Appl. Phys. B (2012) 109:295–309
DOI 10.1007/s00340-012-5128-0

Applied Physics B
Lasers and Optics

On numerical integration with high-order quadratures: with application to the Rayleigh–Sommerfeld integral

W. A. B. Evans · A. Torre

Received: 22 December 2011 / Revised: 26 April 2012 / Published online: 18 September 2012

Abstract The paper focusses on the advantages of using high-order Gauss–Legendre quadratures for the precise evaluation of integrals with both smooth and rapidly changing integrands. Aspects of their precision are analysed in the light of Gauss' error formula. Some "test examples" are considered and evaluated in multiple precision to ≈200 significant decimal digits with David Bailey's multiprecision package to eliminate truncation/rounding errors. The increase of precision on doubling the number of subintervals is analysed, the relevant quadrature attribute being the precision increment. In order to exemplify the advantages that high-order quadrature afford, the technique is then used to evaluate several plots of the Rayleigh–Sommerfeld diffraction integral for axi-symmetric source fields defined on a planar aperture. A comparison of the high-order quadrature method against various FFT-based methods is finally given.

1 Introduction

Integrals abound in virtually all areas of theoretical and experimental physics. Often they formally give an exact solution to, for instance, boundary-value problems solved by the Green's function method. Also, it is often unclear how best to numerically evaluate them nor does one have much idea of the accuracy of the values as evaluated by "simple" quadrature rules.

W. A. B. Evans (✉)
School of Physical Sciences, University of Kent,
Canterbury CT2 7NH, UK
e-mail: w.a.b.evans@kent.ac.uk

A. Torre
ENEA UTAPRAD-MAT, Laboratorio di Modellistica
Matematica, via E. Fermi 45, 00044 Frascati (Rome), Italy
e-mail: amalia.torre@enea.it

Herein we shall consider the precision (or accuracy) of integrals of a continuous non-singular integrand, $f(x)$, over a finite interval $[a, b]$ viz.

$$I = \int_a^b f(x)\mathrm{d}x \tag{1}$$

as evaluated numerically with various quadrature rules. For more accuracy the interval $[a, b]$ can be divided into subintervals and the quadrature rule applied in each subinterval.

In Sect. 2, we define terms that are later used in the subsequent text. Section 3 gives a quadrature error analysis based on Gauss' error formula whilst Sect. 4 presents some simple numerical examples that illustrate the error behaviour in Gauss–Legendre quadratures. Section 5 presents an overview of diffraction theory leading to the R–S integral. In Sect. 6 several plots of the R–S amplitude from the near to the far field are displayed.

A comparison of the high-order quadrature method against various FFT-based methods specifically in connection with the R–S integral is presented in Sect. 7. Concluding notes are finally given in Sect. 8. A listing, along with related comments, of a Fortran program that can be used to evaluate the R–S integral at a specified point is provided in Appendix.

2 Precision, precision increment and intrinsic precision

Let us briefly review some precision definitions that shall often feature later in the text.

(a) As is conventional, by *precision*, P, we shall mean the number of significant decimal digits in whatever approximated quantity, a, under consideration. It is related to the fractional error, $|\delta a/a|$, where $\delta a = a - a_{\text{exact}}$, by

$$P \approx -\log_{10}\left|\frac{\delta a}{a}\right| \text{ i.e. } \left|\frac{\delta a}{a}\right| \approx 10^{-P} \tag{2}$$

(b) Whilst assuming the integrand, $f(x)$, is evaluated without error, the numerical quadrature results in (1) are subject to two other distinct "errors" viz. "truncation/round-off error" and "quadrature error". To numerically "see" the latter we must eliminate the former, which can be made negligibly small by working with a very large, ideally infinite, mantissa. Herein we used David Bailey [1, 2]'s impressive multiprecision package, MPFUN, to enable computations at a mantissa size equivalent to ~ 250 decimal digits, i.e. about 830 mantissa bits. This made the fractional error due to truncation and round-off be of order $\sim 10^{-250}$ which is negligible as compared with the quadrature error for the high-order quadratures investigated (see Sect. 4). The remaining fractional error at ultra-high or infinite precision is just the quadrature fractional error, 10^{-P} where P is the quadrature precision. P can thus be unambiguously identified and is our main concern in this paper.

(c) It shall be convenient here to define the *precision increment*, ΔP, as the number of additional significant decimal digits in the answer for the integral when the size of the subintervals is halved and their number thereby doubled. As argued later in Sect. 3, this proves to be mainly a function of the quadrature rule used and largely independent of the integrand function. As mentioned above in (b), to numerically identify the precision increment one must work with a very long (ideally infinite) mantissa.

(d) By *initial precision* we shall mean the precision obtained by the quadrature when applied over the full interval, $[a, b]$. This depends on both the quadrature rule and the integrand function. Occasionally (with, for example, rapidly changing integrands) the initial precision may be zero, i.e. without a single correct leading decimal digit in the answer. In this case one must ensure a greater number of subintervals to yield at least a few significant decimal digits, i.e. to obtain $P > 0$. Having got $P > 0$, the "full" quadrature *precision increment* can, in our experience, thereafter be expected with every subsequent doubling of the subinterval numbers, subject, always, to the mantissa being sufficiently large for the quadrature precision improvement to be unambiguously identified without being compromised by truncation/round-off errors.

3 Error and precision estimates in quadratures

Let the integrand function, $f(x)$, in (1) be smooth and non-singular. With the use of the integral mean value theorem, Gauss showed that the quadrature fractional error, $\delta I/I$, expected in an N-point Gauss–Legendre evaluation of I is of the form (see Davis and Rabinowitz [3], p 75)

$$\left|\frac{\delta I}{I}\right|_{\text{Gauss}-\text{Legendre}} = 10^{-P} \approx \frac{(b-a)^{2N+1} f^{(2N)}(\bar{x})}{(2N+1)!(b-a) f(\tilde{x})} \propto \frac{(b-a)^{2N}}{(2N+1)!} \tag{3}$$

where we have used the notation

$$f^{(2N)}(\bar{x}) \equiv \left[\left(\frac{\partial}{\partial x}\right)^{2N} f(x)\right]_{x=\bar{x}} \tag{4}$$

and $\bar{x}$ and $\tilde{x}$ are both 'mean values' of x within the integration "volume", $[a, b]$

For integrands that are polynomials of order $(2N-1)$ (or less), i.e. $f(x) = p_{2N-1}(x)$, it is clear from (3) that $f^{(2N)}(\bar{x}) = 0$ wherever $\bar{x}$ might be within the $[a, b]$ interval. Consequently, the quadrature error will be zero and, indeed, the latter form, $p_{2N-1}(x)$, has been termed the Quadrature-Defining Functional Form [4] for the N-point Gauss–Legendre quadrature since it can be constructed purely from the requirement that it should be exact for any polynomial integrand of order $\leq (2N-1)$. Other quadrature rules stemming from other Quadrature-Defining Functional Forms catering for various types of end-point singular, though integrable, behaviours have been presented by Harris and Evans [5]. In this paper we shall be concerned only with more general integrands that are not exactly integrated by the quadrature, i.e. with non-zero quadrature error.

From (3) we see that the fractional error diminishes rapidly with increasing N because of the $1/(2N+1)!$ factor. On the other hand it is proportional to $(b-a)^{2N}$, i.e. the interval length raised to the $2N$th power. With M (equal) subintervals (3) applies to each subinterval with the interval length evidently being $(b-a)/M$. Therefore one is led to expect, at least for smooth "polynomial-like" functions

$$\frac{\text{Fractional error with } M_1 \text{ subintervals}}{\text{Fractional error with } M_2 \text{ subintervals}} \sim \frac{10^{-P_{M_1}}}{10^{-P_{M_2}}} \sim \left(\frac{M_2}{M_1}\right)^{2N}. \tag{5}$$

where P_{M_1} and P_{M_2} are the precisions of the answers obtained with M_1 and M_2 subintervals, respectively. On taking logarithms

$$P_{M_2} - P_{M_1} = \log_{10}((M_2/M_1)^{2N}) = 2N \log_{10}(M_2/M_1) \tag{6}$$

Note this depends only on N and the ratio M_2/M_1 which argues for the precision increment to be largely independent of the integrand function, $f(x)$.

For the case when $M_2/M_1 = 2$ the left-hand-side of (6) gives the theoretical precision increment as defined in Sect. 2(c) and based on the Gauss error formula viz.

$$\Delta P = \log_{10}(2^{2N}) = 2N\log_{10}(2) \approx 0.6N \tag{7}$$

which, for the 100-point Gauss–Legendre quadrature, becomes $\Delta P \approx 60$.

By the same reasoning if $M_2/M_1 = 3$ one would expect from (6) to gain $2N\log_{10}(3) \approx 0.95 \times N$, i.e. around 95 additional significant decimal digits (always assuming the mantissa length of the computation is sufficiently long for the improvement to be identifiable).

