Computing Science Technical Report No. 52

A Tutorial on Galerkin's Method, using B-splines, for Solving Differential Equations

N. L. Schryer

September 1976

A Tutorial on Galerkin's Method, using B-splines, for Solving Differential Equations.

N. L. Schryer

Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

This note is a tutorial description of Galerkin's method, and its implementation using B-splines, for solving linear one-dimensional self-adjoint boundary value problems. The emphasis is on motivating and making clear what Galerkin's method is, what it does, what it is useful for and what must be done to produce a practical program for implementing it. The generalization of Galerkin's method to other equations, including nonlinear and non-self-adjoint equations, is discussed and motivated.

Galerkin's method with B-splines allows approximation of the solution of the equation to within $O(h^k)$, where h is the mesh spacing used and $k \geqslant 2$, the order of the B-spline, is any integer the user desires. For most problems, the "optimal" order k is between 4 and 6. This higher order rate of convergence makes Galerkin's method faster and much cheaper to use than finite differences.

An automatic and reliable error estimation scheme is presented for use with Galerkin's method using B-splines. Several sample problems are then solved and the numerical results discussed.

September 17, 1976

A Tutorial on Galerkin's Method, using B-splines, for Solving Differential Equations.

N. L. Schryer

Bell Laboratories
Murray Hill, New Jersey 07974

1. Introduction.

This note is a tutorial description of Galerkin's method, and its implementation using B-splines, for solving linear one-dimensional self-adjoint (divergence form) boundary value problems. The emphasis is on motivating and making clear what Galerkin's method is, what it does, what it is useful for and what must be done to produce a practical program for implementing it. The generalization of Galerkin's method to other equations, including nonlinear and non-self-adjoint equations, is discussed and motivated. A detailed mathematical analysis of Galerkin's method, which this paper tries to avoid as much as possible, is available in [23]. Anyone interested in the theoretical underpinnings of Galerkin's method is urged to browse through that excellent text.

The equation studied is

$$(a(x)y'(x))' + b(x)y(x) + c(x) = 0 \text{ on } (L,R) \tag{1.1}$$

subject to boundary conditions

$$\begin{aligned} \alpha_L y(L) + \beta_L y'(L) &= \gamma_L \\ \alpha_R y(R) + \beta_R y'(R) &= \gamma_R \end{aligned} \tag{1.2}$$

where $a(x) > 0$, $b(x) \leqslant 0$, and $c(x)$ are given functions on (L,R), $\alpha_L, \alpha_R, \beta_L, \beta_R, \gamma_L$, and γ_R are given constants obeying

$$\begin{aligned} &\alpha_L \beta_L \leqslant 0 \text{ and } (\alpha_L, \beta_L) \neq (0,0) \\ &\alpha_R \beta_R \geqslant 0 \text{ and } (\alpha_R, \beta_R) \neq (0,0). \end{aligned} \tag{1.3}$$

Under reasonable assumptions about the smoothness of the coefficients *a*, *b* and *c*, the solution of (1.1) subject to (1.2) and (1.3) is known to exist and be unique. If (1.3) is violated, the solution may not exist, and even if it does, it may not be unique.

Problems of this type arise in many situations [16,19,20] and their solution must be accomplished cheaply, accurately and reliably. The numerical solution of (1.1) has traditionally been accomplished using classical finite differences [14]. If the mesh spacing is *h*, then the error in the finite difference approximation to the solution is $O(h^2)$. The popularity of finite difference methods resulted mainly from their flexibility and ease of implementation, as well as the fact that there was no known real alternative method for practical problems.

However, there is now an alternative method to finite differences - Galerkin's method using B-splines. It is easy to implement (although not as easy as most finite difference techniques), extremely flexible and efficient, and very robust.

Galerkin's method [2,3,12,21,23] is a technique for finding the "best" approximate solution of (1.1) in any space of functions the user specifies. Galerkin's method basically finds the projection of the true solution onto the space given to it. The space of piecewise polynomials given by B-splines [6,7,9] can be used to approximate almost any function very accurately. Thus, the use of Galerkin's method to find the best approximate solution of (1.1) over the space of B-splines is a very robust and accurate numerical solution technique.

In particular, Galerkin's method with B-splines of order $k \geqslant 2$ allows approximation of the solution of (1.1) to within $O(h^k)$, where h is the mesh spacing used. For most problems, the "optimal" order k (the one which minimizes the cost of solving the problem) is typically between 4 and 6. These higher order methods are faster and much cheaper to use than finite differences.

Throughout this paper the size of a function f over an interval $[L,R]$ is measured by the maximum norm

$$||f|| \equiv \underset{x \in [L,R]}{Max} |f(x)| \tag{1.4}$$

and the size of a vector $v = (v_1, \cdots, v_N)$ is measured by the maximum norm

$$||v|| \equiv \underset{i=1,\ldots,N}{Max} |v_i| \tag{1.5}$$

These are the *only* norms we shall use or define for functions and vectors.

Section 2 discusses the definition and properties of B-splines. Section 3 discusses the Rayleigh-Ritz method and shows that it converges. Section 4 defines and motivates Galerkin's method. Section 5 shows how Galerkin's method is actually implemented on a digital computer. Section 6 describes a robust and reliable technique for estimating the error in the Galerkin solution, so the user of Galerkin's method can tell how good an answer has been obtained. Finally, section 7 presents several numerical examples.

2. B-splines

The way in which the approximate numerical solution of (1.1) is to be represented is a very important decision. The choice of representation affects the entire solution process. Specifically, we would like to choose a space of functions, out of which we will try to obtain the element closest to the solution of (1.1). This space should have several nice properties, including being (1) easy to work with and (2) capable of approximating the solution accurately.

Such a representation exists - expansion in B-splines of order k [6,7,9]. This is a method for representing functions by piecewise polynomials, that is, polynomials of degree $k-1$ or less over each sub-interval of a mesh or grid. Here the integer k is any number $k \geqslant 2$ the user desires. The piecewise polynomial representation is required to satisfy certain continuity restrictions at the end points of each mesh sub-interval. Specifically, let $\pi = \{x_1, \cdots, x_N\}$, where $L = x_1 \leqslant x_2 \leqslant \cdots \leqslant x_N = R$, be a **grid** on the interval (L,R). Let m_i be the **multiplicity** of x_i, or the number of times x_i appears in the list π. The space of B-splines of **order** k defined on the mesh π is defined to be the collection of all functions f

(2.1) which are polynomials of degree $< k$ on each interval (x_i, x_{i+1}) for $i=1, \ldots, N-1$,

(2.2) for which $d^{k-1-m_i} f(x_i) / dx^{k-1-m_i}$ exists and is continuous at each x_i, for $i=1, \ldots, N$, when viewed as a function defined only on $[L,R]$, and

(2.3) for which $f \equiv 0$ outside $[L,R]$.

The multiplicity m_i of a point x_i is restricted to be in the range $1 \leqslant m_i \leqslant k$. For $m_i = 1$ we have $d^{k-2} f / dx^{k-2}$ continuous at x_i. This is the most continuity which can be imposed at x_i without making f a polynomial of degree $k-1$ on (x_{i-1}, x_{i+1}). For $m_i = k$ the condition that $d^{-1} f / dx^{-1}$ be continuous is interpreted to mean that f is continuous from the right (but not necessarily from the left) at $x = x_i$, for $x_i < R$, and continuous from the left if $x_i = R$. This means that B-splines are continuous at the end points of the mesh when viewed as functions defined only on $[L,R]$. This collection of functions is denoted by $B_{\pi,k}$. These $B_{\pi,k}$ spaces have rather nice approximation properties, as summed up by deBoor [6], in the case when $m_1 = k = m_N$:

Theorem 2.1

Let f be any function with $f^{(0)}$ through $f^{(k)}$ continuous on $[L,R]$, where $f^{(j)}$ denotes the j^{th} derivative of f. Let $h = |\pi| \equiv \underset{i=1,\ldots,N-1}{Max} |x_{i+1} - x_i|$ be the largest mesh interval length. Then there is an element g of $B_{\pi,k}$ so that

$$|| f^{(j)}(x) - g^{(j)}(x) || \leqslant C(k,f)\ h^{k-j}$$

for $0 \leqslant j \leqslant k$, where $C(k,f)$ represents a constant which depends only upon k and f, but not h.

That is, as $h \rightarrow 0$, the error in the best B-spline approximation to f goes to zero like h^k; the error in its derivative behaves like h^{k-1}; etc.

Note that this theorem makes no assumption about the relative spacing of the mesh points of π in order to get $O(h^k)$ error. However, finite difference methods require a uniform mesh to achieve an error of $O(h^2)$. Any deviation from a uniform mesh results in $O(h)$ error for such methods. In many problems, the ability to grade the mesh with B-splines and still get $O(h^k)$ error is a decided advantage.

In practice, k is usually taken to be 4, 6, 8 or even 10, depending on what the function f looks like and how much accuracy is desired. k is usually taken to be even due to the rather natural way in which such splines arise and their smoothing properties when used to approximate functions described by discrete data [5]. Typically, the more accuracy desired, the larger the value of k should be. For example, if $k=8$ and the mesh length h is halved, then Theorem 2.1 indicates that the error should decrease by a factor of $2^8 = 256$. However, as we shall see, the work needed to solve a problem using B-splines is $O(Nk^3)$. Thus, a $k=8$ solution will cost 8 times as much as a $k=4$ solution for the **same** mesh. Hence, the optimal k results from minimizing $O(N_k k^3)$, where N_k is the number of mesh points needed to solve the problem to the desired accuracy using a k order B-spline. This optimization is highly problem dependent.

A computationally convenient basis exists for the spaces $B_{\pi,k}$. The dimension of $B_{\pi,k}$ is $N-k$ and the basis consists of elements $B_i(x)$, $i=1, \ldots, N-k$. A complete description of the B_i is given in [9] and [7]. Briefly, when the multiplicities of the first and last mesh points are both k, so that

$$x_1 = \cdots = x_k$$

and

$$x_{N-k+1} = \cdots = x_N$$

then the main properties of the $B_i(x)$ follow:

(2.4) Each B_i is non-zero only on $[x_i, x_{i+k}]$ and is identically zero elsewhere, as well as at $x_1, \ldots, x_{i-1}$ and $x_{i+k+1}, \ldots, x_N$, even if they are in $[x_i, x_{i+1}]$.

(2.5) The sum $B_1(x) + \cdots + B_{N-k}(x)$ is identically one.

(2.6) Each B_i obeys $0 \leqslant B_i(x) \leqslant 1$ everywhere and possesses only one maximum.

These are very nice properties and will be used in later sections.

The convergence result of Theorem 2.1 is independent of the multiplicities m_i of the interior points x_i ($k < i \leqslant N-k$) of the mesh. Usually, for smooth functions f, $m_i=1$ is taken for all these interior (that is, strictly between L and R) mesh points.

The end points of the mesh typically have multiplicity k since the function f usually has $f(L) \neq 0$ and $f(R) \neq 0$, and the elements of $B_{\pi,k}$ cannot be non-zero at L and R, unless

$m_1 = k = m_N$ because of (2.2) and (2.3). In fact, relations (2.2)-(2.5) show that the only B_i which are not zero at L and R are B_1 and B_{N-k}, and these values are respectively simply $B_1(L) = 1$ and $B_{N-k}(R) = 1$.

If the function f has a discontinuity in say its j^{th} derivative, at x_i, then $m_i = k-j$ is chosen because this allows the elements of $B_{\pi,k}$ to have the same behavior. If a smaller multiplicity were chosen, the j^{th} derivative of all the elements of $B_{\pi,k}$ would be continuous at x_i, and the best fit to f from $B_{\pi,k}$ would not be very good at x_i.

Another important property of B-splines is their numerical stability or *condition*. Since any B-spline f is of the form $f = \sum_{i=1}^{N-k} a_i B_i$ and each B_i obeys $0 \leqslant B_i \leqslant 1$ we see that if $||f||$ is small compared with $||a||$, then many significant digits are lost when computing f from $a_1, \cdots, a_{N-k}$ in floating-point arithmetic [25]. Specifically,

$$d \leqslant Log_{10}(||a|| / ||\sum_{i=1}^{N-k} a_i B_i||) \tag{2.7}$$

decimal digits are lost, due to cancellation, in evaluating f. In [6] de Boor shows that

$$||\sum_{i=1}^{N-k} a_i B_i|| \geqslant C_k ||a|| \tag{2.8}$$

where C_k is a constant depending *only* upon k, and therefore

$$d \leqslant Log_{10}(C_k^{-1}).$$

In particular, he shows for a uniform mesh, one where all the mesh intervals have the same length, that

$$C_k \approx 10^{-k/5} \tag{2.8}$$

Thus, when evaluating a B-spline defined on a uniform, or nearly uniform, mesh, we would expect to lose no more than about $k/5$ decimal digits. This is a very satisfactory result since it indicates that, at least for uniform meshes, the conditioning of the B-spline basis is independent of the size of the mesh.

3. Rayleigh-Ritz Method

The Rayleigh-Ritz method for solving (1.1) approximately is extremely powerful and is directly applicable to problems substantially more complex than (1.1). It has a very rich and successful history [2,3,4,16,20,21].

This section begins by introducing the Rayleigh-Ritz method for a simple, classical problem where a variational or "energy" principle can be applied. The convergence properties of the Rayleigh-Ritz method for this canonical problem are obtained simply and concisely.

