12. Integrals and Quadrature Formulas

Up to now we have been solving ODEs, PDEs, and related eigenvalue problems. Now suppose that we are faced with the simpler task of evaluating an integral such as

$$I = \int_{-1}^{1} f(x)\, dx. \tag{12.1}$$

How could we compute I by a spectral method?

One approach is to note that an integral is the special case of an ODE $u' = f(x, u)$ in which f is independent of u. Thus (12.1) can be restated as the initial value problem

$$u'(x) = f(x), \qquad u(-1) = 0, \quad x > -1, \tag{12.2}$$

where our goal is to evaluate $I = u(1)$. For this we can set up a spectral method on $[-1, 1]$ on our usual Chebyshev grid. To impose the boundary condition $u(-1) = 0$, we strip off the last row and column of the differentiation matrix D_N in the usual manner described in Chapter 7. If $\widetilde{D}_N$ is the resulting matrix of dimension $N \times N$, we are left with the linear system of equations

$$\widetilde{D}_N v = f$$

with $f = (f(x_0), \ldots, f(x_{N-1}))^T$. Our approximation to I is given by $I_N = v_0$.

In fact, since we care only about the first component of v, there is no need to solve the whole system of equations. If we let w^T denote the first

row of $\widetilde{D}_N^{-1}$, a row vector of length N, then another formula for the same approximation is

$$I_N = w^T f. \tag{12.3}$$

Speaking abstractly, integration over $[-1,1]$ is a linear functional $I(f)$, and a linear numerical approximation to I based on discrete data will constitute another linear functional $I_N(f)$. Equation (12.3) expresses the fact that any linear functional is equivalent to an inner product with some weight vector w—the *Riesz representation theorem* [LiLo97].

Program 30, whose output is labeled Output 30a, illustrates the success of this method by integrating $|x|^3$, $\exp(-x^{-2})$, $1/(1+x^2)$, and x^{10}, the same functions that we examined in Chapter 6 in connection with the convergence of spectral differentiation. Spectral accuracy just as in Output 12 (p. 58) is evident. Note that x^{10} is integrated exactly for $N \geq 11$.

The method just described made use of our existing spectral differentiation matrix. An alternative and better approach is to start from our fundamental spectral philosophy:

- *Find the polynomial of degree* $\leq N$ *such that* $p(x_j) = f_j$, $0 \leq j \leq N$.
- *Set* $I_N = \int_{-1}^{1} p(x)\,dx$.

This formulation must be different, for it will integrate x^{10} exactly for $N \geq 10$ rather than $N \geq 11$. (It makes use of the value $f(-1)$, which the previous method ignored.) In fact, this new strategy goes by the name of *Clenshaw–Curtis quadrature* [ClCu60]. In the field of numerical integration [DaRa84, KrUe98], it can be classed as the formula of optimal order based on the fixed set of Chebyshev nodes $\{x_j\}$—as opposed to the Gauss formula of optimal order based on optimally chosen nodes, which we shall discuss in a moment.

One way to compute the Clenshaw–Curtis approximation would be by using the FFT methods of Chapter 8. Given a function $f(x)$ defined on $[-1,1]$, consider the self-reciprocal function $\mathsf{f}(z)$ defined on the unit circle $|z| = 1$ by the 2-to-1 pointwise equivalence $x = \mathrm{Re}\, z$ of (8.1). If $p(x) = \sum_{n=0}^{N} a_n T_n(x)$ is the polynomial interpolant to $f(x)$ in the Chebyshev points $\{x_j\}$, then $p(x)$ corresponds pointwise to the self-reciprocal Laurent polynomial interpolant $\mathsf{p}(z) = \frac{1}{2}\sum_{n=0}^{N} a_n(z^n + z^{-n})$ to $\mathsf{f}(z)$ in roots of unity $\{z_j\}$. Since $x = \frac{1}{2}(z+z^{-1})$ and $dx/dz = \frac{1}{2}(1 - z^{-2})$, we compute

$$\begin{aligned}\int_{-1}^{1} p(x)\,dx &= \int_{-1}^{1} p(z)\,dz\,\frac{dx}{dz} \\ &= \frac{1}{4}\sum_{n=0}^{N} a_n \int_{-1}^{1} (z^n + z^{-n})(1 - z^{-2})\,dz\end{aligned}$$