With rapidly oscillating integrands the above will still apply provided the density of subintervals/abscissae is sufficiently large in the interval regions of rapid change to enable some non-vanishing degree of precision i.e. $P > 0$.

There is also the possibility of matching the distribution of subinterval sizes, so that they are smaller in integrand regions of rapid change and vici versa. This "tuning" can improve the accuracy without increasing the load on the CPU but is beyond the remit of this paper.

The above testifies that there is much to be gained by increasing the number of subintervals when using quadratures of high order.

4 Some high-order quadrature examples

To practically illustrate the features to be described in Sect. 3 we shall here analyse two simple but contrasting integrals. First consider the integral,

$$\int_0^1 \mathrm{d}x \frac{4}{1+x^2} = \pi, \tag{10}$$

as evaluated with David Bailey's multiprecision package [1, 2], MPFUN, that was developed as an extension to the FORTRAN language. Working to a specified precision of 250 decimal digits, one finds the results, as evaluated with a 100-point Gauss–Legendre quadrature to be (quoting only the leading 200 digits with the insignificant decimal digits greyed out)

```
   Number of subintervals=1     : 132 significant decimal digits
3.14159265358979323846264338327950288419716939937510582097494459230781
64062862089986280348253421170679821480865132823066470938446095510655 43
59614923977308102273601384208175591979201298537606216404335917,

   Number of subintervals=2     : 184 significant decimal digits
3.14159265358979323846264338327950288419716939937510582097494459230781
64062862089986280348253421170679821480865132823066470938446095505822 31
72535940812848111745028410270193852110555964471578328864781692,

   Number of subintervals=3     : exact to all digits shown (>201)
3.14159265358979323846264338327950288419716939937510582097494459230781
64062862089986280348253421170679821480865132823066470938446095505822 31
72535940812848111745028410270193852110555964462294895493038195,
```

We mention that the above considerations also apply to closed N-point Newton–Cotes quadratures. In the case of the order, N, being odd, the analogue of the Gauss fractional error formula, (3), is

$$\left|\frac{\delta I}{I}\right|_{\text{Newton}-\text{Cotes}} = 10^{-P} \approx \frac{(b-a)^{N+1} f^{(N+1)}(\bar{x})}{(N+1)!(b-a)f(\tilde{x})} \propto \frac{(b-a)^N}{(N+1)!} \tag{8}$$

which, analagously, leads to a precision increment of the form

$$\Delta P = N\log_{10}(2) \approx 0.3N \tag{9}$$

which similarly scales with N though only half the size of the precision increment of the N-point Gauss–Legendre quadrature.

Note that the integrand, $4/(1+x^2)$, is a very smooth, "polynomial-like" function over the integration domain, $[0, 1]$, though not itself a polynomial of finite order, which accounts for the very large initial precision of 132. With 2 subintervals we obtain a further 52 significant digits in accord with ≈ 60 as predicted by the Gauss' error formula. If we doubled again we would need to compute with a larger mantissa to be able to "identify" the next precision increment of significant decimal digits.

In the above integral, the integrand function was "smooth", i.e. "slowly varying". So, it might be thought that there would be no comparable increase in precision if a rapidly varying integrand was integrated by the same method. Accordingly we investigated a simple integral with such a rapidly oscillating integrand viz.

$$\int_0^1 \mathrm{d}x \sin(2000x) = \frac{1}{2000}[1 - \cos(2000)] \tag{11}$$

by the same 100-point Gauss–Legendre quadrature. The results to 200 decimal digits are given below. As before the figures in black denote significant decimal digits with the insignificant digits greyed out. The first value, obtained with Maple, is exact to all shown digits.

```
    Exact value: (using Maple_14)
6.83729774550415664889447683466044947512589644944335508283739737851596
17394240192843624432830352303808028362983851818021225346966617758065 90
81825966174484532302012756070073982560069641198311882300968 39e-4

   Number of subintervals=4     : 0 significant decimal digits
1.59330941239588679898301389885960215332526835113857101804182899508465
22512699764294056236800752799536042302396194334639449389902524210727 92
37669646267351456301039871603127829310651129714623608986025141e-2

   Number of subintervals=8     : 21 significant decimal digits
6.8372977455041566488950473080014498309070860949434881314179685502573 8
61202742719067547311577221333837695637263658339865967515386786935755 98
30941447746108613346720721414250849126949973872253883206710238e-4

   Number of subintervals=16    : 75 significant decimal digits
6.83729774550415664889447683466044947512589644944335508283739737851596
17394258415738786583067638165055105646020375354576280520120357695440 53
45536916723574282209985218093536398022829161787269120119714 62e-4
```

It will be observed that for $N = 4$ (in fact $N \leq 5$) the obtained value is so wide of the mark so as not to yield even a single significant decimal digit. This, in a sense, is a consequence of having more turning points than abscissae per interval (or subinterval) within which the quadrature is applied. Evidently a polynomial of order $(2N - 1)$ can manage (at most) $(2N - 2)$ turning points so that decreasing the interval to increase the density of abscissae is imperative to get answers with some precision, $P > 0$. In the above example we used eight subintervals to get an adequate density to follow the rapid oscillations of the $\sin(2000x)$ integrand yielding an answer with a precision of 21. Thereafter, as can be seen, the precision gains enormously, by ≈ 60 digits for $N = 100$, with every succeeding doubling of the subinterval number, in accord with (7).

Note the fact that $\Delta P \propto N$ for Gauss–Legendre quadratures is overwhelmingly important if precision is the goal. Suppose the CPU time taken to evaluate the integrand function is τ, and the 100-point Gauss quadrature was applied over M subintervals. Naturally this would take a time of $100\,M\tau$. If it was decided to do the same task with a quadrature of lesser order, e.g. a 10-point Gauss quadrature, then, to stick with $100M$ integrand evaluations, one would need apply the quadrature over $10M$ subintervals. The precision, being proportional to $(2N + 1)!$, would naturally favour the 100-point quadrature, but what of the precision increment? For the 100-point quadrature this would be ≈ 60 as argued above, and for the 10-point quadrature it would be $2 \times 10 \log_{10}(2) \approx 6$. Hence on doubling the number of subintervals (which is tantamount to doubling the number of integrand evaluations and, thereby, doubling the CPU time to $200M\tau$) about 60 additional significant decimal digits would be expected from the 100-point quadrature, whereas only ≈ 6 could be expected from the 10-point quadrature. This is the fundamental reason why, for precision and speed of computation, it is compelling to work with high-order quadratures that have the highest initial precision and the highest precision increment.

To illustrate this we revisit the above integral and re-evaluate it with a Gauss–Legendre 10-point quadrature (with weights and abscissae evaluated to a precision of 250 significant decimal digits) obtaining the results.

```
   Number of subintervals=80     : 0 significant decimal digits
7.89662022695276069907645786037449752319223900069971186403331061975370
73580701588536952680653337163342161658489496852255876971185336811729 37
57737096576750260517278627428829152198527901725806212451686711e-4

   Number of subintervals=160    : 5 significant decimal digits
6.83730184112462489722368128414637543861586925682318662809660646477863
87947054754491796383450709382848122105792885320239675546830394058443 39
84941130868755671661790982559516632189813633384992909999855730e-4

   Number of subintervals=320    : 11 significant decimal digits
6.83729774549869030716969429185545697669308412024140431766025262331166
98893146265981872054921416936360547073850682050937275946819565434022 07
56069968530403015382733365635234206633490289718462859924128717e-4
```

Note that with 80 subintervals (that corresponds to just 8 subintervals of the higher order quadrature), not a single significant decimal digit was obtained, whereas, for the 100-point quadrature, 21 were obtained with 8 subintervals. Both of these entail the evaluation of the integrand 800 times. With 160 subintervals of the 10-point quadrature one obtains just 5 significant decimal digits and with 320 subintervals one obtains 11. Note that 11–5 = 6, i.e. 11 is greater than 5 by the precision increment, 6, of the 10-point Gauss–Legendre quadrature, in accord with the discussions above.

The authors have numerically verified the above precision attributes in diverse test integrals with both smooth and rapidly changing integrands with the aid of David Bailey's multiprecision package, MPFUN [1, 2]. The weights and abscissae for the Gauss–Legendre quadratures were evaluated at a mantissa length equivalent to ~250 decimal digits, as were the quadrature evaluations that utilised them. In all cases the values obtained were qualitatively as predicted by the Gauss' error formula, (3).

5 Overview of diffraction theory

Before describing the results of the numerical analysis of the Rayleigh–Sommerfeld diffraction integral, let us briefly review the basics of diffraction theory. Ever since the work of Huygens [6] there has been intense debate on how to properly account for the evolution or propagation in systems (light, sound etc.) that obey the wave equation, see, for example, Born and Wolf [7], Jackson [8], Goodman [9], Baker and Copson [10], Copson [11], Luneburg [12]

$$\left(\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right) u(\boldsymbol{r},t) = \Box^2 u(\boldsymbol{r},t) = 0 \tag{12}$$

A rigorous solution would demand that one solves the Dirichlet boundary value problem within domains where the wave equation applies. Towards the end of the nineteenth century this was independently solved by Sommerfeld [13, 14] and by Rayleigh [15] through the application of Green's function methods to the boundary value problem associated with steady, monochromatic solutions of the wave equation.

