Notes on numerical integraton

Kevin Long

Texas Tech University

These notes supplement Ch 6 of Ackleh *et al.* by adding my own "big picture" view of the subject, a little theory, a little more practical information about how we actually find Gauss rules, and some cool stuff about Euler-Maclaurin and integration of periodic functions.

The fundamental idea in numerical integration of a function $f(x) \in C^k$ is this: construct a simple approximation to the function, and integrate the approximation instead of the function. The approximations used will almost always be interpolants rather than, say, least-squares approximants, for the simple reason that computation of a least-squares approximant requires integrations which, given that we're trying to approximate an integral, we presumably don't know how to do.

Numerical integration is also called quadrature. Numerical integration in higher dimensions is sometimes called cubature.

1 Building blocks

I will summarize a few results from Ackleh *et. al.*, chapter 4.

The Lagrange interpolating polynomial and remainder

Assume a set of nodes $\{x_i\}_{i=0}^N$, $x_i \in [a,b]$ has been chosen, and that $f(x)$ is evaluated at these nodes. Define the Lagrange basis functions,

$$l_i(x) = \prod_{j=0, j\neq i}^{N} \frac{x - x_j}{x_i - x_j}.$$

The Lagrange interpolating polynomial is then

$$p_N(x) = \sum_{i=0}^{N} f(x_i) l_i(x)$$

and the remainder is

$$R(x) = f(x) - p_N(x) = \frac{f^{(N+1)}(\xi(x))}{(N+1)!} \prod_{i=0}^{N} (x - x_i).$$

Recall that $\xi(x)$ is an unknown point somewhere in the interval (a,b).

The remainder formula is important: the error in an approximate integration will be the integral of the remainder. Because we don't know $\xi(x)$ we can't compute the error exactly, but we can bound it. Better yet, we can use the functional form of the remainder to spot ways of reducing the error.

The Hermite interpolating polynomial and remainder

Assume a set of nodes $\{x_i\}_{i=1}^N$, $x_i \in [a,b]$ has been chosen, and that $f(x)$ and $f'(x)$ are evaluated at these nodes. Define the Hermite basis functions,

$$h_i(x) = \left[1 - 2l_i'(x_i)(x - x_i)\right] (l_i(x))^2$$

$$\widetilde{h}_i(x) = (x - x_i) (l_i(x))^2$$

The Hermite interpolating polynomial is then

$$\eta_N(x) = \sum_{i=1}^{N} \left[f(x_i)h_i(x) + f'(x_i)\widetilde{h}_i(x)\right]$$

and the remainder is

$$R_N(x) = f(x) - \eta_N(x) = \frac{f^{(2N)}(\xi(x))}{(2N)!} \left(\prod_{i=1}^{N}(x - x_i)\right)^2.$$

As usual, $\xi(x)$ is some point in (a,b).

Comments

- Ackleh *et. al.* use $H_N(x)$ for the Hermite interpolating polynomial. However, this leads to confusion with another (and more important) family of functions, the Hermite polynomials, the n-th member of which is conventionally denoted $H_n(x)$. To make matters worse, there is a quadrature rule (Gauss-Hermite quadrature) whose analysis uses *both* families of polynomials named after Charles Hermite. Therefore, I have used $\eta_N(x)$ for the Hermite *interpolating* polynomials (η is H in the Greek alphabet) and reserve $H_n(x)$ exclusively for the Hermite polynomials.
- Note the difference in starting index: 0 for the Lagrange polynomials, 1 for the Hermite interpolating polynomials. The functions $l_i(x)$ in the formulas for Hermite interpolation are identical to the Lagrange basis functions, but indexed from 1 rather than 0.

