Numerical Integration in a Symbolic Context [1]

K.O. Geddes
Department of Computer Science

Research Report CS-86-47
October 1986

Numerical Integration in a Symbolic Context [1]

K.O. Geddes

Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

ABSTRACT

Techniques for numerical integration within a symbolic computation environment are discussed. The goal is to develop a fully automated numerical integration code that handles infinite intervals of integration and that handles various types of integrand singularities. Such a code should also be able to compute to arbitrarily high precision. For the case of an analytic integrand on a finite interval, a Clenshaw-Curtis quadrature routine is used. A concept of general (non-Taylor) series expansions forms the basis of techniques for identifying transformations that may yield an analytic integrand. For the case when no transformation is successful, the general series expansion is used to represent the integrand and it is directly integrated to move beyond the singular point. The latter technique relies on a powerful symbolic integrator that can express integrals in terms of special functions.

[1] This research work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant A8967.

1. Introduction

The problem of computing a numerical value for a definite integral has been well studied by numerical analysts. There are several widely used numerical codes for this task, such as those in the IMSL and NAG libraries[IMS79, Gro82]. See [Don75] for an extensive bibliography on numerical integration techniques. The purpose of this paper is to examine the increased generality that can be achieved by a numerical integration code that exploits the power of symbolic computation. Numerical integration codes typically require the interval of integration to be finite and the integrand to be finite at every point of the interval. (For a discussion of numerical techniques for handling singularities see[Don75a]). A symbolic computation environment is a natural environment in which to treat singularities. By using variable transformations and other techniques, it is possible to automatically handle the numerical integration problem in the presence of various types of singularities. Moreover, integrals can be computed to arbitrarily high precision.

A symbolic computation system typically includes a powerful facility for the symbolic indefinite integration of elementary functions, ultimately based on the Risch integration algorithm[Ris69]. Also, many definite integrals (not otherwise handled by indefinite integration) can be expressed symbolically using the techniques discussed by Wang[Wan71]. Nonetheless, many integration problems of practical interest fail to be solved by these techniques. A symbolic computation system should include a facility to "evaluate in floating-point mode" those definite integrals left unevaluated by the symbolic techniques. This paper is a report on such a facility being developed for the Maple system[Cha83, Cha85].

In[Fat81], Fateman discusses the problem of symbolic/numeric integration. He indicates several approaches that are offered by a symbolic computation environment. However, it is left to the user to try various techniques much as in the traditional mode, except that the power of a symbolic "calculator" is available. In contrast, the approach taken here is to present a general algorithmic scheme by which a wide class of numerical integration problems can be solved automatically.

In section 2, the control procedure for numerical integration is presented, and there is a brief discussion of the method currently being used for the case of an analytic integrand on a finite interval. Section 3 outlines the concept of "general series expansions" which forms the basis for handling singularities. A method for identifying variable transformations to transform the problem into one with an analytic integrand is presented in section 4. If transformation to an analytic integrand fails, the method discussed in section 5 is to integrate the "general series" approximation to move beyond the singular point. Successful integration of such series relies on the power of the symbolic integrator, typically including integrals expressed in terms of special functions such as the error function, exponential integral, sine integral, etc.

2. The Case of an Analytic Integrand on a Finite Interval

Underlying the techniques presented in this paper, we need an efficient routine for numerical integration in the "simple" case where the interval of integration is finite and the integrand has no singularities on the interval. Most traditional numerical integration methods could serve for this purpose, but the present investigation is being carried out in the following context. The automatic integration code is being developed within a symbolic computation system (Maple) so that powerful algebraic manipulations can be applied to handle various types of singularities. The floating-point arithmetic currently supplied in Maple does not exploit hardware floating-point facilities, but rather it is a software-coded arbitrary-precision floating-point facility. As a consequence, floating-point operations are relatively slow. It is therefore highly desirable to use a numerical integration method which minimizes the number of function evaluations required to achieve a specified accuracy, and the method must also extend gracefully to arbitrarily high precision.

Based on these considerations, the Clenshaw-Curtis quadrature method as implemented by Gentleman[Gen72,Ged79] was found to be a very good choice. This method is at its best when the integrand is analytic in a "sufficient" region of the complex plane surrounding the interval of integration. Using the techniques presented in this paper, it is possible to ensure analyticity of the integrand which gets passed to the numerical method. Even if the sub-problem of integrating an analytic integrand on a finite interval were coded in an environment using hardware floating-point arithmetic, there is evidence in the literature to suggest that the Clenshaw-Curtis quadrature method remains very competitive with other known methods.