Figure 1 shows the geometry of the problem for the specific case herein considered. The circular aperture of radius, a, lies in the source (or aperture) plane, $z = 0$, and a monochromatic plane wave with wave-number, $k_0 = 2\pi/\lambda_0$, is incident normally on it from behind. The domain of Green's theorem is the "half-space", $z > 0$. The amplitude, $u(\boldsymbol{r})$, where $\boldsymbol{r} = x\hat{\boldsymbol{i}} + y\hat{\boldsymbol{j}}$, is given on the planar surface, $z = 0$. The observation point, P, is at a distance, z_0, from the source plane and at a radial distance, $x_0\hat{\boldsymbol{i}} + y_0\hat{\boldsymbol{j}}$, from the perpendicular axis, $x = y = 0$. Then the position vector of P will be $\boldsymbol{r_0} = x_0\hat{\boldsymbol{i}} + y_0\hat{\boldsymbol{j}} + z_0\hat{\boldsymbol{k}}$. Note also that if $u(\boldsymbol{r})$ is axi-symmetric on the source-plane, S, then, by symmetry, its evolution, i.e. $u(\boldsymbol{r_0}, \boldsymbol{r}, z)$ will be axi-symmetric and thereby independent of the polar angle, θ.

Essentially what Rayleigh and Sommerfeld did was solve the boundary value problem for steady solutions of the time-independent wave equation, i.e. the *Helmholtz*

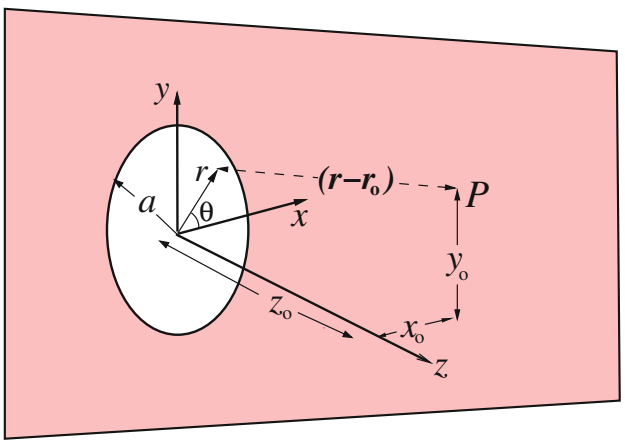

Fig. 1 Circular aperture diffraction geometry

equation, where the monochromatic amplitude is steady and everywhere $\propto e^{i\omega_0 t}$

$$-\left(k_0^2+\nabla^2\right)u(\boldsymbol{r})=0 \quad \text{where } k_0=\frac{\omega_0}{c} \tag{13}$$

As stated above, to solve the current problem we shall need a Green's function obeying

$$-\left(k_0^2+\nabla^2\right)g(k_0,\boldsymbol{r_0},\boldsymbol{r})=\delta(\boldsymbol{r}-\boldsymbol{r_0}) \tag{14}$$

As is well-known, the intrinsic Green's function satisfying (14) and subject to Sommerfeld boundary conditions, i.e. "outgoing waves" $\propto e^{ik_0 r}$ in the far-field, may be shown to be (see for example Goodman [9])

$$g(k_0,|\boldsymbol{r}-\boldsymbol{r_0}|)=\frac{e^{ik_0|\boldsymbol{r}-\boldsymbol{r_0}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r_0}|} \tag{15}$$

To solve the boundary value problem requires a form of Green's theorem for Helmholtz's equation. This follows from the identity

$$g(\nabla^2+k_0^2)u-u(\nabla^2+k_0^2)g=\nabla.(g\nabla u-u\nabla g) \tag{16}$$

whence, by Gauss' divergence theorem, and using (13) and (14)

$$u(\boldsymbol{r_0})=\oint_S \boldsymbol{dA}.(g(k_0,|\boldsymbol{r}-\boldsymbol{r_0}|)\nabla u-u\nabla g(k_0,|\boldsymbol{r}-\boldsymbol{r_0}|)) \tag{17}$$

The above gives the solution in terms of $u(\boldsymbol{r_s})$ and the normal component of ∇u on the bounding surface, $\boldsymbol{r}=\boldsymbol{r}_s$. In certain idealised situations we can find a complementary function which can be added to $g(k_0,|\boldsymbol{r}-\boldsymbol{r_0}|)$ so as to yield a special (Dirichlet) Green's function, $w(k_0,\boldsymbol{r_0},\boldsymbol{r})$, which has the property of vanishing on the relevant part of the planar bounding surface, $z=0$, and sufficiently convergent as $|\boldsymbol{r}|\to\infty$ for the contribution from the hemispherical surface to be proven zero as its radius$\to\infty$. Given such a Green's function the first term on the r.h.s. of (17) will be zero, so enabling a solution that depends only on $u(\boldsymbol{r}_s)$ viz.

$$u(\boldsymbol{r_0})=-\oint_S \boldsymbol{dA}.\nabla w(k_0,\boldsymbol{r_0},\boldsymbol{r})_{\boldsymbol{r}=\boldsymbol{r}_s}u(\boldsymbol{r}_s) \tag{18}$$

Such special Green's functions normally involve "image" Green's functions that cancel the values of the intrinsic Green's function, $g(k_0,|\boldsymbol{r}-\boldsymbol{r_0}|)$, at every relevant point on the bounding surface.

In the current situation the domain of Green's theorem is the half-space, $z>0$. A Dirichlet Green's function would be required to be zero on the source (or aperture) plane, $z=0$. This can be achieved by subtracting $g(k_0,\bar{\boldsymbol{r}}_{\boldsymbol{0}},\boldsymbol{r})$, where $\bar{\boldsymbol{r}}_{\boldsymbol{0}}[\equiv(x_0,y_0,-z_0)]$ is the image of $\boldsymbol{r_0}$ in the aperture-plane, i.e.

$$w(k_0,\boldsymbol{r_0},\boldsymbol{r})=\frac{e^{ik_0|\boldsymbol{r}-\boldsymbol{r}_{\boldsymbol{0}}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r_0}|}-\frac{e^{ik_0|\boldsymbol{r}-\bar{\boldsymbol{r}}_{\boldsymbol{0}}|}}{4\pi|\boldsymbol{r}-\bar{\boldsymbol{r}}_{\boldsymbol{0}}|} \tag{19}$$

Note that $g(k_0,\bar{\boldsymbol{r}}_{\boldsymbol{0}},\boldsymbol{r})$ satisfies the homogeneous Helmholtz equation at all points within the domain of Green's theorem and therefore is a perfectly valid complementary function, a multiple of which can be added or subtracted to the intrinsic Green's function, $g(k_0,|\boldsymbol{r}-\boldsymbol{r_0}|)$. Furthermore, $w(k_0,\boldsymbol{r_0},\boldsymbol{r})$ can easily be seen to decay as $\sim 1/|\boldsymbol{r}|^2$ as $|\boldsymbol{r}|\to\infty$ which is sufficient to ensure that the contribution from the large hemispherical surface at infinity will be zero. Hence using $w(k_0,\boldsymbol{r_0},\boldsymbol{r})$ and $\boldsymbol{dA}\equiv-\hat{\boldsymbol{k}}\mathrm{d}x\,\mathrm{d}y$ on the source plane, one obtains from (18)

$$\begin{aligned}u(\boldsymbol{r_0})=&\iint \mathrm{d}x\,\mathrm{d}y\left[\left(\frac{\partial}{\partial|\boldsymbol{r}-\boldsymbol{r_0}|}\frac{e^{ik_0|\boldsymbol{r}-\boldsymbol{r_0}|}}{4\pi|\boldsymbol{r}-\boldsymbol{r_0}|}\right)\left(\frac{z-z_0}{|\boldsymbol{r}-\boldsymbol{r_0}|}\right)\right]_{z=0}u(x,y)\\&-\iint \mathrm{d}x\,\mathrm{d}y\left[\left(\frac{\partial}{\partial|\boldsymbol{r}-\bar{\boldsymbol{r}}_{\boldsymbol{0}}|}\frac{e^{ik_0|\boldsymbol{r}-\bar{\boldsymbol{r}}_{\boldsymbol{0}}|}}{4\pi|\boldsymbol{r}-\bar{\boldsymbol{r}}_{\boldsymbol{0}}|}\right)\left(\frac{z+z_0}{|\boldsymbol{r}-\bar{\boldsymbol{r}}_{\boldsymbol{0}}|}\right)\right]_{z=0}u(x,y),\end{aligned} \tag{20}$$

i.e. the solution is a quadrature that applies within the "half-space", $z>0$. The boundary values of the scalar amplitude are assumed given on the boundary plane, $z=0$, and to tend to zero on the bounding source-plane as the radial distance tends to infinity. This then gives the celebrated Rayleigh–Sommerfeld quadrature solution for a scalar wave amplitude under the current boundary conditions viz.