Consider the simple but classical problem of Poisson's Equation (which is (1.1) with $a = 1$, $b = 0$ and $c = -f$),

$$y'' = f(x) \quad \text{on } (0,1) \tag{3.1}$$

subject to

$$y(0) = 0 = y(1) \tag{3.2}$$

where f is some sufficiently smooth function on $[0,1]$. Physically, this may be viewed, for example, as either finding the electrostatic potential y given the charge distribution f or finding the displacement of an elastic string, clamped at the ends, subject to a transverse force f [8]. An equivalent formulation of these problems is the following "variational" principle:

Find a smooth function y which obeys (3.2) and minimizes the "energy" $F(w)$ given by

$$F(w) = \int_0^1 \left(\frac{1}{2}(w')^2 + wf \right) dx. \tag{3.3}$$

This formulation of the problem uses the principle of "least energy" [8] to find the potential, or displacement, y. It is rather easy to see that (3.3) and (3.1) are equivalent. For consider the function $g(\epsilon) \equiv F(y+\epsilon\eta)$ where ϵ is a real number and η is some smooth function which satisfies (3.2). Then the statement that y minimizes $F(w)$ implies that $g(0)$ is the minimum of g and thus $g'(0)=0$, since y makes $F(w)$ smallest over *all* smooth functions, including those of the form $y+\epsilon\eta$. But $g'(0)=0$ means that

$$\int_0^1 (y'\eta' + \eta f)\, dx = 0$$

Using integration by parts and (3.2) we then have

$$\int_0^1 (-y'' + f)\, \eta\, dx = 0 \tag{3.4}$$

for *any* suitably smooth function η satisfying (3.2). This clearly means that (3.1) must hold. The reverse argument is just as easy and shows that any solution of (3.1) is a minimizer of (3.3).

Now there is a venerable and powerful technique for approximately finding functions y which minimize functionals like $F(w)$. It is called the Rayleigh-Ritz method [8] and it has been used for many years as a tool for minimizing such functionals. The idea is quite straightforward - Pick a few *basis functions* $w_1, \ldots, w_N$ which are smooth and satisfy (3.2). Then any linear combination w of the w_i of the form

$$w = a_1 w_1 + \cdots + a_N w_N \tag{3.5}$$

is also smooth and satisfies (3.2). All linear combinations of the form (3.5) form a *linear subspace* of the space of all smooth functions. This space is called the *span* of $w_1, \ldots, w_N$ and is denoted by $< w_1, \ldots, w_N >$. The span is called a linear space since the sum of any two elements in it is still in it, also any scalar multiple of an element remains in it. If the w_i are judiciously chosen, then minimizing F over the large subset (sub-space) $< w_1, \ldots, w_N >$ of the space of all smooth functions should give a good approximation to the solution of (3.1)-(3.2). Specifically, we choose $a_1, \ldots, a_N$ to minimize the function

$$I(a_1, \ldots, a_N) \equiv F(a_1 w_1 + \cdots + a_N w_N). \tag{3.6}$$

Minimizing (3.6) is an easy task since I is a quadratic function of its arguments. The minimum of I is attained at the point where

$$\frac{\partial I}{\partial a_i}(a_1, \ldots, a_N) = 0, \quad i=1, \ldots, N$$

which leads to the equations

$$0 = \int_0^1 \left[\sum_j a_j w_j' w_i' + w_i f \right] dx, \quad i=1, \ldots, N. \tag{3.7}$$

Now consider the symmetric, positive-definite (as we shall see later) matrix A given by

$$A_{ij} = \int_0^1 w_i' w_j'\, dx \tag{3.8}$$

and the vector $b = (-\int_0^1 w_1 f\, dx, \ldots, -\int_0^1 w_N f\, dx)$. Then the solution a of the linear system of algebraic equations

$$A\ a = b \tag{3.9}$$

is the vector of coefficients a for the linear combination (3.5) which minimizes (3.6). The numerical solution of the above system (3.9) is a well-understood and well-posed problem and is easily solved [13,25].

Thus, the Rayleigh-Ritz method of minimizing (3.3) over finite dimensional subspaces of smooth functions reduces the original problem (3.3), whose generalizations nobody knows how to solve directly and constructively, to a problem (3.9) which can be easily solved. Moreover, if the basis-functions w_i are well chosen, the function v which results from minimizing (3.6) should be quite close to the solution y of (3.1) and (3.2).

If B-splines are taken as the basis-functions w_i, then, since they can approximate any smooth function w to within $O(h^k)$, it is clear that they can approximate the solution y at least that well. The question is:

Does the Rayleigh-Ritz method produce a B-spline which is as accurate as can be achieved, $O(h^k)$?

The answer is, of course, yes. The proof of such results [2,3,4,24] is in general a rather complicated matter. However, a weak form of the convergence result can be easily obtained for the simple example under consideration (3.1)-(3.2).

Let π be any mesh on $[0,1]$ with the multiplicity of x_1 and x_N both k. Then, since the Rayleigh-Ritz approximate solution $\hat{y} = \sum_{i=1}^{N-k} a_i B_i$ which minimizes $I(a_1, \ldots, a_{N-k})$ must also satisfy the boundary conditions (3.2), we see that $a_1 = 0 = a_{N-k}$. Thus, the subspace of $B_{\pi,k}$ which obeys (3.2) is just the span of $B_2, \ldots, B_{N-k-1}$ and the Rayleigh-Ritz basis functions w_i are simply B_{i+1} for $i=1, \ldots, N-k-2$.

It is assumed that any mesh π is chosen so that

$$Max\ |x_{i+1} - x_i| \,/\, Min\ |x_{i+1} - x_i| \leq \mu \tag{3.10}$$

where μ is some fixed constant. This relation simply assures that the mesh π cannot stray too far from being uniform. This restriction on the mesh is important and will be used in later sections.

Now note that (3.3) is equivalent to (3.4), which in turn is equivalent, using integration by parts, to

$$\int_0^1 (y'\eta' + f\eta)\, dx = 0, \tag{3.11}$$

But, from (3.7), the Rayleigh-Ritz solution is just the solution $\hat{y}$ of

$$\int_0^1 (\hat{y}'B_i' + f B_i)\, dx = 0, \quad i=2, \ldots, N-k-1, \tag{3.12}$$

where $\hat{y} \epsilon < B_2, \ldots, B_{N-k-1} >$. Since (3.11) must hold, in particular, for $\eta = B_i$, we then have, subtracting (3.12) from (3.11) with $\eta = B_i$,

$$\int_0^1 (y' - \hat{y}')B_i'\, dx = 0. \tag{3.13}$$

This relation states that $\hat{y}$ is also the solution of

$$\underset{\hat{y} \epsilon < B_2, \ldots, B_{N-k-1} >}{Min} \int_0^1 (y' - \hat{y}')^2\, dx. \tag{3.14}$$

That is, $\hat{y}$ minimizes the above measure of the difference between $\hat{y}$ and the solution y. **This is the first key observation:**

> *The Rayleigh-Ritz method does in fact minimize some measure of the error in the approximate solution, even though it doesn't minimize the error $||y - \hat{y}||$ itself.*

We know from Theorem 2.1 that

$$\underset{\hat{v} \in B_{\pi,k}}{Min} \int_0^1 (y' - \hat{v}')^2 \, dx = O(h^{k-1})^2$$

A natural question now arises:

> *Is the minimum achieved in (3.14) also $O(h^{k-1})^2$?*

We can show that the answer is yes as follows. By Theorem 2.1 there is an element v of $B_{\pi,k}$ for which both $||y(x) - v(x)|| = O(h^k)$ and $||y'(x) - v'(x)|| = O(h^{k-1})$. However, Theorem 2.1 does not guarantee that v will satisfy the boundary conditions (3.2). We must show that v comes within $O(h^{k-1})$ of satisfying (3.2). Let $v = \sum_{i=1}^{N-k} b_i B_i$, then since $B_1(0) = 1$ and all other $B_i(0) = 0$ we have $b_1 = O(h^k)$. Similarly, $b_{N-k} = O(h^k)$. Thus, by simply setting $b_1 = 0 = b_{N-k}$ we obtain $\hat{v} \epsilon < B_2, \ldots, B_{N-k-1} >$, given by $\hat{v} = \sum_{i=2}^{N-k-1} b_i B_i$, which differs from v by $b_1 B_1(x) + b_{N-k} B_{N-k}(x)$. We shall show that $\hat{v}$ is the function we are looking for. Anyone not interested in the proof of this result may skip the next paragraph.

We have [7] that (2.1)-(2.5) imply

$$B_1(x) = (x_{k+1} - x)^{k-1} / (x_{k+1} - x_1)^{k-1} \text{ and}$$
$$B_{N-k}(x) = (x - x_{N-k})^{k-1} / (x_N - x_{N-k})^{k-1}$$

Thus,

$$\begin{aligned} |v' - \hat{v}'| &\leqslant |b_1||B_1'(x)| + |b_{N-k}||B_{N-k}'(x)| \\ &= O(h^k)(k-1)\left[\frac{(x_{k+1} - x)^{k-2}}{(x_{k+1} - x_1)^{k-1}} + \frac{(x - x_{N-k})^{k-2}}{(x_N - x_{N-k})^{k-1}} \right] \\ &= O(h^{k-1}) \end{aligned}$$

making use of the mesh restriction (3.14). This gives us $||v' - \hat{v}'|| = O(h^{k-1})$ and thus

$$\begin{aligned} \int_0^1 (y' - \hat{v}')^2 \, dx &= \int_0^1 (y' - v' + O(h^{k-1}))^2 \, dx \\ &= \int_0^1 (y' - v')^2 \, dx + 2O(h^{k-1}) \int_0^1 (y' - v') \, dx + \int_0^1 O(h^{k-1})^2 \, dx \\ &= O(h^{k-1})^2. \end{aligned}$$

This shows that

$$\int_0^1 (y' - \hat{v}')^2 dx = O(h^{k-1})^2$$

for some element $\hat{v}$ of $< B_2, \ldots, B_{N-k-1} >$ and we then have

$$\underset{\hat{v} \epsilon < B_2, \ldots, B_{N-k-1} >}{Min} \int_0^1 (y' - \hat{v}')^2 dx = O(h^{k-1})^2 \qquad (3.15)$$

With this knowledge we can ask how big the error $y - \hat{y}$ is in terms of $y' - \hat{y}'$, which we know has been made as small as possible. Let $f = y - \hat{y}$. The starting point is the derivative relation, which uses the fact that $f(0) = 0$,

$$f(x) = \int_0^x f'(\xi)\, d\xi$$

which gives

$$|f(x)| \leqslant \int_0^x |f'(\xi)|\, d\xi.$$

We can then apply the Cauchy-Schwarz inequality [8]

$$\int_0^1 |f(x)\, g(x)|\, dx \leqslant (\int_0^1 f^2 dx)^{1/2} (\int_0^1 g^2 dx)^{1/2}$$

to the right-hand-side of the above relation, with $g \equiv 1$, to get

$$|f(x)| \leqslant x^{1/2} (\int_0^x (f'(\xi))^2 d\xi)^{1/2}$$

which in turn gives

$$||y - \hat{y}|| \leqslant (\int_0^1 (y' - \hat{y}')^2 dx)^{1/2} \tag{3.16}$$

This is the second key observation:

The error in $\hat{y}$ is bounded above by a measure of the error in $\hat{y}'$, and the latter is minimized by $\hat{y}$.

By combining (3.15) and (3.16) we have

$$||y - \hat{y}|| = O(h^{k-1}) \tag{3.17}$$

This shows that the Rayleigh-Ritz method converges when applied to (3.1)-(3.2) and even shows that the convergence rate is at least as fast as h^{k-1}.

The same result, (3.17), can be easily obtained for (1.1) subject to $y(0) = 0 = y(1)$ using precisely the same techniques as were used above for (3.1)-(3.2). The interested reader is invited to obtain this result as an exercise. The "energy" associated with the equation

$$\begin{aligned} &(ay')' + by + c = 0 \quad \text{on } [0,1] \\ &y(0) = 0 = y(1) \end{aligned} \tag{3.18}$$

is

$$\int_0^1 [\frac{1}{2} a (y')^2 - \frac{1}{2} b y^2 - cy]\, dx \tag{3.19}$$

and a function y solves (3.18) if and only if it also minimizes (3.19). By assuming that both $||a||$ and $||b||$ are finite, and that $a(x) \geqslant \sigma > 0$ for some constant σ, the error in the Rayleigh-Ritz solution of (3.19) can easily be shown to be no greater than $O(h^{k-1})$ using precisely the same arguments as were used above. Such a proof of $O(h^{k-1})$ convergence for the Rayleigh-Ritz solution of (3.18), under the above assumptions, is presented in Appendix 1.

We have seen that the error in the Rayleigh-Ritz approximate solution of (1.1) subject to $y(L) = 0 = y(R)$ is at most $O(h^{k-1})$.

The actual rate of convergence for the Rayleigh-Ritz solution of (3.18) is $O(h^k)$

The proof of such results is rather deep and complex [24], and we must be content to use the above simple arguments to obtain $O(h^{k-1})$ convergence rate estimates.

4. Galerkin's Method

So far we have discussed the Rayleigh-Ritz method, but what is Galerkin's method? This section first formulates the general Galerkin equations. Finally, a general statement is made about the relation between Galerkin's method and variational principles, and why Galerkin's method may be viewed as an extension of them.

The starting point for Rayleigh-Ritz is an "energy" functional like (3.3) or (3.19) and it results in equations like (3.12) or more generally

$$\int_0^1 (-a\hat{y}'B_i' + b\hat{y}B_i + cB_i)\,dx = 0 \tag{4.1}$$

for (3.19).