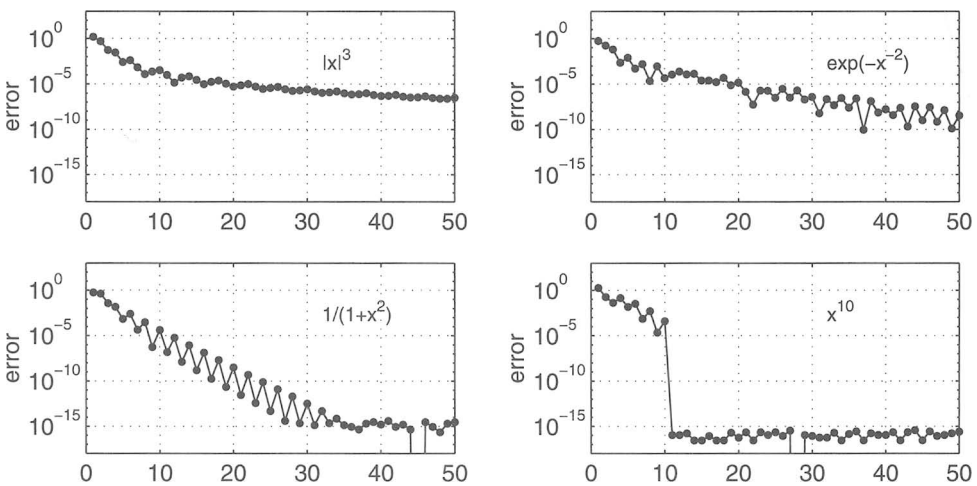
Program 30

```
% p30.m - spectral integration, ODE style (compare p12.m)

% Computation: various values of N, four functions:
  Nmax = 50; E = zeros(4,Nmax); clf
  for N = 1:Nmax; i = 1:N;
    [D,x] = cheb(N); x = x(i); Di = inv(D(i,i)); w = Di(1,:);
    f = abs(x).^3;      E(1,N) = abs(w*f - .5);
    f = exp(-x.^(-2)); E(2,N) = abs(w*f - ...
                            2*(exp(-1)+sqrt(pi)*(erf(1)-1)));
    f = 1./(1+x.^2);    E(3,N) = abs(w*f - pi/2);
    f = x.^10;          E(4,N) = abs(w*f - 2/11);
  end

% Plot results:
  labels = {'|x|^3','exp(-x^{-2})','1/(1+x^2)','x^{10}'};
  for iplot = 1:4,
    subplot(3,2,iplot)
    semilogy(E(iplot,:)+1e-100,'.','markersize',12), hold on
    plot(E(iplot,:)+1e-100)
    axis([0 Nmax 1e-18 1e3]), grid on
    set(gca,'xtick',0:10:Nmax,'ytick',(10).^(-15:5:0))
    ylabel error, text(32,.004,labels(iplot))
  end
```

Output 30a



Output 30: *Integration of* (12.1) *via ODE: error vs. N. Compare Output* 12 *(p.* 58*).*

$$= \frac{1}{4}\sum_{n=0}^{N} a_n \int_{-1}^{1} (z^n - z^{n-2} + z^{-n} - z^{-n-2})$$

$$= \frac{1}{4}\sum_{n=0}^{N} a_n \left(\frac{z^{n+1} + z^{-n-1}}{n+1}\Big|_{-1}^{1} - \frac{z^{n-1} + z^{-n+1}}{n-1}\Big|_{-1}^{1} \right)$$

$$= \sum_{\substack{n=0 \\ n \text{ even}}}^{N} a_n \left(\frac{1}{n+1} - \frac{1}{n-1} \right) = \sum_{\substack{n=0 \\ n \text{ even}}}^{N} \frac{2a_n}{1-n^2}.$$

Thus to implement Clenshaw–Curtis quadrature, we can use the FFT to determine the coefficients $\{a_n\}$ as in Chapter 8, then sum the results over even values of n with the weights $2/(1-n^2)$.

