Integration by Parts for Oscillatory Integrals. DRAFT

Richard J. Fateman
Computer Science Division
Electrical Engineering and Computer Sciences
University of California at Berkeley

December 13, 2011

Abstract

Assume we have a definite integral we wish to evaluate, but it looks nasty because the integrand is wildly oscillatory and the usual numerical techniques based on sampling will not work well. The approach we take is to reformulate the *indefinite* integral so it doesn't wiggle so much. That is, we compute another form for the anti-derivative. Using the fundamental theorem of integral calculus (when appropriate) allows us to evaluate the definite integral without any subdivisions, sample points, etc. Our starting point is integration by parts. In one formulation we develop an asymptotic series. In another, we approximate the non-oscillatory part in terms of simpler functions to aid in computing a (now approximate) indefinite integral.

1 Oscillatory Integrals

Consider that our integrand looks like $f(\omega x)g(x)$ where f is oscillatory, and the ω is a frequency, or akin to a frequency. For example, it might be $f(x) = \cos(\exp(\omega x))$. We are going to compute another form for its anti-derivative.

For specificity, let's choose a "Bessel function of the first kind" $J_n(\omega x)$.

For any of the Bessel functions it turns out that we can use a nice integration formula like this [we can ignore "plus a constant"]:

$$\int z^{n+1} J_n(\omega z)dz = (1/\omega)z^{n+1}J_{n+1}(\omega z). \tag{1}$$

Observe the term on the right appears reduced by a factor of ω compared to the original on the left, leading us to believe that we can take an integration problem and run it through integration-by-parts to grab some chunk of the answer and leave the rest as something multiplied by $1/\omega$. By repeating this we can grab another chunk of the answer and leave the rest as something multiplied by $1/\omega^2$, etc.

This can work as follows. To integrate

$$\int g(z)J_n(\omega z)dz \tag{2}$$

by parts (i.e. $\int udv = uv - \int vdu$), let

$$dv = z^{n+1}J_n(\omega z)dz.$$

Then by formula (1) above,

$$v = (1/\omega)z^{n+1}J_{n+1}(\omega z).$$

To finish setting up our task we must make the integrand look like $\int udv$ so we divide by the "borrowed" z^{n+1} we used to make the integration of dv possible. That is, we set

$$u = g(z)/z^{n+1}$$

and by differentiation

$$du = \frac{zg'(z) - (n+1)g(z)}{z^{n+2}} dz.$$

Rearranging pieces in $uv - \int vdu$ a bit, we get

$$\int g(z)J_n(\omega z)dz = \frac{1}{\omega} g(z)J_{n+1}(\omega z) - \frac{1}{\omega} \int \frac{g'(z) - (n+1)g(z)}{z} J_{n+1}(\omega z)dz. \tag{3}$$

Note that we have pulled out a chunk of the integral as a simple formula and also that the remaining integral on the right has the same form as the left. It differs in the index of the Bessel function which is increased by one, and the function g is more complicated. Note that the new g is still easy to differentiate, which is all that is required for integration by parts. Also observe there is a division by z which suggests that if this factor is not cancelled out with some factor of the numerator there may be a difficulty in evaluating this new $g(z)$ at $z = 0$ should a definite integration require that. Finally, note that *we have not made any approximations in these equations.* The idea is that if we iterate this K times, the final integral will be multiplied by ω^{-K} and for sufficiently large ω and K, this final term will be negligible. However, if we have some other way of computing this final term exactly (instead of neglecting it!) then the formula is exact.

By analogy with equation (1), we can take advantage of

$$\int \cos(\omega z)dz = (1/\omega)\sin(\omega z) \quad \text{and} \quad \int \sin(\omega z)\mathrm{dz} = (-1/\omega)\cos(\omega \mathrm{z})$$

or encompassing both of these in one formula via $\exp(iz) = \cos z + i \sin z$:

$$\int \exp(i\omega z)dz = (-i/\omega)\exp(i\omega z).$$

More generally, consider

$$\int \exp(i\omega f(z))dz = \frac{-i}{\omega f'(z)} \exp(i\omega f(z)). \tag{4}$$

This accounts for both $\sin(\omega f(z))$ and $\cos(\omega f(z))$ for any differentiable $f(z)$. While the resulting formula is correct, the reformulation on the right is useful mainly if we are given an $f(z)$ that is indeed a slowly varying function in the range of integration, and also that, since we are dividing by $f'(z)$, this quantity should not be zero. If, contrary to expectations, the term $(1/(\omega f'(z))$ is large, the contribution of that "remainder" integral may be large. Instead of discarding it, it should be evaluated some other way.