2 The Big Picture

Some basic terminology

- A *quadrature rule* is a formula such as
$$Q_N(f) = \sum_{i=1}^{N} w_i f(x_i)$$
where w_i are the *weights* and x_i are the *quadrature points*, also sometimes called *nodes*, or (especially in older books) the *abcissas.* Note that Ackleh *et al* use α_i for the weights. The notation w_i is more common. Quadrature rules are usually stated on some fixed interval, typically $[-1,1]$ or $[0,1]$. The weights are normalized so that
$$Q_N(1) = \sum_{i=1}^{N} w_i \cdot 1 = \int_{-1}^{1} 1 \cdot dx$$
(assuming a reference interval $[-1,1]$). You then map from the reference interval to the desired interval $[a,b]$ by making the affine transformation (assuming a reference interval $[-1,1]$),
$$y_i = \frac{a+b}{2} + \frac{b-a}{2} x_i$$
$$v_i = \frac{b-a}{2} w_i.$$
Instead of memorizing this transformation, learn how to derive it from scratch: you want -1 to map to a, 1 to map to b, and normalize so that your weights v_i sum up to $\int_a^b 1 \cdot dx$.

- A *composite quadrature rule* is formed by partitioning $[a, b]$ into nonoverlapping subintervals

$$[a, b] = [a_0, a_N] = [a_0, a_1] \cup [a_1, a_2] \cup \cdots \cup [a_{N-1}, a_N]$$

  and then applying a basic rule on each subinterval. The subintervals need not all have the same length, but often they will.

- A *closed* rule has nodes at the endpoints a and b. An *open* rule does not.

Types of methods

- *Newton-Cotes* methods were the earliest to be developed, and some are still useful. You might have seen the simplest NC methods – the midpoint rule, trapezoidal rule, and Simpson's rule – in your calculus course. NC methods are easy to use for hand calculations because the nodes are evenly spaced and the weights are rational numbers. However, Newton-Cotes methods behave badly at higher polynomial order (for the same reason that interpolants with equally-spaced nodes can be wildly inaccurate) and so except for the composite rules based on the lowest-order NC methods (basically the three mentioned previously: midpoint, trapezoidal, and Simpson), Newton-Cotes methods are now largely obsolete. It's still worth knowing a little about them because:

  – Interestingly, the composite trapezoidal rule proves to be the *optimal* rule for a small but important class of functions: the periodic functions.
  – NC rules are the starting point for a family of differential equation solvers, the multistep predictor-corrector methods.
  – Composite trapezoidal and composite midpoint are the starting points for the Romberg methods, described below.
  – The trapezoidal rule and Simpson's rule are easy to explain at the undergraduate level, and are a good recommendation for non-experts

- *Gaussian quadrature* methods adjust the positions of the nodes, as well as the weights assigned to each node. Compared to a NC method with the same number of nodes, this results in roughly double the degree of precision for the same amount of work. Unlike NC methods, Gaussian methods are provably convergent (for 1D integrals) as order is increased. Historically, the drawbacks had been that the nodes and weights are more complicated to compute than in a NC rule, and that typically both the nodes and weights are irrational numbers making hand calculation impractical. There are now efficient methods for computing Gauss points and weights, and a computer doesn't consider an 8-byte approximation to $1/\sqrt{3}$ to be any more complicated than an 8-byte approximation to $4/3$. Thus, Gaussian methods are now a good default choice, robust and accurate for typical quadrature problems. Some other interesting facts:

  – Gaussian methods are (for 1D integrals) closely related to orthogonal polynomials and eigenvalue problems.
  – It is possible to develop Gaussian methods specialized to integrals with certain integrable singularities at the endpoints.
  – Gauss points also arise in certain control problems; for example, the optimal location of actuators to control transverse motion of a beam can be shown to be the Gauss quadrature points.
  – Finding Gauss points on tetrahedra and on higher-dimensional simplices (4D domains are used for problems in general relativity and 6D, 9D, 12D, etc domains are used in quantum chemistry) is an ongoing research area.

- *Richardson extrapolation* methods do repeated refinements of a low-order composite rule (trapezoidal or midpoint) and extrapolate the results to interval size $\to 0$. Comparison of different steps in the refinement sequence can provide on-the-fly error estimates; if you have an accuracy goal, you keep refining until you either reach the desired accuracy or get tired of waiting for an answer.