$$\begin{aligned}u(\boldsymbol{r_0})=&\iint_{z=0}\mathrm{d}x\,\mathrm{d}y\left(\frac{e^{ik_0|\boldsymbol{r}-\boldsymbol{r_0}|}}{2\pi|\boldsymbol{r}-\boldsymbol{r_0}|^2}\right)\\&\times(1-ik_0|\boldsymbol{r}-\boldsymbol{r_0}|)\left(\frac{z_0}{|\boldsymbol{r}-\boldsymbol{r_0}|}\right)u(x,y)\end{aligned} \tag{21a}$$

$$\begin{aligned}u(\boldsymbol{r_0})=&\iint_{z=0}r\mathrm{d}r\,\mathrm{d}\theta\left(\frac{e^{ik_0|\boldsymbol{r}-\boldsymbol{r_0}|}}{2\pi|\boldsymbol{r}-\boldsymbol{r_0}|^2}\right)\\&\times(1-ik_0|\boldsymbol{r}-\boldsymbol{r_0}|)\left(\frac{z_0}{|\boldsymbol{r}-\boldsymbol{r_0}|}\right)u(r,\theta)\end{aligned} \tag{21b}$$

where $\boldsymbol{r}=x\hat{\boldsymbol{i}}+y\hat{\boldsymbol{j}}\equiv r\cos(\theta)\hat{\boldsymbol{i}}+r\sin(\theta)\hat{\boldsymbol{j}}$ specifies a point in cartesian or polar co-ordinates on the source plane, $z=0$. Equation (21a) and (b) show the forms that the R–S integral takes if the boundary integration over the source plane is done in cartesian or polar co-ordinates, respectively.

The evaluation of such integrals that involve rapidly oscillating integrands has been problematic in the past. This necessitated approximations to be made, the simplest of which was to ignore the "1" term in the central bracket of the integrand of (21a) and (b) whenever $k_0|\boldsymbol{r}-\boldsymbol{r_0}|\gg 1$

(Fresnel–Kirchoff approximation[16]). To be valid everywhere we would need

$$k_0|\boldsymbol{r}-\boldsymbol{r_0}|_{min} \gg 1 \tag{22a}$$

which, as $|\boldsymbol{r}-\boldsymbol{r_0}|$ is everywhere $\geq z_0$, would be satisfied if

$$k_0 z_0 \gg 1 \quad \text{i.e.} \quad \zeta = \frac{1}{k_0 z_0} \ll 1 \tag{22b}$$

where we have introduced the dimensionless parameter, ζ. If $\zeta \ll 1$ we are fully justified in ignoring the "1" in the central bracket whereas if $\zeta \gg 1$ the converse is true.

Several examples of solutions and their ζ-parameter values will be given in Sect. 6.

Other approximations were due to Fresnel et al. [17], etc. that had their own domains of validity (for a succinct account see the report of Pritchett [18]). With the advent of fast desktop PC's such approximations need no longer be made as the exact precise amplitudes can be calculated relatively quickly.

5.1 Rayleigh–Sommerfeld integral and the Fresnel approximation

According to the notation defined in Fig. 1, which, in particular, depicts a circular aperture of radius a, we see that

$$\begin{aligned}\boldsymbol{r}-\boldsymbol{r_0} &= (r\cos(\theta)-x_0)\hat{\boldsymbol{i}} + (r\sin(\theta)-y_0)\hat{\boldsymbol{j}} - z_0\hat{\boldsymbol{k}} \\ &= \boldsymbol{r}-\boldsymbol{\rho_0} - z_0\hat{\boldsymbol{k}}\end{aligned} \tag{23}$$

where $\boldsymbol{\rho_0} = x_0\hat{\boldsymbol{i}} + y_0\hat{\boldsymbol{j}}$ is the vector radius in the observation plane, $z = z_0$. Hence, with the hypothesis that z_0 is very large compared to other lengths, one seeks an expansion in inverse powers of z_0 viz.

$$|\boldsymbol{r}-\boldsymbol{r_0}| = (z_0^2 + |\boldsymbol{r}-\boldsymbol{\rho_0}|^2)^{1/2} \tag{24}$$

$$|\boldsymbol{r}-\boldsymbol{r_0}| = z_0\left[1+\frac{|\boldsymbol{r}-\boldsymbol{\rho_0}|^2}{z_0^2}\right]^{1/2} \tag{25}$$

$$|\boldsymbol{r}-\boldsymbol{r_0}| \approx z_0\left[1+\frac{1}{2}\left(\frac{|\boldsymbol{r}-\boldsymbol{\rho_0}|^2}{z_0^2}\right) - \frac{1}{8}\left(\frac{|\boldsymbol{r}-\boldsymbol{\rho_0}|^2}{z_0^2}\right) + \cdots\right] \tag{26}$$

whose convergence demands, by the ratio test, that

$$\frac{|\boldsymbol{r}-\boldsymbol{\rho_0}|^2}{z_0^2} < 1$$

be true for all values of $\boldsymbol{r}$ within the aperture. Taking the worst case ($r = a$ with $\boldsymbol{r}$ and $\boldsymbol{\rho_0}$ anti-parallel) one concludes the series will be convergent only if

$$\frac{(r+\rho_0)^2}{z_0^2} < \frac{(a+\rho_0)^2}{z_0^2} < 1 \quad \text{or, simply} \quad \left|\frac{a+\rho_0}{z_0}\right| < 1. \tag{27}$$

Subject to the above (Fresnel) constraint the factors $|\boldsymbol{r}-\boldsymbol{r_0}|$ may be replaced by z_0 everywhere except in the exponential of (21) where we need keep terms up to quadratic order, i.e. the leading two terms as shown in (26) (Fresnel approximation). This is justified if the maximum phase change due to the next higher-order, i.e. quartic term in the expansion be much less than 1 radian. For this we require

$$\frac{k_0 z_0}{8}\left(\frac{|\boldsymbol{r}-\boldsymbol{\rho_0}|^2}{z_0^2}\right)^2 \leq \frac{k_0 z_0}{8}\left(\frac{(r+\rho_0)^2}{z_0^2}\right)^2 \ll 1 \tag{28}$$

$$\text{i.e. } z_0^3 \gg \frac{\pi(r+\rho_0)^4}{4\lambda_0} \approx \frac{\pi\rho_0^4}{4\lambda_0} \tag{29}$$

Within these Fresnel–Kirchoff approximations one may neglect the "1" term in the central bracket of the integrand of (21b). Then, specifically referring to the circular aperture of Fig. 1, the appropriate R–S integral, (21b), becomes

$$\begin{aligned}u(\boldsymbol{r_0}) = \frac{-ik_0}{z_0}\int_0^a r\mathrm{d}r \int_0^{2\pi}\frac{\mathrm{d}\theta}{2\pi} \\ \times \exp\left(ik_0 z_0\left[1+\frac{r^2+\rho_0^2-2r(x_0\cos(\theta)+y_0\sin(\theta))}{2z_0^2}\right]\right)u(r,\theta)\end{aligned}$$

For axi-symmetric initial amplitudes, $u(r,\theta) \equiv u(r)$, the integration over the polar angle eventually yields

$$\begin{aligned}u(\boldsymbol{r_0}) = \frac{-ik_0}{z_0}\exp\left(ik_0 z_0 + i\frac{k_0\rho_0^2}{2z_0}\right)\int_0^a r\,\mathrm{d}r \\ \times \exp\left(\frac{ik_0 r^2}{2z_0}\right)J_0\left(\frac{k_0 r\rho_0}{z_0}\right)u(r)\end{aligned} \tag{30}$$

where the amplitude is given in terms of a one-dimensional quadrature involving the zeroth order Bessel function of the first kind, $J_0(\ldots)$ the above will, by (29), be valid as long as $\rho_0 \ll \lambda_0^{1/4} z_0^{3/4}$. If we can further assume (Fraunhofer approximation) that, for every point in the aperture,

$$\exp\left(\frac{ik_0 r^2}{2z_0}\right) \approx 1 \quad \text{i.e.} \quad z_0 \gg \frac{k_0 a^2}{2} \equiv \frac{\pi a^2}{\lambda_0}, \tag{31}$$

i.e. $z_0 \gg$ Rayleigh length $\equiv a^2/\lambda_0$, then (30) becomes

$$u(\boldsymbol{r_0}) = \frac{-ik_0}{z_0}\exp\left(ik_0 z_0 + i\frac{k_0\rho_0^2}{2z_0}\right)\int_0^a r\,\mathrm{d}r J_0\left(\frac{k_0 r\rho_0}{z_0}\right)u(r) \tag{32}$$

In the important case of the aperture being uniformly illuminated from behind the aperture plane one may put $u(r) = Uh(a - r)$ where $h(\ldots)$ denotes the Heaviside step function. Bearing in mind the identity, $xJ_0(x) = d/dx(xJ_1(x))$, the r-integration can be done analytically to yield

$$u(\boldsymbol{r_0}) = \frac{-ia}{\rho_0}\exp\left(ik_0 z_0 + i\frac{k_0\rho_0^2}{2z_0}\right)J_1\left(\frac{k_0\rho_0 a}{z_0}\right)U \tag{33}$$

and, hence, taking the moduli,

$$|u(\boldsymbol{r_0})| = \frac{a}{\rho_0}J_1\left(\frac{k_0\rho_0 a}{z_0}\right)|U| \tag{34}$$

which is the well-known result of Airy [17] that ascribes the radii of the dark concentric rings of the "top-hat" circular aperture diffraction pattern to the zeros of the first order Bessel function, $J_1(x)$.