We can derive additional useful formulas by noting that all we are requiring is some (presumably oscillatory) function y that satisfies

$$y'(\omega z) = \omega a(z)y(\omega z) \tag{5}$$

or in the case of Bessel functions (not just of the first kind), or similar indexed special functions of physics, something like this:

$$y'_n(\omega z) = \omega a(z)y_{n+r}(\omega z). \tag{6}$$

Note the occurrence of $\omega a(z)$, not $a(\omega z)$. The latter would be far more general, but cannot always be manipulated to attenuate the oscillation by integration by parts. The additional "$+r$" might step y_n upward or downward.

Implementation of a formula like (6) takes a few lines in a typical computer algebra program. (See the Appendix for a sample). It is more difficult to decide how many iterations (and their powers of $1/\omega$) to expand, since the truncated series that is generated is asymptotic in nature, and will not converge for fixed ω and all z. Alternatively, we could try to prove that the remainder term is small. Evaluating the magnitude of the error cannot be done simply by looking at the anti-derivative, but requires consideration of limits of the definite integral, something we haven't previously mentioned in this short note. There are many variations and elaborations to speed up the processing. For example, if we are ultimately computing a definite integral, we do not really need to compute the various symbolic derivatives, but just the values of the derivatives at two points. This can be done by two runs of automatic differentiation[1].

Just to be explicit on our "akin to frequency" example, consider integrating $f(x)g(x)$ where $f(x) = \sin(\exp(\omega x))$. In order to make the parts fit together we set

$$dv = \sin(\exp(\omega x)) \times \exp(\omega x) \times \omega \times dx$$

and thus

$$v = -\cos(\exp(\omega x)).$$

Correspondingly,

$$u = \frac{g(x)}{\exp(\omega x) \times \omega}.$$

We mention again that we get into trouble if uv evaluated at an endpoint involves a division by zero or encounters some singularity.

2 Another approach: Integration by parts to lower order J, for even n

Given the identity

$$J_{n-1}(\omega x) - J_{n+1}(\omega x) = \frac{2}{\omega}\frac{d}{dx}J_n(\omega x)$$

we can write, for $n > 1$,

$$\int J_n(\omega x)dx = -\frac{2}{\omega}J_{n-1}(\omega x) + \int J_{n-2}(\omega x)$$

which recursively drives the integral immediately into representations in terms of *lower order* Bessel functions which may be preferable. Also note that we have a simple formula for the integral J_1 as a Bessel function, and a more complicated one for J_0. (see, in Appendix 2, s1[0] and s0[0], respectively.)

Now consider again defining rewriting

$$\int g(z)J_n(\omega z)dz$$

for integration by parts letting $dv = J_n(\omega z)dz$ and thus $v = -2/\omega(J_{n-1} + J_{n-3} + \cdots)$ the appropriate integration from the formula above. Then

$$\int g(z)J_n(\omega z)dz \;=\; g(z)v - \frac{2}{\omega}\int g'(z)\{J_{n-1} + J_{n-3} + ...)dx.$$

The integral on the right, if multiplied through, is of the same form as the left, but is attenuated by the $2/\omega$ factor. If n starts out as an even integer, we can ultimately reduce the result to integration of J_1, and we need no more complexity than that offered by Bessel functions. For odd n we are left with a task of integrating an nth derivative of $g(x)$ by a product of Bessel J and Struve H functions.

3 Another approach: Reduction to J_0 and J_1 first

We have casually mixed higher-order Bessel functions in with our discussion as though they were more-or-less essential. Since it is possible to rewrite all of the J_n functions of natural-number order n in terms of J_0 and J_0, perhaps that should be our first step, noting that $J_n = 2(n-1)/xJ_{n-1} - J_{n-2}$. Thus we can easily derive:

$$J_5(x) \;=\; \frac{\left(x^4 - 72\,x^2 + 384\right)\,J_1(x) + \left(12\,x^3 - 192\,x\right)\,J_0(x)}{x^4}.$$

[1]www.autodiff.org

Therefore

$$\int g(z)J_n(\omega z)dz \;=\; \int g(z)f_{0,n}(z)J_0(\omega z)dz + \int g(z)f_{1,n}(z)J_0(\omega z)dz$$

where $f_{0,n}$ and $f_{1,n}$ are the appropriate coefficients in the recurrence.

This does not immediately solve the problem, nor does the reduction necessarily result in a simpler problem unless one's criterion of simplicity is to remove when possible any higher order Bessel functions while favoring the rational-function multipliers.

4 Another approach, expansion in $x^n J_n(\omega x)$

This is another style of approach suitable for exploration in a computer algebra system. Again we use Bessel functions.

