Spectral Interpolation

James F. Epperson

April 15, 2014

Introduction, Algorithms, and Examples

In Chapter 4 we constructed some least squares approximations to functions, using orthogonal polynomials as our basis functions, and found that the accuracy of the approximations was quite good. In these pages we will look at what is generally known as "spectral interpolation," by which we mean interpolation using special nodes and polynomials. Our goal is to see if we can obtain a significant improvement in the accuracy of our approximation without a significant increase in computational cost.

We will use the Chebyshev polynomials $T_k(x)$, introduced in §4.11, as our basic building blocks, so we will restrict ourselves to the interval $[-1, 1]$. We will use two sets of nodes:

1. The Lobatto nodes:

$$x_k = -\cos\left(\frac{k\pi}{N}\right), \quad k = 0, 1, \ldots, N \tag{1}$$

These are new, and were first introduced by the Dutch mathematician Rehuel Lobatto (1797–1866) in the early 1850s as part of a quadrature scheme.[1]

2. The Chebyshev nodes:

$$x_k = -\cos\left[\frac{(2k+1)\pi}{2N+2}\right], \quad k = 0, 1, \ldots, N \tag{2}$$

These were introduced in §4.12.3.

The minus signs are used to force the indexing to be from left to right on the interval $[-1, 1]$.

Remember that the interpolating polynomial is unique, so what we produce here is the same as we would get from the Lagrange (§4.1) or Newton (§4.2) forms in Chapter

[1]Rehuel Lobatto (6/6/1797–2/9/1866) was a Dutch mathematician, born in Amsterdam to a Portuguese family of Jewish extraction. Athough he never completed an ordinary doctorate, he published over 60 articles in various journals between 1823 and his death in 1866. In 1842, after receiving an honorary doctorate from the University of Gronigen in 1834, he became professor of mathematics at the newly founded Royal Academy in Delft. His work on quadrature was published as "Lessen over de differentiaal-en integraalrekening" in the early 1850s.

4. The advantage of the spectral approach is in the choices of nodes. (The choice of the Chebyshev polynomials can lead to some computational savings, but those issues will not be addressed.)

Much of what we do here is based on material in [1] and [2].

The algorithms for computing the interpolants can be summarized as follows.

1. Lobatto nodes: The polynomial is defined by

$$P_N(x) = \sum_{n=0}^{N}{}'' b_n T_n(x) \tag{3}$$

where the coefficients are defined by

$$b_n = \frac{2}{N} \sum_{k=0}^{N}{}'' f(x_k) T_n(x_k) \tag{4}$$

The double-prime notation means that the first and last entries of the sum are multiplied by $\frac{1}{2}$.

2. Chebyshev nodes: The polynomial is defined by

$$Q_N(x) = \sum_{n=0}^{N}{}' c_n T_n(x) \tag{5}$$

where the coefficients are defined by

$$c_n = \frac{2}{N+1} \sum_{k=0}^{N}{}'' f(x_k) T_n(x_k) \tag{6}$$

The prime notation means that the first entry of the sum is multiplied by $\frac{1}{2}$. We will derive these formulas later in this document, based on material in [2].

Let's begin by looking at some examples.

Example: Consider $f(x) = e^x$ over the interval $[-1, 1]$. Using the computations outlined above, we can easily construct spectral interpolates using either grid. The errors, as estimated by sampling at 2000 equally-spaced points, are given in Table 1 for $N = 4,\ 8,\ 12,\ 16$; Figure 1 shows the interpolate and function and the error plots for both sets of nodes, for $N = 8$ (in all these figures, the results for the Lobatto nodes are on the left, and for the Chebyshev nodes they are on the right). Note that, in all the examples, the errors at the nodes (marked with a red 'o') are exactly zero.

Table 1: Errors in spectral interpolation to $f(x) = e^x$.