This method works, but it is more elaborate than necessary, for by pursuing the algebra a little further, one can determine the Clenshaw–Curtis weights analytically. Rather than write down the results in formulas, we encapsulate them in a MATLAB program:

clencurt.m

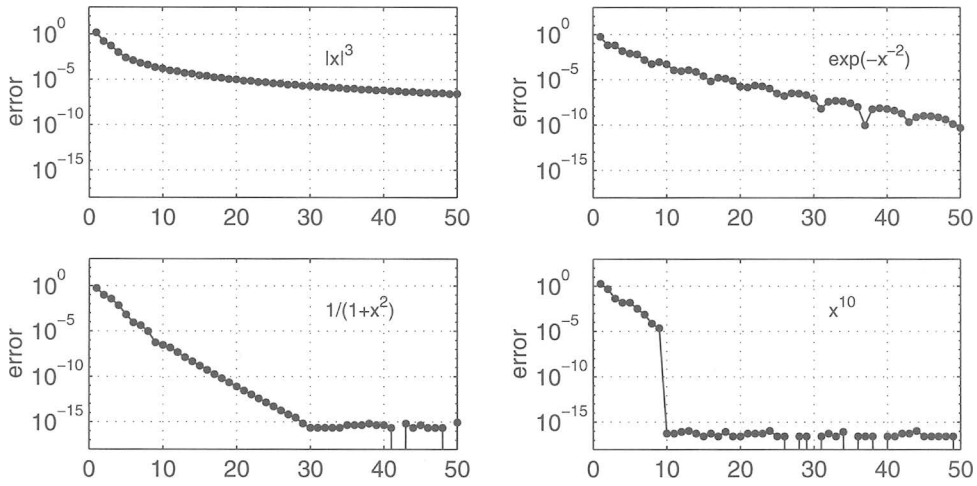
```
% CLENCURT  nodes x (Chebyshev points) and weights w
%           for Clenshaw-Curtis quadrature

  function [x,w] = clencurt(N)
  theta = pi*(0:N)'/N; x = cos(theta);
  w = zeros(1,N+1); ii = 2:N; v = ones(N-1,1);
  if mod(N,2)==0
    w(1) = 1/(N^2-1); w(N+1) = w(1);
    for k=1:N/2-1, v = v - 2*cos(2*k*theta(ii))/(4*k^2-1); end
    v = v - cos(N*theta(ii))/(N^2-1);;
  else
    w(1) = 1/N^2; w(N+1) = w(1);
    for k=1:(N-1)/2, v = v - 2*cos(2*k*theta(ii))/(4*k^2-1); end
  end
  w(ii) = 2*v/N;
```

Output 30b shows the results obtained by modifying Program 30 to use `clencurt`. They are marginally more accurate than before, and much cleaner.

The convergence rates exhibited in Outputs 30a and 30b are excellent—this is spectral accuracy of the kind showcased throughout this book. Nevertheless, we can do better. If we use a Gaussian formula, then the integral will be exact for polynomials of degree $2N-1$, not just N or $N-1$. For this we must

Output 30b



Output 30b: *Clenshaw–Curtis integration of* (12.1). *These results are generated by Program* 30 *except with the line beginning* `[D,x] = ...` *replaced by the command* `[x,w] = clencurt(N)`.

take $\{x_j\}$ to be not Chebyshev points but *Legendre points*, that is, roots of Legendre polynomials in $(-1, 1)$. These points and the associated weights can be computed numerically by solving a tridiagonal matrix eigenvalue problem [GoWe69, TrBa97]. The next, surprisingly short program specifies the details.