6 Results

We now present some results of evaluating the diffraction integral of a plane wave incident normally from behind onto a circular aperture of radius a in the source plane, $z = 0$. This is generally assumed to produce a circular "top-hat" boundary value function emanating from the circular aperture. The R–S double integral, (21b), was evaluated in Fortran double precision with a nested application of two 100-point Gauss-Legendre quadratures over the two finite domains, $[0 \le r \le a]$ and $[0 \le \theta \le 2\pi]$ (see listing in Appendix).

Although the intrinsic precision of the 100-point Gauss-Legendre quadrature was far larger than 16, the Fortran double precision mantissa length caps the attainable precision, P to a ceiling of $\le$16. The actual precision attainable depends on the conditioning of the integrand and can be less than 16 due to the accumulation of "round-off" errors whilst doing the quadrature sums. We investigated this using the program listed in Appendix. We obtained a precision of $\approx$12 significant decimal digits for the R–S amplitude with virtually all the N-point Gauss–Legendre quadratures we used. Our observations are consistent with there being $\approx$4 digits of precision being lost due to round-off accumulation in the nested quadrature summations. This then leaves us with a precision of $(16 - 4) \sim 12$ which accounts for what is typically seen.

Despite this ceiling there is still an advantage when using high-order quadrature in double-precision. Imagine one plots the intrinsic precision attainable with an N-point Gauss–Legendre quadrature when integrating integrands that range from being smooth and slowly varying to being singular and rapidly changing. At the singular end of this range the large N quadrature would still manage an intrinsic precision $\ge$16 over a much superior range of functions than could be got with a smaller N, i.e. lower-order quadrature. The double precision truncation result would thereby give answers with $P \sim 12$ over a much wider range than with a quadrature of inferior order. This is more than adequate for graphical plotting purposes.

To produce a 200-point plot normally took $\sim$3.2 s corresponding to a CPU time of $\sim$16 ms per plot point. These timings were for a "top-hat" $u(r, \theta)$ which is "ultra-simple" so that the quoted times should be regarded as lower bounds.

All computations were done on an INTEL PC with a 2.6 GHz CPU running the Linux operating system (kernel version 2.6.21) with `gfortran` compiler.

One of the simplest things one can do is evaluate the "on-axis" (i.e. at $x_0 = y_0 = 0$) intensity for a uniformly illuminated circular aperture from the far-field to the near-field. Thus Fig. 2 shows the on-axis intensity, $|u(z_0)|^2$ plotted for a "hard-edged" circular aperture of radius a, as a function of z_0/a over a logarithmic range extending from 0.01 to 1,000. This compares with Fig. 2a of Gillen and Guha [19] who conjectured that the number of maxima that can be seen in such a plot is a/λ_0 (which is 8 here, and was 10 in Gillan and Guha's plot). In both cases the conjecture appears valid.

Figure 3 shows various "near field" radial profiles emanating from a circular "top-hat" initial amplitude where the observation plane is very close to the source plane. All were evaluated by 100-point Gauss–Legendre

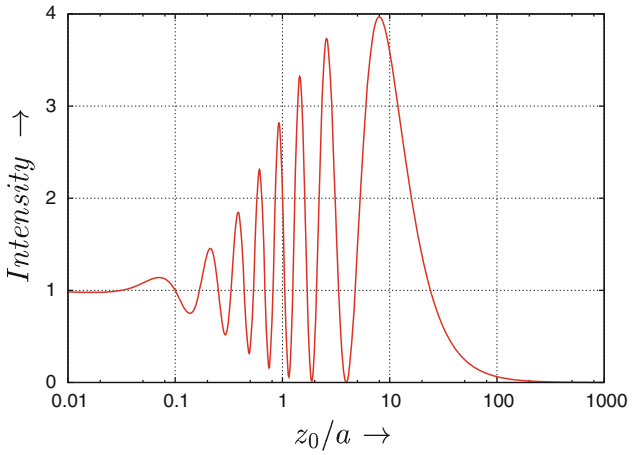

Fig. 2 The plot shows the on-axis intensity, $|u(z_0)|^2$, plotted as a function of z_0/a over a logarithmic range extending from 0.01 to 1,000 with $\lambda_0/a = 0.125$ implying ζ-parameters from 0.0000199 to 1.99 which, as λ_0/a is kept constant, takes us from the far field, $z_0/a \gg 1$ to the near field, $z_0/a \ll 1$ for a "circular top-hat" aperture function of radius, a. The plot compares with the similar result of Gillen and Guha [19] (see their Fig. 2a). Note also that, as the point moves towards the source plane, i.e. as z_0/a decreases, the first intensity minimum at $z_0/a \approx 3.93$ is a virtually complete extinction of the intensity

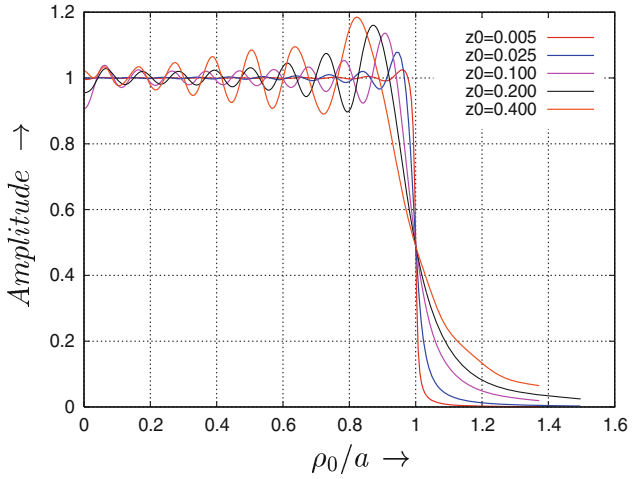

Fig. 3 Various near-field radial profiles for a "circular top-hat" axisymmetric profile, $u(r) = h(a - r)$, i.e. 1 within the aperture and zero otherwise, for values of z_0/a from 0.005 to 0.4, $\lambda_0/a = 0.1$, and with the ζ-parameters ranging over $0.04 \rightarrow 3.18$

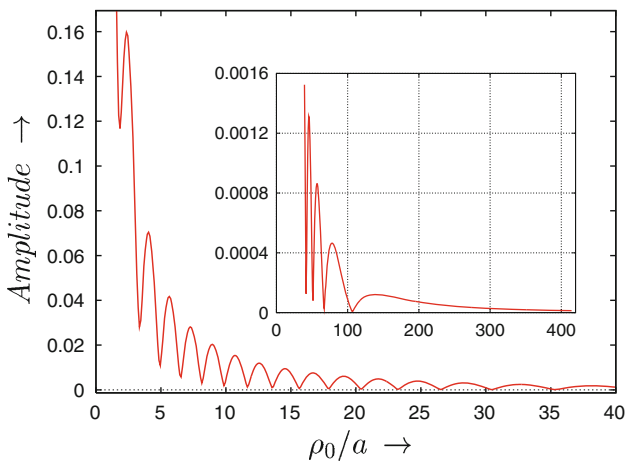

Fig. 4 The exact R–S amplitude is plotted to the distant outskirts for $z_0/a = 30$, $\lambda_0/a = 0.1$ (ζ-parameter 0.00053) for $0 \le \rho_0/a \le 400$. The inset shows the last four minima (above 40) in the far outskirts before a monotonic decay ensues

quadrature and checked for accuracy by doubling the number of subintervals. In all cases the accuracy was such that the plots could not be graphically distinguished.