Consider approximation of $g(x)$ as a polynomial in x. Depending on the function and the range of the integral, it may be prudent to use (say) Chebyshev polynomials of modest degree, and approximate the integral in sections. This tradeoff is widely discussed in the literature, for example Clenshaw-Curtis vs. Gaussian [17]. In any case, we have (approximately) reduced the integration problem of Bessel functions of any non-negative integer order times $g(x)$ to the case of integrands of the form $x^m J_n(\omega x)$.

Given derivative properties $d/dx(xJ_1(\omega x)) = \omega x J_0(\omega x)$ and $d/dx(J_0(\omega x)) = -\omega J_1(\omega x)$, the formulas in Appendix 2 can be derived by judicious use of integration by parts. (Or see the table computed apparently in part by the Maple computer algebra system by Rosenheinrich [16]. We expect that such printed tabular results, while easily "browsed" by humans, are less useful than the digital versions, or the programs that can be used to generate them.) In two places our program must use Struve "H" functions to resolve end-cases in which we cannot express the integrals solely in terms of Bessel functions, and in another end-case, which may appear at most $k = n - 1$ times, but may not appear at all: the program simply requires an integral of an isolated Bessel function of order k. For this purpose we could leave the integral in place or use a hypergeometric function, since

$$\int J_k(ax)dx \;=\; \frac{2^{-k}(ax)^k \, {}_1F_2\left(\frac{k}{2}; \frac{k}{2}+1, k+1; -\frac{1}{4}a^2x^2\right)}{k^2\Gamma(k)}.$$

An example in which the result does not include any of the extra end-case notions is `r[4,3]`, which is the integral of $x^4 J_3(ax)$ and is exactly

$$-\frac{\left(8\,a\,x^3 - 48\,a\,x\right)\,J_1(a\,x) + \left(-a^4\,x^4 + \left(8\,a^2+16\right)\,x^2\right)\,J_0(a\,x)}{a^5}.$$

In these neat cases, to compute a definite integral (assuming the fundamental theorem of integral calculus can be applied), sometimes only two Bessel functions (of order 0, 1) need be evaluated at each endpoint. It is important to note that this result is valid regardless of the "frequency" of oscillation, and regardless of the number of terms in the polynomial approximation. That is, integrating $(x^3 + x^9)J_6(ax)$ requires no additional Bessel evaluations. In some cases, exactly two additional Struve-H evaluations are needed, and where the reduction strategy requires it, possibly some ${}_1F_2$ terms.

In order to emphasize the simplicity of this method, the program in Appendix 2 returns expressions that may need to undergo some expansion and collection of like terms (a call to "`ratsimp`" will suffice).

As an example, say we wish to compute $\int_{-1}^{1} \sin(x) * \exp(x) * J_4(1000x)dx$. The slowly varying term, $\sin(x) * \exp(x)$ can be approximated on that interval by a polynomial which we computed as a truncated Chebyshev series in Maxima:

```
chebseries (0.99305989521824,   1.228320669845807,
            0.49479528302661,   0.072410556016733, -0.0020812668931913 ,
           -0.0022506166845858, -3.4644735201735521e-4, -2.3209155509095646e-5,
      ...)
```

and then converted to an ordinary polynomial, which looks something like this (we have taken some liberties with the actual computed output to make the display more compact): $x + x^2 + 0.3333333333366\,x^3 - 0.033333333368561\,x^5 - 0.011111111110922\,x^6 - 0.0015873014195317\,x^7 + 4.4091300264470177\times 10^{-5}\,x^9 + 8.8183438004848522\times 10^{-6}\,x^{10} + 8.022060865230587\times 10^{-7}\,x^{11} 1.0638784430187349\times 10^{-8}\,x^{13} - 1.4674197568765272\times 10^{-9}\,x^{14} + \cdots$.

Given the setting in our Chebyshev conversion which are keyed to double-float precision, there are 11 monomial terms; we will need to compute values for J_0, J_1, H_0 and H_1 each at 2 arguments, -1000 and +1000, and a bunch of polynomials, plus two of the ${}_1F_2$ forms.