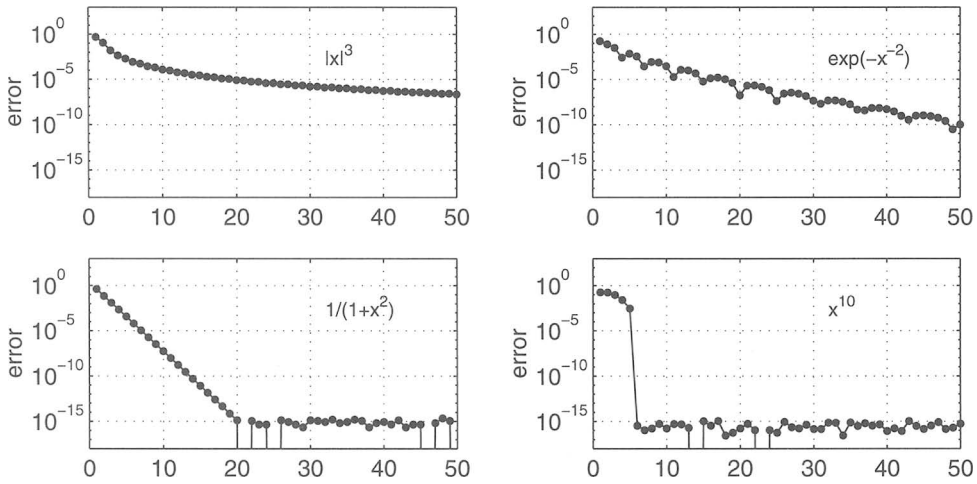
gauss.m

```
% GAUSS  nodes x (Legendre points) and weights w
%        for Gauss quadrature

  function [x,w] = gauss(N)
  beta = .5./sqrt(1-(2*(1:N-1)).^(-2));
  T = diag(beta,1) + diag(beta,-1);
  [V,D] = eig(T);
  x = diag(D); [x,i] = sort(x);
  w = 2*V(1,i).^2;
```

Output 30c shows the results obtained with Gauss quadrature. Note that for the smoother functions, the convergence surpasses that of Outputs 30a and 30b, but there is not much difference for the functions that are less smooth.

Output 30c



Output 30c: *Gauss integration of* (12.1). *Here we run Program* 30 *again, but with the command* `[x,w] = gauss(N)`. *Note that the convergence is now faster for the smoother functions* f.

Gauss quadrature has genuine advantages over Clenshaw–Curtis quadrature for definite integrals. However, most applications of spectral methods involve the solution of differential equations. For these problems, Gauss quadrature is still relevant if one solves the problem by a Galerkin formulation, but it is less relevant for solutions by collocation, as in this book. Some practitioners feel strongly that Galerkin formulations are superior; others feel they require extra effort for little gain. For better or worse, the present book concentrates on collocation, and we shall make no further use of Gauss quadrature.

All of the discussion in this chapter has considered integration by Chebyshev spectral methods and their variants, not Fourier methods. What about the latter? Are there problems where we wish to calculate integrals over periodic domains, and do Fourier spectral methods provide a useful technique for such problems?

The answer is smashingly yes. Suppose we wish to evaluate an integral

$$I = \int_0^{2\pi} f(\theta)\, d\theta, \tag{12.4}$$

where f is 2π-periodic. According to the usual spectral collocation philosophy, we will construct a trigonometric interpolant in equispaced points and then integrate the interpolant. In this integral, all the nonconstant terms will integrate to zero, leaving us with just the constant term. That is, periodic

Fourier integration reduces to the *periodic trapezoid rule*,

$$I_N = \frac{2\pi}{N} \sum_{j=1}^{N} f(\theta_j), \tag{12.5}$$

with $\theta_j = j\pi/N$ as usual. Our weight vector w is a multiple of $(1, 1, 1, \ldots, 1)^T$.

