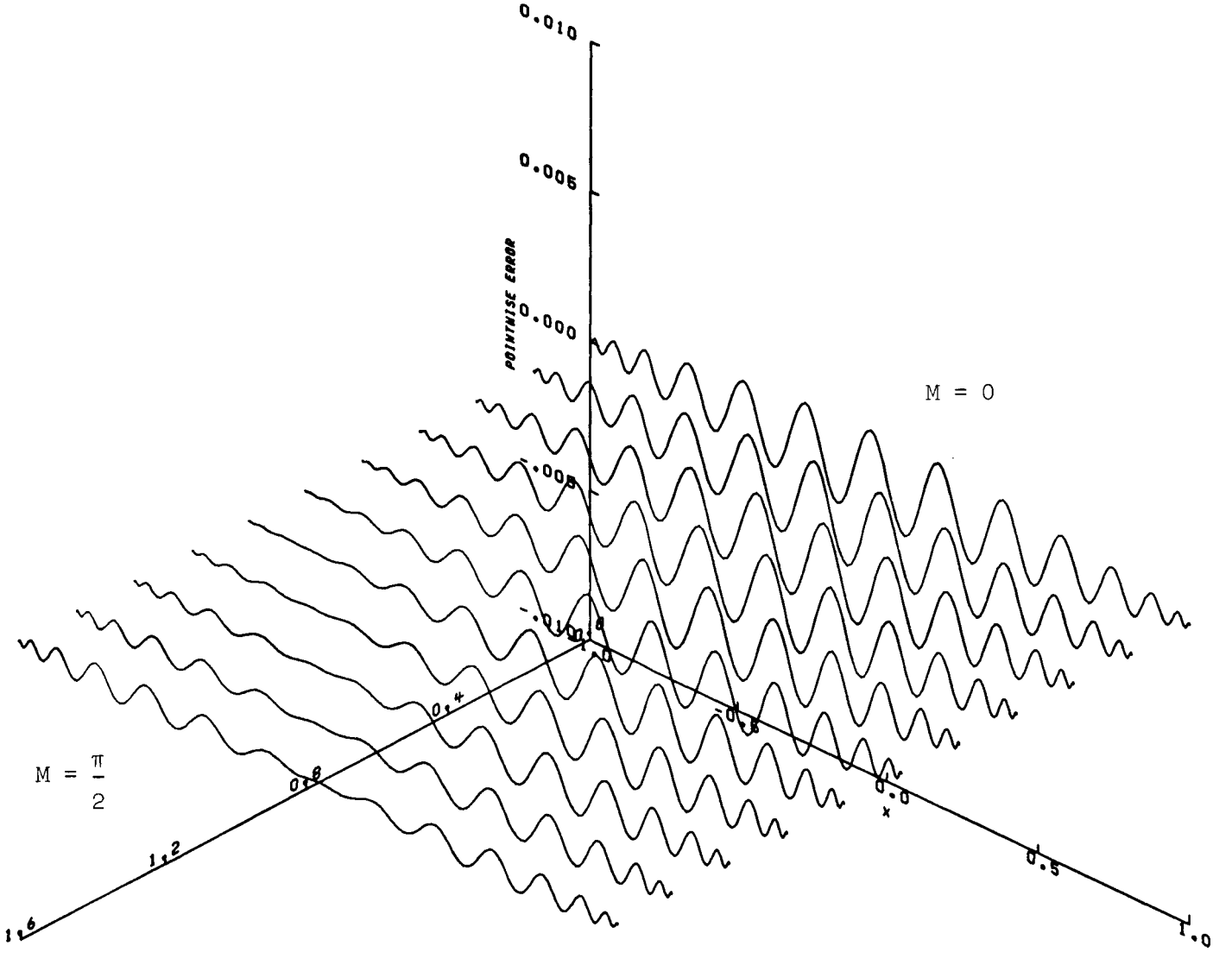

Figure 1a. Pseudospectral Chebyshev approximation to $\sin(20x - M)$, $0 < M < \frac{\pi}{2}$, with 28 nodes.

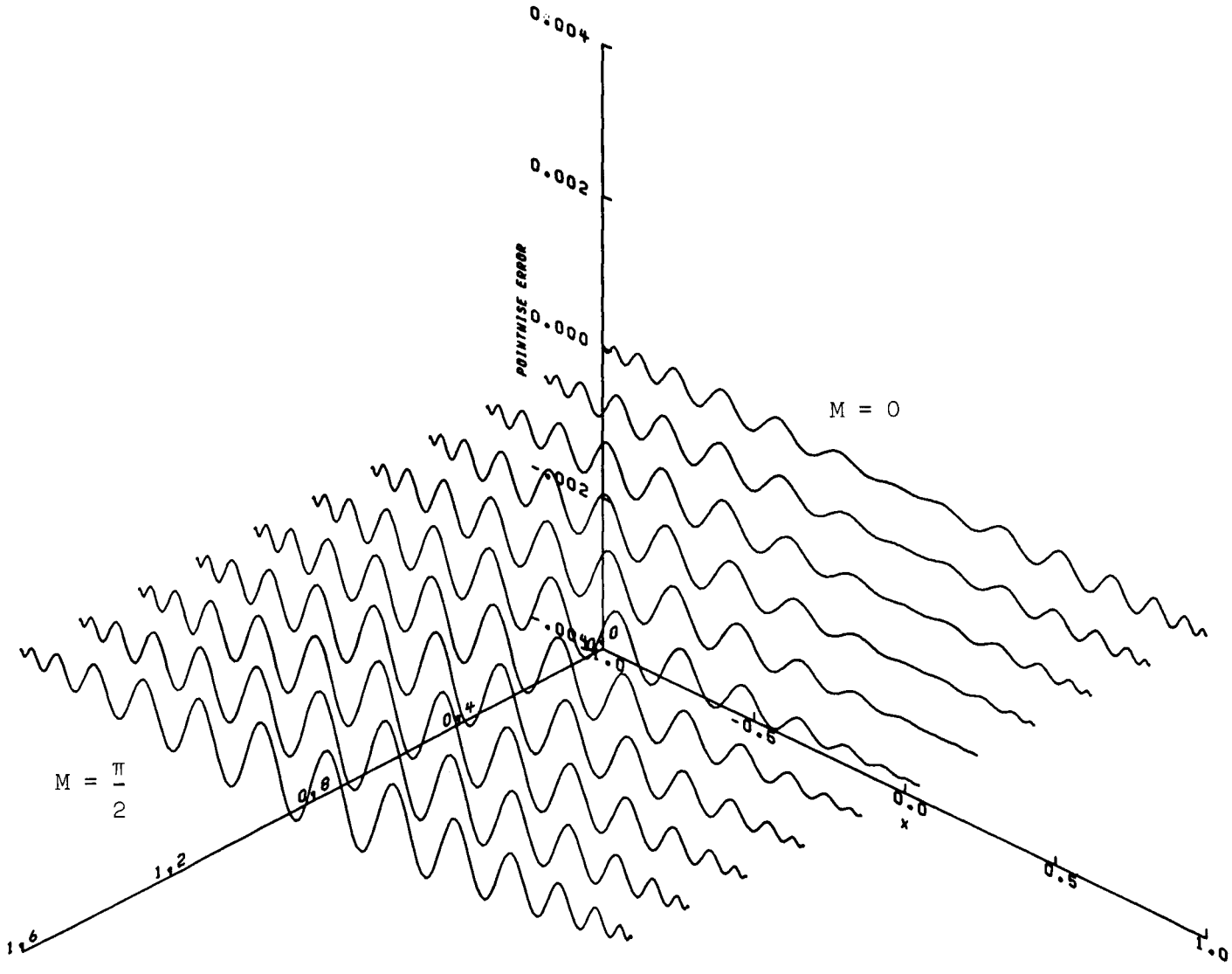

Figure 1b. Pseudospectral Chebyshev approximation to $\sin(20x - M)$, $0 < M < \frac{\pi}{2}$, with $N = 29$.

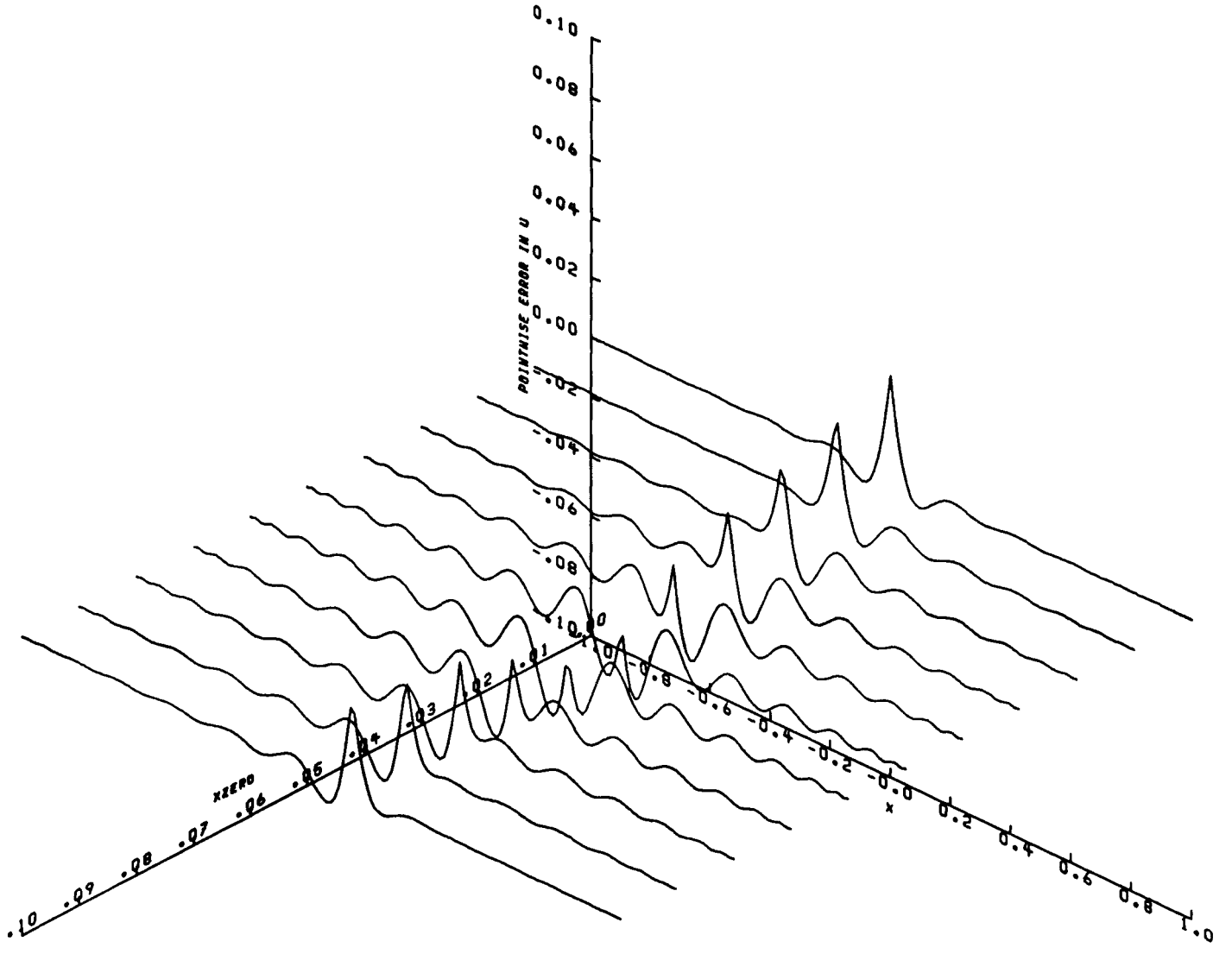

Figure 2. The error for pseudospectral approximation to $|x - x_0|$, $0.05 < x_0 < 0.05$ with 29 nodes. The error is plotted as a function of x.