Galerkin's method starts from another point of view but arrives at precisely the same result as Rayleigh-Ritz gives. Specifically, Galerkin's method for (3.18) is to solve

$$\int_0^1 ((a\hat{y}')' + b\hat{y} + c)B_i\,dx = 0,\ i=2,\ .\ .\ .\ ,N-k-1 \tag{4.2}$$

for $\hat{y}$. The idea is simply to make the error in the differential equation (3.18) small by making it "orthogonal" to $< B_2, \ldots, B_{N-k-1} >$. Actually (4.2) is not used in practice, rather integration by parts is used to convert it into

$$\int_0^1 (-a\hat{y}'B_i' + b\hat{y}B_i + cB_i)\,dx = 0,$$

using $y(0) = 0 = y(1)$, which is identical to the Rayleigh-Ritz equations (4.1) for (3.18). In fact,

> *Whenever there is an energy functional for a differential equation, the Rayleigh-Ritz method applied to that differential equation is the same as Galerkin's method for that equation.*

To show the general nonlinear self-adjoint case [2] and the equivalence of Rayleigh-Ritz and Galerkin's method, the two formulations are presented below for the equation

$$(ay')' = f(x,y) \quad \text{on} \quad (0,1) \tag{4.3}$$

subject to $y(0) = 0 = y(1)$. The Rayleigh-Ritz functional for (4.3) is [2]

$$\int_0^1 \left[\frac{1}{2}a(w')^2 + \int_0^{w(x)} f(x,\xi)\,d\xi \right] dx \tag{4.4}$$

and its minimum over $< B_2, \ldots, B_{N-k-1} >$, $\hat{y}$, is attained when

$$0 = \int_0^1 (a\hat{y}'B_i' + f(x,\hat{y})\,B_i)\,dx \tag{4.5}$$

Thus, (4.5) are the Rayleigh-Ritz equations for (4.3). The Galerkin equations for (4.3) are

$$\int_0^1 ((a\hat{y}')' - f(x,\hat{y}))\,B_i\,dx = 0 \tag{4.6}$$

which, using integration by parts, is exactly (4.5). Thus, for this very general class of equations, the Rayleigh-Ritz and Galerkin solutions are identical, and a proof of convergence for Rayleigh-Ritz also shows that Galerkin's method is convergent.

Since Rayleigh-Ritz requires an "energy" functional, its use is limited to differential equations which have such functionals. Galerkin's method does not need or use an energy functional and can thus be applied to equations where Rayleigh-Ritz cannot. For example, the general, non-self-adjoint, linear two-point boundary value problem

$$ay''+by'+cy+d = 0 \text{ on } (0,1) \tag{4.7}$$

subject to $y(0) = 0 = y(1)$ has no known Rayleigh-Ritz functional. However, Galerkin's method for (4.7) is trivial to write down:

$$0 = \int_0^1 B_i\,(a\hat{y}''+b\hat{y}'+c\hat{y}+d)\,dx,\quad i=2,\ldots,N-k-1 \tag{4.8}$$

which, as usual, is re-written as

$$0 = \int_0^1 \left[-(aB_i)'\hat{y}'+B_ib\hat{y}'+B_ic\hat{y}+B_id\right]dx \tag{4.9}$$

This technique converges as $O(h^k)$ even for this non-self-adjoint equation [21, 24].

In general, even for nonlinear differential equations subject to $y(0) = 0 = y(1)$, writing down the Galerkin equations is trivial and may be summed up as:

$$0 = \int_0^1 (\textit{the error in the differential equation})B_i\,dx,\quad i=2,\ldots,N-k-1. \tag{4.10}$$

If the differential equation is nonlinear, then so is the Galerkin system of equations, obviously. Thus, at least conceptually, writing down the Galerkin equations is a very easy matter. It is in this sense that Galerkin's method may be viewed as an extension of the Rayleigh-Ritz method:

> *Galerkin's method can be applied to literally any differential equation, but when applied to a differential equation with an "energy" functional, it agrees exactly with the Rayleigh-Ritz solution.*

The above discussion has shown that Galerkin's method is rather simple to describe, even for nonlinear equations, and that it converges for simple, linear equations like (1.1) subject to $y(0) = 0 = y(1)$.

We now need to formulate the Galerkin equations for the general case of (1.1) subject to (1.2). For $i=2,\ldots,N-k-1$ the following usual Galerkin equations hold

$$0 = \int_L^R (-a\hat{y}'B_i'+b\hat{y}B_i+cB_i)\,dx \tag{4.11}$$

When we write

$$\hat{y} \equiv \sum_i a_iB_i \tag{4.12}$$

relation (4.11) becomes the system of linear equations

$$\sum_{j=1}^{N-k} a_j\int_L^R (-bB_iB_j+aB_i'B_j')\,dx = \int_L^R cB_i\,dx,\quad i=2,\ldots,N-k-1. \tag{4.13}$$

Equations (4.13) may be viewed as determining $a_2,\ldots,a_{N-k-1}$. Equation (4.13) cannot be used for $i=1$ and $i=N-k$ since the integration by parts which led to (4.11) is not valid for these values of i. We now need equations for determining a_1 and a_{N-k}.

The previous Galerkin formulations of this section were based on the assumption that the boundary conditions were homogeneous Dirichlet, that is $y(L) = 0 = y(R)$. These boundary conditions determined the coefficients a_1 and a_{N-k} in the Galerkin solution, namely $a_1 = 0 = a_{N-k}$. The same principle still holds, even for boundary conditions like (1.2). The boundary conditions do determine a_1 and a_{N-k}, although not as simply as in the $y(L) = 0 = y(R)$ case.

In determining a_1, there are two basic cases to consider. The first case is when $\beta_L = 0$, which gives $y(L) = \gamma_L / \alpha_L$. In this case, then, $a_1 = y(L)$ is known and any appearance of a_1 in (4.13) can be moved to the right hand side of those equations and the equation for a_1 is simply

$$a_1 = \gamma_L / \alpha_L . \tag{4.14}$$

The second boundary case is when $\beta_L \neq 0$. Here the value of $\hat{y}(L) = a_1$ is unknown *a priori.* A natural and logical, but improper, way to specify a_1 would be to force the boundary condition (1.2) to hold *exactly* at $x=L$:

$$\alpha_L y(L) + \beta_L y'(L) = \gamma_L \tag{4.15}$$

It is easily seen that this cannot be a proper thing to do, for (4.15) would mean that

$$\alpha_L e(L) + \beta_L e'(L) = 0 \tag{4.16}$$

where $e = y - \hat{y}$ is the error. But we want, and expect, e to be $O(h^k)$ and know that e' cannot in general be better than $O(h^{k-1})$. Thus, since in general $\alpha_L \neq 0$, (4.16) implies that

$$e(L) = -\frac{\beta_L}{\alpha_L} e'(L) = O(h^{k-1}) \tag{4.17}$$

and forcing the boundary conditions to hold exactly at $x=L$ results in a lower convergence rate than is possible, expected and optimal.

A similar argument can be made against doing the next most "obvious" thing, namely forcing the boundary conditions to hold in the "Galerkin" sense by requiring

$$\int_L^R (\alpha_L \hat{y} + \beta_L \hat{y}' - \gamma_L) B_1 \, dx = 0 \tag{4.18}$$

to hold. Thus, the boundary conditions alone, in a vacuum, are not sufficient to determine a_1.

The only other information available is that provided by the differential equation (1.1) itself. This observation results in the following formulation of the equation for a_1, and requires a detailed look at the derivation of the Galerkin equations.

The starting point for Galerkin's equations for (1.1) is the relation

$$0 = \int_L^R ((a\hat{y}')' + b\hat{y} + c) B_i \, dx$$

which, using integration by parts, becomes

$$0 = \int_L^R (-a\hat{y}'B_i' + b\hat{y}B_i + cB_i) \, dx + aB_i\hat{y}' \Big|_L^R . \tag{4.19}$$

For $i=2, \ldots, N-k-1$ we have $B_i(L) = 0 = B_i(R)$, by (2.2)-(2.6), and thus for $i=2, \ldots, N-k-1$ equations (4.19) are exactly (4.11). The previous discussion of the boundary conditions at $x=L$, when $\beta_L \neq 0$, then leads to a natural, and healthy, interest in (4.19) for $i=1$. For in the case where $\beta_L \neq 0$ we can solve (1.2) for

$$y'(L) = \frac{\gamma_L}{\beta_L} - \frac{\alpha_L}{\beta_L} y(L) \tag{4.20}$$

Equation (4.19), as written, contains no information about the boundary conditions. However, by replacing $\hat{y}'(L)$ by the right hand side of (4.20) in (4.19) for $i=1$ gives

$$0 = \int_{L}^{R} (-a\hat{y}'B_1' + b\hat{y}B_1 + cB_1)\,dx - a(L)\left(\frac{\gamma_L}{\beta_L} - \frac{\alpha_L}{\beta_L}\hat{y}(L)\right) \tag{4.21}$$

This is a slightly re-written Galerkin equation for $i=1$ which does contain information about the boundary conditions.

Equation (4.21) is the correct equation for determining a_1 when $\beta_L \neq 0$ and a similar relation holds for a_{N-k} when $\beta_R \neq 0$. This scheme is shown to be convergent as $O(h^k)$ in [4].

Now that we have the equations for the Galerkin solution $\hat{y}$, namely (4.11) and either (4.14) or (4.21), we need to find out what equations they represent for $a_1, \ldots, a_{N-k}$. Let δ_{ij} denote the standard Kronecker delta function, that is $\delta_{ij} \equiv 0$ for all $i \neq j$ and $\delta_{ii} \equiv 1$. Also, let $(f,g) \equiv \int_{L}^{R} f(x)g(x)\,dx$ for any functions f and g. Then we have the following four cases, where care has been taken to write the equations so that their symmetric and banded structure is clear:

Case 1: $\beta_L = 0 = \beta_R$

$$a_1 = \gamma_L/\alpha_L$$

and for $i = 2, \ldots, N-k-1$,

$$\begin{aligned}\sum_{j=2}^{N-k-1} a_j[(aB_j',B_i') - (bB_j,B_i)] = {}&\\ (c,B_i) + \frac{\gamma_L}{\alpha_L}(-(aB_1',B_i') + (bB_1,B_i)) +{}& \\ \frac{\gamma_R}{\alpha_R}(-(aB_{N-k}',B_i') + (bB_{N-k},B_i))&\end{aligned} \tag{4.22}$$

and finally,

$$a_{N-k} = \gamma_R/\alpha_R$$

Case 2: $\beta_L \neq 0 = \beta_R$.

For $i = 1, \ldots, N-k-1$,

$$\begin{aligned}\sum_{j=1}^{N-k-1} a_j[(aB_j',B_i') - (bB_j,B_i) - \delta_{i1}\delta_{j1}a(L)\frac{\alpha_L}{\beta_L}] = {}&\\ (c,B_i) + \frac{\gamma_R}{\alpha_R}(-(aB_{N-k}',B_i') + (bB_{N-k},B_i)) -{}& \\ \delta_{i1}a(L)\frac{\gamma_L}{\beta_L}&\end{aligned} \tag{4.23}$$

and finally,

$$a_{N-k} = \gamma_R/\alpha_R$$

Case 3: $\beta_L = 0 \neq \beta_R$.

$$a_1 = \gamma_L/\alpha_L$$

and for $i = 2, \ldots, N-k$,

$$\sum_{j=2}^{N-k} a_j[(aB_j',B_i') - (bB_j,B_i) + a(R)\delta_{i,N-k}\delta_{j,N-k}\frac{\alpha_R}{\beta_R}] =$$

$$(c,B_i) + \frac{\gamma_L}{\alpha_L}(- (aB_1',B_i') + (bB_1,B_i)) + \qquad \delta_{i,N-k} a(R) \frac{\gamma_R}{\beta_R} \tag{4.24}$$

Case 4: $\beta_L \neq 0 \neq \beta_R$.

For $i = 1, \ldots, N-k$,

$$\sum_{j=1}^{N-k} a_j [(aB_j',B_i') - (bB_j,B_i) - \delta_{i1}\delta_{j1} a(L)\frac{\alpha_L}{\beta_L} + \delta_{i,N-k}\delta_{j,N-k} a(R)\frac{\alpha_R}{\beta_R}] =$$

$$(c,B_i) + \delta_{i,N-k} a(R)\frac{\gamma_R}{\beta_R} - \qquad \delta_{i1} a(L)\frac{\gamma_L}{\beta_L} \tag{4.25}$$

Equations (4.23)-(4.25) all have the form

$$Ga = b \tag{4.26}$$

where G is an $N-k$ by $N-k$ matrix and b is a known $N-k$ vector. Such systems are easily solved, see [13] for example. The system of linear algebraic equations given by (4.26) has some very nice properties. First, it is *symmetric*, that is, $G_{ij} = G_{ji}$ for all i and j. This means that we only need to store G_{ij} for $j \leqslant i$. The matrix G is also *banded* in that $G_{ij} = 0$ if $|i-j| \geqslant k$, this property coming from the fact that each B-spline basis function B_i is only non-zero on the interval $[x_i,x_{i+k}]$. The matrix G is also *positive-definite*. This is easily seen in the case of (1.1) subject to $y(L) = 0 = y(R)$: The statement that G is positive definite means that for any non-zero vector a we have $a'Ga > 0$. If there is a vector a so that $a'Ga \leqslant 0$, then for $v = \sum_{i=2}^{N-k-1} a_i B_i(x)$ we have $0 \geqslant a'Ga = \int_L^R (av'B_i' - bvB_i)\,dx$, for $i=2, \ldots, N-k-1$, which simply means that $0 \geqslant \int_L^R (a(v')^2 - bv^2)\,dx$. But since $b \leqslant 0$ on (L,R) we then have $0 \geqslant \int_L^R a(v')^2\,dx$ which implies that $v' \equiv 0$ on $[L,R]$ since $a(x) > 0$ there. Having $v(L) \equiv 0$ and $v'(x) \equiv 0$ on (L,R) gives $v(x) \equiv 0$ on $[L,R]$ and hence $a_1 = \cdots = a_{N-k} = 0$. Thus, the only vector a such that $a'Ga \leqslant 0$ is a=0 and G is positive-definite.

Now a banded, symmetric and positive-definite matrix is an especially nice matrix to deal with and the solution of such systems of equations as (4.26) can be achieved cheaply and accurately [13]. Specifically, only the elements G_{ij} for $i=1, \ldots, N-k$ and $i-k+1 \leqslant j \leqslant i$ need be stored. This requires about $k(N-k)$ storage locations. The system of equations (4.26) can then be solved in about $k^2(N-k)$ operations.

5. Implementation of Galerkin's Method

It is good to know that the Galerkin equations can be solved cheaply and accurately, once they are formed, and that the resulting approximate solution converges to the true solution of the problem like $O(h^k)$.