For smooth integrands, for the usual reasons analyzed in Chapter 4, it follows that the periodic trapezoid rule converges extraordinarily fast. For illustration, suppose we use (12.5) to determine the perimeter of an ellipse of length 2 and width 1, which is given by the integral

$$\int_0^{2\pi} \left(\tfrac{1}{4}\sin^2\theta + \cos^2\theta\right)^{1/2} d\theta.$$

The single line of MATLAB

```
t=2*pi*(1:N)/N; I=2*pi*mean(sqrt(.25*sin(t).^2+cos(t).^2))
```

is enough to carry out this computation, and with $N = 25$, we get $I_N = 4.84422411027386$, which is correct except in the last digit. (The number in question is $4E(3/4)$, where E is the complete elliptic integral of the second kind [AbSt65]; in MATLAB, `[K,E] = ellipke(3/4)`, `perimeter = 4*E`.) For more on this phenomenon of rapid convergence of the periodic trapezoid rule, see [DaRa84], [Hen86], and Exercise 12.6.

There is a special context in which integrals over periodic domains regularly arise: as contour integrals in the complex plane. This is a beautiful subject which, although off the beaten track of spectral methods, is a standard tool in computational complex analysis. If $f(z)$ is an analytic function in the closed unit disk, for example, then its Taylor series converges there, and the Taylor coefficients can be computed by Cauchy integrals:

$$f(z) = \sum_{j=0}^{\infty} a_j z^j, \qquad a_j = \frac{1}{2\pi i} \int_{|z|=1} z^{-1-j} f(z)\, dz, \tag{12.6}$$

where the contour of integration is the unit circle traversed once counterclockwise. (If $f(z)$ is merely analytic in a neighborhood of the unit circle, not throughout the disk, the formulas generalize to a *Laurent series*, convergent in an annulus, with terms $-\infty < j < \infty$.) Setting $z = e^{i\theta}$, with $dz = iz\,d\theta$, shows that an equivalent expression for a_j is

$$a_j = \frac{1}{2\pi} \int_0^{2\pi} e^{-ij\theta} f(e^{i\theta})\, d\theta. \tag{12.7}$$

Thus each coefficient of a Taylor series can be evaluated accurately by the periodic trapezoid rule. What is more remarkable is that a whole collection of

coefficients can be evaluated simultaneously by the FFT (Exercise 12.7). This observation forms the basis of fast algorithms for problems in computational complex analysis as diverse as differentiation, integration, analytic continuation, zerofinding, computation of transforms, evaluation of special functions, and conformal mapping [Hen79, Hen86].

Here is an example involving just one trapezoid rule integral, not the FFT. One of the most familiar of special functions is the gamma function $\Gamma(z)$, the complex generalization of the factorial function, which satisfies $\Gamma(n+1) = n!$ for each integer $n \geq 0$. $\Gamma(z)$ has a pole at each of the nonpositive integers, but $1/\Gamma(z)$ is analytic for all z and is given by a contour integral formula due to Hankel (see equation (8.8.23) of [Hil62]),

$$\frac{1}{\Gamma(z)} = \frac{1}{2\pi i} \int_C e^t t^{-z}\, dt, \tag{12.8}$$

where C is a contour in the complex plane that begins at $-\infty - 0i$ (just below the branch cut of t^{-z} on the negative real axis), winds counterclockwise once around the origin, and ends at $-\infty + 0i$ (just above). Since the integrand decays exponentially as $\mathrm{Re}\, t \to -\infty$, we can get results as accurate as we like by replacing C by a bounded contour that begins and ends sufficiently far out on the negative real axis. Specifically, Program 31 takes C to be the circle of radius $r = 16$ centered at $c = -11$. If we define $t = c + re^{i\theta}$, then we have $dt = ire^{i\theta} = i(t-c)$, and the integral becomes