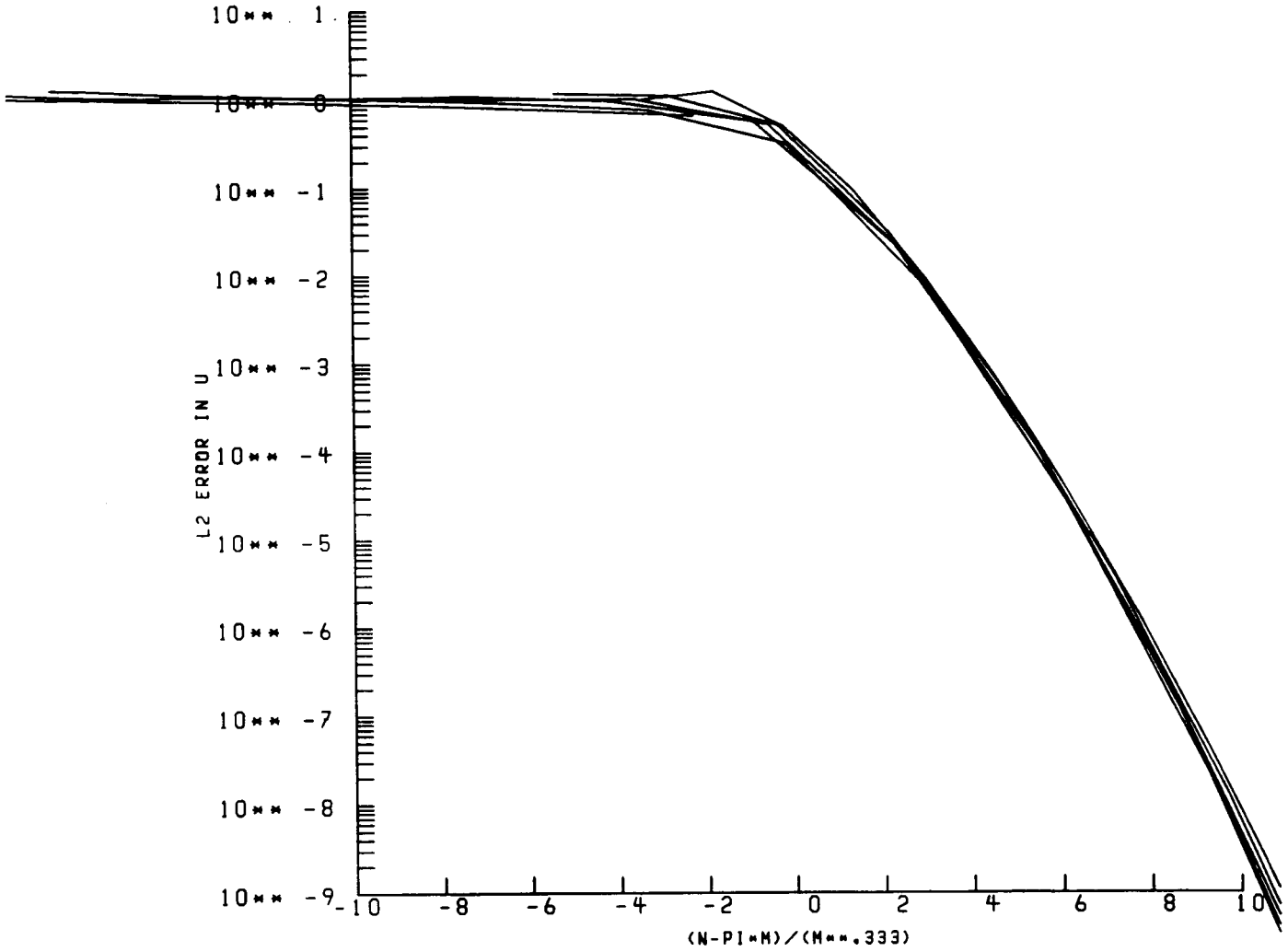

Figure 3a. Pseudospectral Chebyshev approximation to $\sin(M\pi x)$. The different graphs represent $M = 2, 4, 8, 16,$ and 32. The L^2 error is plotted as a function of $\frac{N - \pi M}{M^{1/3}}$ with several N.

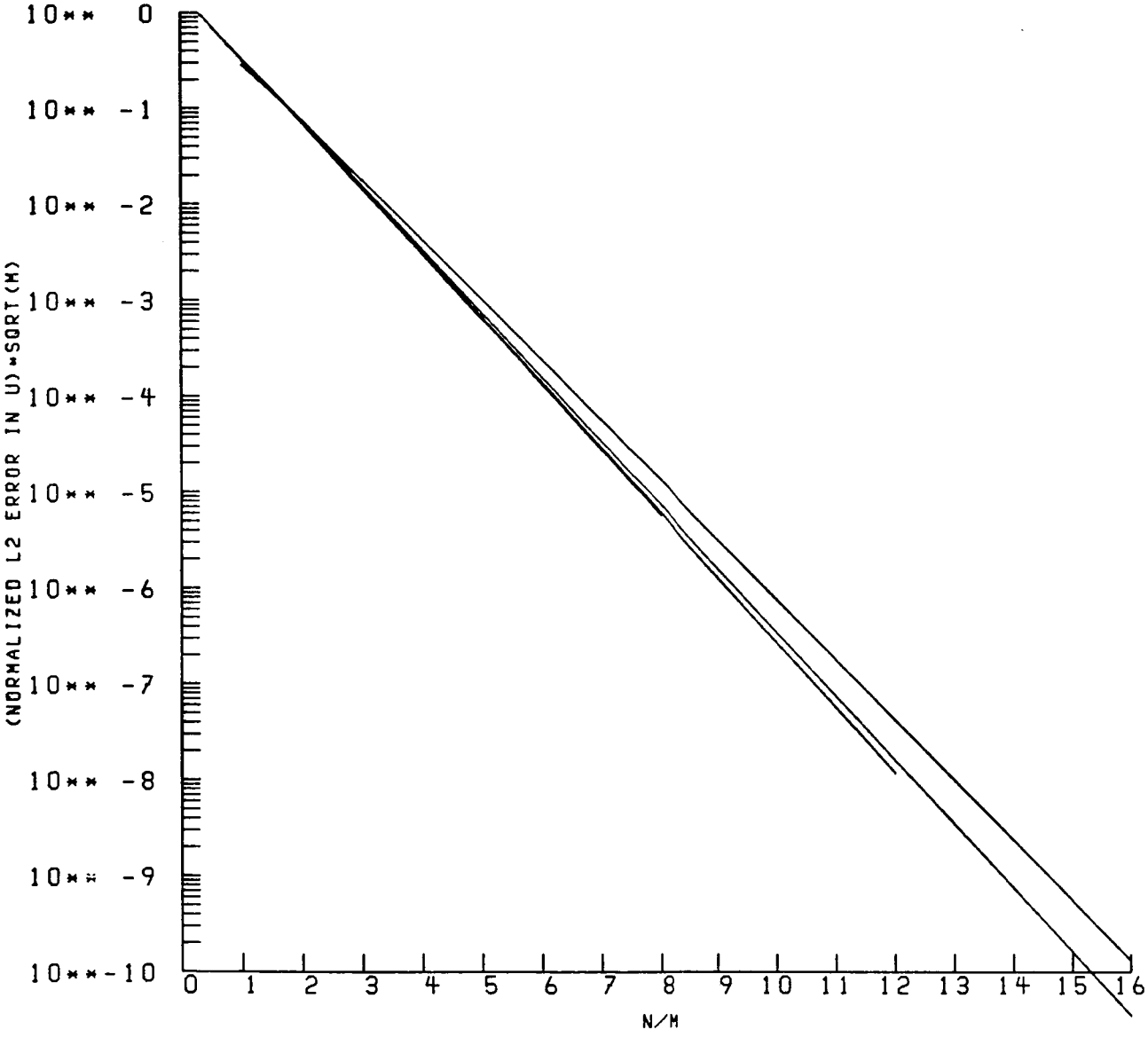

Figure 3b. Pseudospectral Chebyshev approximation to tanh(Mx) for M = 2, 8, 16, 32. We plot normalized L^2 error as a function of N/M for several M.

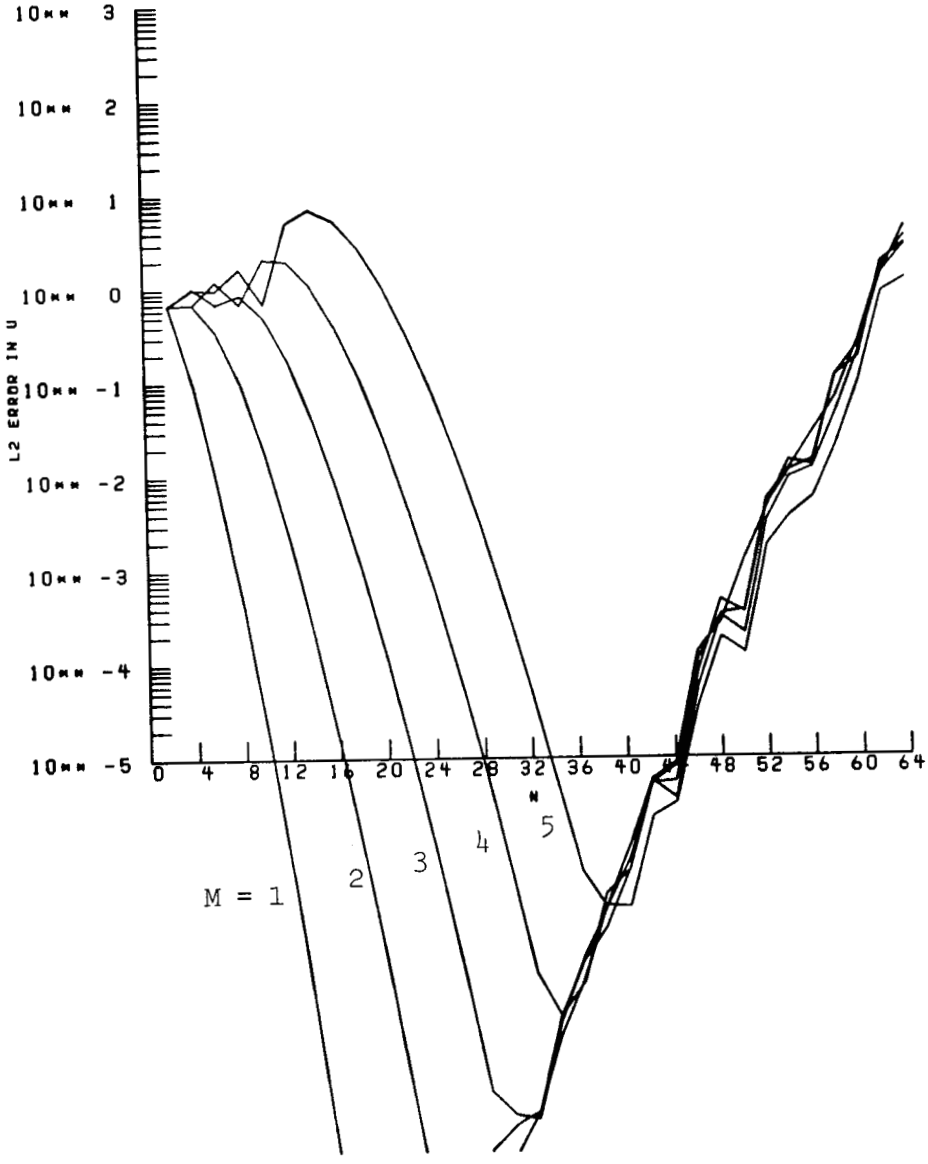

Figure 4. Collocation based on uniformly spaced points for $f(x) = \sin(M\pi x)$, $M = 1,\ldots,5$, and nodes.