N	Error (Lobatto nodes)	Error (Chebyshev nodes)
4	1.0659518054e-03	6.3969948255e-04
8	2.2029401858e-08	1.2190087162e-08
12	8.0380146983e-14	4.7073456244e-14
16	5.5511151231e-15	2.0428103653e-14

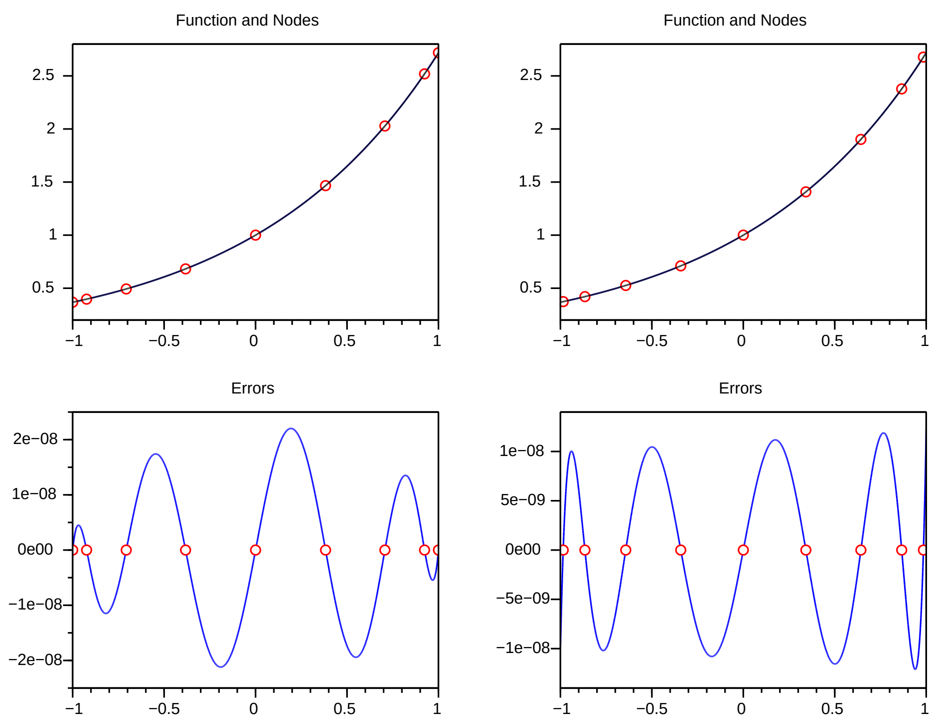

Figure 1: Plots of spectral interpolation to $f(x) = e^x$.

Example: Consider now $f(x) = \frac{1}{1+25x^2}$, which we referred to as "the Runge example" in Chapter 4. Our usual interpolation had a bit of difficulty with this function. How do the spectral methods fare? Table 2 gives the same error information as for our first example. The errors are not as small as above, but the accuracy is much better than with Newton interpolation. The figure this timne is for $N = 16$.

Table 2: Errors in spectral interpolation to $f(x) = \frac{1}{1+25x^2}$.

N	Error (Lobatto nodes)	Error (Chebyshev nodes)
4	4.5998051841e-01	4.0201674194e-01
8	2.0468170483e-01	1.7083373973e-01
12	8.4395954222e-02	6.9215707808e-02
16	3.6712899069e-02	3.2613370682e-02