The rationale for truncating the Chebyshev series at a particular point must be based on estimations of acceptable accuracy, and the precision of the coefficients are (in the context of a CAS) computable to higher accuracy if needed. A discussion which can be found elsewhere (e.g. www.chebfun.org) [18]. It is also plausible to just do the expansion without knowing anything about $g(x)$ by simply using a polynomial with coefficients corresponding to an indexed indeterminate, e.g. a_0, a_1, ... although in that case it is not possible to decide how many terms are necessary for adequate accuracy; one could attempt to estimate this by measuring the computed correction from additional terms, looking for convergence. Rather than expanding the integration in terms of the monomials x^k we could try to find simple formulas in terms of Chebyshev polynomials directly. While Rosenheimrich [16] tabulates such integrals, he presents the results in the monomial representation for the first 16 Chebyshev polynomials of the first kind. Is there a neat formula or recurrence? A brief search suggests that any formula, even one that would use Chebyshev polynomials of the first and second kind (usually T and U), would not be particularly neat.

5 Other forms of integrands

There are a slew of other forms involving Bessel functions, for example expressions which include a product of Bessel functions (times something else). An interesting paper by Deun [2] explores algorithms for the (definite) integral of some such forms, collecting results from a number of earlier sources.

Watson's classic [19] contains numerous additional nuggets.

Current computer algebra systems can produce some of these formulas on demand, but some are more elusive.

6 Concluding notes

We claim no great originality for the ideas here, just a simplicity of presentation. Our intent is to show that, given a computer algebra system, it is (relatively speaking) a piece of cake to do some reformulations of what may be initially presented as numerical integration problems, although perhaps these problems might be generalized to allow the propagation of parameters (like ω). This allows us to see a continuum of solutions rather than a single value as we would get from traditional quadrature.

The references include papers on computer algebra systems and efforts to build tools for symbol-numeric integration [4, 7, 13], to which this kind of tool can be added. A more recent paper [5] has more symbolic system references. The remaining items in the bibliography are selected from a substantial literature mostly concerned with purely numerical approaches to oscillatory problems; some of them touch on asymptotic series such as these, but mostly to discount them as too complicated (true if done by hand). We suggest that an interested reader start with the easily-available on-line survey reference by Iserles *et al* [10], or Olver [15] chapters 1 and 2.

7 Appendix 1: Maxima program for asymptotic expansion

Here is an example of an *extremely bare-bones* program to generate a representation of an integral of $h(x)J_n(\omega x)$ ignoring terms after the `lim` power of ω. A slight modification allows us to keep track of the remainder (as an anti-derivative), or to compute the next term or two to gauge the magnitude of the error.

```
s(i,w,n,lim):= if (i=lim) then 0 else 1/w *(g[i]*J[n+1](w*x)-s(i+1,w,n+1,lim))$

intbypartsJ(h,x,n,w,lim):= block([],kill(g),
  g[0]:h,
  g[i]:=diff(g[i-1],x)-n/x*g[i-1],
  s(0,w,n,lim))$
```

To compute 3 terms of $\int \sin(x) J_3(\omega x) dx$ we evaluate `intbypartsJ(sin(x),x,3,omega,3)`. After some massaging to arrange the display, the result is

$$P(x,\omega) = \frac{\sin x\, J_4(\omega\, x)}{\omega} - \frac{\cos x\, J_5(\omega\, x)}{\omega^2} + \frac{4\,\sin x\, J_5(\omega\, x)}{\omega^2\, x} - \frac{9\,\cos x\, J_6(\omega\, x)}{\omega^3\, x} - \frac{\sin x\, J_6(\omega\, x)}{\omega^3} + \frac{24\,\sin x\, J_6(\omega\, x)}{\omega^3\, x^2}$$

The value of $P(2,1000) - P(1,1000)$ is -0.0000144073, and is computed in a negligible time. The integral can be computed by quadrature, for example by `quad_qag`, easily called from Maxima. The timing for the quadrature depends on the setting of methods and error tolerances, but the program takes about a second to find an answer of -0.0000144072. They won't always be this close in value, especially for smaller ωx, but for fixed x, increasing ω becomes more accurate. This is apparent from viewing the (truncation) error from discarding the remainder, which can also be computed by our computer algebra system. In the example above, the remainder expression is

$$-\int \frac{\left(\left(87\, x - x^3\right)\,\cos x + \left(-192 + 15\, x^2\right)\,\sin x\right)\; J_6(\omega\, x)}{\omega^3\, x^3} dx.$$

This can be approximated by the technique we've been discussing.