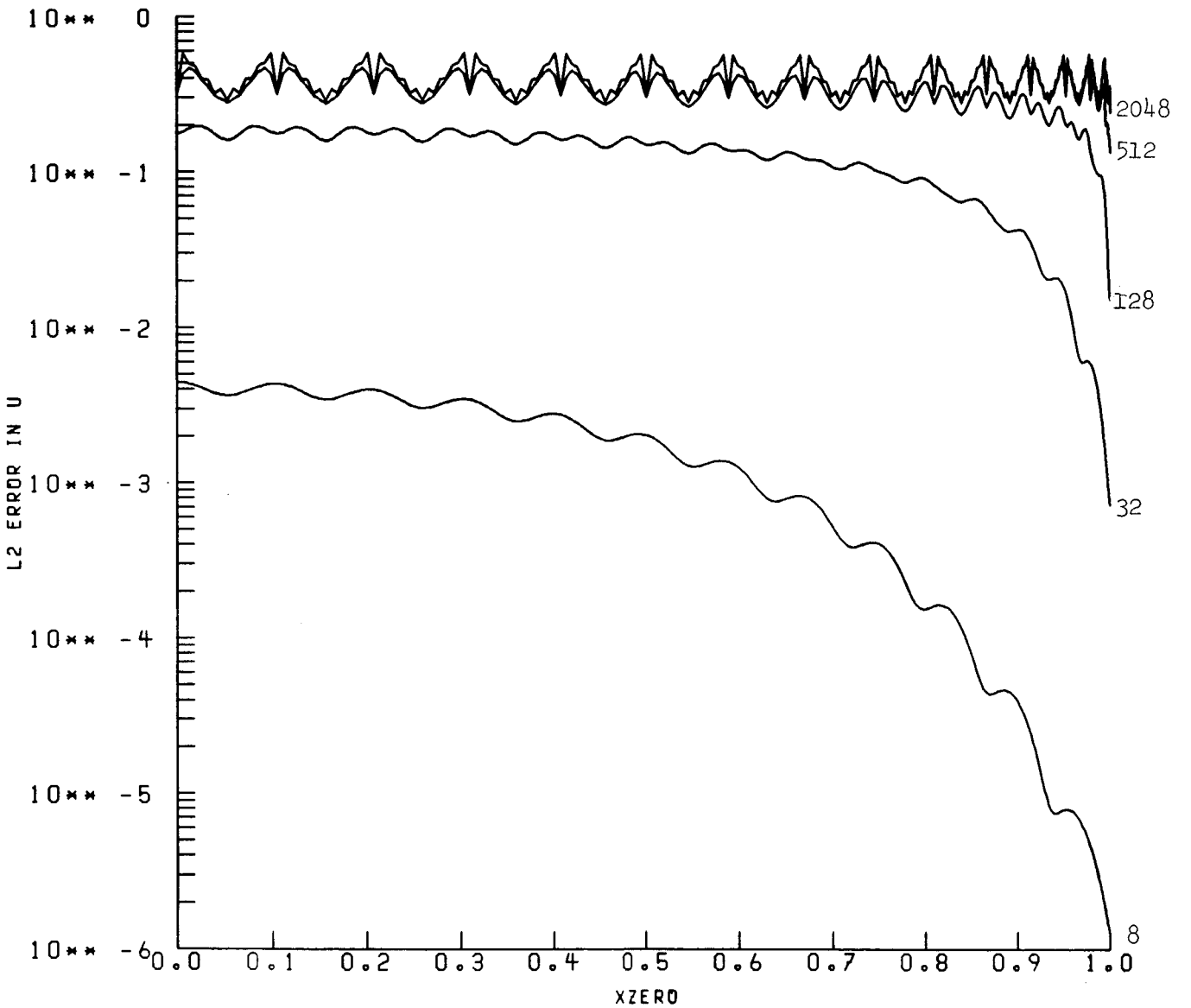

Figure 5a. Pseudospectral collocation with 31 nodes for $f(x) = \tanh(M(x - x_0))$ with $M = 8, 32, 512, 2048$. x_0 varies between the center, $x_0 = 0$, and the edge, $x_0 = 1$.

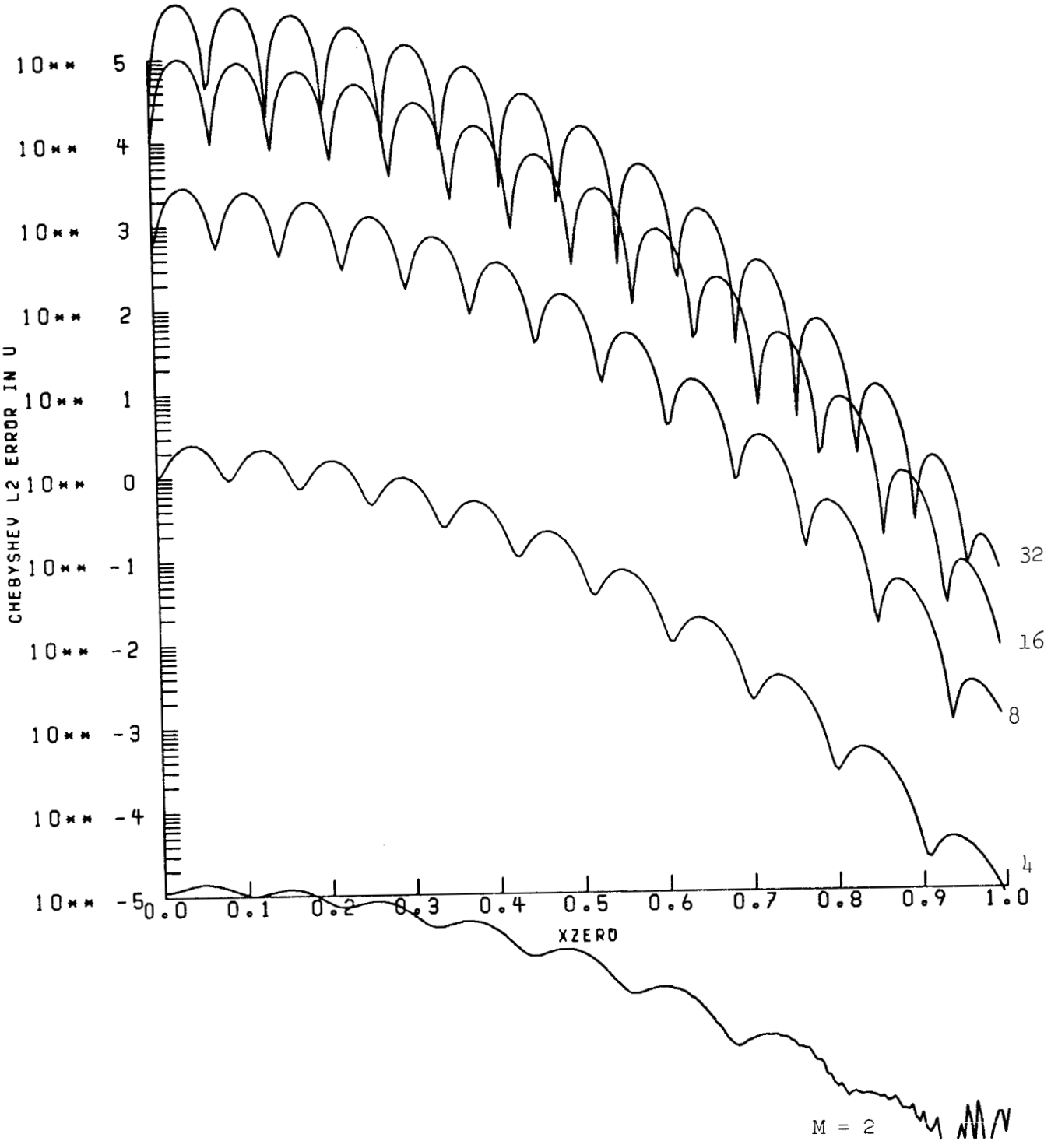

Figure 5b. Same case as Figure 5a but using uniformly spaced collocation points. Now M = 2, 4, 8, 16, and 32. The L^2 error is the same Chebyshev norm as in Figure 5a.

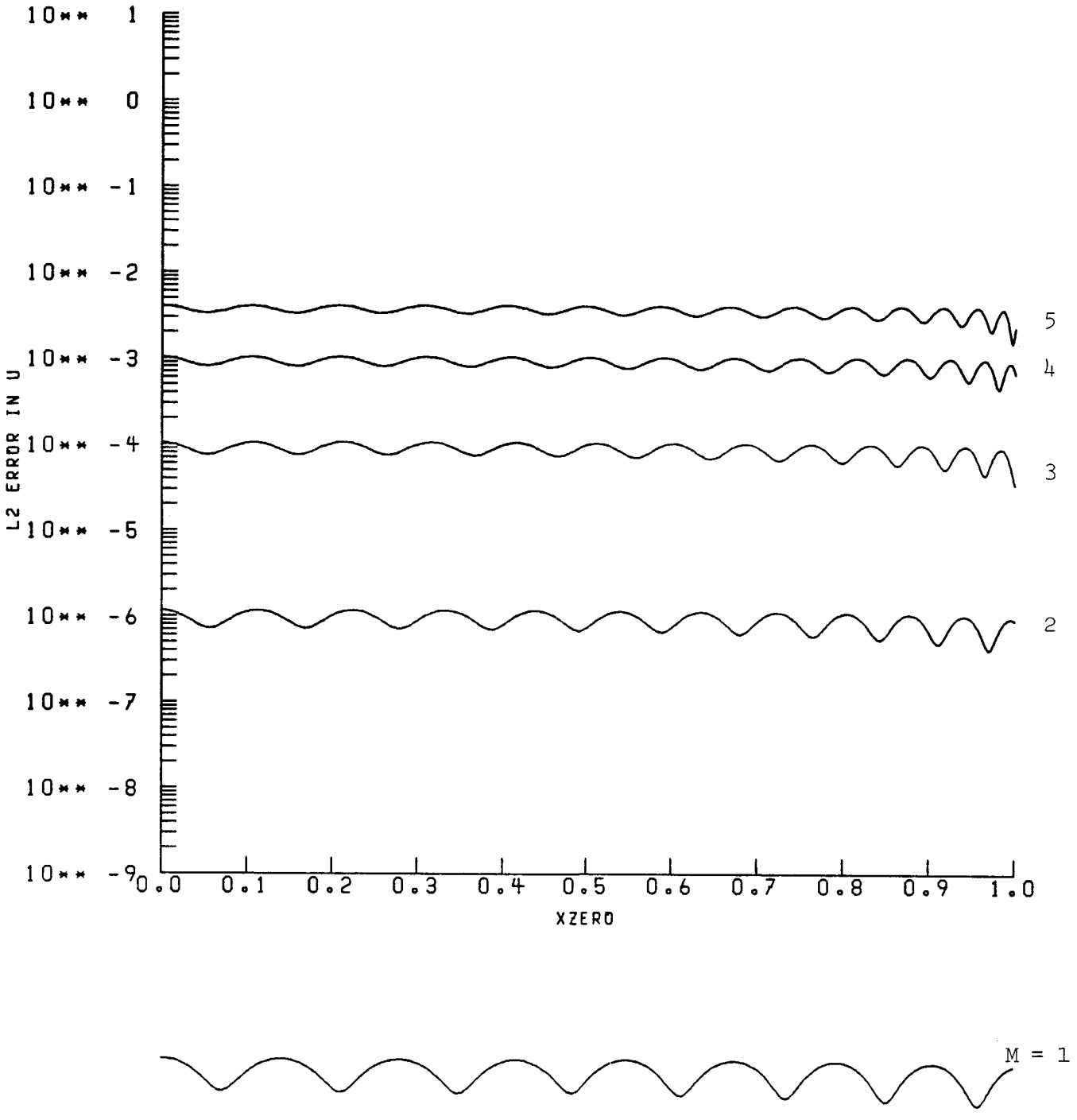

Figure 6. Same as Figure 5a for the function $f(x) = \tanh(Q(x - x_0))$,

$$Q = \frac{M\pi}{2}\sqrt{\frac{M^2 + 1}{M^2(1 - x_0^2) + 1}}, \quad M = 1,\ldots,5.$$