Figure 4 plots the exact R-F amplitude at what might be considered to be a "middle-field" distance of $z_0/a = 30$, $\lambda/a = 0.1$ and with ρ_0/a varying from 0 to about 400 in the distant outskirts. The main graph shows all of the 15 minima in the range $[0 \le \rho_0/a \le 40]$, whereas the inset shows the last 4 minima that are found above 40 the 19th or last minimum being at $\rho_0/a \approx 106.6$ corresponding to an angle of $\tan^{-1}(106.6/30) \approx 1.30$ radians. After this the amplitude is seen to decay monotonically to zero in an exponential-like manner. This is reminiscent of Fig. 2 where there were a finite number of minima as one

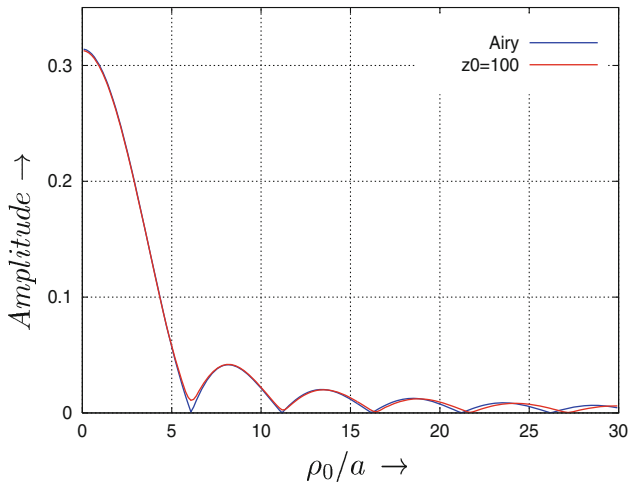

Fig. 5 Large-angle diffraction amplitude from a uniformly illuminated circular aperture of radius, a, with $z_0/a = 100$ and $\lambda_0/a = 0.1$ (ζ-parameter 0.000159). Note how the oscillations in the exact R–S amplitude (shown in *red*) extend their "inter-minimum" distance as they decay at the larger ρ_0/a values, i.e. at large angles

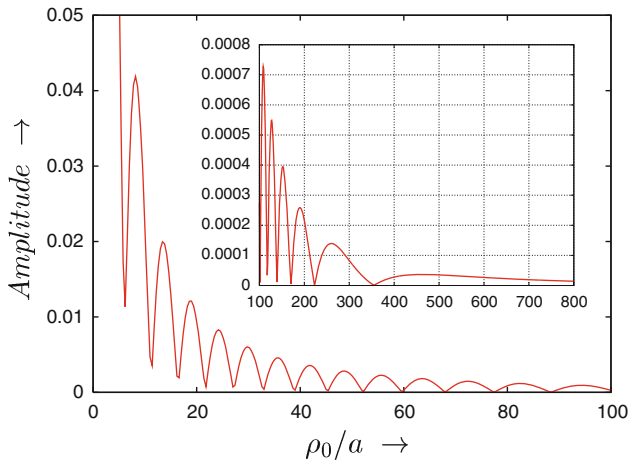

Fig. 6 Large-angle diffraction amplitude from a uniformly illuminated circular aperture of radius, a, with $z_0/a = 100$ and $\lambda_0/a = 0.1$ (ζ-parameter 0.000159). The inset continues the main plot to the region, $\rho_0/a \ge 100$. Note how the oscillations in the exact R–S amplitude extend their "wavelength" as they decay at large ρ_0/a values. They also cease, the last one being at $\rho_0/a = 355.114$, and monotonically decay thereafter

approached the origin along the z axis. It can be conjectured that a similar behaviour will be seen for all radial profiles in their distant outskirts with those in the far-field having the most minima before the onset of monotonic decay.

Figure 5, with $z_0/a = 100$ may be considered as being at the threshold of the "far-field" regime, $z_0\lambda_0/a^2 \gg 1$. Note how well the minima of the R–S amplitude, shown in red, corresponds with the zeros of the Airy function, $u(r_0) = \frac{a}{\rho_0} J_1\left(\frac{k_0 \rho_0 a}{z_0}\right)$, that is shown in blue.

One of the flexible advantages of the R–S quadrature method is that it can evaluate the amplitude at any point in the observation plane. The accuracy of its value depends only on the accuracy of the quadrature at that point. This makes it ideal to study how the amplitude behaves in the large angle regime. Figure 6 shows the behaviour of the amplitude at large angles, i.e. ρ_0 is larger or comparable with z_0 so that $\tan^{-1}(\rho_0/z_0)$ is not small—so that the Fresnel conditions do not apply. Note that, at the larger angles, the "spacing" between adjacent minima of the R–S amplitude oscillations extends as they decay at large ρ_0-values. Indeed, as was seen in Fig. 4, the decay ceases being oscillatory and enters a regime of monotonic decay instead. This radial profile plot also appears to have a finite number of minima (19) followed by an exponential-like decay as one moves radially off-axis into the far outskirts.

Figure 7 shows both the exact R–S solution and the Airy far-field approximation for $z_0 = 10,000a$, $\lambda_0 = 0.1a$ over a range $0 \le \rho_0/a \le 6,000$ (or even up to 180,000 including the inset). Note that from (29) we expect the agreement with R–S to be good only for $\rho_0 \ll \lambda_0^{1/4} z_0^{3/4} \sim 562$ for this example. At 600 we can see that the Airy approximation begins to be spatially phase-shifted slightly ahead of the exact R–S value which is a symptom of the inter-minimum R–S spacing increasing slightly at the larger angles, i.e. $\sim \tan^{-1}(6,000/10,000) = 0.5404$ radians. The inset continues the R–S solution into the far radial outskirts and, as in Fig. 4, one finds it ending in a region of monotonic decay after the last minimum at $\rho_0/a > 35,450$. This region of monotonic decay at large radial distances that is

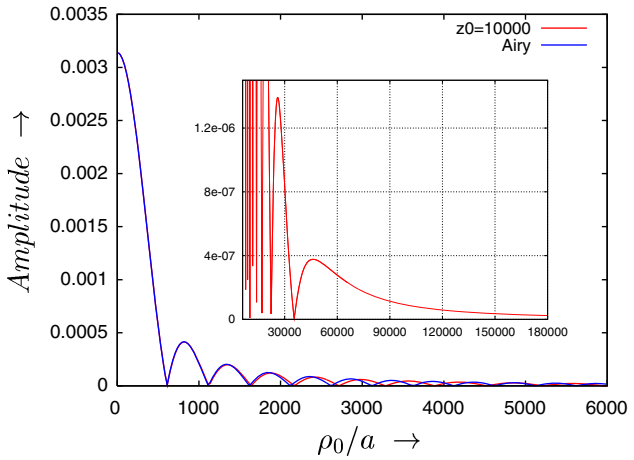

Fig. 7 The exact R–S amplitude (*red*) and the Airy approximation (*blue*) compared, in the far-field, for $z_0/a = 10,000$, $\lambda_0/a = 0.1$ (ζ-parameter = 0.0000016) for $0 \le \rho_0/a \le 6,000$. Note there is excellent agreement before the value $\lambda_0^{1/4} z_0^{3/4} \sim 562$ but slowly deteriorates thereafter insofar as the exact R–S amplitude extends its inter-minimum spacing as compared with the inter-zero spacing of the Airy approximation. The inset is a continuation of the R–S solution from $\rho_0/a = 6,000$ into the far radial outskirts

present in all the plots we have done is reminiscent of evanescent wave fields. It seems probable that this can be viewed as collimating the radiant energy so that it stays within a paraxial regime—an effect akin to the familiar total internal reflection phenomenon discussed in many texts on Optics.

7 Relative merits of high-order quadratures and FFT-based methods from a Rayleigh–Sommerfeld perspective

FFT-based evaluation of the RS integral is commonly found in the literature. We consider it as compared with the high-order evaluation of concern here. A clear exposition of the FFT-based approach can be found in Veerman [20], Shen and Wang [21]. Here we make the following observations:

1. As seen in Sect. 3 a high-order (high N) quadrature acting over a typical interval has a fractional error, see (3), which decreases rapidly as a function of N due to the factor $(2N+1)!$ in its denominator. The smallness of this term makes the *initial precision* with high-order quadratures often sufficient for most purposes without needing any interval division into subintervals to boost the precision further. Of course, if working in Fortran double precision, any higher than 16 quadrature precision gets truncated to that level of $\sim$16 significant decimals though the benefit of the *increased range* of singular integrands that can be handled without interval subdivision remains.
2. In order to obtain the amplitude at a single point with the quadrature method the function `fun1(theta)` is called $(MN)^2$ times where N is the quadrature order (typically 100 for our work in Sect. 6) and M is the subinterval number which, for simplicity, we took to be the same for both the θ- and r-integrations. A high-order quadrature with $M = 1$ gives an adequate precision over most regimes, but sometimes, at extreme near or outskirt points and with a rapidly changing integrand, a larger M may be required. Hence typically we require N^2 calls to `function fun1(theta)` (see listing in Appendix) per amplitude evaluation. If our aim was to generate data for plotting then we would need to do this for (say) 200 points. So, typically, $200 \times N^2$ evaluations of the R–S integrand would be required.