But exactly how does one set-up the Galerkin equations?

The matrix G and the right-hand-side b of (4.26) involve the integration of many functions. One way of computing them would be to call a good automatic quadrature routine for computing integrals as accurately as possible. However, this would be an expensive over-kill of the problem since a typical user only wants the solution of his problem accurate to somewhere between 3 and 6 decimal places. Thus, it would be nice if the necessary integrals could be computed only as accurately as needed in order to give a computed Galerkin solution which is accurate to $O(h^k)$.

To get some idea of how this might be accomplished, consider the canonical problem of Poisson's equation

$$y'' = f \quad \text{on } (0,1) \tag{5.1}$$

subject to $y(0) = 0 = y(1)$. The Galerkin matrix G in this case is simply

$$G_{ij} = \int_0^1 B_i'(x) B_j'(x)\, dx$$

and the integrands are all simply piecewise polynomials of degree $2(k-2)$. Thus, all we need to do is, piecewise, integrate polynomials of degree $2(k-2) = 2k-4$. By far and away the easiest way to do this is to use Gauss-Legendre quadrature [18] on each B-spline mesh interval $[x_i, x_{i+1}]$. Since an m point Gauss-Legendre quadrature rule is exact for polynomials of degree $\leqslant 2m-1$, we see that a $k-1$ point Gaussian quadrature rule is exact for polynomials of degree $\leqslant 2k-3$. Thus, a $k-1$ point Gauss-Legendre quadrature rule, applied to each B-spline mesh interval, will exactly compute the Galerkin matrix G for the simple equation (5.1). This raises an interesting question:

> *Is a $k-1$ point Gauss-Legendre quadrature rule, applied over each B-spline mesh interval, always accurate enough to compute an approximate Galerkin matrix G^* and right-hand-side b^* so that the computed Galerkin solution a^* of*
>
> $$G^* a^* = b^* \tag{5.2}$$
>
> *still gives*
>
> $$\| y(x) - \sum_{i=1}^{N-k} a_i^* B_i(x) \| = O(h^k)$$
>
> *where y is the solution of (1.1)-(1.2)?*

The answer to this question is, of course, yes. We shall now show why it is true, and also why no fewer than $k-1$ points can be used in the quadrature rule. Thus, we shall see that precisely a $k-1$ point Gauss-Legendre quadrature rule should be used to compute the integrals.

The first step is to study the sensitivity of the Galerkin solution coefficients a of (4.26) to perturbations in G and b. Later we shall study the perturbations of G and b introduced by only approximately computing the integrals. We know that if the right-hand-side b is perturbed by δb then the solutions of the two equations

$$Ga = b \quad \text{and} \quad G(a + \delta a) = b + \delta b$$

differ by $\delta a = G^{-1} \delta b$.

Now the norm of a matrix A is simply defined in terms of the vector norm to be

$$\|A\| \equiv \mathop{Max}_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

It is an easy matter to see that

$$\|AB\| \leqslant \|A\| \, \|B\|$$

and

$$\|A + B\| \leqslant \|A\| + \|B\|,$$

these relations being inherited from the same relationships for the vector norm. Thus, $\|\delta a\| \leqslant \|G^{-1}\| \, \|\delta b\|$. Also, since $\|b\| \leqslant \|G\| \, \|a\|$ we can divide these two relations to obtain [13]

$$\frac{||\delta a||}{||a||} \leqslant ||G||\,||G^{-1}||\frac{||\delta b||}{||b||}. \qquad (5.3)$$

The quantity $||G||\,||G^{-1}||$ in (5.3) is called the *condition number* of G and is denoted by $Cond(G)$. The condition number of G clearly relates the relative change in the right-hand-side b to the relative change in the solution a. If $Cond(G)$ is not too large, then a small change in b will result in only a small change in a. Similarly it can be shown [13] that perturbing G by δG as in the equations

$$Ga = b \text{ and } (G+\delta G)(a+\delta a) = b$$

gives, for sufficiently small $||\delta G||$,

$$\frac{||\delta a||}{||a||} \leqslant Cond(G)\frac{||\delta G||}{||G||}. \qquad (5.4)$$

Thus, perturbing either G or b by a relative change of ϵ, results in a relative change in a of at most $Cond(G)\epsilon$. This tells us how sensitive the solution a of (4.26) is to changes in either the matrix G or the right-hand-side b.

We now study the perturbations introduced by approximate integration. By using a $k-1$ point Gauss-Legendre quadrature rule, we obtain all integrals accurate to $O(h^{2(k-1)})$ and thus the perturbations are all $O(h^{2(k-1)})$. Thus, we may write

$$\frac{||a-a^{*}||}{||a||} \leqslant Cond(G)O(h^{2(k-1)}) \qquad (5.5)$$

and if we can show that $Cond(G)$ doesn't grow too fast with $h \rightarrow 0$, then (5.5) will show that the computed Galerkin solution coefficients a^{*} stay close to the exact Galerkin solution coefficients.

As a start we first consider Poisson's equation (5.1). To estimate $Cond(G)$ we first find an upper bound on $||G||$. This is obtained by noting that for any vector a we have

$$|\sum_j G_{ij}a_j| = |\sum_j a_j\int_0^1 B_i'B_j'\,dx| = |\sum_{|i-j|\leqslant k} a_j\int_{x_i}^{x_{i+k}} B_i'B_j'\,dx| \leqslant ||a||O(h^{-1})$$

since $||B_i'|| = O(h^{-1})$, see Appendix 2. This gives

$$||G|| = O(h^{-1}). \qquad (5.6)$$

We next need an upper bound on $||G^{-1}||$. This is obtained by noting that for any vector a we have

$$\sum_{i,j}a_ia_jG_{ij} = \int_0^1 \sum_{i,j}a_ia_jB_i'B_j'\,dx = \int_0^1 (\sum_i a_iB_i')^2\,dx \geqslant ||\sum_i a_i B_i||^2 \qquad (5.7)$$

using the same argument that led to (3.16). But we know from section 2, equation (2.8), that

$$||\sum_i a_i B_i|| \geqslant C_k||a|| \qquad (5.8)$$

for any vector a. When this result is coupled with (5.7) we see that

$$\sum_{i,j}a_ia_jG_{ij} \geqslant C_k^2||a||^2. \qquad (5.9)$$

But, since G is positive-definite,

$$\sum_{i,j}a_ia_jG_{ij} = |\sum_i a_i(\sum_j G_{ij}a_j)| \leqslant ||a||O(h^{-1})||Ga||,$$

since $N=O(h^{-1})$, and by combining this with (5.9) we see

$$||Ga|| \geqslant C_k^2 O(h)||a|| \qquad (5.10)$$

which gives

$$||G^{-1}|| \leqslant C_k^{-2} O(h^{-1}). \qquad (5.11)$$

By combining (5.6) with (5.11) we obtain the very nice result that

$$Cond(G) = O(h^{-2}) \qquad (5.12)$$

Thus, the Galerkin equations for (5.1) are well conditioned. In fact, the Galerkin equations condition number of $O(h^{-2})$, is the same order as that for centered finite differences [22].

The interested reader is encouraged to show that the same result - $Cond(G) = O(h^{-2})$ - holds for the Galerkin matrix of (1.1) subject to $y(L) = 0 = y(R)$, under the same assumptions as were made in section 3 to make the proof of the $O(h^{k-1})$ Rayleigh-Ritz convergence rate for (1.1) easy to obtain, as an exercise. The proof follows the above outline exactly. Such a proof is presented in Appendix 3. A general proof that $Cond(G) = O(h^{-2})$ is given in [22].

Also, by (5.5), we then have

$$\frac{||a-a^*||}{||a||} = O(h^{2(k-2)}) = O(h^k h^{k-4})$$

which shows that, for $k \geqslant 4$,

$$||a-a^*|| = O(h^k). \qquad (5.13)$$

This shows that the error in the computed Galerkin coefficients a^* is on the order of the error in the Galerkin solution itself. Since

$$\begin{aligned} ||y^*-y|| &\leqslant ||y^*-\hat{y}||+||\hat{y}-y|| \\ &= ||\sum_i (a_i^*-a_i)B_i||+O(h^k). \end{aligned}$$

we may use (2.5) and (2.6) to see, for $k \geqslant 4$,

$$||y^*-y|| \leqslant ||a-a^*||+O(h^k) = O(h^k), \qquad (5.14)$$

This establishes that, for $k \geqslant 4$, using a $k-1$ point Gauss-Legendre quadrature rule to compute the necessary integrals results in a computed approximate Galerkin solution which converges at the correct rate. The same result holds for $k=2$ and 3, but a more detailed argument must be used [23].

If $k-1$ quadrature points per mesh interval is sufficient to guarantee the correct rate of convergence of the computed Galerkin solution, can fewer than $k-1$ points be used in the quadrature rule to give the same result? It is easy to see that no fewer than $k-1$ quadrature points can be used by considering the sample problem (5.1) again. In that case, the Galerkin matrix is

$$G_{ij} = \int_0^1 B_i' B_j' \, dx$$

which is to be computed by say a $k-2$ point Gauss-Legendre quadrature rule. Let the quadrature rule have abscissae ξ_m and weights ω_m, $m=1, \ldots, k-2$. Then for any function $g(x)$ on $[-1,+1]$ we have

$$\int_{-1}^{+1} g(x)dx \approx \sum_{i=1}^{k-2} \omega_i g(\xi_i)+O(||g^{(2(k-2))}||)$$

Now consider the simplest mesh possible, that is, just one mesh interval, $x_1 = \cdots = x_k = 0$, $x_{k+1} = \cdots = x_{2k} = 1$. In this case, the approximate Galerkin matrix is, ignoring a factor of 2,

$$G^*_{ij} = \sum_{m=1}^{k-2} \omega_m B_i'(x_m) B_j'(x_m), \tag{5.15}$$

where $x_m = (1 + \xi_m) / 2$. We shall show that this approximate Galerkin matrix is singular, and hence that a less than $k-1$ point quadrature rule cannot in general be sufficient to compute an accurate Galerkin matrix. To show that G^* is singular, it is enough to exhibit a non-zero vector $a = (0, a_2, \ldots, a_{N-k-1}, 0)$ such that $G^* a = 0$. Now we know that $G_{ij} = \int_0^1 B_i' B_j' \, dx$ and thus

$$\sum_j G_{ij} a_j = \int_0^1 B_i' (\sum_j a_j B_j') \, dx \tag{5.16}$$

We can exhibit the appropriate vector a by constructing a non-zero function $v = \sum_i a_i B_i$ which is a polynomial of degree $\leqslant k-1$, whose first derivative vanishes at the x_m for $m=1, \ldots, k-2$, and for which $v(0) = 0 = v(1)$. A polynomial of degree $k-2$ which vanishes at the quadrature points is given by

$$z(x) \equiv (x - x_1) \cdots (x - x_{k-2})$$

The polynomial of degree $k-1$ given by

$$v(x) \equiv \int_0^x z(\eta) \, d\eta \tag{5.17}$$

has $v'(x_m) = z(x_m) = 0$ for $m=1, \ldots, k-2$, $v(0)=0$ and $v(1) = \int_0^1 z(\eta) \, d\eta = 0$ by virtue of the fact that the x_m are the Gaussian quadrature points for the interval $[0,1]$. The polynomial v is not zero since the coefficient of x^{k-1} is $1/(k-1)$. Thus, the non-zero polynomial v given by (5.17) provides the necessary non-zero coefficient vector a for which $G^* a = 0$ and G^* is in fact singular. These results may be summed up as follows:

> *Precisely $k-1$ Gauss-Legendre quadrature points should be used per mesh interval to compute the integrals. The use of more points would increase the cost, but not the convergence rate, while the use of fewer quadrature points could result in a singular matrix.*

We now have a complete formulation of Galerkin's method - the equations and a method for forming the integrals which make up the equations. The most obvious way to form the Galerkin equations is then basically to compute G_{ij} for $j \leqslant i$ by doing each integral, piece by piece, in its turn, as in the code

```
For $i=2, \ldots, N-k-1$
    {
    For $j=Max(i-k+1, 1), \ldots, i$
        {
        Compute $\int_{x_i}^{x_{i+k}} (aB_i'B_j' - bB_iB_j) dx$
        using $k-1$ Gaussian quadrature points on each mesh interval.
        }
    }
```

This then requires roughly $(N-k)\frac{k}{2}(k^2-1)$ evaluations of $a(x)$ and $b(x)$, and as many multiplications, to compute all integrals. However, this is exceedingly wasteful in that both $a(x)$ and $b(x)$ are really only evaluated at the quadrature points of each interval, or about

$(N-k)(k-1)$ points. This could of course be remedied by evaluating and storing the necessary $a(x)$ and $b(x)$ values just before entering the outer loop. However, this would require about $2(N-k)(k-1)$ storage locations, which is more than that used by the Galerkin matrix itself, which uses precisely $k(N-k)$ locations. Another way would be to evaluate and store the a and b values and all the non-zero B_j and B_j' at all the Gaussian quadrature points of all the intervals $(x_l,x_{l+1}), \ldots, (x_{l+k-1},x_{l+k})$ at the top of the outer loop. This would require $2k(k-1)$ storage locations for storing the a and b values, but requires $2k^2(k-1)$ words to store all the non-zero B_j and B_j' values.

By re-ordering the loops to do the operations *interval by interval* we can construct the equations in the same number of operations and yet use less storage. On any interval (x_l,x_{l+1}) the only B_j which are non-zero there are for $j=l-k+1, \ldots, l$. Thus, we can form the Galerkin equations by

```
For $l=1, \ldots, N-k$
    {
    Evaluate and store $a$,$b$ and all non-zero $B_i$ and $B_i'$
    at the $k-1$ quadrature points of $(x_l,x_{l+1})$.
    For $i=l-k+1, \ldots, l$
        {
        For $j=l-k+1, \ldots, i$
            {
            Compute $\int_{x_l}^{x_{l+1}} (aB_i'B_j' - bB_iB_j)\,dx$
            using a $k-1$ point Gaussian quadrature rule.
            }
        }
    }
```

This uses only

$$\frac{k}{2}(k^2-1)(N-k) \tag{5.18}$$

operations, $(N-k)(k-1)$ evaluations of a and b, and $2(k-1)+2(k-1)k = 2(k^2-1)$ scratch storage locations.