- The composite trapezoidal rule with Richardson extrapolation is known as the *Romberg* method. Romberg quadrature is closely related to an important method for solving ordinary differential equations, the Bulirsch-Stoer method.
- Romberg methods are based on a theorem, the *Euler-Maclaurin summation formula*, that is interesting in its own right. Euler-Maclaurin relates an integral to a finite sum based on iterated integrations by parts. However, in general the Euler-Maclaurin procedure *does not converge* as the number of integrations by parts is taken to infinity. This leads to the fruitful concept of an asymptotic sequence: a sequence of approximations that diverges if taken to the limit, but which can produce a surprisingly accurate approximation at one of its intermediate steps. You might have seen Stirling's formula for the factorial: $\log(n!) \approx n \log n - n + \frac{1}{2} \log(2\pi n)$. This very accurate approximation is the start of an asymptotic series that does not, in fact, converge.
- Certain of the error terms in the Euler-Maclaurin formula vanish identically for integrals of periodic functions. A proof quickly follows that the composite trapezoidal rule is extremely accurate for periodic functions.

Some useful methods we won't cover

You should have heard these names in case you run into a problem for which they are appropriate:

- *Gauss-Kronrod* methods use variants of Gaussian quadrature rules that are constructed so that a subset of points re-used through a sequence of refinements. This lets Richardson extrapolation and automatic error control be used with moderate-order Gauss rules rather than low-order NC.
- *Adaptive* methods refine differently on different subintervals, refining more on subintervals where the integration is more difficult. Refinement can be through increasing the number of subintervals while holding order fixed (called h-refinement) or increasing the order while fixing the subinterval size (called p-refinement), or a combination of the two (called, you guessed it, hp-refinement).
- Numerical integrals over domains of high spatial dimension are important in many fields including physics, finance, and reliability engineering. Unfortunately, they are very hard. *Monte Carlo* methods use random sampling to deal with the "curse of dimensionality": the exponential growth in the number of quadrature points as the dimension of the domain of integration increases. Cleverer variants of the MC method choose samples from a distribution that is adapted on-the-fly to explore preferentially the parts of the domain that have been seen to contribute most significantly to the integral.
- *Space-filling curve* and *sparse cubature* methods are alternate ways of dealing with the curse of dimensionality. They can converge more rapidly than Monte Carlo. The development of good methods for approximation of integrals in high dimensional domains is an ongoing research area.

3 Practical calculation of Gauss points

The obvious way to find the Gauss points is to use a numerical root-finding method to compute the zeros of the Legendre polynomials. A more robust method based on numerical linear algebra was developed by Golub and Welsch; most calculations now use variants of this method rather than root-finding.

The foundation of the method is the recurrence relation for the Legendre polynomials,

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x).$$

This formula can be used to compute the Legendre polynomials, as follows. Seed the recurrence with $P_{-1} = 0$ and $P_0 = 1$, then

$$P_1 = xP_0 - 0 \cdot P_{-1} = x$$

$$2P_2 = 3xP_1 - P_0 = 3x^2 - 1$$

$$3P_3 = 5xP_2 - 2P_1$$

and so on. This is in fact the most efficient way to compute numerical values of the Legendre polynomials. In the spirit of taking a simple problem and making it complicated, let's write this recurrence algorithm as a linear system of equations for the vector

$$q(x) = (P_0(x),\ P_1(x),\ \cdots,\ P_N(x))\,.$$

Putting the initialization $P_0 = 1$ in the first row and the three-term recurrence in each subsequent row forms a lower-triangular $(N+1) \times (N+1)$ system of equations

$$\begin{pmatrix} 1 & & & & & & & \\ -x & 1 & & & & & & \\ 1 & -3x & 2 & & & & & \\ & 2 & -5x & 3 & & & & \\ & & & \ddots & & & & \\ & & & & N-2 & -(2N-3)x & N-1 & \\ & & & & & N-1 & -(2N-1)x & N \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ \\ \\ \vdots \\ P_N \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ \\ \\ \vdots \\ 0 \end{pmatrix}.$$