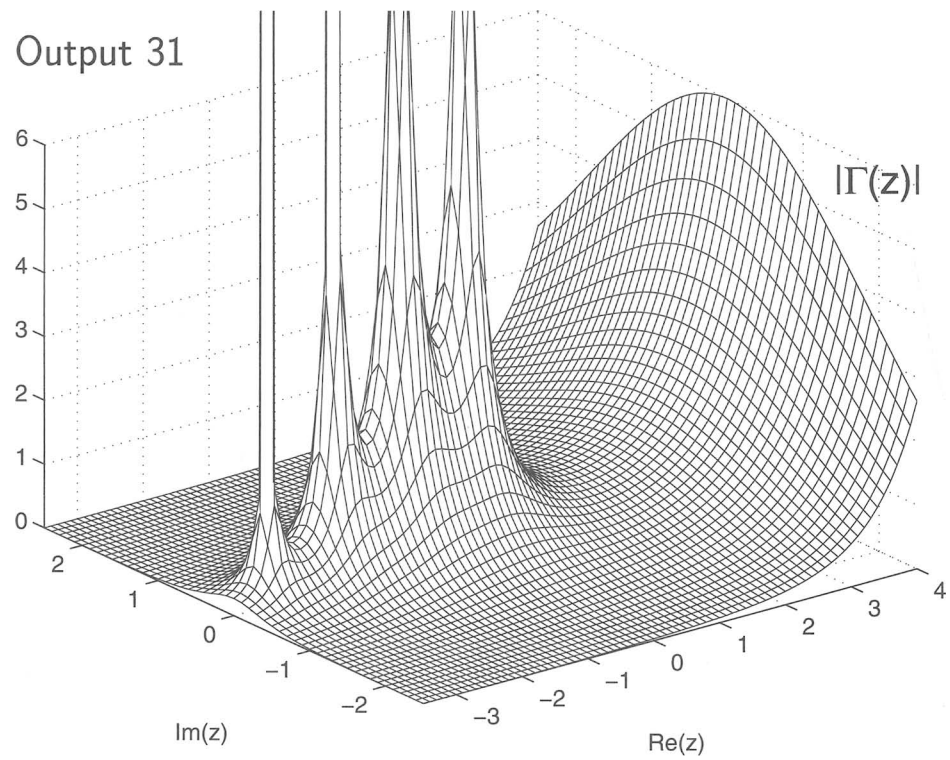
$$\frac{1}{\Gamma(z)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^t t^{-z} (t-c)\, d\theta. \tag{12.9}$$

If we evaluate this by the trapezoid rule (12.5), we find that $1/\Gamma(z)$ is approximated simply by the mean value of $e^t t^{-z}(t-c)$ over equispaced points on the contour C. It couldn't be much simpler! Output 31 inverts the result to show the familiar shape of the gamma function generated to high accuracy by this technique.

Summary of This Chapter. The natural spectral method for numerical integration in Chebyshev points is Clenshaw–Curtis quadrature, defined by integrating the polynomial interpolant, and it is spectrally accurate. A higher order of spectral accuracy can be achieved by Gauss quadrature, based on interpolation in Legendre points instead, and this is the basis of many Galerkin spectral methods. The natural spectral integration formula on a periodic interval or a closed contour in the complex plane is the trapezoid rule, and in conjunction with the FFT, this has powerful applications in complex analysis.