The above outline for evaluating the Galerkin matrix, as well as the right-hand-side b, can be implemented in less than 35 FORTRAN statements, with the treatment of the boundary conditions requiring less than 35 FORTRAN statements as well. This is roughly twice as much FORTRAN code as the implementation of a centered finite difference scheme for solving (1.1)-(1.2). Thus, the benefits of Galerkin's method - higher order rate of convergence, greater accuracy, the ability to use a non-uniform mesh - come at a price which is not too high for the implementor of such techniques.

Galerkin's method, using B-splines, has been implemented for solving the boundary value problem for systems of linear differential equations in a single variable. This package is in use as the core of the time-varying partial differential equation solver POST [20].

6. Error Estimation

Now that we have a technique available for finding an approximate solution of (1.1)-(1.2) accurate to $O(h^k)$, and it is thus known to converge as $h \rightarrow 0$ to the true solution of the problem, a very important practical question arises:

For a given mesh π, how accurate is the computed Galerkin solution $\hat{y}$ based upon that mesh?

This is a tidy paraphrase of the eternal question asked by all users of computer software (and, sometimes, hardware):

I just spent X Kilobucks getting this basket of numbers, how good are they?

There is, in general, no way to guarantee, in a finite amount of computer time and memory (Dollars), that a given Galerkin solution is accurate to a certain amount. No matter how much treasure is expended, the coefficients a, b and c of the differential equation can only be sampled at a finite number of points, yet the solution y of (1.1)-(1.2) is very strongly dependent upon the value of these coefficients everywhere. Thus, no matter how carefully the coefficients are sampled, they may be sufficiently "kinky" between the sampled points that the solution y is radically different from $\hat{y}$.

Even though the above question is, in general, unanswerable in a definitive sense, users will continue to ask it and many will demand some sort of statement about the accuracy of the computed approximate solutions. This section makes a "reasonable" attempt to respond to this unanswerable question.

We want to estimate the error in a given computed Galerkin solution $\hat{y}$ to (1.1)-(1.2) over a mesh π. We know that the error $||y-\hat{y}|| = O(h^k)$, which simply means that there is a constant C so that as $h \rightarrow 0$ we have

$$||y-\hat{y}|| \leqslant Ch^k. \tag{6.1}$$

If we can estimate the constant C, then by (6.1) we will have an estimate for $||y-\hat{y}||$. The actual, observed behavior of the error as $h \rightarrow 0$ is

$$||y-\hat{y}|| = Ch^k. \tag{6.2}$$

This is an asymptotic statement about the observed behavior of the error as the mesh spacing approaches zero, but we shall assume that it holds for all meshes. Let π_1 and π_2 be two meshes, with $|\pi_2| < |\pi_1|$, and let y_i be the computed Galerkin approximate solution over the mesh π_i for $i=1,2$. If we set $\sigma = |\pi_2|/|\pi_1| < 1$, then we obtain

$$||y-y_2|| = C|\pi_2|^k = \sigma^k||y-y_1|| \tag{6.3}$$

and thus

$$||y-y_1|| \leqslant ||y-y_2||+||y_2-y_1||=\sigma^k||y-y_1||+||y_2-y_1||.$$

We also have from the triangle inequality,

$$||y-y_1|| \geqslant ||y_2-y_1||-||y-y_2|| = ||y_2-y_1||-\sigma^k||y-y_1||$$

which when combined with the previous inequality gives

$$||y_2-y_1||-\sigma^k||y-y_1|| \leqslant ||y-y_1|| \leqslant ||y_2-y_1||+\sigma^k||y-y_1|| \tag{6.4}$$

This relation gives

$$\frac{||y_2-y_1||}{1+\sigma^k} \leqslant ||y-y_1|| \leqslant \frac{||y_2-y_1||}{1-\sigma^k}. \tag{6.5}$$

Since typically $\sigma^k << 1$ we see from (6.5) that a very good estimate (upper bound) for $||y-y_1||$ is given by

$$||y-y_1|| \approx \frac{||y_2-y_1||}{1-\sigma^k} \tag{6.6}$$

The estimate given by (6.6) is obviously computable since both y_1 and y_2 are known B-splines. The estimate given by (6.6) is very accurate and reliable, as we shall see in section 7. It is accurate, asymptotically, because (6.2) *does* hold, in practice, very well. It is reliable because when (6.2) does not hold, the difference $||y_2 - y_1||$ is quite likely to be large, giving a large estimate of $||y - y_1||$ by (6.6).

The value of $||y_2 - y_1||$ need not be computed exactly, a good estimate for it will do just as well. Let π be the mesh which is the union of all the points in both π_1 and π_2. Then on each interval of the mesh π, the difference $y_1(x) - y_2(x)$ is just a polynomial of degree $k-1$. So estimating $||y_2 - y_1||$ can be reduced to the problem of estimating the maximum absolute value of a polynomial over an interval.

The Bernstein Inequality [1] may be used to estimate the norm of a polynomial over an interval. Specifically, the Bernstein Inequality states that for any polynomial $P_n(x)$ of degree n on the interval $[-1, +1]$, the trigonometric polynomial $T_n(\theta) = P_n(\cos(\theta))$ on $[-\pi, +\pi]$ obeys

$$||T_n'|| \leqslant n \, ||T_n||$$

If we sample $T_n(\theta)$ at m equally spaced points θ_i on $[0, \pi]$, then for any $\theta \epsilon [0, \pi]$ we see that, by the Mean Value Theorem,

$$|T_n(\theta) - T_n(\theta_i)| \leqslant ||T_n'|| \; |\theta - \theta_i|.$$

Thus,

$$||T_n|| \leqslant \underset{i = 1, \ldots, m}{Max} |T_n(\theta_i)| + n \, ||T_n|| \frac{\pi}{2(m-1)}$$

and for $m \geqslant n\pi + 1$, we see that

$$||T_n|| \leqslant 2 \underset{i = 1, \ldots, m}{Max} |T_n(\theta_i)|$$

Consequently, if we search the interval $[-1, +1]$ at $x_i = \cos(\theta_i)$, for $i = 1, \ldots, m$, we obtain

$$\underset{i = 1, \ldots, m}{Max} |P_n(x_i)| \leqslant ||P_n|| \leqslant 2 \underset{i = 1, \ldots, m}{Max} |P_n(x_i)|$$

and the norm of P_n may be estimated to within a factor of 2 by using a search pattern of $n\pi + 1$ points. Such a search, over each interval of the mesh π, is sufficient for our needs, and results in the computation of the error estimates to within a factor of 2.

The very first user complaint about using (6.6) to estimate the error in the Galerkin solution is that the Galerkin equations must be formed and solved for two different meshes in order to estimate the error in one of them. Just how big an overhead does (6.6) require above just solving for y_2, the numerical solution which will be used because it is the more accurate of y_1 and y_2? Well, typically $k \geqslant 4$ and $\sigma \equiv |\pi_2| / |\pi_1| \approx 2/3$, and thus $\sigma^k \approx 1/5$. Since the cost of obtaining a Galerkin solution is proportional (5.18) to the number of points in the mesh, the ratio of the cost of computing both y_1 and y_2 to just computing y_2 is 5/3. This, at first glance, appears to be a rather large price to pay for error estimates. However, the alternative is the relatively cheap computation of a pile of numbers of absolutely unknown accuracy. The situation may be summed up as:

We can get the (potentially) wrong answer cheaply, the correct answer costs more to obtain.

Another error estimation scheme that can be used is obtained from (6.6) by noting that $||y - y_2|| \approx \sigma^k ||y - y_1||$ and hence, from (6.6),

$$||y - y_2|| \approx \sigma^k \frac{||y_2 - y_1||}{1 - \sigma^k} \qquad (6.7)$$

This relation can be used to estimate the error in the most accurate approximate solution we have computed, y_2, rather than the least accurate, y_1. This seems like a good idea, to use the best information available, and in the limit as $h \rightarrow 0$, it is a good idea. However, (6.7) suffers from a serious defect. If the solution is not approximated well by either y_1 or y_2, and we have, for example, $\sigma = 2/3$ and $k = 6$, then $\sigma^k \approx 10^{-1}$ and we will estimate that

$$||y-y_2|| \approx ||y_2-y_1||/10$$

Thus, although both y_1 and y_2 may only be good to 10^{-1}, (6.7) will estimate $||y-y_2|| \approx 10^{-2}$. The problem of course is that the estimate (6.7) may be small (for $\sigma^k << 1$) when $||y_2-y_1||$ and $||y-y_2||$ are quite large. In section 7 we shall see that (6.7) must indeed be used with some care, but that (6.6) is extremely reliable. In any case, no matter which error estimate is believed, the more accurate y_2 values are used as the solution.

7. Numerical Examples

This section applies Galerkin's method, using B-splines, to two problems. The first example serves several purposes. First, it consists of 2 differential equations, each holding on one side of an "interface". Such interface problems occur frequently in practice and it is useful to see how Galerkin's method handles them. Second, it is a problem to which the exact answer is known and we can examine the convergence rate of Galerkin's method, and the error estimation schemes of section 6, for this simple problem. The last purpose served by the first example is to exhibit a "superconvergence" property [10,11,24] of Galerkin's method. We can then note that the error estimation schemes of section 6 are robust enough to detect and exhibit this property as well. The second example is cooked up to "break" the error estimation schemes of section 6. The extent to which those schemes break down is discussed and compared, showing that the second error estimation scheme of section 6 is in fact rather unreliable, while the first scheme is quite reliable.

The previous sections have described Galerkin's method for a rather clean, textbook problem, (1.1)-(1.2). Problems in real life are often not that simple. The first sample problem illustrates a common "unclean" problem which can be laundered into the form of (1.1)-(1.2). Consider the problem

$$\begin{aligned} y_1'' &= 0 \quad \text{on } (-1,0) \\ y_2'' - \frac{y_2}{2} + \frac{e^{x/2}}{4} &= 0 \quad \text{on } (0,+1) \end{aligned} \tag{7.1}$$

with $y_1(-1) = 0$ and $y_2(+1) = \sqrt{e} = \sqrt{2.7...}$, subject to interface conditions

$$y_1(0) = y_2(0) \tag{7.2a}$$

$$y_1'(0) = 2y_2'(0). \tag{7.2b}$$

The solution of this problem is

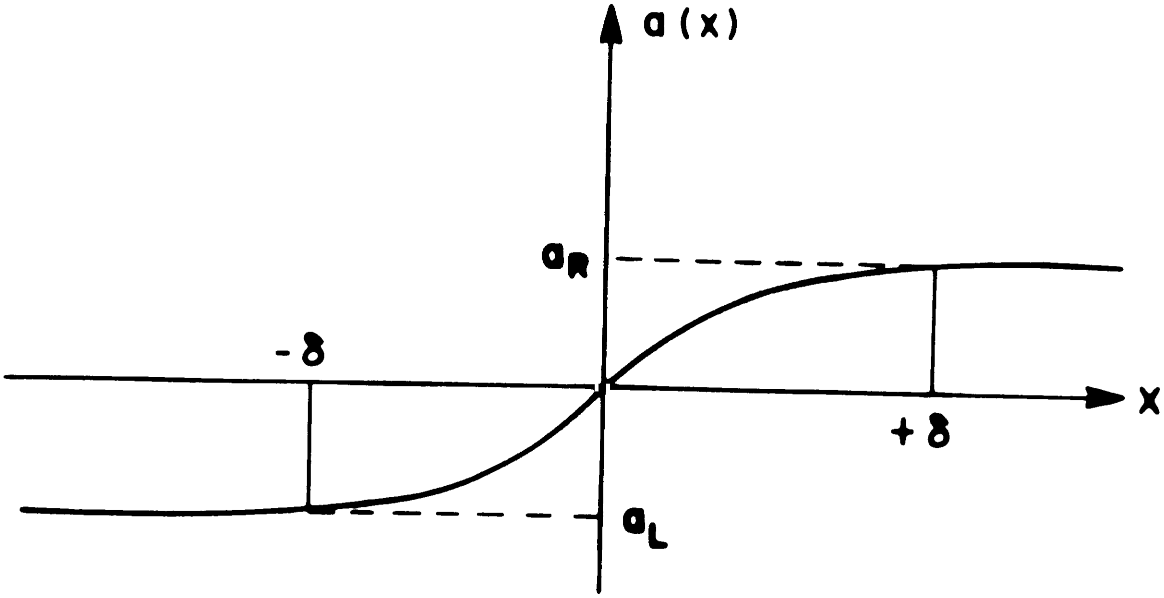
$$\begin{aligned} y_1(x) &= x+1 \quad \text{on } (-1,0) \\ y_2(x) &= e^{x/2} \quad \text{on } (0,+1). \end{aligned} \tag{7.3}$$

Interface conditions (7.2) do not fit into the problem formulation (1.1)-(1.2), which only allows conditions to be imposed on the solution of the differential equation at the end points of the interval.