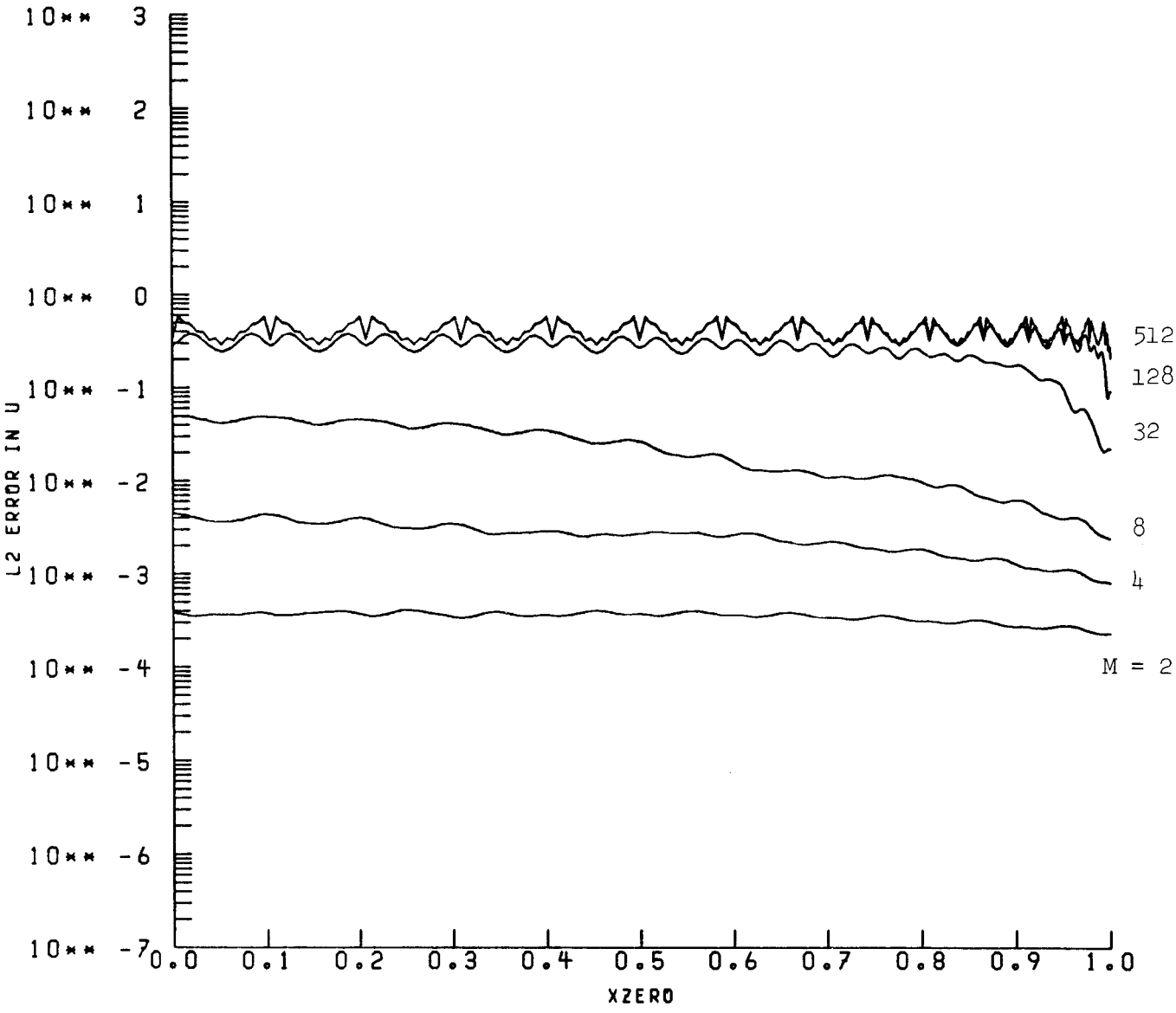

Figure 7a. Chebyshev collocation using 31 nodes for the function given in (4.1).

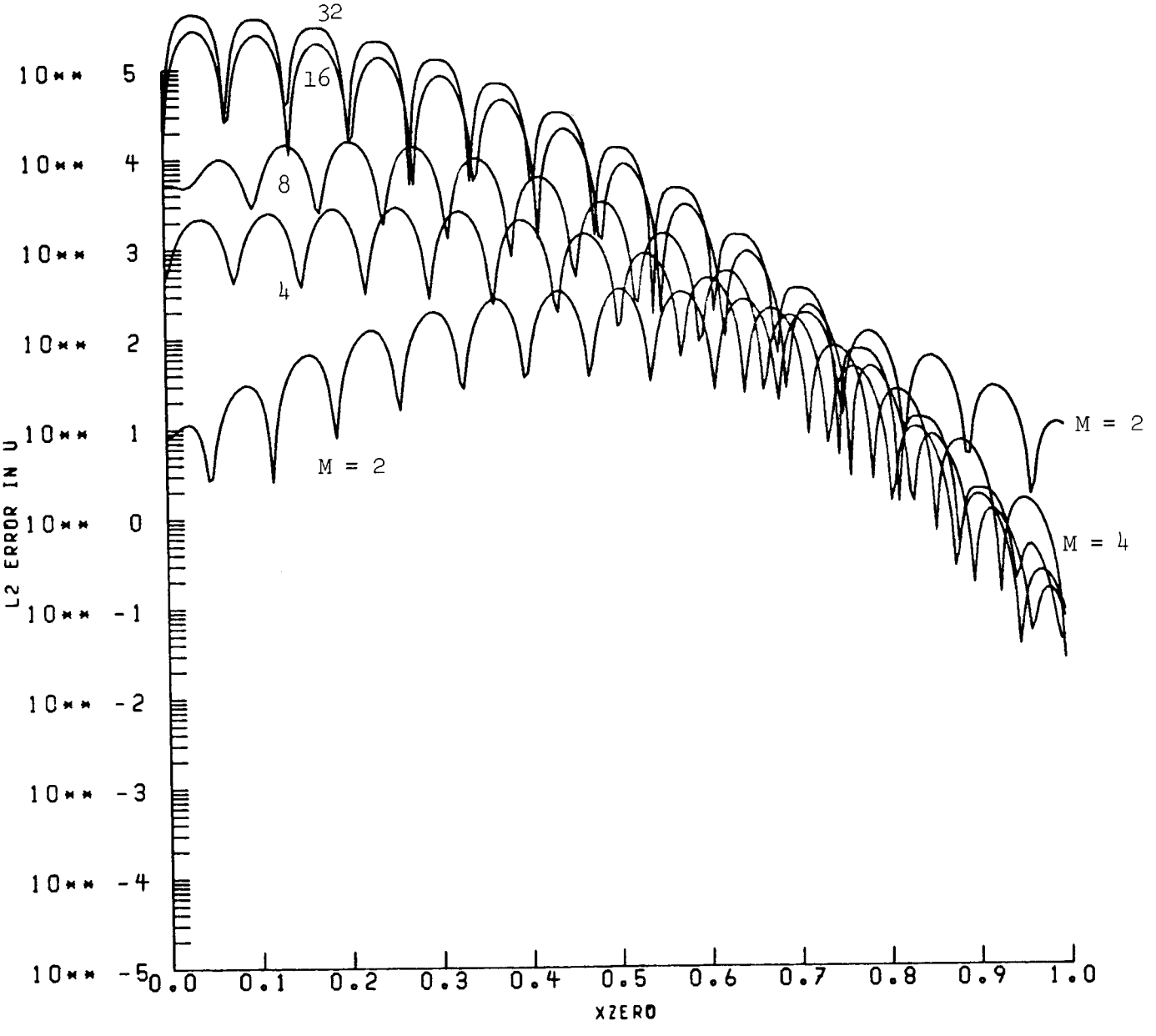

Figure 7b. Same as Figure 7a but using uniformly spaced nodes and M = 2, 4, 8, 16, 32.

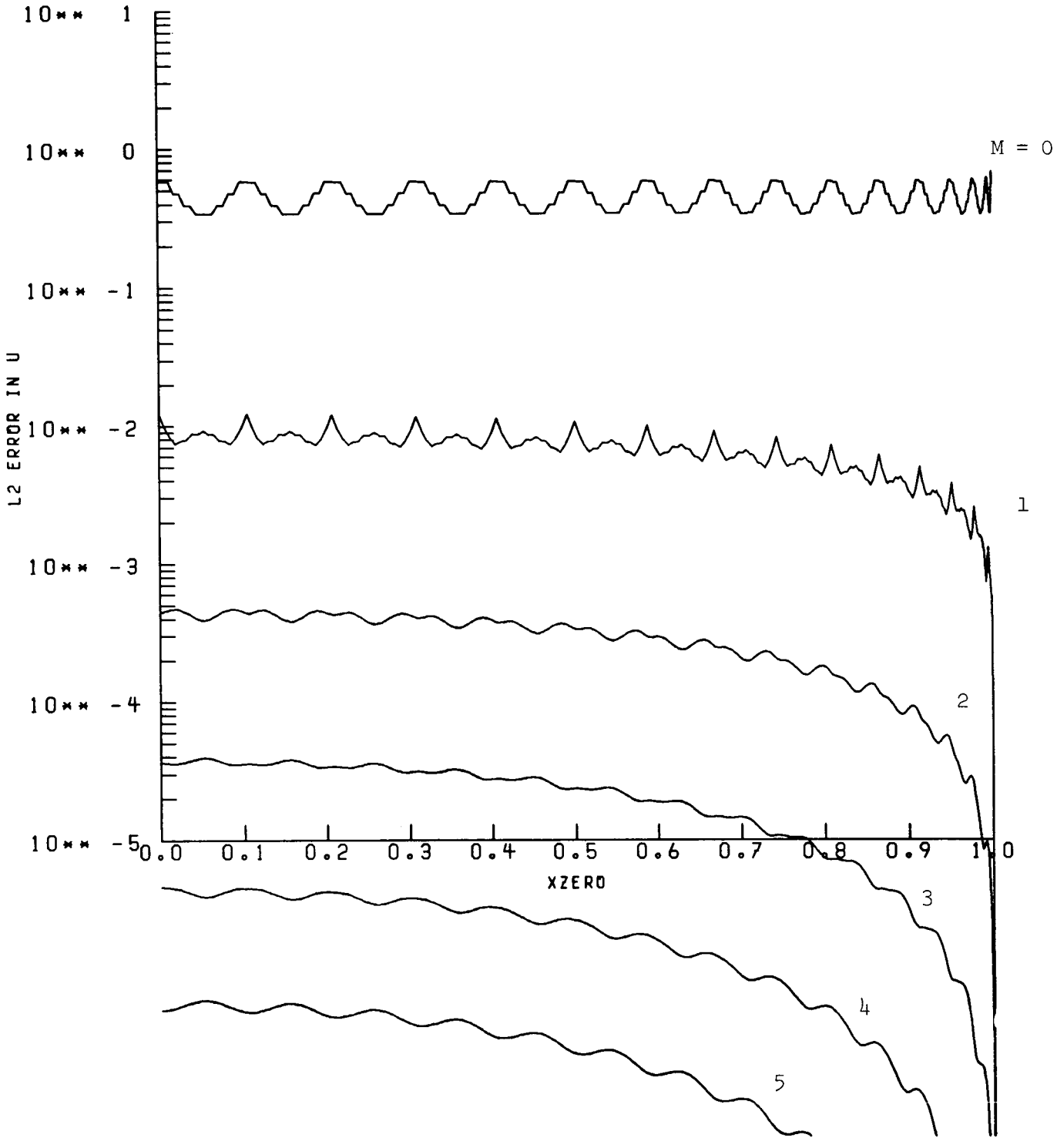

Figure 8a. Chebyshev collocation for $f(x) = \mathrm{sgn}(x - x_0)\,|x - x_0|^M$, $M = 0,\ldots,5$, using 30 nodes. We let x_0 vary between 0 and 1.