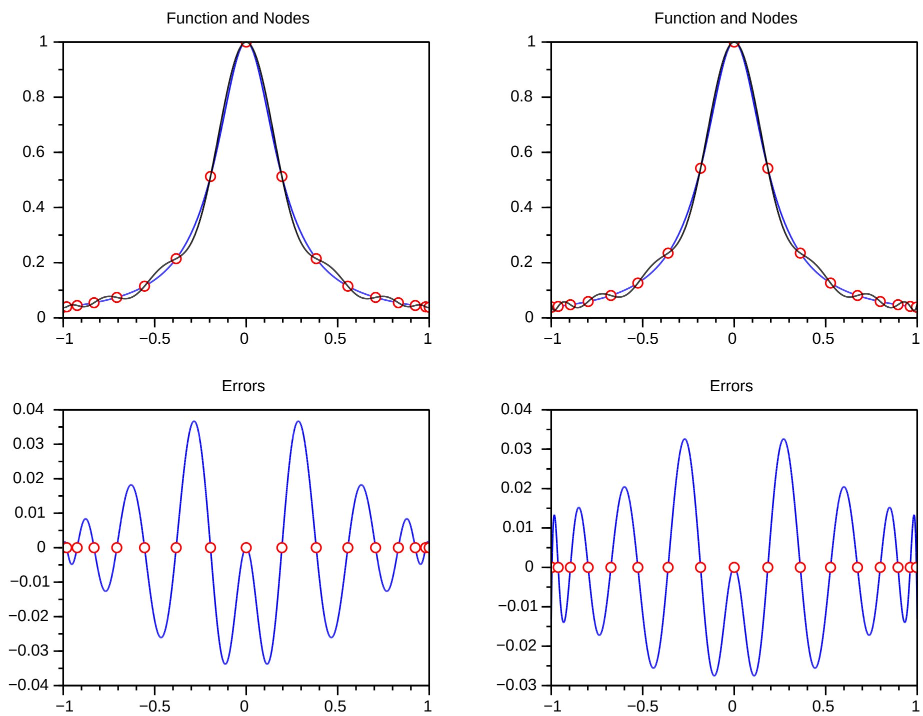

Figure 2: Plots of spectral interpolation to $f(x) = \frac{1}{1+25x^2}$, for $N = 16$.

Example: Consider now $f(x) = \frac{e^{10x}}{1+e^{10x}}$, which is often referred to as a "sigmoid" function, because it is roughly S-shaped. Table 3 gives the results of our interpolations, and Fig. 3 shows the function, interpolate, and the error for $N = 8$.

Table 3: Errors in spectral interpolation to $f(x) = \frac{e^{10x}}{1+e^{10x}}$.

N	Error (Lobatto nodes)	Error (Chebyshev nodes)
4	2.0418793004e-01	1.7700643107e-01
8	7.7954779351e-02	6.3182166100e-02
12	2.5604089310e-02	2.0190187202e-02
16	7.8768570291e-03	6.1405010074e-03

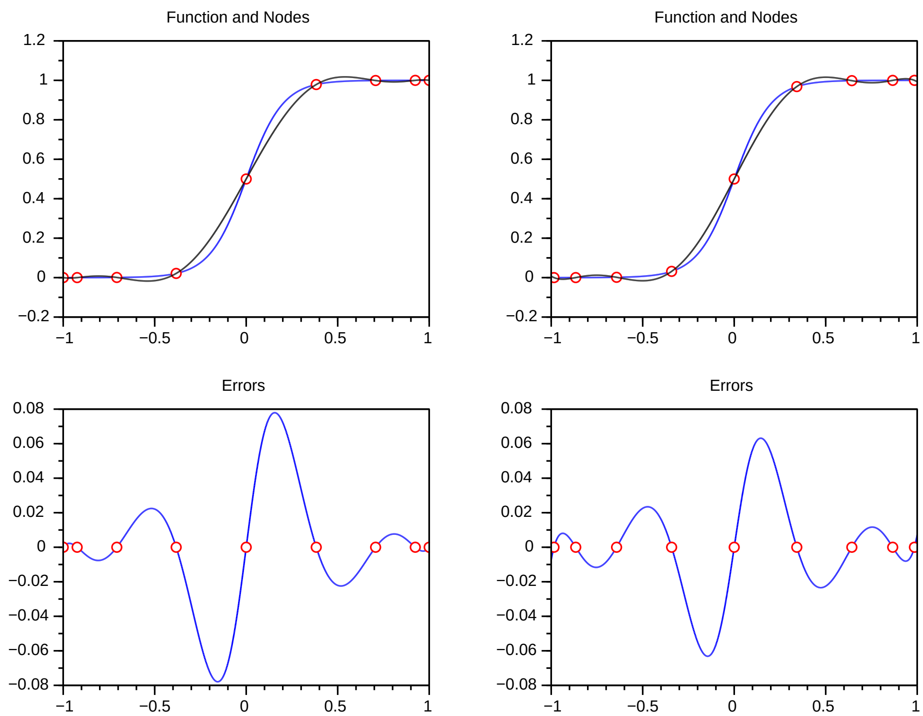

Figure 3: Plots of spectral interpolation to $f(x) = \frac{e^{10x}}{1+e^{10x}}$, for $N = 8$.

Example: Consider now the function $f(x) = 1 - \beta e^x - \beta e^{-x}$, for $\beta = \frac{e}{e^2+1}$. This is the exact solution for the first example in §11.1 of the text. We get the errors in Table 4. Fig. 4 shows the solution and errors for $N = 8$.