Solving this system by backsubstitution goes through exactly the same steps as the recurrence; in practice, you'd simply use the recurrence. The advantage of the linear algebraic formulation becomes clear when we consider the problem of finding the roots of $P_N(x)$. If we set $P_N(x) = 0$ then we can strike out the last column from the system. The assignment $P_0 = 1$ in a linear recurrence is arbitrary, so we can also strike out the first row of equations and let P_0 "float" at least temporarily. The resulting $N \times N$ equation set is

$$\begin{pmatrix} -x & 1 & & & & \\ 1 & -3x & 2 & & & \\ & 2 & -5x & 3 & & \\ & & & \ddots & & \\ & & & & -(2N-3)x & N-1 \\ & & & & N-1 & -(2N-1)x \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \\ \vdots \\ P_{N-2} \\ P_{N-1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \\ \\ \vdots \\ 0 \end{pmatrix}.$$

This can be written compactly if we define the tridiagonal matrix A with elements $A_{i,i+1} = i+1$, $A_{i+1,i} = i+1$ and the diagonal matrix B with elements $B_{ii} = 2i+1$ (where indices start with zero). Then the equations are the generalized eigenvalue problem,

$$Aq = xBq$$

which can be turned into an ordinary eigenvalue problem,

$$B^{-1}Aq = xq.$$

Unfortunately, the problem's matrix is no longer symmetric. An improvement is to write the diagonal B as D^TD, where $D_{ii} = \sqrt{B_{ii}}$. Define an auxiliary vector $r = Dq$ so that the eigenvalue problem becomes

$$AD^{-1}Dq = xD^TDq$$

$$AD^{-1}r = xD^Tr$$

$$D^{-T}AD^{-1}r = xr$$

which is a symmetric eigenvalue problem. With a symmetric formulation all eigenvalues will remain real even in the presence of roundoff error, making this formulation more suitable for numerical calculation of high-order quadrature rules.

The eigenvalues x are those points at which $P_N(x) = 0$, *i.e.*, the roots we're looking for. The n-th eigenvector is $q(x_n) = \begin{pmatrix} P_0(x_n) & P_1(x_n) & \cdots & P_{N-2}(x_n) & P_{N-1}(x_n) \end{pmatrix}^T$, or the Legendre polynomials evaluated at the roots. As always in an eigenvalue problem, the eigenvectors are determined only up to a multiplicative constant so we can scale each eigenvector so that $P_0(x_n) = 1$, thereby satisfying the first row of equations from the original system.

Calculation of the weights

Having found the Gauss-Legendre points, the weights can be determined by the requirement that each Legendre polynomial P_0 through P_{N-1} be integrated exactly:

$$\sum_{i=1}^{N} w_i P_n(x_i) = \int_{-1}^{1} P_n(x)\,dx = \begin{cases} 2 & n = 0 \\ 0 & n > 0 \end{cases}.$$

Define Q to be the matrix whose columns are the eigenvectors q, and the weights are obtained by solving the linear equations

$$Q \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}.$$

For this to work, it is important that the eigenvectors be scaled so that each eigenvector has $P_0(x_n) = 1$. If not, the Legendre polynomials at different nodes are scaled by different, effectively random, constants.

Example

We compute the Gauss-Legendre points and weights for $N = 4$. The matrices A and B are

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 3 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 7 \end{pmatrix}$$

and so we must find the eigenvalues and eigenvectors of

$$D^{-T} A D^{-1} = \begin{pmatrix} 0 & \frac{1}{\sqrt{3}} & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & \frac{2}{\sqrt{15}} & 0 \\ 0 & \frac{2}{\sqrt{15}} & 0 & \frac{3}{\sqrt{35}} \\ 0 & 0 & \frac{3}{\sqrt{35}} & 0 \end{pmatrix}.$$

The points are the eigenvalues, and the weights are obtained by solving $Qw = (2, 0, \cdots, 0)^T$. The calculation must be done numerically. We find the four quadrature points and their weights, to 15 digits:

x_i	w_i
$\pm$0.86113 63115 94053	0.34785 48451 37454
$\pm$0.33998 10435 84856	0.65214 51548 62546