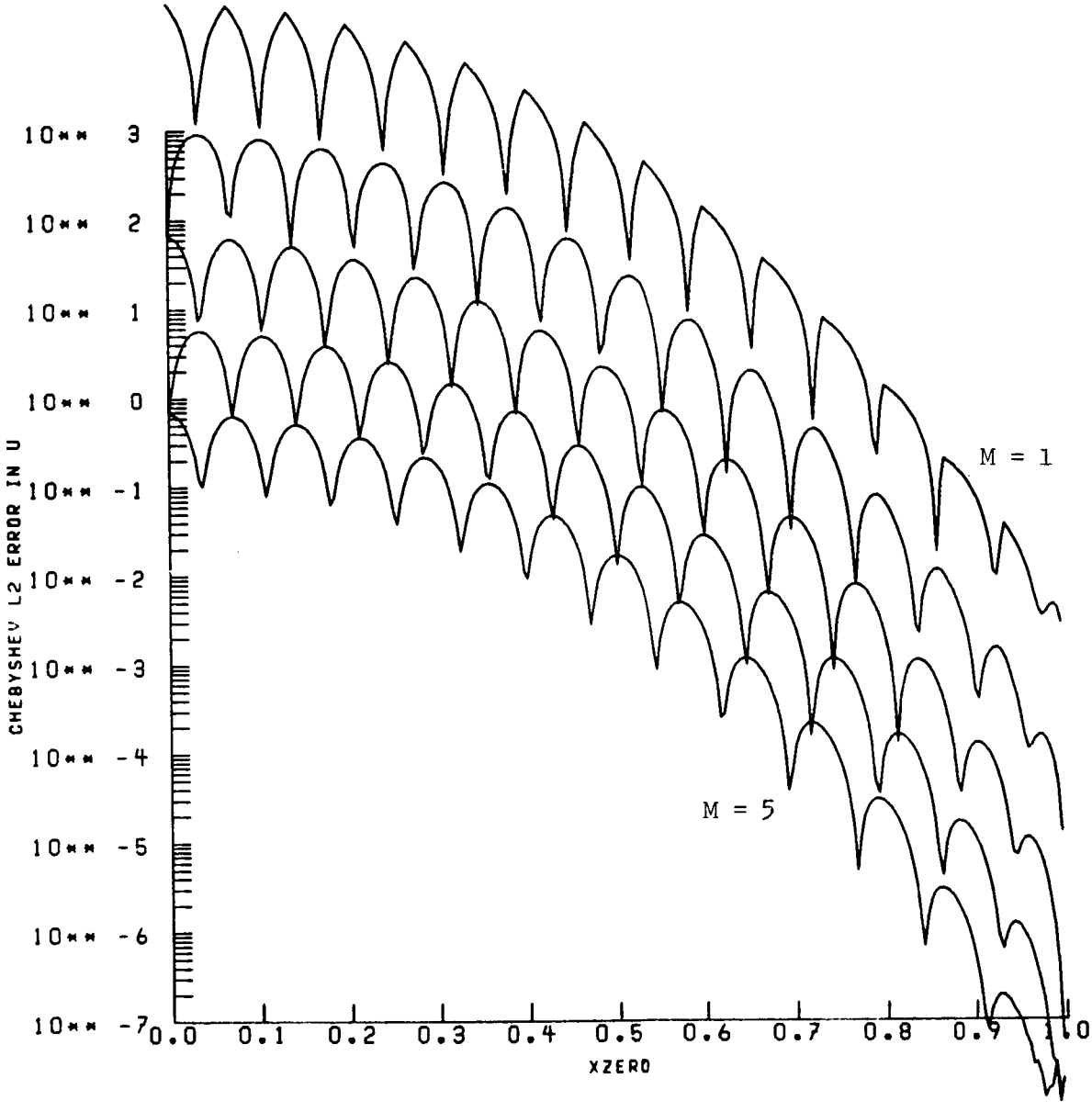

Figure 8b. Same as Figure 8a but using uniformly spaced collocation nodes but Chebyshev norm.

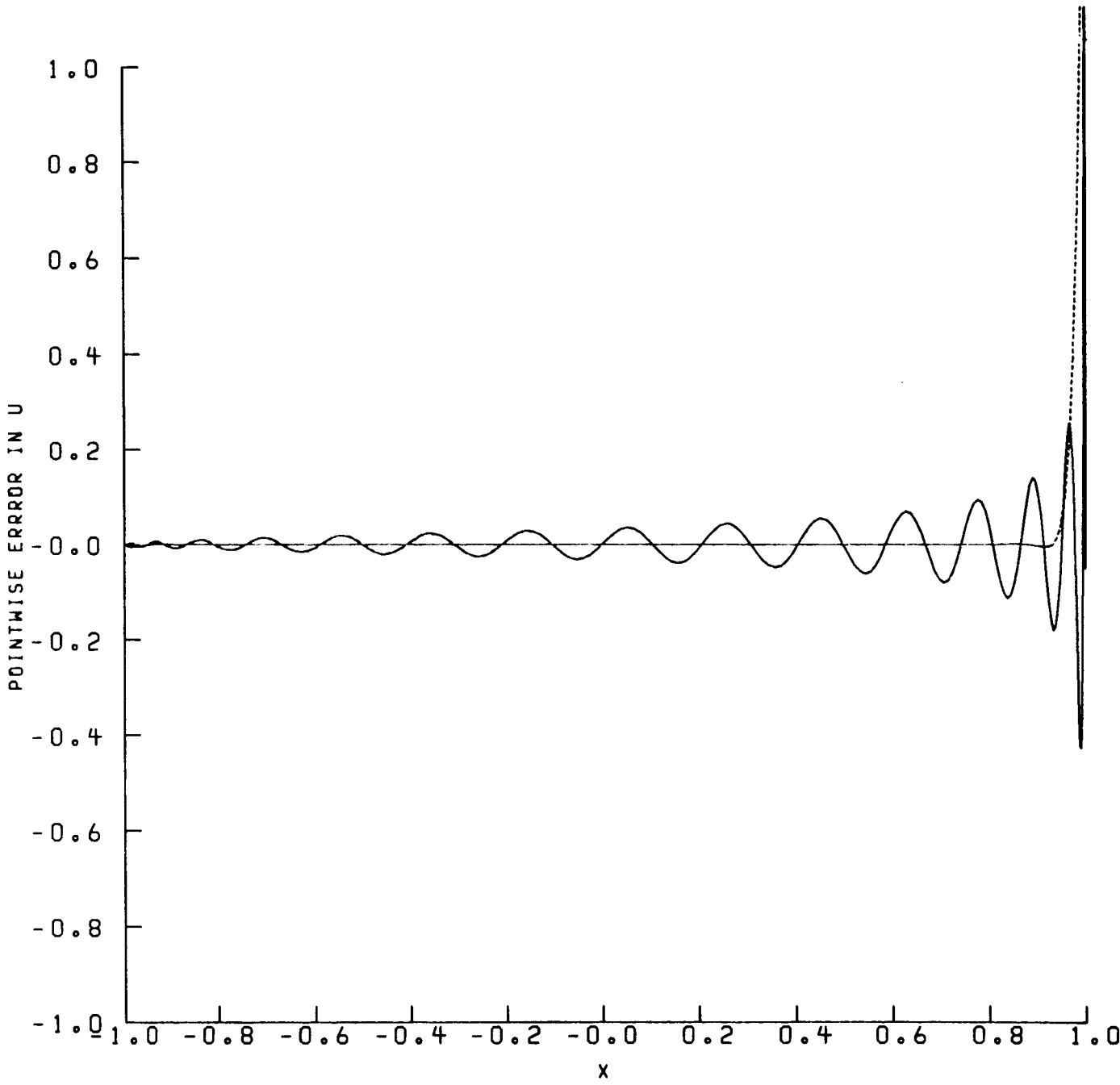

Figure 9a. We consider the function $f(x) = 1$ except at $x = 1$ when $f(x) = 0$. We graph the error for (solid line) Chebyshev nodes and (dotted line) uniformly spaced nodes, both using 31 collocation points.

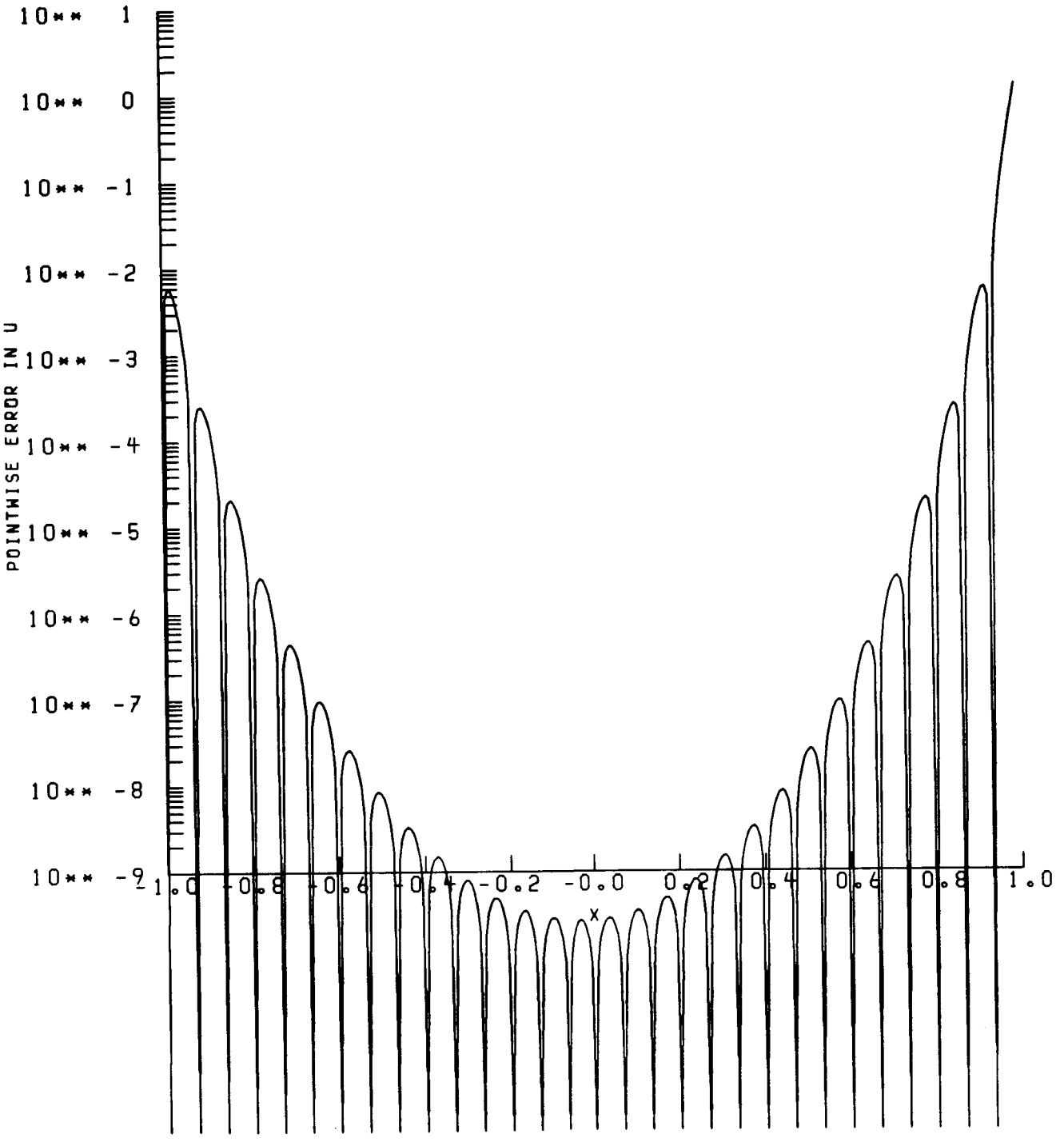

Figure 9b. The uniformly spaced nodes of Figure 9a plotted on a logarithmic scale.