To see how Galerkin's method can be used to solve (7.1)-(7.2) requires a little analysis, and handwaving. It is instructive to see how such interface problems arise in practice. In electrostatics problems [17] where there are two materials of different dielectric constants in contact, the dielectric constant is in fact *continuous* across the point of contact. However, the size of the transition region where the dielectric constant changes from one constant value to the other is exceedingly small. Let us assume that $a(x)$ represents this essentially piecewise constant dielectric constant and that the space-charge present is given by $f(x)$. Then Poisson's equation for the electrostatic potential y is

$$(a(x)y')' = f \quad \text{on } (-1,+1) \tag{7.4}$$

where the interval $(-1,+1)$ has been chosen for no particular reason. Further, let us assume that $a(x)$ looks like



where a_L and a_R represent the two dielectric constants and 2δ is the (small) width of the transition region. Then by taking $\int_{-\delta}^{+\delta}$ of (7.4) we have

$$ay' \,|_{-\delta}^{+\delta} = \int_{-\delta}^{+\delta} f(x)dx = O(\delta)$$

which gives

$$a_R y_R' (0) - a_L y_L' (0) = O(\delta). \tag{7.5}$$

The physical argument then says that the potential y is insensitive to the choice of δ in (7.4), so long as δ is small, and thus that we might as well take $\delta = 0$, and literally make $a(x)$ a step-function. But $\delta = 0$ gives, from (7.5),

$$a_R y_R'(0) = a_L y_L'(0), \tag{7.6}$$

which has precisely the same form as (7.2b). Thus, such interface conditions are a *consequence,* in the limit as $\delta \to 0$, of solving the differential equation, rather than a *condition* placed upon the solution. Specifically, it shows that a jump discontinuity in the coefficient, a, of y'' implies an interface condition of the same *form* as (7.2b). The trick is to re-write (7.1) so that its coefficient a of y'' obeys $a_L = 1$ and $a_R = 2$ at its jump discontinuity. It is also good to note that the above argument which led to (7.6) is independent of (did not use) the boundary conditions on y at $x = -1$ and $+1$.

Now suppose that Galerkin's method was applied, blindly, to an equation like (7.4), with a knot placed at 0. What would happen? Well, Galerkin's method will sample $a(x)$ at $k-1$ Gaussian quadrature points strictly inside the mesh intervals on either side of 0. From that information Galerkin's method cannot tell whether $a(x)$ is continuous or discontinuous at 0. In fact, for a given mesh, the Galerkin solution will be *independent* of what $a(x)$ does between the quadrature points immediately to the left and right of 0. Thus, we might as well assume that $a(x)$ is continuous, but changing from a_L to a_R over an exceedingly small interval. The argument which led to (7.6) then says that we should obtain the Galerkin approximation to the solution of (7.4) with $a_L y_1'(0) = a_R y_2'(0)$. This is precisely what we want of course.

However, we can just as well assume that $a(x)$ changes smoothly, not abruptly, from a_L to a_R between the Gaussian quadrature points on either side of the origin. In this case, we are trying to approximate another, slightly different, solution.

How does Galerkin's method know which solution to approximate?

The main difference between the two solutions, for $a(x)$ smooth and discontinuous, is of course in the continuity of y' at 0, the former being smooth and the latter being discontinuous. Remember that the continuity of a B-spline at a knot x_i is determined by the multiplicity m_i of that knot in the mesh. Specifically, for any B-spline u, we have that $u^{(0)}(x_i), \ldots, u^{(k-m_i-1)}(x_i)$ are all continuous at x_i, but all higher derivatives of u may be discontinuous at x_i. Recall that this forced, in general, the multiplicity of the first and last mesh points to be k and allowed $m_i = 1$ at all other knots of u where u is *smooth*. Here however we have a point where the solution is not smooth at a mesh point. Thus,

> *It is the multiplicity of the knot at* 0 *which determines the type of solution we are trying to approximate.*

Instead of choosing the multiplicity of 0 to be 1, we take it to be $k-1$, giving only $u^{(0)}$ continuous. With this choice for the multiplicity of 0 in the mesh, we can expect to find the solution y of (7.4) accurate to $O(h^k)$.

We can now re-formulate (7.1)-(7.2) as

$$\begin{aligned} &y_1'' = 0 \quad \text{on } (-1,0) \\ &2\,y_2'' - y_2 + \frac{e^{x/2}}{2} = 0 \quad \text{on } (0,+1), \end{aligned} \tag{7.7}$$

with $y_1(-1) = 0$, $y_2(+1) = \sqrt{e}$ and the knot at 0 of multiplicity $k-1$ in the B-spline mesh. This gives the ratio of the coefficients of y'' on the left and right sides of 0 as 2. The above handwaving then indicates that we should have $y_1'(0) = 2y_2'(0)$ which is precisely (7.2b). Note that (7.2a) is guaranteed by the choice of $k-1$ for the multiplicity of 0 in the mesh.

Figures 1,2 and 3 show the true error and the two error estimates of section 6 for $k = 2$, 4 and 6 of the Galerkin solution of (7.7), or (7.1)-(7.2), using N equally spaced mesh points on $(-1,+1)$. Thus, $h = 2/(N-1)$ and we should have the error $\approx C/(N-1)^k$ for some constant C. As the plots show, both error estimation schemes of section 6 give excellent results for this example. The rate of convergence for y_2 is as expected - $O(h^k)$ - since the slope of ln (error in y_2) versus ln $(N-1)$ is $-k$.

However, the error in y_1 is *too* good. In fact, the error for y_1 when $k = 6$ is at the rounding error level (about 10^{-18} for the Honeywell HIS 6070) for all N ! For $k = 2$ the error in y_1 is clearly $O(h^2)$ which is no surprise. For $k = 4$, the error in y_1 is clearly $O(h^6)$. This is an example of a "**superconvergence**" result. It is known [10,11,24] that the error *at the mesh points* x_i of the B-spline Galerkin solution is $O(h^{2(k-1)})$. This means for our problem that $\hat{y}_1(0)$, 0 being a mesh point, is accurate to $O(h^{2(k-1)})$. We also know that $\hat{y}_1(-1) = y_1(-1)$. The error in $\hat{y}_1$ at 0 and -1 is thus $O(h^{2(k-1)})$. But (7.7a) means that y_1 should be a straight line, and the Galerkin solution has this property as well. Thus $y_1 - \hat{y}_1$, being a straight line which is $O(h^{2(k-1)})$ at its end points, is simply $O(h^{2(k-1)})$. This result certainly agrees with the numerical evidence presented in figures 1-3. This shows that the error estimation schemes of section

6 can detect convergence rates which are different from the expected $O(h^k)$.

The above example exhibits all kinds of nice behavior - Galerkin's method works for such interface problems, the rate of convergence is as expected, or better, and it shows that the error estimation schemes of section 6 work properly.

While it is nice to see that a numerical technique works as advertised, it is much more interesting, fun and instructive to find out where the technique breaks down. The second example is designed to break the error estimation schemes of section 6. This is easily accomplished by trying to approximate a spiky function like $y = \sin(x)^m$, for some large m, on $[0, \pi]$. The equation

$$y'' - (sin(x)^m + 1)y + \sin(x)^{2m} + (m^2+1)\sin(x)^m - m(m-1)\sin(x)^{m-2} = 0 \qquad (7.8)$$

on $(0, \pi)$ subject to $y(0) = 0 = y(\pi)$, has the solution $y = \sin(x)^m$. Note that for large m, (7.8) is nearly the equation $y'' - y = 0$, for all x not near $\pi/2$, which has the solution $y = 0$, the wrong answer.

Clearly, by choosing m sufficiently large we can fool any technique for approximately solving (7.8), and/or estimating the error in the solution, which does not use the point $\pi/2$ in its computations. Using this test equation it is easy to see that the second error estimation scheme of section 6 is quite unreliable. By choosing $m = 10$, we obtain the results shown in figures 4-6. As those plots show, the second error estimation scheme sometimes grossly underestimates the error. The amount of underestimation goes up with k, as the handwaving in section 6 said it should, and the error for $k = 6$ is underestimated by two orders of magnitude! On the other hand, the worst underestimate by the first error estimation scheme of section 6 was, by less than a factor of 3, for $k = 6$.

Acknowledgments

James McKenna and James L. Blue have provided ample constructive criticism and have participated in a number of enlightening catharses. Blue is also responsible for the observation that interface conditions, discussed in section 7, are a natural consequence of the differential equations, rather than conditions to be placed upon their solution.

Bibliography

[1] E.W. Cheney, **Introduction to Approximation Theory,** McGraw-Hill, New York, 1966.

[2] P. G. Ciarlet, M.H. Schultz and R.S. Varga, "Numerical Methods of High-Order Accuracy for Nonlinear Boundary Value Problems I. One Dimensional Problem.", Numer. Math., 9, 394-430(1967).

[3] P. G. Ciarlet, M.H. Schultz and R.S. Varga, "Numerical Methods of High-Order Accuracy for Nonlinear Boundary Value Problems II. Nonlinear Boundary Conditions", Numer. Math., 11, 331-345(1968).

[4] P.G. Ciarlet, M.H. Schultz and R.S. Varga, "Numerical Methods of Higher-Order Accuracy for Nonlinear Boundary Value Problems III. Eigenvalue Problems", Numer. Math., 12, 120-133(1968).

[5] C. deBoor, "Best Approximation Properties of Spline Functions of Odd Degree", J. Math. and Mech., 12, 747-749(1963).

[6] C. de Boor, "On Uniform Approximation by Splines", J. Approx. Th., 1, 219-235(1968).

[7] C. de Boor, "On Calculating with B-splines", J. Approx. Th., 6, 50-62(1972).

[8] R. Courant and D. Hilbert, **Methods of Mathematical Physics,** Vol. 1, Interscience, New York, 1966.

[9] H.B. Curry and I.J. Schoenberg, "On Polya Frequency Functions IV: The Fundamental Spline Functions and their Limits", J. of Anal. and Math., 17, 71-107(1966).

[10] J. Douglas, T. DuPont and M.F. Wheeler, "An L_∞ Estimate and a Superconvergence Result for A Galerkin Method For Elliptic Equations Based on Tensor Products of Piecewise Polynomials", RAIRO, 8, 66-66(1974).

[11] T. DuPont, "A Unified Theory of Superconvergence for Galerkin Methods for Two-Point Boundary Problems", SIAM J. Numer. Anal., 13, 362-368(1976).

[12] G. Fix, "Higher-Order Rayleigh-Ritz Approximations", J. Math. and Mech., 18, 645-657(1969).

[13] G. Forsythe and C. Moler, **Computer Solution of Linear Algebraic Systems,** Prentice-Hall, 1967.

[14] G. Forsythe and W. Wasow, **Finite Difference Methods for Partial Differential Equations** , Wiley, New York, 1959.

[15] M.J. Marsden, "On Uniform Spline Approximation", J. Approx. Th., 6, 247-253(1972).

[16] J. McKenna and N.L. Schryer, "Analysis of Field-Aided Charge-Coupled Device Transfer", BSTJ, 54, 667-685(1975).

[17] P.M. Morse and H. Feshbach, **Methods of Theoretical Physics,** Mc Graw-Hill, New York, 1953.

[18] R.A. Sack and A.F Donovan, "An Algorithm for Gaussian Quadrature given Modified Moments", Numer. Math., 18, 465-478(1972).

[19] N.L. Schryer and L.R. Walker, "The Motion of 180^o Domain Walls in Uniform dc Magnetic Fields", J. Appl. Physics, 45, No. 12, 5406-5421(1974).

[20] N.L. Schryer, "Numerical Solution of Time-Varying Partial Differential Equations in One Space Variable", Bell Laboratories Computing Science Technical Report #53, 1976.

[21] M.H. Schultz, "The Galerkin Method for Non-Self-Adjoint Differential Equations", J. Math. Anal. and Appl., 28, 647-651(1969).

[22] M.H. Schultz, "The Condition of a Class of Rayleigh-Ritz-Galerkin Matrices", Bull. AMS, 76, 840-844(1970).

[23] G. Strang and G. Fix, **An Analysis of the Finite Element Method,** Prentice-Hall, New York, 1973.

[24] M.F. Wheeler, "L_∞ Estimates of Optimal Orders for Galerkin Methods for One-Dimensional Second Order Parabolic and Hyperbolic Equations", SIAM J. Num. Analy., 10, 908-913(1973).

[25] J.H. Wilkinson, **Rounding Errors in Algebraic Processes,** Prentice-Hall, New York, 1963.

Appendix 1

The Convergence of Rayleigh-Ritz for (3.18).

The Rayleigh-Ritz solution, $\hat{y}$, of (3.18) is the minimizer of (3.19) over $< B_2, \ldots, B_{N-k-1} >$. Thus, as before, $\hat{y}$ is the solution of

$$\int_0^1 (a\hat{y}'B_i' - b\hat{y}B_i - cB_i)\, dx = 0 \qquad \text{(A1.1)}$$

while the solution of (3.18) certainly satisfies

$$\int_0^1 (ay'B_i' - byB_i - cB_i)\, dx = 0 \qquad \text{(A1.2)}$$

by taking $\int_0^1 B_i\, dx$ of (3.18) and using integration by parts. Subtracting (A1.1) from (A1.2) we see that

$$\int_0^1 (a(y-\hat{y})'B_i' - b(y-\hat{y})B_i)\, dx = 0 \qquad \text{(A1.3)}$$

Thus, $\hat{y}$ minimizes the functional

$$F(\hat{y}) = \int_0^1 (a(y'-\hat{y}')^2 - b(y-\hat{y})^2)\, dx \qquad \text{(A1.4)}$$

over $< B_2, \ldots, B_{N-k-1} >$. Since $||a||$ and $||b||$ are assumed to be finite, we can show, precisely as before, that the minimum of this functional is $O(h^{k-1})$, and thus

$$F(\hat{y}) = O(h^{k-1})^2. \qquad \text{(A1.5)}$$

Now all we need to note is that, since $b \leqslant 0$ on $(0,1)$,

$$F(\hat{y}) \geqslant \int_0^1 a\, (y'-\hat{y}')^2 dx.$$

This in turn, since $a \geqslant \sigma$ on $(0,1)$, gives

$$F(\hat{y}) \geqslant \sigma \int_0^1 (y'-\hat{y}')^2 dx \qquad \text{(A1.6)}$$

Thus, by (3.16), (A1.5) and (A1.6),

$$||y-\hat{y}|| \leqslant (\frac{1}{\sigma} F(\hat{y}))^{\frac{1}{2}} = O(h^{k-1})$$

and we are done.

Appendix 2

To show $||B_i'(x)|| = O(h^{-1})$.