Program 31

```
% p31.m - gamma function via complex integral, trapezoid rule
  N = 70; theta = -pi + (2*pi/N)*(.5:N-.5)';
  c = -11;                        % center of circle of integration
  r = 16;                         % radius of circle of integration
  x = -3.5:.1:4; y = -2.5:.1:2.5;
  [xx,yy] = meshgrid(x,y); zz = xx + 1i*yy; gaminv = 0*zz;
  for i = 1:N
    t = c + r*exp(1i*theta(i));
    gaminv = gaminv + exp(t)*t.^(-zz)*(t-c);
  end
  gaminv = gaminv/N; gam = 1./gaminv; clf, mesh(xx,yy,abs(gam))
  axis([-3.5 4 -2.5 2.5 0 6]), xlabel Re(z), ylabel Im(z)
  text(4,-1.4,5.5,'|\Gamma(z)|','fontsize',20), colormap([0 0 0])
```



Output 31: *Computation of the gamma function by a 70-point trapezoid rule approximation to the contour integral* (12.8). *At most points of the grid, the computed result is accurate to* 8 *digits.*

Exercises

12.1. Perform a comparative study of Chebyshev vs. Legendre points. To make the comparisons as close as possible, define Chebyshev points via zeros rather than extrema as in (6.1): $x_j = \cos((j-1/2)\pi/N)$, $j = 1, 2, \ldots, N$. Plot the two sets of points for $N = 5$, 10, 15, and find a graphical way to compare their locations as $N \to \infty$. Modify Programs 9 and 10 to use Legendre instead of Chebyshev points, and discuss how the results compare with those of Outputs 9 and 10.

12.2. Write a MATLAB program to implement (6.8) and (6.9) and construct the differentiation matrix D_N associated with an arbitrary set of distinct points $x_0, \ldots, x_N$. Combine it with `gauss` to create a function that computes the matrix D_N associated with Legendre points in $(-1, 1)$. Print results for $N = 1, 2, 3, 4$.

12.3. Suppose you didn't know about Clenshaw–Curtis quadrature and had to reinvent it. One approach would be to find the weights by setting up and solving an appropriate system of linear equations in Vandermonde form. Describe the mathematics of this process, and then implement it with the help of MATLAB's command `vander`. Compare the weight vectors w obtained in this manner with those delivered by `clencurt` for $N = 4$, 8, and 128.

12.4. Write a program based on a Chebyshev spectral method to compute the indefinite integral $f(x) = \int_0^x \sin(6s^{2.5})ds$ for $0 \le x \le 2$. The program should plot values at (shifted) Chebyshev points and the curve of the polynomial interpolant between these values, and print the error $f(1)_{\text{computed}} - f(1)_{\text{exact}}$. Produce results for $N = 10, 20, 30, 40, 50$. Comment on the accuracy as a function of N and on how the accuracy appears to depend on the local number of points per wavelength.

12.5. To 10 digits, what is the perimeter of the superellipse defined by the equation $x^4 + y^4 = 1$? To 10 digits, what exponent α has the property that the curve defined by the equation $|x|^\alpha + |y|^\alpha = 1$ has perimeter equal to 7?

12.6. Suppose the 2π-periodic function $f(x)$ extends to an analytic function in the strip $|\text{Im}(z)| < a$ in the complex plane for some $a > 0$. From results of Chapter 4, derive an estimate for the error in evaluating $\int_{-\pi}^{\pi} f(x)\,dx$ by the trapezoid rule with step size h. Perform the integration numerically for the function $f(x) = (1 + \sin^2(x/2))^{-1}$ of Program 7 (p. 35). Does the actual convergence behavior match your estimate?

12.7. Use the FFT in N points to calculate the first 20 Taylor series coefficients of $f(z) = \log(1 + \frac{1}{2}z)$. What is the asymptotic convergence factor as $N \to \infty$? Can you explain this number?

12.8. What symmetry property does $1/\Gamma(z)$ satisfy with respect to the real axis? When c is real as in Program 31, the computed estimates of $1/\Gamma(z)$ will satisfy the same symmetry property. If c is moved off the real axis, however, the magnitude of the resulting loss of symmetry can be used to give some idea of the error in the computation. Try this with $c = -11 + i$ and produce a contour plot of the error estimate with contours at $10^{-5}, 10^{-6}, 10^{-7}, \ldots$. How does your contour plot change if N is increased to 100?