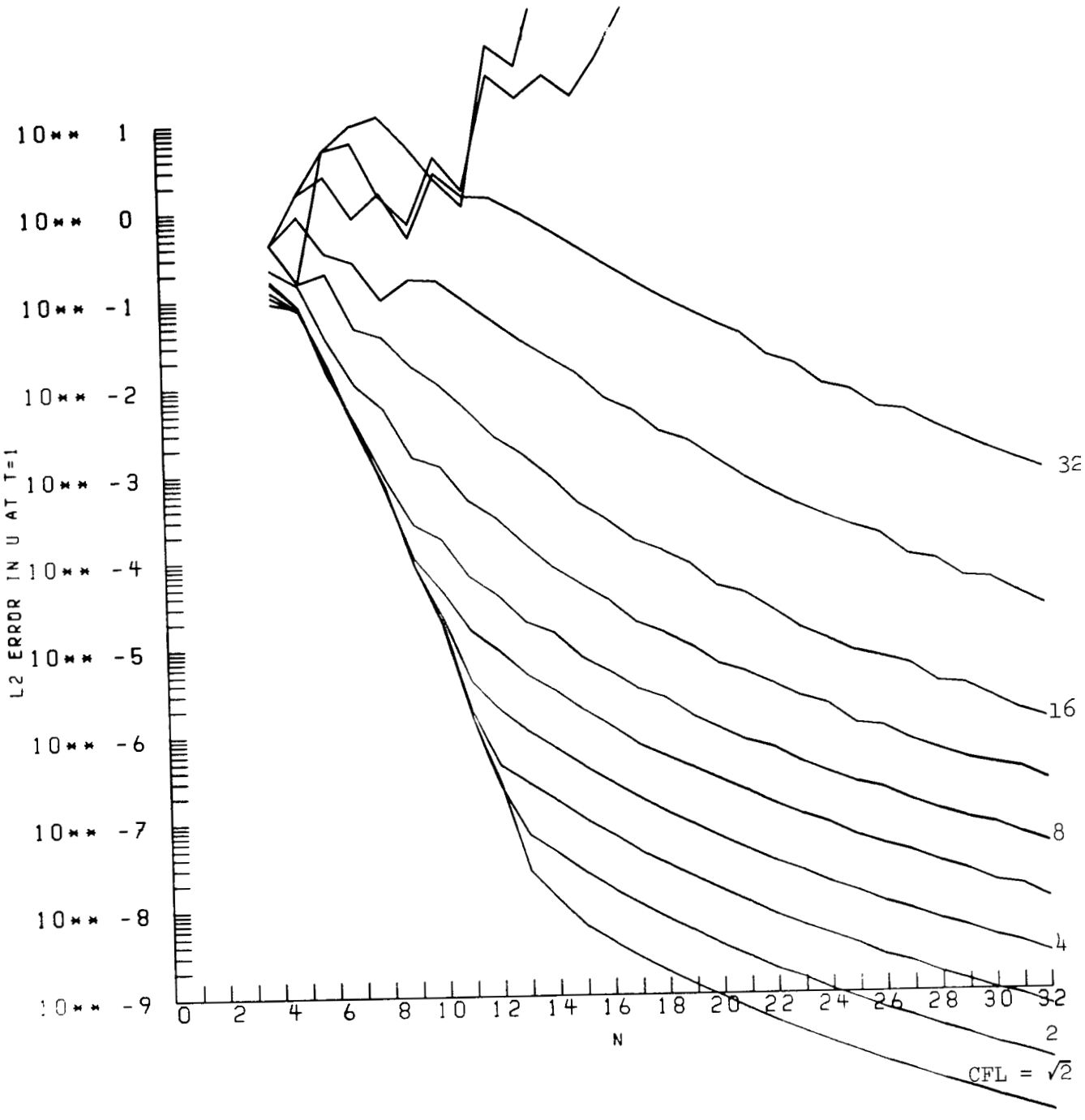

Figure 10a. Pseudospectral approximation to (4.1) with $f(x) = \sin\pi x$. A four-stage fourth-order Runge-Kutta formula is used and boundary conditions are imposed after every stage. Each graph represents a different time step, i.e., CFL number with an increase of $\sqrt{2}$ between graphs. The L^2 error at $t = 1$ is given as a function of N.

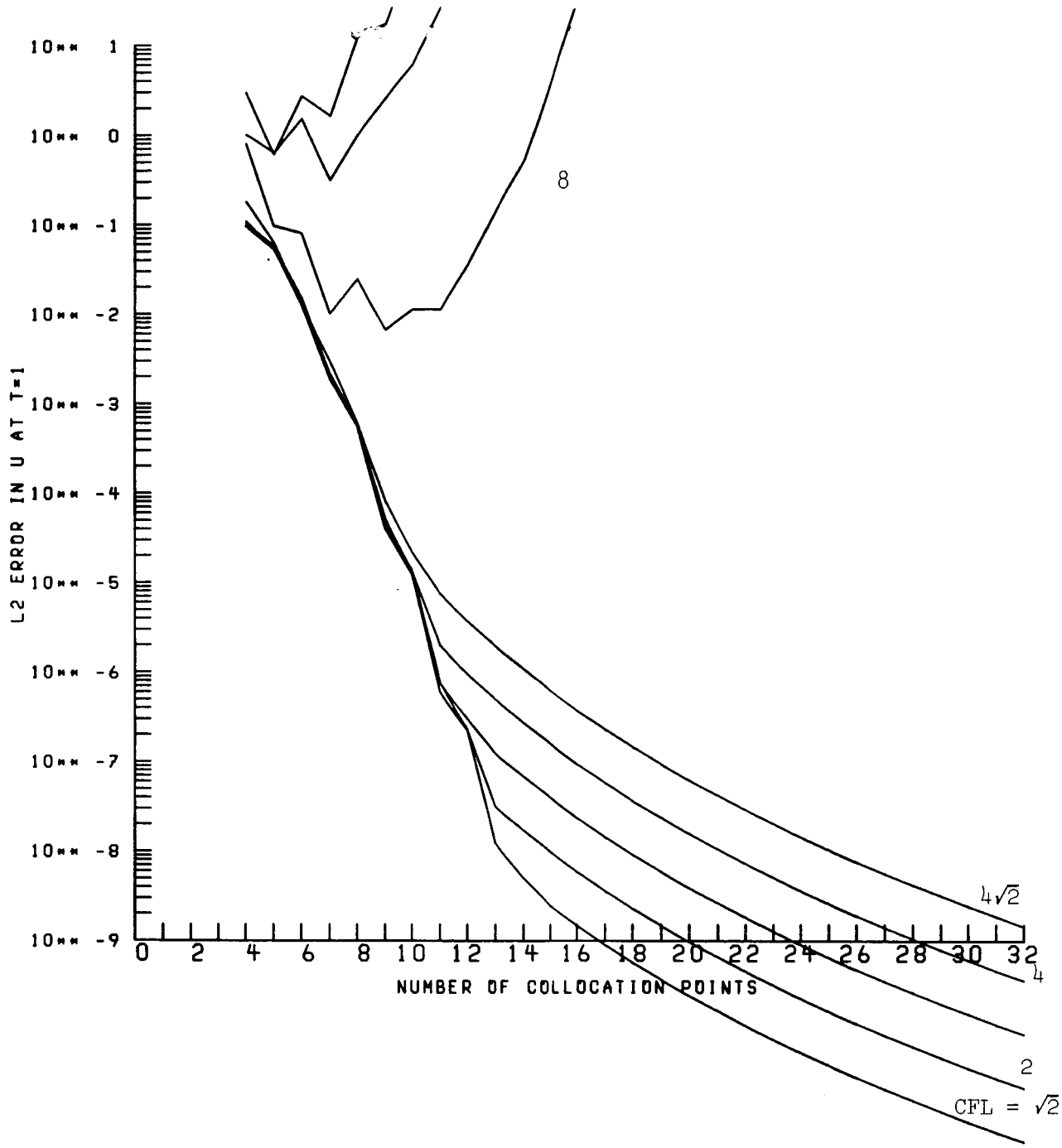

Figure 10b. Pseudospectral approximation to (4.1) with $f(x) = \sin\pi x$. A four-stage fourth-order Runge-Kutta formula is used but with the boundary condition imposed only once after the completion of the four Runge-Kutta stages.