8 Appendix 2: Maxima program for integrating $x^m J_n(ax)$

(Note: the notation in Maxima for $J_n(x)$ is `bessel_j(n,x)`,

```
r[m,n]:= /* integral of x^m*J_n[a*x], Maxima program. */

  if (n=0) then s0[m] else
  if (n=1) then s1[m] else
  if (m=0) then 'integrate(J_n[a*x),x) else
   2*(n-1)/a*r[m-1,n-1]- r[m,n-2]$

/* s0[m]:= integral of x^m*J_0[a*x] */

s0[1]: (x *J[1](a*x))/a$
s0[0]:  x/2 * (%pi*J[1](a*x)*H[0](a*x) + J[0](a*x)*(2-%pi*H[1](a*x)))$
s0[n]:=(x^n* J[1](a*x)/a +  (n-1)*x^(n-1)* J[0](a*x)/a^2 - (n-1)^2 *s0[n-2])/a^2$

/* s1[m]:=  integral of x^m*J_1[a*x] */

s1[0]: (1-J[0](a*x))
/a$
s1[1]:  (%pi*x*J[1](a*x)* H[0](a*x) - J[0](a*x)* H[1](a*x))/(2*a)$
s1[n]:= -x^n/a*J[0](a*x)+n/a*s0[n-1]$
```

References

[1] S.M. Chase and L.D. Fosdick, "An algorithm for Filon Quadrature," *CACM 12 no 8* August 1969 453–457.

[2] Joris Van Deun and Ronald Cools, "Algorithm 858: Computing Infinite Range Integrals of an Arbitrary Product of Bessel Functions, *ACM Trans. Math. Softw. 32*, no 4, Dec. 2006, 580–596.

[3] G.A. Evans, An alternative method for irregular oscillatory integrals over a finite range, *Internat. J. Comput. Math. 53* (1994) 185-193.

[4] R.J. Fateman, "Computer Algebra and Numerical Integration," Proc. SYMSAC'81 (ISSAC), August, 1981, 228–232.

[5] R. Fateman, "Revisiting numeric/symbolic indefinite integration of rational functions, and extensions", `http://www.cs.berkeley.edu/~fateman/papers/integ.pdf` or `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.8514`

[6] L.N.G. Filon, "On a quadrature formula for trigonometric integrals," *Proc. Roy Soc. Edinburgh 49* 1928-29 38.

[7] K.O. Geddes, "Numerical Integration in a Symbolic Context" Proc. SYMSAC-86, (ISSAC) July, 1986, 185–191.

[8] Arieh Iserles and Syvert P Norsett. "Efficient quadrature of highly oscillatory integrals using derivatives," *Proc. R. Soc. A 8* May 2005 vol. 461 no. 2057 1383—1399 .

[9] A. Iserles and S. P. Norsett, "On Quadrature Methods for Highly Oscillatory Integrals and Their Implementation" *BIT Numerical Mathematics* Volume 44, Number 4, 755—772, DOI: 10.1007/s10543-004-5243-3

[10] A. Iserles, S.P. Nrsett, and S. Olver, "Highly oscillatory quadrature: The story so far", Proceedings of ENUMATH, Santiago de Compostela (2005) (A. Bermudez de Castro et al., eds.), Springer-Verlag, Berlin, 2006, 97-118. MR 2303638 (2008j:65029), `http://www.cs.ox.ac.uk/people/sheehan.olver/papers/storysofar.pdf`

[11] David Levin. "Procedures for computing one- and two-dimensional integrals of functions with rapid irregular oscillations," *Math. Comput. 38 (1982)* 531—538.

[12] David Levin, "Fast integration of rapidly oscillatory functions," *J. Computational and Applied Math. 67 (1996)*, 95—101.

[13] Andrew Moylan, `http://sites.google.com/site/andrewjmoylan/levinintegrate`. Also see arxiv `0710.3140v1.pdf`.

[14] Newton Institute Seminars on Highly Oscillatory Problems, `http://www.newton.ac.uk/webseminars/pg+ws/2007/hop/`

[15] Sheehan Olver, "Numerical approximation of highly oscillatory integrals", Dept of Appl. Math and Theoretical PHys, Cambridge,CB30WA, UK.(34 p) http://www.cs.ox.ac.uk/people/sheehan.olver/papers/smithknightessay.pdf

[16] Werner Rosenheinrich, "Tables of Some Indefinite Integrals of Bessel Functions Oct. 18,2011 (109 pages) `http://www.fh-jena.de/~rsh/Forschung/Stoer/besint.pdf`.

[17] Lloyd N. Trefethen, "Is Gauss Quadrature Better than Clenshaw-Curtis?" *SIAM Review 50* no 1, (2008) pp 67-87.

[18] L. N. Trefethen and others, "Chebfun Version 4.0, The Chebfun Development Team," 2011, http://www.maths.ox.ac.uk/chebfun/.

[19] G. N. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, New York. 1966.

[20] Shuhuang Xiang, Weihua Gui, Pinghua Mo, “Numerical quadrature for Bessel transformations,” *Appl. Numer. Math. 58* (2008) 1247—1261.