Table 4: Errors in spectral interpolation to $f(x) = 1 - \beta e^x - \beta e^{-x}$, for $\beta = \frac{e}{e^2+1}$.

N	Error (Lobatto nodes)	Error (Chebyshev nodes)
4	4.5105466649e-05	5.8554408689e-05
8	6.6349846450e-10	7.1497261044e-10
12	2.1926904736e-15	2.1649348980e-15
16	7.7715611724e-16	1.3877787808e-15

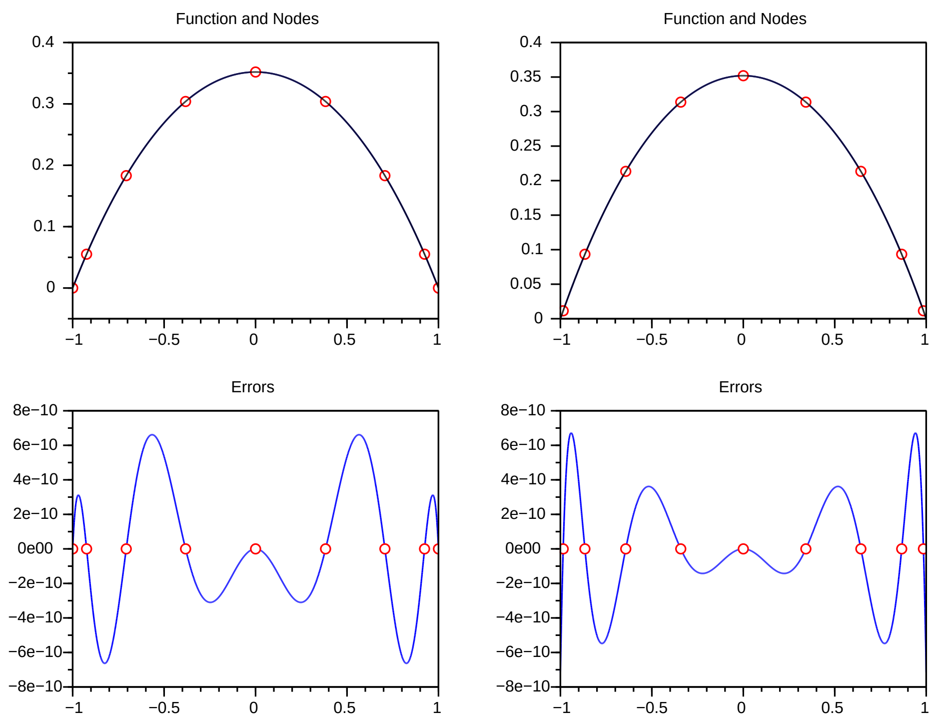

Figure 4: Plots of spectral interpolation to $f(x) = 1 - \beta e^x - \beta e^{-x}$, for $N = 8$.

Exponential convergence

In spectral methods one often reads of “exponential convergence.” What does this mean? We can illustrate the idea with some more data from our examples.

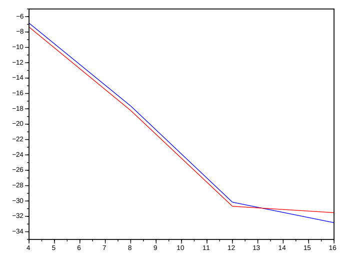

Figure 5: $\log E_N$ vs. N for $f(x) = e^x$, and $N = 4, 8, 12, 16$.

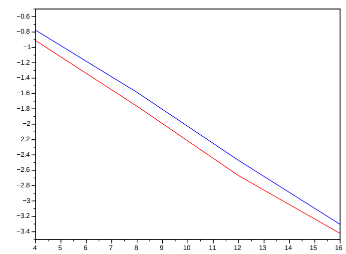

Figure 6: $\log E_N$ vs. N for $f(x) = \frac{1}{1+25x^2}$, and $N = 4, 8, 12, 16$.