   In the FFT *plane-wave propagation* method, see Veerman et al. [20], done with a grid of $L \times L$ points, where typically $L \sim 512$, one would need evaluate the R–S initial amplitude 512^2 times on the source-plane which would give the amplitude at the same number,

i.e. 512^2, of points on the observation plane in a CPU-time that ultimately scales with L according to $L^2 \log_2(L)$ as the FFT is 2-dimensional. Note also that 2 FFT's need to be done, a forward FFT into $\boldsymbol{k}$-space and an inverse FFT. This gives one L^2 points in the grid of the observation plane from which what is needed is extracted to (say) provide data for a plot. Then a comparison of the CPU effort required for the two cases is summarised by

$$\frac{\text{Time for ``L} \times \text{L''FFT plane wave evaluation}}{\text{Time for quadrature evaluation of 200 plot points}} = \frac{K_{\text{FFT}} L^2 \log_2(\text{L})}{K_{\text{QUAD}} 200 N^2} \tag{35}$$

where K_{FFT} and K_{QUAD} are proportionality constants. If typical working values, $N \sim 100$ and $L \sim 512$, are input one finds

$$\frac{K_{\text{FFT}} \times L^2 \times \log_2(L)}{K_{\text{QUAD}} \times 200 \times N^2} = \frac{K_{\text{FFT}} \times 512^2 \times 9}{K_{\text{QUAD}} \times 200 \times 100^2} = \frac{K_{\text{FFT}}}{K_{\text{QUAD}}} \times 1.18$$

which appears to favour the quadrature method especially as K_{QUAD} is certain to be much smaller than K_{FFT}, since the evaluation of the R–S integrand is relatively much shorter (see listings of the functions `fun(r)` and `fun1(theta)` in Appendix). Also far fewer amplitude evaluations are required as $N \ll L^2$.

3. If precision is the goal then Gaussian quadrature can evaluate the integral value to a far greater precision than any FFT-based method. This is because the FFT imposes constraints on the representation of the integral insofar as all the abscissae need to be equally spaced and padded out with zeros so that one ends up with an integer power of 2 as the total abscissae number. Nearly always the integral is represented by the common 'histogram' or 'midpoint-rule' (see, for example, Atkinson [22]) where the fractional error is easily shown to be proportional to $(interval)^1$. As discussed in Sect. 3 this implies an N-point Newton–Cotes precision increment of $N \log_{10}(2)$ (for N odd) which, for $N = 1$ (mid-point rule) gives $\approx \log_{10}(2^1) = 0.3$ i.e. 0.3 additional significant decimal digits on doubling the subinterval number. This compares poorly with the N-point Gauss–Legendre value of $2N \log_{10}(2) \approx 0.6N$, being only half the latter when the order, N, is very large (there are also 'instabilities', see (4), with evaluating the weights of large order Newton–Cotes quadratures). Shen and Wang [21] in their FFT-based approach have used Simpson's rule to improve the accuracy over the common 'midpoint-rule'. Both, of course, have equally spaced abscissae, thereby lending themselves to representing an integral within the constraints of the FFT, with Simpson's rule implying unequal weights proportional to $1, 4, 2, 4, 2, \ldots, 2, 4, 2, 4, 1$ to accompany the abscissae. Simpson's rule is a 3rd-order *closed* Newton–Cotes quadrature which has a fractional error term proportional to $(\text{interval})^3$. This implies, by the same argument as in Sect. 3, that its *precision increment* is $\approx 3 \log_{10}(2) \approx 0.9$ additional significant decimal digits which, though poor, is three times larger than the 'midpoint-rule'. However, it seems improbable that such a small precision increment could be clearly seen even if the '$L \times L$' mesh-size in the FFT method was increased fourfold (by doubling L).
4. Presumably, with N equally spaced abscissae, the situation described in (3) above could potentially be improved, whilst keeping the abscissae equally-spaced, by using a higher-order Newton–Cotes quadrature. By the same argument as in Sect. 3 (9), the expected increase in precision on doubling the number of subintervals with a 100-point Newton–Cotes quadrature would be $N \log_{10}(2)$, i.e. a precision increment of about 30. However, in contrast to Gauss–Legendre quadrature, some weights in the higher-order Newton–Cotes quadrature rules are extremely large (of the order of 10^{23} for the 100-point quadrature!) and, at orders ≥ 11, are also of mixed sign, i.e. negative as well as positive. This is problematic as it causes ill-conditioning at the higher orders so that the practicality of doing this is questionable even with a formidable mantissa length. Basden and Lucchese [23] have investigated this situation and concluded that Newton–Cotes quadratures are only useable for $N \leq 16$ as, at higher N values, "instability" develops amongst the weights. By contrast, all weights of the Gauss–Legendre quadrature are positive with all its abscissae lying within the integration interval. This follows as its expansion functions viz. $1, x^2, x^4, \ldots, x^{2N-2}$ constitute a Tchebycheff system (see Davis and Rabinowitz [3, p 96–99] , Karlin and Studden [24]) over the interval, $[0, 1]$.
5. To check the accuracy of a numerical answer one normally resorts to repeating the calculation with a finer mesh size in the FFT method or with more subintervals in the Gauss quadrature case. We simply remark that this is most easy in the latter case as all we need do is alter the M-value in the calls to the quadrature subroutine, **cquad**, and recompute (see the listing of the **cquad** subroutine in Appendix). In the FFT case it seems to us that to do the equivalent poses problems in that the FFT is not suitable to

evaluate the amplitude at a single observation point at a time without evaluating the "full" set of L^2 points of the transform. This implies that checking the accuracy in the FFT case becomes a seriously tedious task involving much additional code.

6. It can be seen, see Appendix, that to evaluate a double integral in a straightforward manner involves calling the **cquad** subroutine twice in a **nested** manner, i.e. whereby the first function subroutine, `fun(r)`, calls the second function subroutine, `fun1(theta)`. Hence the **cquad** subroutine is called recursively and the programmer should check that the compiler understands how to handle recursive calls. Some (older) compilers do not, in the which case the work-around is to compile two identical **cquad** subroutines but to give them different names. This forces the two subroutines to exist in distinct "memory domains", so preventing the calculations within one `call cquad(...)` overwriting those of the other.
7. Here we simply point out that the number of subintervals in each of these quadratures need not be equal. For example when integrating over area in polar coordinates, $|dA| \equiv r\,\mathrm{d}r\,\mathrm{d}\theta$, it would be wasteful to use a large M for the θ-integration at tiny values of the radial distance, r, as $r \to 0$ and, conversely, at a larger r, a larger M might be needed. In the examples given in Sect. 6 we have used the same M for both quadratures. Clearly there is much scope for "tuning" here which will not be persued further in this paper.

It is seen from the above that most considerations favour the quadrature method. As the latter does not attempt any Fourier transform it can rival FFT-based methods in speed as well as being far better in accuracy. That said, we have been most impressed by the timings reported by Shen and Wang [21] on their FFT-DI and their FFT-AS methods.

8 Conclusions

It has herein been argued, and verified by examples done in multi-precision, that quadratures of high order, such as Gauss-Legendre, are eminently suitable for the evaluation of integrals to high precision—be they "smooth" and well-behaved or, indeed, rapidly oscillating as in several optical applications. This is due to their large, $\propto N$, precision increment (see Sect. 2c).

The technique has been appplied to evaluate the Rayleigh–Sommerfeld integral solution to the external Dirichlet problem for the wave equation in the half-space, $z > 0$, and subject to sources on the boundary plane, $z = 0$. The results shown in Sect. 6 were done with a 100-point Gauss-Legendre quadrature in 'Fortran double-precision' that, despite a truncation ceiling at $P \sim 16$, gives precisions of $P \sim 14$ over a wider *range* of singular, rapidly changing integrand functions than quadratures of lower order can manage. In rare cases at singular spatial points where the quadrature precision, $P < 12$, a re-evaluation with double the number of subintervals, M, restored the required precision of $P \sim 12$ that was adequate for our plotting needs.

The quadrature algorithm presented in the Appendix accesses its weights and abscissae from the `/wtsabs/` common block as it proceeds, thereby making it economical of memory space. It is also argued in Sect. 7 that its flexibility and simplicity makes it preferable to FFT-based methods for most tasks.

Acknowledgments WABE is indebted to Prof. Mainardi for alerting him to David Bailey's multi-precision package, MPFUN, and acknowledges some useful past discussions on quadrature topics with Prof. A.C. Genz and the late Prof. P. Rabinowitz. AT thanks Dr. Omar El Gawhary for interesting discussions on paraxial propagation-related issues. The authors also convey their thanks to an anonymous referee whose comments and suggestions have indubitably improved the paper.