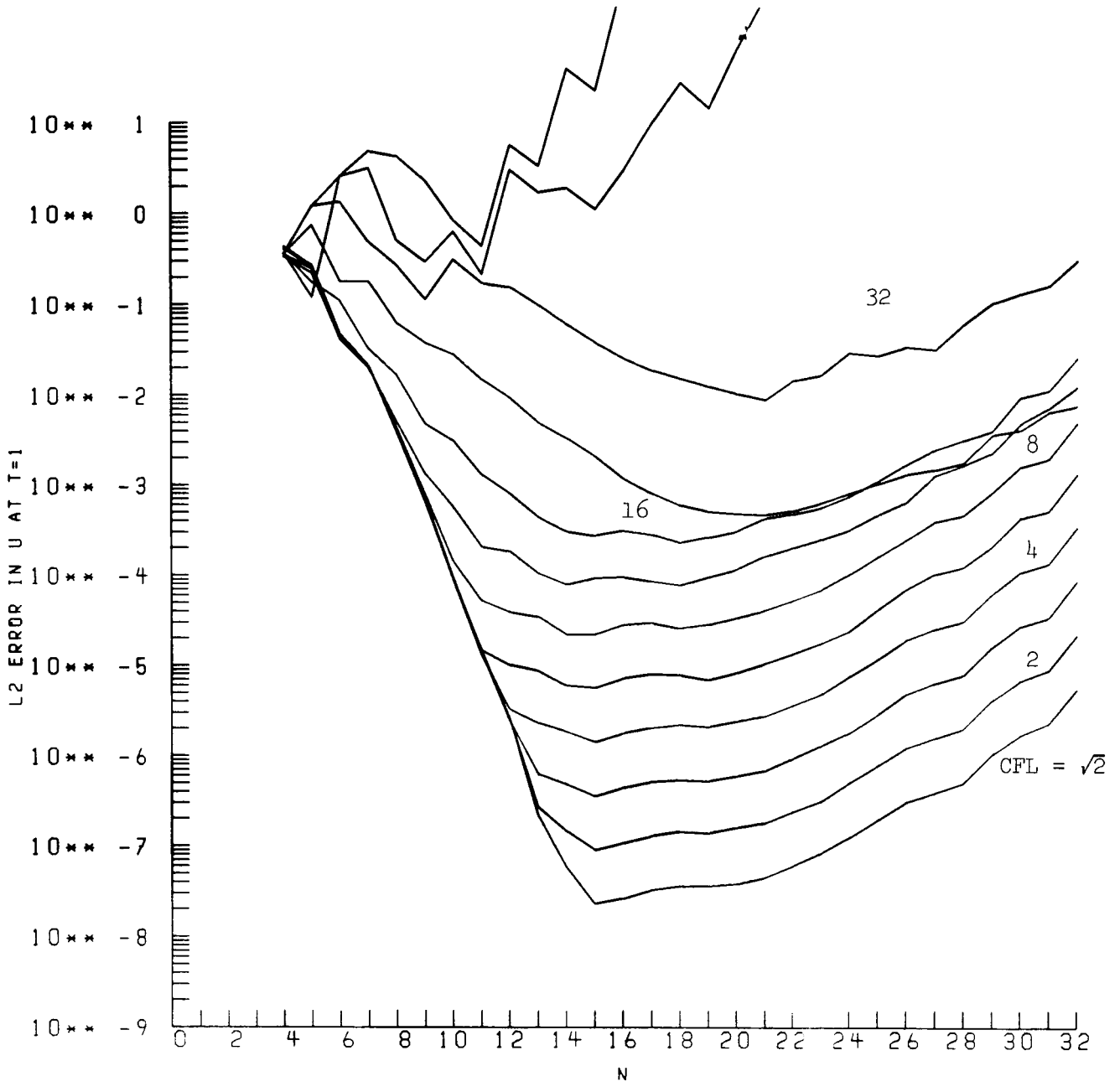

Figure 11a. Pseudospectral Chebyshev approximation to (4.1) with $f(x) = \sin\pi x$, 4 applications of the boundary conditions and $\beta = 2$, i.e., mesh is twice as coarse as a Chebyshev grid near the boundary. Each graph represents a different CFL number, increasing by a factor of $\sqrt{2}$.

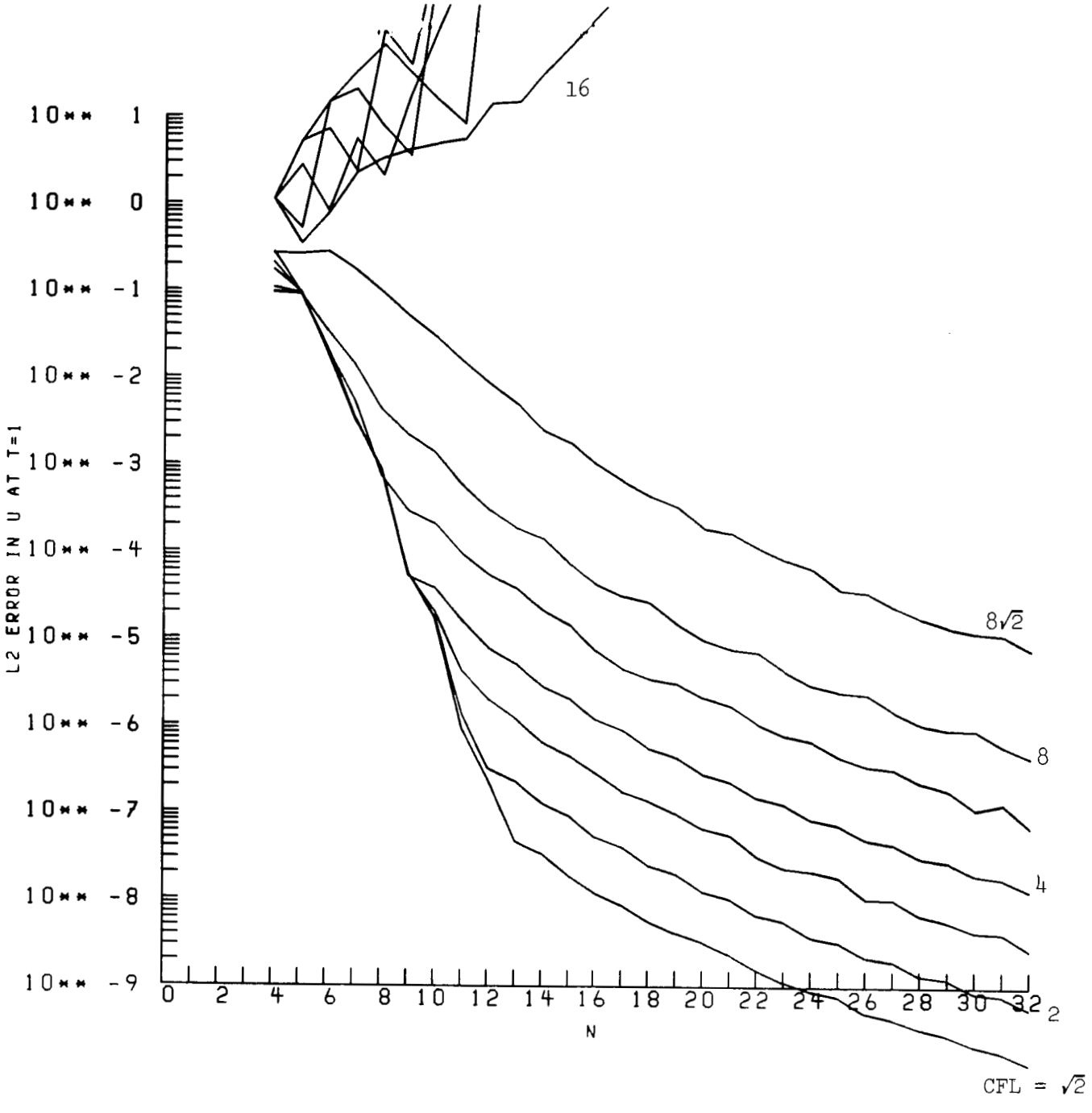

Figure 11b. Pseudospectral Chebyshev approximation to (4.1) with $f(x) = \sin\pi x$, 4 applications of the boundary conditons and $\beta = \frac{1}{2}$, i.e., twice the density of Chebyshev spacing near the boundary. Each graph represents a different CFL number, increasing by a factor of $\sqrt{2}$.

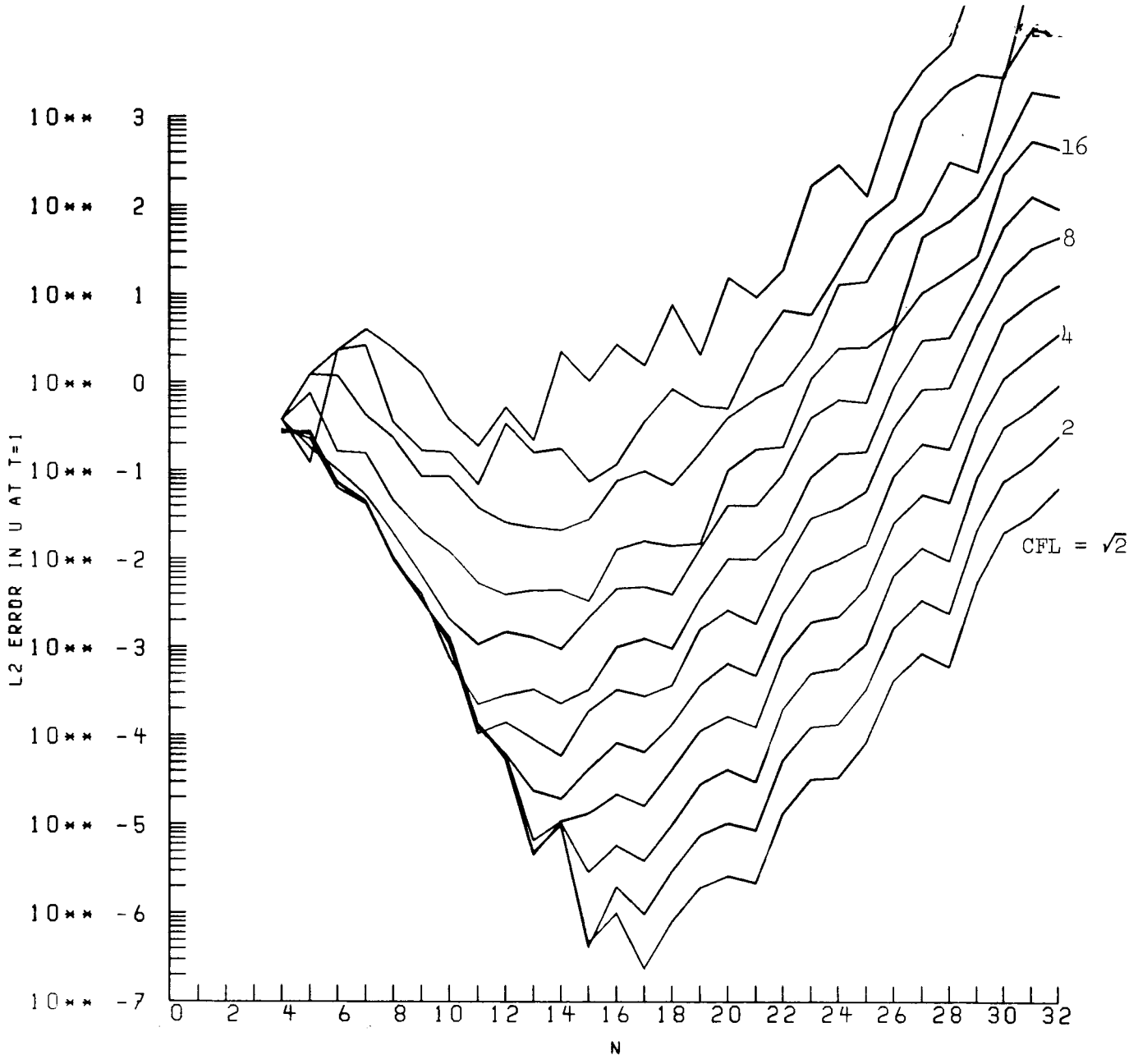

Figure 12. Same as Figure 10a but with uniformly spaced nodes. Based on double precision on the CRAY-XMP.