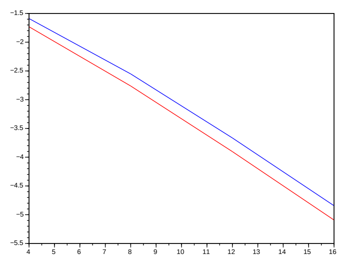

Figure 7: $\log E_N$ vs. N for $f(x) = \frac{e^{10x}}{1+e^{10x}}$, and $N = 4, 8, 12, 16$.

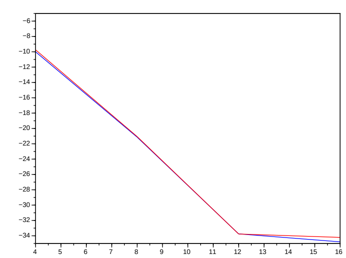

Figure 8: $\log E_N$ vs. N for $f(x) = 1 - \beta e^x - \beta e^{-x}$, and $N = 4, 8, 12, 16$.

If we plot the logarithm of the error versus N for, say, the Runge function $f(x) = \frac{1}{1+25x^2}$, we get the plot in Fig. 6. Note that both lines are nearly straight lines, implying that

$$\log |E_N| \approx aN + b$$

for some constants a and b, with $a < 0$ (since the slope is obviously negative). We therefore have

$$|E_N| \approx Be^{aN},$$

for $B = e^b$. In other words, the error decays (recall $a < 0$) *exponentially* with N. (We have not proved this, only demonstrated it by example; but it is rigorously true.) Is this an artifact of the particular example? No, it is not, as the graphs plus the material in the previous section indicate. All four example functions show the expected rapid exponential convergence. (The apparent “flattening” of the error for the first and last examples can be explained, below.) This is why spectral methods for various differential equation problems deliver such high accuracy for such a small investment in computational cost.

Where does this high accuracy come from? Recall §4.12.3, where the text presented the Chebyshev nodes (2), and Theorem 4.11 in particular, in which we proved that the error in interpolation using the Chebyshev nodes was bounded according to

$$\|f - Q_N\|_\infty \leq \frac{1}{2^N(N+1)!}\|f^{(N+1)}\|_\infty; \tag{7}$$

here, f is the function being interpolated, and Q_N is the interpolate using $N+1$ nodes. The exponential convergence comes from the 2^N factor together with the factorial in the denominator. A similar result holds for the Lobatto nodes, as we shall now prove.

Theorem 1 *Let $f \in C^{N+1}([a,b])$ and let the nodes $x_k \in [-1,1]$ be the Lobatto nodes (1). Then, for each $x \in [-1,1]$,*

$$\|f - P_N\|_\infty \leq \frac{N^2}{2^{N-1}(N+1)!}\|f^{(N+1)}\|_\infty, \tag{8}$$

where P_N is the interpolate defined by the values $f(x_k)$, $0 \leq k \leq N$.

Proof: The Polynomial Interpolation Error Theorem (Theorem 4.3) says that there is a value $\xi_x \in [-1,1]$ such that

$$f(x) - P_N(x) = \frac{w_N(x)}{(N+1)!}f^{(N+1)}(\xi_x),$$

where

$$w_N(x) = \prod_{k=0}^{N}(x - x_k).$$

Clearly the key is going to be bounding w_N.

Let’s assume that N is even, so $N+1$ is odd[2]. The “interior” nodes are the points where $T_N(x) = \pm 1$, i.e., the points where the “ monic polynomial” $\hat{T}_N$ hits its extrema, thus $\hat{T}_N'(x) = 0$. Thus

$$w_N(x) = (x^2 - 1)\hat{T}_N'(x).$$

But recall Theorem 4.10, part 3: $T_n(x) = 2^{n-1}x^n +$ lower-order terms. Therefore $\hat{T}_N'(x) = N\left(\frac{1}{2^{N-1}}\right)x^{N-1} +$ lower-order terms, or, $\hat{T}_N'(x) = \left(\frac{N}{2^{N-1}}\right)T_N'(x)$. So, with a little work, we get

$$w_N(x) = \frac{N(x^2-1)}{2^{N-1}}T_N'(x).$$

Direct application of calculus to the definition of the Chebyshev polynomials yields

$$T_N'(x) = -\sin\left(N\cos^{-1}x\right)\left[\frac{-N}{\sqrt{1-x^2}}\right].$$

[2]The proof changes in an obvious way if N is odd.