We have [6], when all $m_i = 1$ for $2 \leqslant i \leqslant N-k-1$,

$$B_i(x) = (x_{i+k} - x_i) \sum_{j=1}^{k+1} \frac{(x - x_{i+j-1})_+^{k-1}}{\prod_{l=1}^{j-1} (x_{i+j-1} - x_{i+l-1}) \prod_{l=j+1}^{k+1} (x_{i+j-1} - x_{i+l-1})}$$

and thus, for $k \geqslant 3$,

$$B_i'(x) = (k-1)(x_{i+k} - x_i) \sum_{j=1}^{k+1} \frac{(x - x_{i+j-1})_+^{k-2}}{\prod_{l=1}^{j-1} (x_{i+j-1} - x_{i+l-1}) \prod_{l=j+1}^{k+1} (x_{i+j-1} - x_{i+l-1})}$$

where ξ_+ is defined to be ξ for $\xi \geqslant 0$ and 0 for $\xi \leqslant 0$. From this we see that, using the mesh restriction (3.10),

$$\begin{aligned}
||B_i'(x)|| &\leqslant (k-1)kh \sum_{j=1}^{k+1} \frac{(x - x_{i+j-1})_+^{k-2}}{\prod_{l=1}^{j-1} (j-l) \prod_{l=j+1}^{k+1} (l-j)(h/\mu)^k} \\
&\leqslant k(k-1)h \sum_{j=1}^{k+1} \frac{(x_{i+k} - x_{i+j-1})_+^{k-2}}{(j-1)!(k+1-j)!(h/\mu)^k} \\
&\leqslant k(k-1)\mu^k h^{-1} \sum_{j=1}^{k+1} \frac{(k+1-j)^{k-2}}{(j-1)!(k+1-j)!} \\
&= O(h^{-1})
\end{aligned}$$

and we are done.

Appendix 3

To show $Cond(G) = O(h^{-2})$.

We have

$$G_{ij} = \int_L^R (\,aB_i'B_j' - bB_iB_j\,)\,dx \tag{A3.1}$$

and for any vector a

$$| \sum_j G_{ij}a_j \,| = | \sum_j a_j \int_L^R (\,aB_i'B_j' - bB_iB_j\,)\,dx \,| \tag{A3.2}$$

$$= | \sum_{|i-j| \leqslant k} a_j \int_L^R (\,aB_i'B_j' - bB_iB_j\,)\,dx \,|$$

But, since $||a||$ and $||b||$ are both finite, we have exactly as before,

$$\int_L^R aB_i'B_j'\,dx = O(h^{-1})$$

and

$$\int_L^R bB_iB_j\,dx = O(h),$$

and we have, using (A3.2),

$$||G|| = O(h^{-1}). \tag{A3.3}$$

To obtain an upper bound on $||G^{-1}||$ we note that, as before,

$$\sum_{i,j} a_i a_j G_{ij} = \int_L^R \sum_{i,j} a_i a_j (\,aB_i'B_j' - bB_iB_j\,)\,dx$$

$$= \int_L^R \left[a(\sum_i a_iB_i')^2 - b(\sum_i a_iB_i)^2 \right] dx$$

$$\geqslant \int_L^R a(\sum_i a_iB_i')^2 dx$$

$$\geqslant \frac{\sigma}{R-L} ||\sum_i a_iB_i||^2$$

using the fact that $b \leqslant 0$. But we also know from (2.8) that

$$||\sum_i a_iB_i|| \geqslant C_k||a||$$

for any vector a. This coupled with the above inequality gives,

$$\sum_{i,j} a_i a_j G_{ij} \geqslant C_k^2\sigma||a||^2/(R-L).$$

From this we see that, exactly as before,

$$||Ga|| \; ||a|| O(h^{-1}) \geqslant C_k^2 \sigma ||a||^2/(R-L)$$

and

$$||Ga|| \geqslant C_k^2 \frac{\sigma}{R-L} O(h) ||a||.$$

Thus,

$$||G^{-1}|| \leqslant C_k^{-2} \frac{R-L}{\sigma} O(h^{-1})$$

and using (A3.3)

$$Cond(G) = O(h^{-2})$$

and we are done.

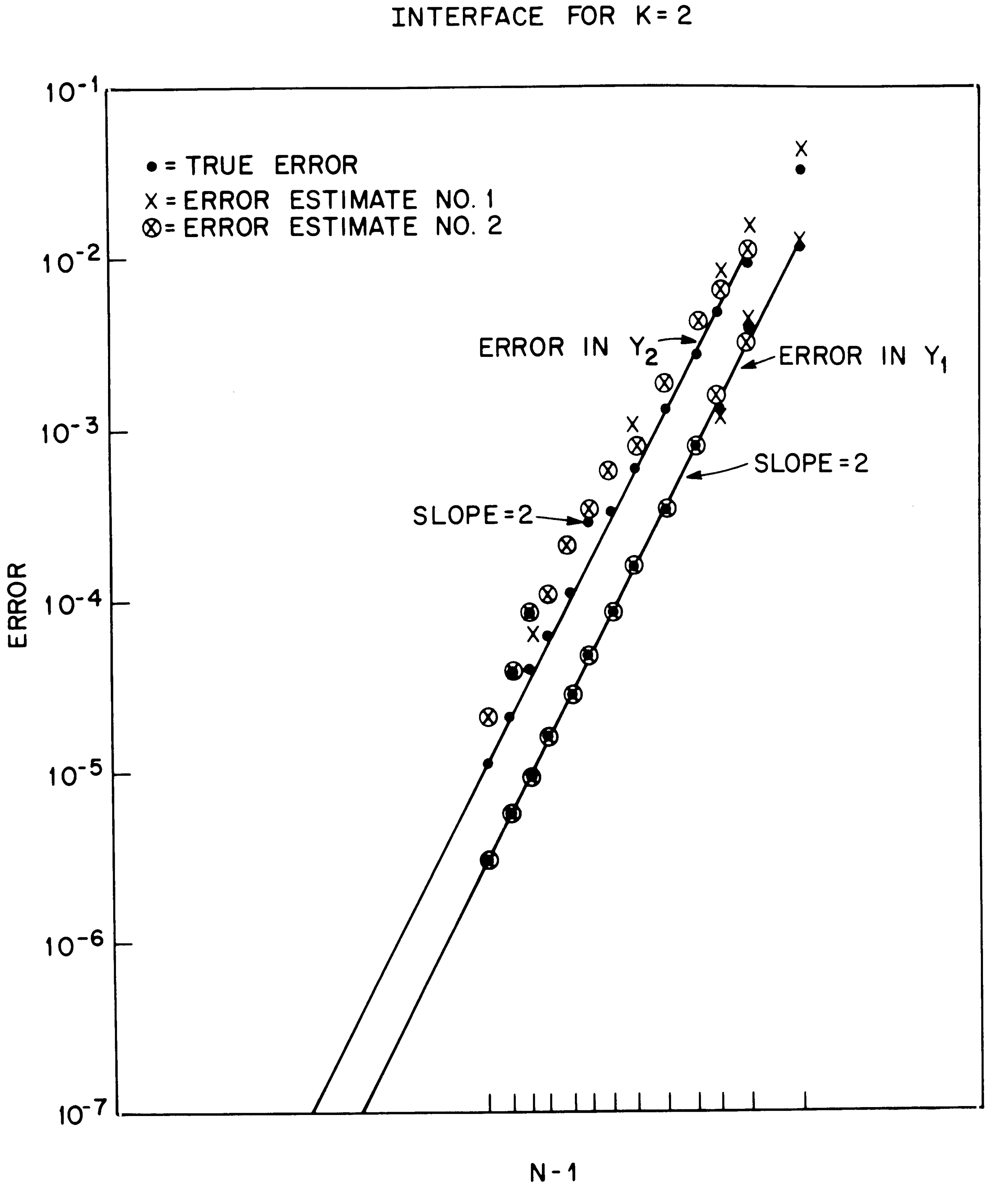

INTERFACE FOR K=2
● = TRUE ERROR
X = ERROR ESTIMATE NO. 1
⊗= ERROR ESTIMATE NO. 2
ERROR IN Y_2
ERROR IN Y_1
SLOPE = 2
SLOPE = 2
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
ERROR
N - 1

FIGURE 1

INTERFACE FOR K=4

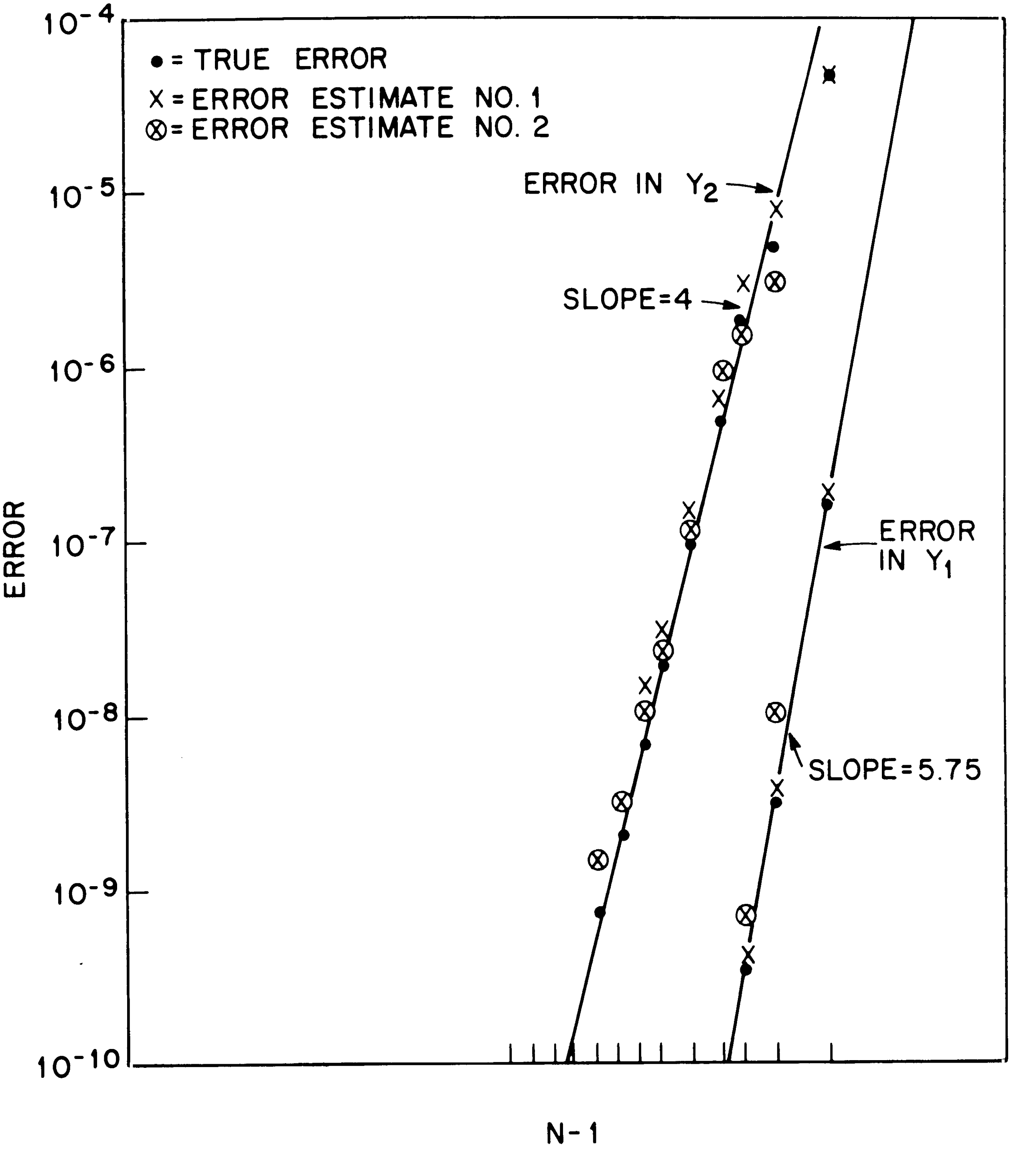
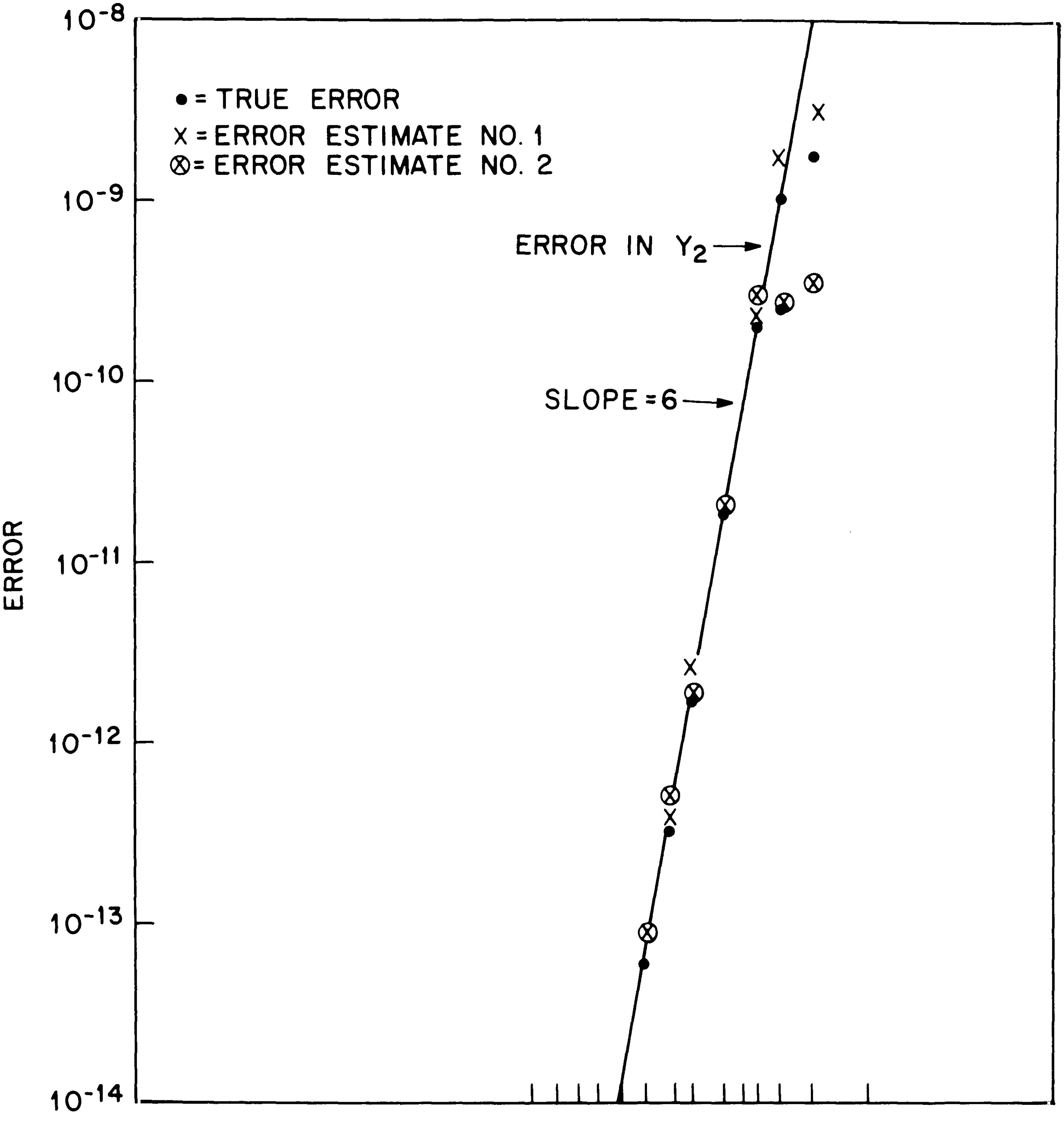