Notes

- All of the classical orthogonal polynomials have an associated three-term recurrence relation, so the Golub-Welsch method generalizes easily to other families of orthogonal polynomials.
- Though this method was first proposed in 1969, it was not widely used until more recently. Why not? A typical PC in 1989 had 640 KB memory. Finding a 200-point Gauss-Legendre rule in this way requires storage of a 200×200 matrix in double precision, requiring 320 KB. That's half of the machine's memory just to store the eigenvectors, so the method wasn't often used in practice. Finding high-order Gauss rules in those days was therefore usually done by rootfinding, and required careful, robust rootfinding code to avoid duplicating solutions. Most people used composite Newton-Cotes or composite low-order Gauss out of convenience. Now that typical computers can easily store and solve the eigenvalue problem, Golub-Welsch is the simpler choice and can be written in a few lines of Matlab or Mathematica.

- The linear solve for the weights can be bypassed by using an identity for Legendre polynomials. See the literature for this and for various other improvements to the method.
- A C++ implementation for any orthogonal polynomial family is available as part of the open-source library *Trilinos*.

4 The Euler-Maclaurin formula

The Euler-Maclaurin formula (EMF) is one of the more remarkable results in classical analysis. It relates integration and discrete summation, and is a source of approximate methods for problems in discrete mathematics arising in analytic number theory, combinatorics, and theoretical physics.

Approximate summation of series

In this course you're seeing the EMF in connection with numerical integration, where it is at the heart of both Romberg integration and the error analysis of the CTR for periodic functions. An equally important use (and the application for which it was originally developed) is in the numerical summation of series. Suppose you wanted to compute

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2}.$$

It can be shown through Fourier analysis that $\zeta(2) = \frac{\pi^2}{6}$, and any $\zeta(2n)$ can be computed through a similar method. In Euler's day, however, the calculation of this sum was a major unsolved problem (known as the Basel problem). Even today, the exact summation of $\zeta(2n+1)$ is an open problem.

Euler's idea was to write $\zeta(2)$ in two parts,

$$\zeta(2) = \sum_{n=1}^{P} \frac{1}{n^2} + \sum_{n=P+1}^{\infty} \frac{1}{n^2}$$

where P is small enough so that we can easily compute the first term directly, and to then approximate the second term by an integral. Use the extended EMF (Theorem 6.4 in Ackleh) with $f(x) = x^{-2}$ to write

$$\int_P^{\infty} \frac{dx}{x^2} = \frac{1}{2P^2} + \sum_{n=P+1}^{\infty} \frac{1}{n^2} + \sum_{j=1}^{M} \frac{B_{2j}}{(2j)!} \frac{(-2)(-3)\cdots(-(2j))}{P^{2j+1}} + \text{error},$$

or, after doing the easy integral and ignoring the error term,

$$\frac{1}{P} = \frac{1}{2P^2} + \sum_{n=P+1}^{\infty} \frac{1}{n^2} + \sum_{j=1}^{M} \frac{B_{2j}}{P^{2j+1}} (-1)^j.$$

We can then write

$$\zeta(2) \approx \sum_{n=1}^{P} \frac{1}{n^2} + \frac{1}{P} - \frac{1}{2P^2} - \sum_{j=1}^{M} \frac{(-1)^j B_{2j}}{P^{2j+1}}.$$

How well does this work? Let $\epsilon_1(P)$ and $\epsilon_2(P)$ be the error after truncating at P terms with and without the EMF correction terms (with $M = 1$). We find

P	ϵ_1	ϵ_2
10	0.0951663	$3.30985e-7$
20	0.0487708	$1.03981e-8$
40	0.0246901	$3.25376e-10$
80	0.0124222	$1.01716e-11$
100	0.009950167	$3.333095e-12$
1000	0.000999500	$3.333331e-16$
10000	0.000099995	$3.333333e-22$

As you can see from the results, a ten-term direct sum corrected by the EMF gives results two orders of magnitude better than a *ten thousand term* direct sum without correction.