So, finally,

$$|w_N(x)| = \frac{N(1-x^2)}{2^{N-1}} \left| \sin\left(N\cos^{-1}x\right)\left[\frac{-N}{\sqrt{1-x^2}}\right]\right| \le \frac{N^2}{2^{N-1}}\left|1-x^2\right|^{3/2} \le \frac{N^2}{2^{N-1}},$$

and we have (8) •

We can use Stirling's Formula (§5.5) to rigorously establish that

$$\|f - Q_N\|_\infty \le A_N \left(\frac{e}{2N+2}\right)^{N+1} \|f^{(N+1)}\|_\infty,$$

where A_N is bounded and decays to zero like $1/\sqrt{N}$. This predicts a *very* high degree of accuracy (for sufficiently smooth f). Similarly, we can show that

$$\|f - P_N\|_\infty \le B_N N^{3/2} \left(\frac{e}{2N+2}\right)^{N+1} \|f^{(N+1)}\|_\infty,$$

where B_N is positive and bounded as $N \to \infty$. Again, this yields very high accuracy for smooth functions f.

The error plots for both the exponential and the ODE solution (Figs. 5 and 8) show what appears to be a decrease in the rate of convergence: The error appears to "flatten out" as we go from the $N = 12$ to the $N = 16$ case, for both examples. Is something going wrong here? Well, yes and no. The approximations here are so accurate that we are into "the noise" in the computation—the rounding error is comparable to the mathematical error—and we have reached the limits of our accuracy. If we could set the code to compute in higher precision, then we would see these error lines both straighten out.

The error bounds suggest that the Chebyshev nodes would be superior to the Lobatto nodes by a factor of about $N^2/2$, but the evidence from our examples suggests neither node set is consistently superior in practice. The Lobatto nodes, because they involve the boundary points ± 1, are generally preferred.

Derivation of Interpolation Formulas

Note: As of 4/4/2014, this section is still very sketchy. The results are valid, but I have yet to produce proofs that I am happy with. I will post this to the web site for the sake of "getting it up there," with the idea that the full proofs can follow, later.

The first thing we need to do is establish a trig identity which is the first step in constructing the interpolation formulas.

Lemma 1 *Define*

$$S_N(\theta) = \sum_{k=0}^{N} \cos\left(k + \frac{1}{2}\right)\theta;$$

then, for all θ and all $N \ge 0$,

$$S_N(\theta) = \frac{\sin\,(N+1)\theta}{2\sin\frac{1}{2}\theta} \tag{9}$$

Proof: Since the identity depends on an integer N, we can use induction. For $N = 0$ we have

$$S_0(\theta) = \sum_{k=0}^{0} \cos\left(k + \frac{1}{2}\right)\theta = \cos\frac{1}{2}\theta = \frac{\cos\frac{1}{2}\theta\sin\frac{1}{2}\theta}{\sin\frac{1}{2}\theta} = \frac{2\cos\frac{1}{2}\theta\sin\frac{1}{2}\theta}{2\sin\frac{1}{2}\theta} = \frac{\sin\theta}{2\sin\frac{1}{2}\theta}$$

and the "inductive hypothesis" is confirmed for $N = 0$.

Now we will assume (9) is true for $N = n$, and use this to prove that it is true for $N = n + 1$, which completes the inductive proof. We have:

$$\begin{aligned} S_{n+1}(\theta) &= S_n(\theta) + \cos\left(n + \frac{3}{2}\right)\theta \\ &= \frac{\sin(n+1)\theta}{2\sin\frac{1}{2}\theta} + \cos\left(n + \frac{3}{2}\right)\theta \\ &= \frac{\sin(n+1)\theta}{2\sin\frac{1}{2}\theta} + \frac{2\cos\left(n + \frac{3}{2}\right)\theta\sin\frac{1}{2}\theta}{2\sin\frac{1}{2}\theta} \end{aligned}$$

But

$$2\sin A\cos B = \sin(A - B) + \sin(A + B),$$

so

$$2\cos\left(n + \frac{3}{2}\right)\theta\sin\frac{1}{2}\theta = -\sin(n+1)\theta + \sin(n+2)\theta.$$

Therefore,

$$S_{n+1}(\theta) = \frac{\sin(n+2)\theta}{2\sin\frac{1}{2}\theta},$$

which was to be proved. •

The formulas used for the interpolations are based on a pair of "discrete orthogonality" results, which we now state. The interpolation formulas then follow almost immediately.