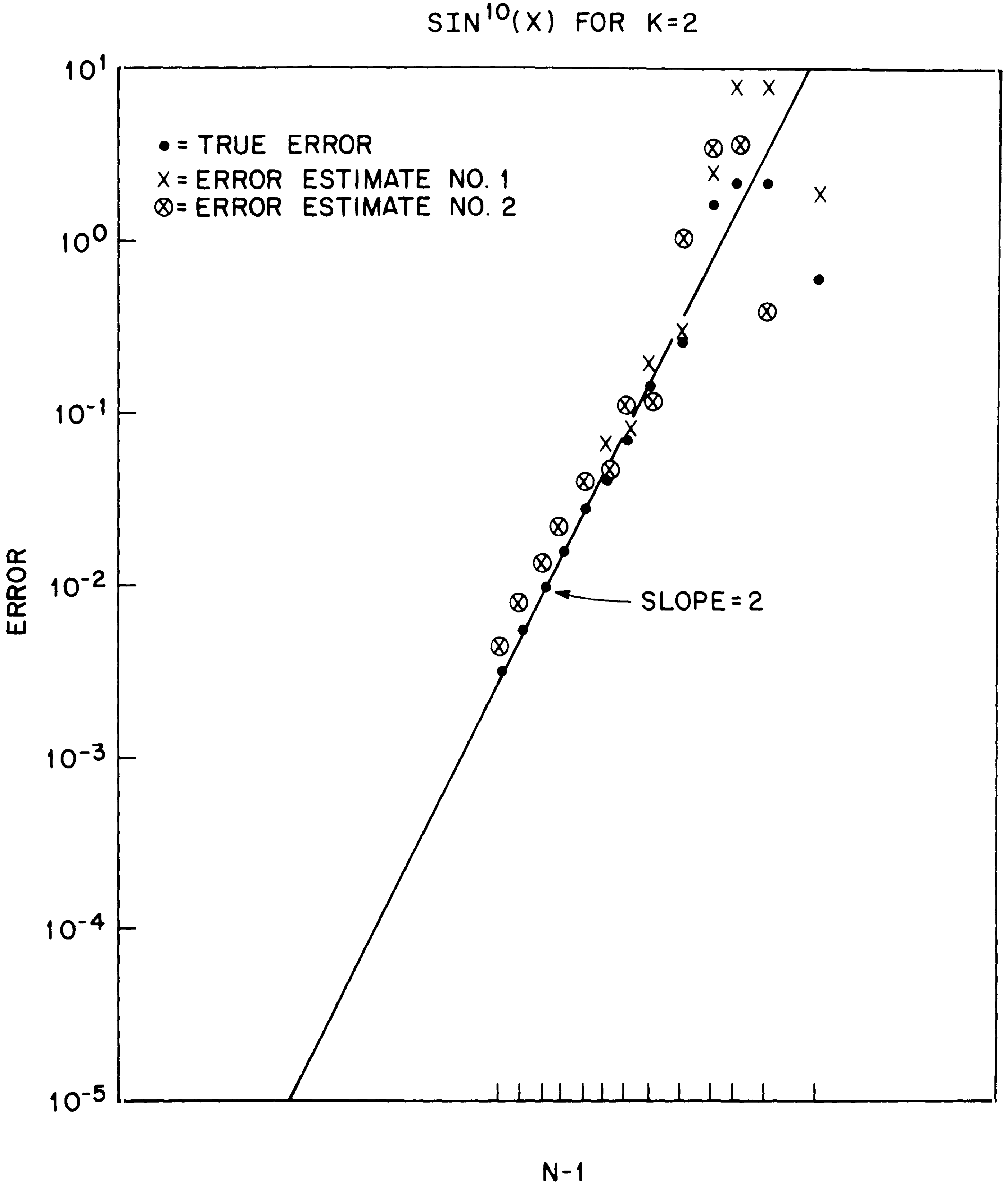
FIGURE 2

INTERFACE FOR K=6



N-1

FIGURE 3



$SIN^{10}(X)$ FOR K=2

FIGURE 4

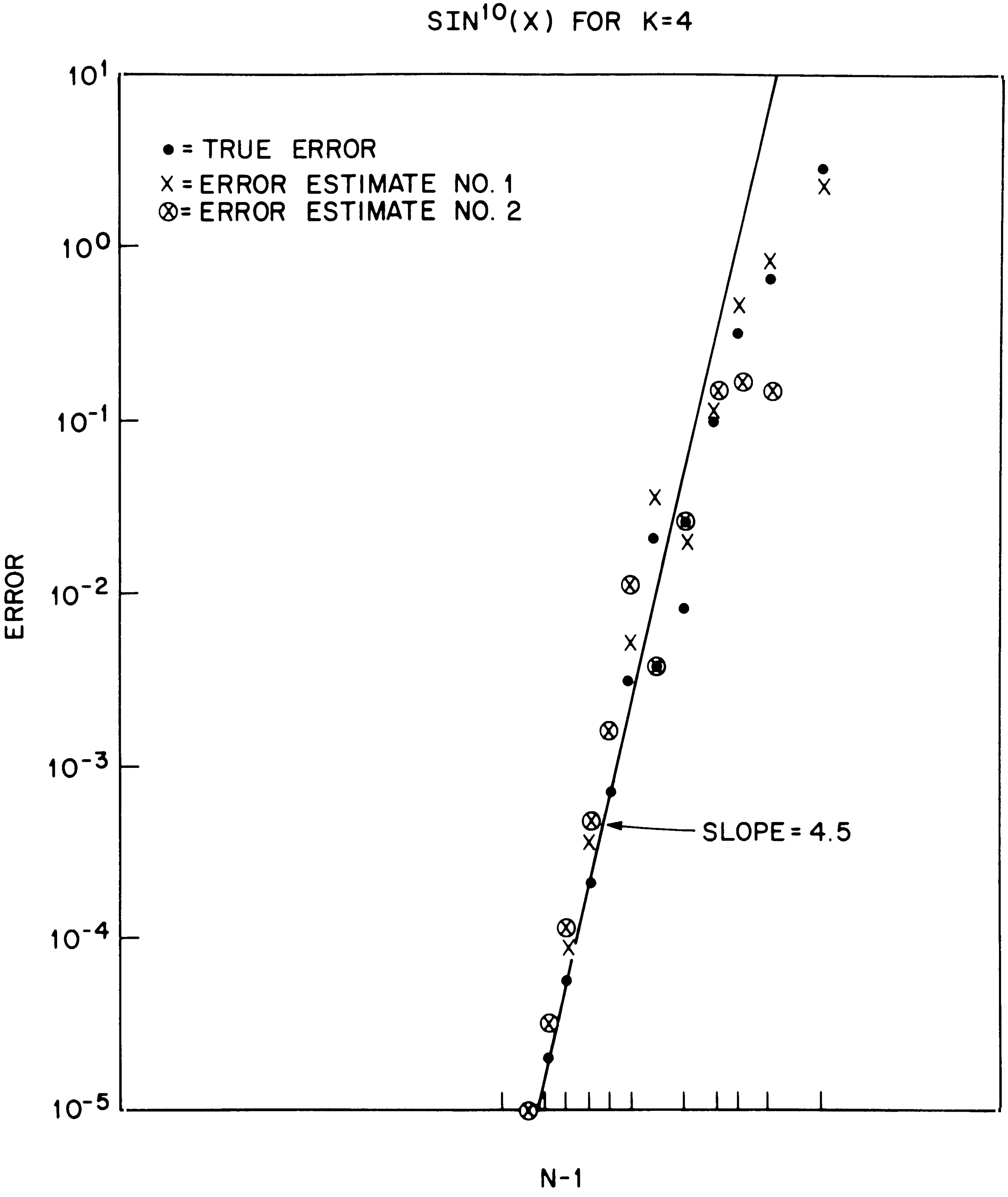

SIN^{10}(X) FOR K=4
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
ERROR
● = TRUE ERROR
X = ERROR ESTIMATE NO. 1
⊗ = ERROR ESTIMATE NO. 2
SLOPE = 4.5
N-1

FIGURE 5

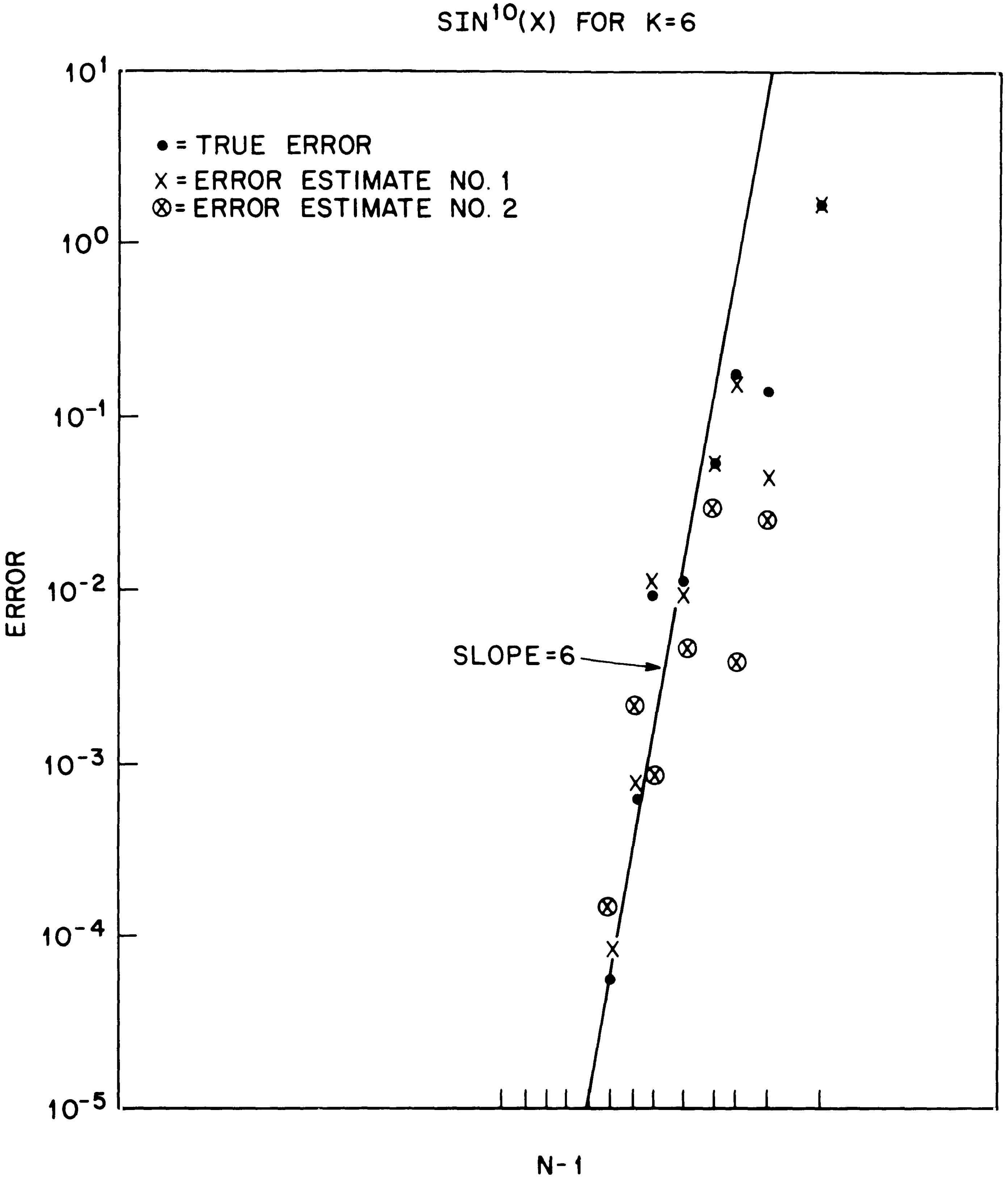

SIN^{10}(X) FOR K=6
● = TRUE ERROR
X = ERROR ESTIMATE NO. 1
⊗ = ERROR ESTIMATE NO. 2
SLOPE=6
ERROR
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
N-1

FIGURE 6