Asymptotic approximations

In the preceding calculations, I used $M = 1$. Would the results have been better had I used a larger M? Surprisingly, no. The sum of boundary terms in the EMF is an example of an asymptotic series: as $M \to \infty$, the sum may in fact diverge (and often does for many problems) but nonetheless its partial sums may still give amazingly good results up to some M.

For a thorough discussion of asymptotic series and how to use them safely, see the book by Bender and Orszag, *Advanced Mathematical Methods for Scientists and Engineers*. Applications of asymptotic methods to combinatorics and analysis of algorithms can be found in Knuth's masterpiece, *The Art of Computer Programming: Volume I, Seminnumerical Methods.*

5 Quadrature for periodic functions

The periodic functions are important in applications. A function is $T-$periodic if, for any x, $f(x+T) = f(x)$. Without loss of generality take $T = 2\pi$. We want to approximate

$$\int_0^{2\pi} f(x)\,dx$$

by quadrature. We will show that the best quadrature method for periodic functions is the composite trapezoidal rule, provided enough points are used to resolve the most significant Fourier modes of the function.

The N-point CTR on $[0, 2\pi]$ is

$$Q_N(f) = \frac{\pi}{N} f(0) + \frac{\pi}{N} f(2\pi) + \frac{2\pi}{N} \sum_{n=1}^{N-1} f\left(\frac{2\pi n}{N}\right)$$

Note that the two endpoints coincide, $f(0) = f(2\pi)$, so that

$$Q_N(f) = \frac{2\pi}{N} \sum_{n=0}^{N-1} f\left(\frac{2\pi n}{N}\right).$$

The CTR on a periodic function is simply the average of equally-spaced points.

Theorem 1. *The N-point CTR on* $[0, 2\pi]$ *is exact for all trigonometric polynomials*

$$\tau_M(x) = \frac{A_0}{2} + \sum_{k=1}^{M} \left(A_k \cos(kx) + B_k \sin(kx)\right)$$

of maximum frequency M up to $N - 1$.

Proof. We know that for all $k > 0$

$$\int_0^{2\pi} \cos(kx)\,dx = \int_0^{2\pi} \sin(kx)\,dx = 0$$

and thus for any M,

$$\int_0^{2\pi} \tau_M(x)\,dx = \frac{A_0}{2}.$$

The CTR with any number of points is exact for constants.

Form the error,

$$E_N(\tau_M) = \int_0^{2\pi} \tau_M(x)\,dx - \left(\frac{2\pi}{N}\right) \sum_{n=0}^{N-1} \tau_M\left(\frac{2\pi n}{M}\right)$$

$$= \left(\frac{2\pi}{N}\right) \sum_{k=0}^{M} \sum_{n=0}^{N-1} \left(A_k \cos(\frac{2\pi kn}{N}) + B_k \sin(\frac{2\pi kn}{N}) \right)$$

$$= \left(\frac{\pi}{N}\right) \sum_{k=0}^{M} \sum_{n=0}^{N-1} \left((A_k + iB_k)\, e^{2\pi ink/N} + (A_k - iB_k)\, e^{-2\pi ink/N} \right)$$

The error will be zero, and thus the CTR will be exact, iff $Q_N(e^{\pm ikx}) = 0$ for all $k = 1, 2, \cdots, M$. We therefore look at the CTR on the complex exponentials,

$$Q_N(e^{\pm ikx}) = \frac{2\pi}{N} \sum_{n=0}^{N-1} e^{\pm 2\pi ink/N}.$$

First notice that when $k = pN$, $p \in \mathbb{Z}$, we have $e^{\pm 2\pi inp} = 1$ and so $Q_N(e^{\pm ipNx}) = 2\pi$. The CTR is thus *not* exact when $k = N$. For $k \neq pN$, we notice that Q_N is a finite geometric series and so

$$\frac{N}{2\pi} Q_N(e^{\pm ikx}) = \frac{1 - e^{\pm 2\pi iNk/N}}{1 - e^{\pm 2\pi ik/N}} = \frac{1 - e^{\pm 2\pi ik}}{1 - e^{\pm 2\pi ik/N}}.$$