Lemma 2 (Discrete orthogonality for the Lobatto nodes) *Define the Lobatto nodes as*

$$\lambda_k = -\cos\left(\frac{k\pi}{N}\right), \quad k = 0, 1, \ldots, N;$$

then for all $N > 0$ *and all* $0 \leq j, k \leq N$*, we have*

$$\sum_{n=0}^{N}{}'' T_j(\lambda_n) T_k(\lambda_n) = K_{\lambda,j,k}\delta_{jk},$$

where

$$K_{\lambda,j,k} = \begin{cases} N & j = k = 0, \\ N & j = k = N, \\ N/2 & 1 \leq j = k \leq N-1, \end{cases}$$

Proof: This follows from some clever manipulations with the trig identity and the definition of the Chebyshev polynomials as cosines. (To be finished, later.)

Lemma 3 (Discrete orthogonality for the Chebyshev nodes) *Define the Chebyshev nodes as*

$$\gamma_k = -\cos\left[\frac{(2k+1)\pi}{2N+2}\right], \quad k = 0, 1, \ldots, N;$$

then for all $N > 0$ and all $0 \le j, k \le N$, we have

$$\sum_{n=0}^{N} T_j(\gamma_n) T_k(\gamma_n) = K_{\gamma,j,k}\delta_{jk},$$

where

$$K_{\gamma,j,k} = \begin{cases} N+1 & j = k = 0, \\ \frac{1}{2}(N+1) & 1 \le j = k \le N, \end{cases}$$

Proof: (To appear, later.)

We can easily use the discrete orthogonality relations to derive the formulas (4) and (6). We'll do it, informally, for the Lobatto nodes—the same argument works for the Chebyshev nodes.

The interpolation polynomial is defined by

$$P_N(x) = \sum_{k=0}^{N}{}'' b_k T_k(x),$$

so, in particular,

$$P_N(\lambda_n) = \sum_{k=0}^{N}{}'' b_k T_k(\lambda_n),$$

and therefore

$$T_j(\lambda_n) P_N(\lambda_n) = \sum_{k=0}^{N}{}'' b_k T_j(\lambda_n) T_k(\lambda_n).$$

This is true for all n, so we sum up over $0 \le n \le N$ to get

$$\sum_{n=0}^{N} T_j(\lambda_n) P_N(\lambda_n) = \sum_{n=0}^{N} \sum_{k=0}^{N}{}'' b_k T_j(\lambda_n) T_k(\lambda_n).$$

Now, the fact that P_N interpolates at the nodes means that $P_N(\lambda_n) = f(\lambda_n)$; if we interchange the sums on the right, we get

$$\sum_{n=0}^{N} T_j(\lambda_n) f(\lambda_n) = \sum_{k=0}^{N} b_k \left(\sum_{n=0}^{N}{}'' T_j(\lambda_n) T_k(\lambda_n) \right).$$

The term in parentheses can be eliminated by the discrete orthogonality relationship to give us

$$\sum_{n=0}^{N} T_j(\lambda_n) f(\lambda_n) = K_{\lambda,j,k} b_k.$$

So, we finally have

$$b_k = \frac{1}{K_{\lambda,j,k}} \sum_{n=0}^{N} T_j(\lambda_n) f(\lambda_n),$$

which is equivalent to (4).

Note: The author finally bought a new laptop in the Winter of 2014, and discovered that his ancient-but-still-serviceable version of MATLAB would not install under Windows 7. While there exist various options to work-around this problem by using various OS emulators (and I intend to pursue those options, eventually), I took this as a good reason to install and learn SciLab (`http://www.scilab.org/`), a freeware program that does much of what MATLAB does. All of the examples in this section were done using SciLab. (I do find MATLAB easier to work with, but it is hard to beat SciLab's price as an individual.)

References

[1] Boyd, John P., *Chebyshev and Fourier Spectral Methods*, Dover Press, 2001.

[2] Fox, L. and Parker, I. B., *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, London, 1968.