Appendix: High-order quadrature program

The FORTRAN program used to evaluate the integrals is listed below. The high-order Gauss–Legendre weights and abscissae are read from data-statements in the **cquad** subroutine or, as here, read from a file in the main program and put in the **wtsabs** common block that is shared with **cquad**.

```
      program RayleighSommerfeld
      implicit real*8(a-h,o-z)
      real*8 k0
      double complex fun,s,aik0
      external fun
      common ar,m
      common /params/k0,z0,x0,y0,rho0,twopi,pi
      common/wtsabs/wtg(100),absg(100),ng2
c  The weights and abscissae should be read in from a file
c  at this point - so they reside in the /wtsabs/ common block
      open(9,file='wts_abs_file')
c  Put statements here to read in ng2 (=N/2) and wtg() and absg()
      read(9,*) ng2
      do i=1,ng2
      read(9,*) wtg(i)
      end do
      do i=1,ng2
      read(9,*) absg(i)
      end do
      close(9)

      zero = 0.
      one = 1.
      pi= 4*atan(one)
      twopi= 2*pi
c  Open file to hold results
      open(8,file='results',status='unknown')
   10 write(*,'('' Enter wavelength and aperture radius [def '',f9.3,
     + 0pf5.2,'']  Exit=^C'')') wavelength,a
      read*, wavelength,a
      k0 = twopi/wavelength
      aik0= dcmplx(zero,k0)
      print*,'Enter x0,y0,z0 to define observation point'
      read*, x0,y0,z0
      rho0= sqrt(x0**2+y0**2)
      print*, 'Enter number, m, of (equal) subintervals'
      read*, m
c  Initialise amplitude, s, to zero.
      s = dcmplx(0.d0,0.d0)
c  Now call the CQUAD complex quadrature subroutine
c  to evaluate the Rayleigh-Sommerfeld amplitude
      call cquad(s,m,zero,a,fun)
c  s now holds the result of the R-S quadrature
c  Print out answer to screen and to file.
      write(*,33) x0,y0,rho0,abs(s),abs(s)**2,dble(s),dimag(s),z0,m
   33 format(0p3f10.5,1p4e17.9,1pe14.6,i3)
      write(8,33) x0,y0,rho0,abs(s),abs(s)**2,dble(s),dimag(s),z0,m
      goto 10
      stop
      end

      subroutine cquad(s,m,xmin,xmax,fun)
      implicit real*8(a-h,o-z)
      double complex s,s1,fun
      external fun
c  the weights and abscissae are read in either by data-statements
c  here or accessed via a common block shared with the calling program
```

```
      common/wtsabs/wtg(100),absg(100),ng2
      c= (xmax-xmin)/(2*m)
      s1= dcmplx(0.,0.)
      do 10 j=1,m
      b= xmin+ (2*j-1)*c
      do 20 i=1,ng2
      s1=s1+wtg(i)*(fun(c*absg(i)+b)+fun(-c*absg(i)+b))
   20 continue
   10 continue
c    the evaluated integral is added to what was previously in s.
      s= s+ c*s1
      return
      end
```

Note that it basically evaluates the integral, $\int_{x\min}^{x\max} \mathrm{fun}(x)\,\mathrm{d}x$, over `M` equal subintervals using weights and abscissae that are "read-in" from a file by the calling program and made available to the **cquad** subroutine via the common block, **wtsabs**.[1]

To evaluate the R–S integral, which is 2-dimensional, the integrand function, $fun1(\boldsymbol{r_0}, r, \theta)$, in general, depends on θ and must be integrated numerically over this polar angle (and including any θ-dependence in $u(r, \theta)$). This gives a result, $fun(\boldsymbol{r}_0, r)$, which is then integrated over the

```
      function fun(r)
c   Evaluates the Rayleigh-Sommerfeld integrand
      implicit real*8(a-h,o-z)
      real*8 k0
      double complex aik0,fun,fun1
      external fun1
      common /params/k0,z0,x0,y0,rho0,twopi,pi
      common ar,m
      ar = r
      rz0dtwopi = r*z0/twopi
      aik0= dcmplx(0.d0,k0)
      fun = dcmplx(0.d0,0.d0)
c   Integrate over polar angle, theta, over [0,twopi]
      call cquad(fun,m,0.d0,twopi,fun1)
      fun= fun*rz0dtwopi
      return
      end

      function fun1(theta)
c   Defines the Rayleigh-Sommerfeld integrand
      implicit real*8(a-h,o-z)
      real*8 k0
      double complex aik0,fun1
      common ar,m
      common /params/k0,z0,x0,y0,rho0,twopi,pi
      r = ar
      aik0= dcmplx(0.d0,k0)
      rmr02= r*r+z0*z0+rho0*rho0-2*r*(x0*cos(theta)+y0*sin(theta))
      rmr0 = sqrt(rmr02)
c  Define, for example, a circular top-hat function of radius "a".
c  As the r-integration upper limit has been defined as "a", we put u=1.
      u=1.
      fun1 = exp(aik0*rmr0)/rmr02*(1./rmr0-aik0)*u
      return
      end
```

[1] The corresponding author can e-mail files of high-order ($N \leq 200$) Gauss–Legendre weights and abscissae evaluated to 200 significant decimals to any programmer wishing to repeat our computations

radial distance, $0 \leq r \leq \infty$ and so cover the whole aperture plane as shown schematically in (36).

$$\text{i.e. } u(\boldsymbol{r_0}) = \int_0^a \mathrm{d}r \underbrace{\int_0^{2\pi} \mathrm{d}\theta \, fun1(\boldsymbol{r_0}, r, \theta)}_{fun(\boldsymbol{r_0}, r)} \tag{36}$$

Here the programmer is alerted to the fact that, if the same quadrature is used for both the θ- and r-integrations, then it is recursively called. Most modern compilers will recognise this and take appropriate action, but there are some compilers that do not—and accordingly produce "nonsensical" output as all of both quadrature evaluations were done in the same quadrature 'memory space'. In such cases the solution is to compile two **cquad** quadrature subroutines, differing only in their calling names, and to call them appropriately in the calling program(s). This forces the compilation of a distinct subroutine for each quadrature.

References

1. D.H. Bailey, A portable high performance multiprecision package. NASA RNR Technical, Report (RNR-90-022) (1993)
2. D.H. Bailey, Multiprecision translation and execution of fortran programs. ACM Trans. Math. Softw. **19**, 288–319 (1993)
3. P.J. Davis, P. Rabinowitz, Methods of Numerical Integration (Academic Press, New York, 1975) pp. 96–99
4. C.N. Pannell, W.A.B. Evans, D.A. Jackson, A new integration technique for flowmeters with chordal paths. Flow. Meas. Instrum. **1**, 216–224 (1990)
5. C.G. Harris, W.A.B. Evans, Extension of numerical quadrature formulae to cater for end point singular behaviours over finite intervals. Int. J. Comput. Math. **B6**, 219–227 (1977)
6. C. Huygens, Trait'e de la lumi'ere, Published in Leyden (1690)
7. M. Born, E. Wolf, *Principles of Optics* (Cambridge U.P., New York, 1999)
8. J.D. Jackson, *Classical Electrodynamics*, 3rd edn. (Wiley, New York, 1998)
9. J.W. Goodman, *Introduction to Fourier Optics* (McGraw Hill, New York, 1996)
10. B.B. Baker, E.T. Copson, *The Mathematical Theory of Huygens' Principle* (Clarendon Press, Oxford, 1939)
11. E.T. Copson, Proc. Roy. Soc. (London), Series A, **186**, 100 (1946).
12. R.K. Luneburg, *Mathematical Theory of Optics* (University of California Press, Berkeley, 1964)
13. A. Sommerfeld, Zur mathematischen theorie der beugungserscheinungen. Nachr. Kgl. Acad. Wiss. Göttingen **4**, 338–342 (1894)
14. A. Sommerfeld, Optics (Academic Press, New York, 1949), p. 201. Translated by O. Laporte and P.A. Moldauer
15. Lord Rayleigh, On the passage of waves through apertures in plane screens, and allied problems. Philos. Mag. **43**, 259–272 (1897)
16. G.R. Kirchoff, Zur theorie der lichtstrahlen. Ann. Phys. (Leipzig) **18**, 663–695 (1883)
17. G.B. Airy, On the diffraction of an object glass with circular aperture. Trans. Camb. Phil. Soc **5**, 283 (1835)
18. T.M. Pritchett, Spectral Solution of the Helmholtz and Paraxial Wave Equations and Classical Diffraction Formulae. Army Research, Report (ARL-TR-3179), pp. 1–19 (2004)
19. G.D. Gillen, S. Guha, Modelling and propagation of near-field patterns: a more complete approach. Am. J. Phys. **72**, 31195–31201 (2004)
20. J.A.C. Veerman, J.J. Rusch, H.P. Urbach, Calculation of the Rayleigh-Sommerfeld diffraction integral by exact integration of the fast oscillating factor. J. Opt. Soc. A **22**, 636–646 (2005)
21. F. Shen, A. Wang, Fast-Fourier-transform based numerical integration method for the Rayleigh-Sommerfeld diffraction formula. Appl. Opt. **45**, 1102–1110 (2006)
22. K. Atkinson, *Elementary Numerical Analysis* (Wiley, New York, 1985)
23. B. Basden, R.R. Lucchese, High order Newton-Cotes integration methods in scattering theory. J. Comput. Phys. **77**, 524–536 (1988)
24. S. Karlin, W.J. Studden, *Tchebycheff Systems with Applications in Analysis and Statistics* (Wiley, New York, 1966)