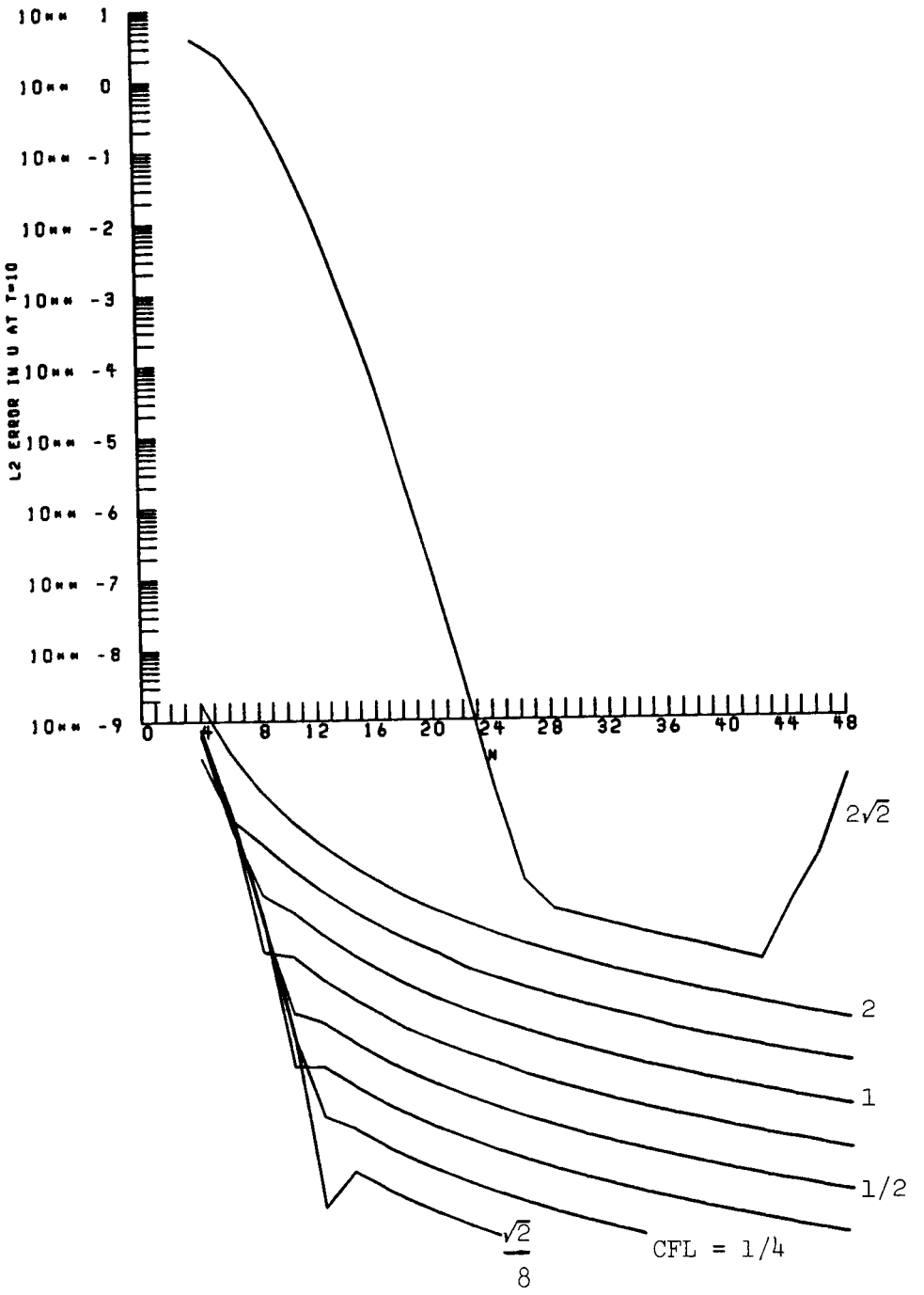

Figure 13a. Pseudospectral Chebyshev approximation to the equation $u_t = -xu_x$. Uses double precision on the CRAY-XMP, CFL = $N\Delta t$.

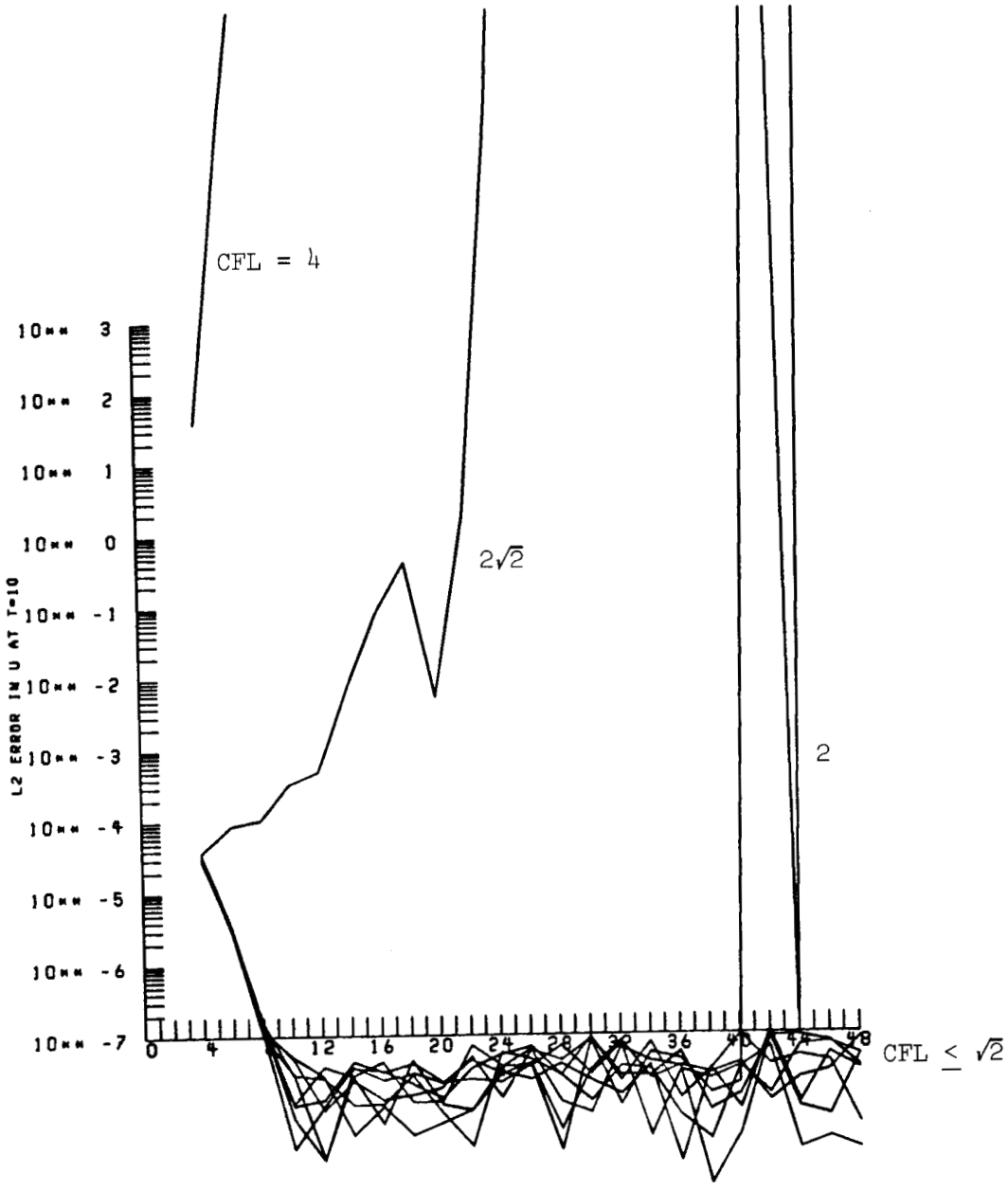

Figure 13b. Same as Figure 13a but using only 5 significant digits.

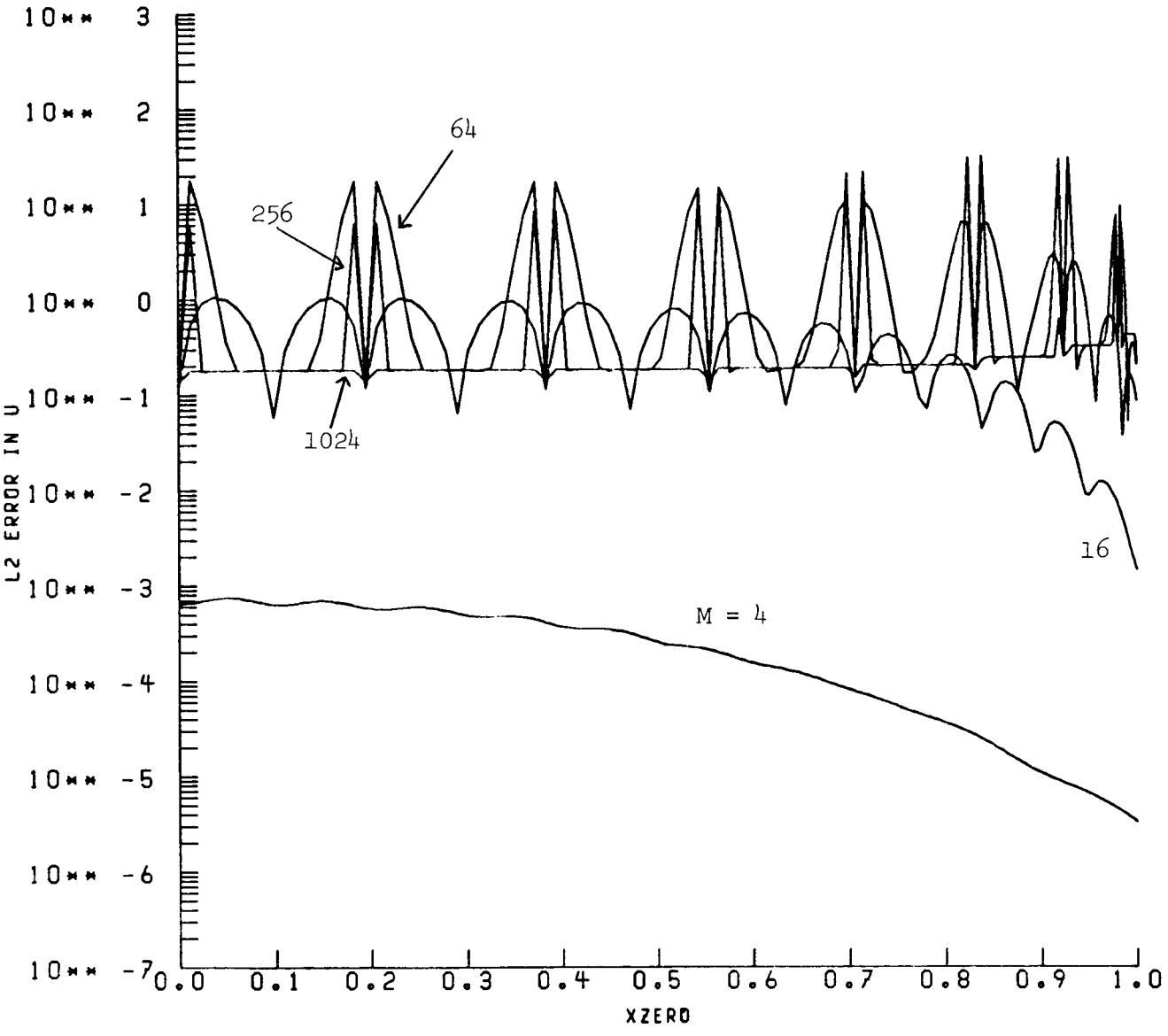

Figure 14. Pseudospectral Chebyshev approximation to the solution of a Poisson equation. The exact solution is $u(x,y) = \sin\pi y \tanh(m(x - x_0))$ with $M = 4, 16, 64, 256, 1024$ and $N = 17$ modes in each direction. We plot the L^2 Chebyshev error as a function of x_0.

Standard Bibliographic Page

1. Report No. NASA CR-178179 ICASE Report No. 86-60	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
GLOBAL COLLOCATION METHODS FOR APPROXIMATION AND THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS	September 1986
	6. Performing Organization Code
7. Author(s) A. Solomonoff and E. Turkel	8. Performing Organization Report No. 86-60
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	10. Work Unit No.
	11. Contract or Grant No. NAS1-17070, NAS1-18107
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	13. Type of Report and Period Covered Contractor Report
	14. Sponsoring Agency Code 505-90-21-01

15. Supplementary Notes

Langley Technical Monitor:
J. C. South

Submitted to Journal of Computational Physics

Final Report

16. Abstract

We apply polynomial interpolation methods both to the approximation of functions and to the numerical solutions of hyperbolic and elliptic partial differential equations. We construct the derivative matrix for a general sequence of the collocation points. The approximate derivative is then found by a matrix times vector multiply. We explore the effects of several factors on the performance of these methods including the effect of diffferent collocation points. We also study the resolution of the schemes for both smooth functions and functions with steep gradients or discontinuities in some derivative. We investigate the accuracy when the gradients occur both near the center of the region and in the vicinity of the boundary. The importance of the aliasing limit on the resolution of the approximation is investigated in detail. We also examine the effect of boundary treatment on the stability and accuracy of the scheme.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
spectral methods, approximation theory	64 - Numerical Analysis Unclassified - unlimited

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	67	A04

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1986