The numerator is always zero, and the denominator is nonzero provided $k/N \notin \mathbb{Z}$ (that case has already been dealt with). We have shown that the N-point CTR will be exact for $k < N$, and will fail to integrate exactly a term with frequency $k = N$. We conclude the N-point CTR is exact for trigonometric polynomials up to maximum frequency $N - 1$. □

Convergence of the CTR for periodic functions

A corrollary to the Weierstrass approximation theorem tells us that any C^0 periodic function can be approximated uniformly by trigonometric polynomials. The weights in the CTR are positive, so using an argument similar to that used to demonstrate convergence of Gaussian quadrature, it can be proved that the CTR is convergent on C^0 periodic functions.

The error in CTR for periodic functions

It is useful to examine the rate of convergence, which can be done using the EMF. The EMF for periodic functions is interesting because the surface terms in each integration by parts cancel exactly, leaving

$$\int_0^{2\pi} f(x)\, dx = Q_N(f) - \frac{2\pi B_{2m+2}}{(2m+2)!} \left(\frac{2\pi}{N}\right)^{2m+2} f^{(2m+2)}(\xi).$$

Therefore, the error is bounded by

$$E_N(f) \leq \left| \frac{2\pi B_{2m+2}}{(2m+2)!} \left(\frac{2\pi}{N}\right)^{2m+2} \right| \left\| f^{(2m+2)} \right\|_\infty.$$

To simplify, we use an identity due to Euler,

$$B_{2n+2} = (-1)^{n+1} \frac{2\,(2n+2)!}{(2\pi)^{2n+2}} \zeta(2n+2)$$

where ζ is the Riemann zeta function

$$\zeta(2m+2) = \sum_{n=1}^{\infty} \frac{1}{n^{2m+2}}.$$

Clearly for all $m > 0$, the zeta function is bounded by $1 < \zeta(2m+2) < \zeta(2) = \frac{\pi^2}{6}$. Using Euler's identity to eliminate the Bernoulli number and factorial gives

$$E_N(f) \leq 2\pi\zeta(2m+2) \left(\frac{1}{N}\right)^{2m+2} \left\| f^{(2m+2)} \right\|_\infty.$$

We have therefore established:

Theorem 2. *The error from CTR quadrature of a periodic function $f \in C^{2k}[0, 2\pi]$, $k \geq 1$, is bounded by*

$$E_N(f) \leq 2\pi\zeta(2k) \left(\frac{1}{N}\right)^{2k} \left\| f^{(2k)} \right\|_\infty .$$

This is a remarkable result. By contrast, the CTR on a non-periodic C^{2k} function, $k \geq 2$, converges as $O(N^{-2})$ regardless of k. For non-periodic functions, the CTR is a moderately efficient, rather robust method, but Gaussian quadrature will usually do better if k is large. For periodic functions, however, the CTR is a *very* good method. Unless some other information about f is known (for example, the location of a peak requiring refined quadrature) the CTR is optimal for a periodic function.

What about CTR on periodic C^∞ functions? The error must decrease with N more rapidly than any power of N, for example, $O(e^{-N})$. The determination of the precise form of the dependence of the error on N is not particularly simple, but an exponential decrease is a good rule of thumb.

Finally, we can use the EMF for an alternate proof of the exactness of the CTR for trigonometric polynomials of frequency less than N. Consider $f(x) = \cos(kx)$. Then $f^{(2m+2)}(k) = k^{2m+2}\cos(kx)$ and, using the bound $\zeta(2k+2) < \zeta(2)$,

$$E_N(f) \leq 2\pi\zeta(2) \left(\frac{k}{N}\right)^{2m+2} .$$

The cosine is C^∞, so this bound is true for any m. The best bound will be obtained by finding the m which minimizes E_N. If $k < N$ we can take $m \to \infty$ and find $E_N \to 0$. Therefore, the CTR is exact for cosines of integer frequency less than N. The same argument goes through for $\sin(kx)$, and the theorem for trigonometric polynomials follows from the linearity of quadrature.