Clenshaw-Curtis quadrature has another significant advantage: it yields an approximation to the indefinite integral in the form of a Chebyshev series expansion. This property can be exploited conveniently within a symbolic computation environment.

In Figure 1, the Maple code for the "control routine" for the numerical integration package is presented. This code tests for infinite limits of integration, and for the presence of end-point singularities using the **taylor** command, and dispatches accordingly. Ultimately, it invokes a procedure called **ccquad** which is a Maple implementation of Gentleman's Clenshaw-Curtis quadrature code.

Prior to the invocation of the quadrature code, an evaluation procedure for the integrand is created via the Maple procedure **makeproc**. The purpose of this code is to generate a procedure which accurately evaluates the integrand at all arguments in the interval, taking care to handle removable singularites and points where floating-point cancellation would lead to loss of accuracy. Space does not permit elaboration of this procedure, but the following example will serve to illustrate. Suppose that the integrand is the expression:

$$f := \frac{x - \sin(x)}{x^3}.$$

A straightforward procedure to evaluate this expression would have two types of difficulties. First, evaluating f at $x = 0$ leads to division by zero due to the removable singularity at zero. If this problem is handled by returning the limiting value at $x = 0$, a second problem is the fact that for argument values near zero, cancellation in floating-point arithmetic will lead to inaccurate evaluation. For example, direct evaluation of f at $x = .00321$ in a 7-decimal-digit floating-point arithmetic yields the result .1813995 compared to the correct result $f(.00321) = .16666658 \cdots$. The procedure **makeproc** analyzes the integrand for removable singularities and for points where loss of accuracy may occur, and creates a procedure which uses a Taylor series expansion near points of difficulty. For the expression *f* above with the floating-point precision set at `Digits := 7`, the procedure created by the invocation `fproc := makeproc(f,x,-1,1,10^(-Digits))` is:

```
fproc := proc (t)
    if abs(t) < 1.1561940 then
    subs(x = t, .1666667+(-.008333333+
        (.0001984127-.000002755732*x^2)
                *x^2)*x^2)
    else
    subs(x = t, (x-sin(x))/x^3)
    fi
end .
```

With this procedure for evaluating the integrand *f*, **fproc**(.00321) yields the value .1666666 in 7-digit arithmetic, which is fully accurate.

Notes on the code in Figure 1:

(1) **evalf** is the Maple function for "evaluation in floating-point mode".

(2) **traperror** is the Maple function that provides an error-trapping facility. The type-checking of the form `type(pleft, taylor)` will return *false* if the result from **taylor** was an error message or if the result from **taylor** was a non-Taylor series.

(3) If infinite limits of integration arise, the problem is dispatched to a routine named **improper**. The details of this algorithm for handling improper integrals are not presented. It simply uses a $t = 1/t$ transformation to transform an infinite limit to zero, first breaking up the interval if the interval includes zero. The transformed problems are then passed back to **control**.

(4) When a singularity is encountered, a transformation or interval splitting is performed if necessary to yield an integration problem in which the singularity appears at the left end-point of the interval. This simplifies the subsequent handling of the singularity by the routine **singular** (see the following sections).

```
control :=  proc (f, x, a, b, eps)
local left,right,h,pleft,pright,fproc,err,
                      nofun,c,r,tol;
if has({a,b}, infinity) then
    RETURN( improper(f,x,a,b,eps) )
fi;
left := evalf(a);  right := evalf(b);
if not type(left,numeric)
    or not type(right,numeric) then
        RETURN( FAIL )
elif left > right then
    RETURN( control(-f,x,right,left,eps) )
fi;

h := right-left;
pleft := traperror( taylor(f, x=left) );
pright := traperror( taylor(f, x=right) );

# If not "pure Taylor series" then there is
#   an end-point singularity.
if not type(pleft,taylor)
  and not type(pright,taylor) then
    # Split interval in two.
    RETURN(control(f,x,left,left+h/2,eps/2)
      + control(f,x,left+h/2,right,eps/2))
elif not type(pright,taylor) then
    # Transform singularity to the left.
    RETURN( control(subs(x=2*right-x,f),x,
                right,2*right-left,eps) )
elif not type(pleft,taylor) then
    # Deal with singularity at the left.
    RETURN( singular(f,x,left,right,eps) )
else
    # Attempt numerical integration.
    fproc := traperror(
                makeproc(f,x,a,b,eps) );
    if fproc = lasterror then
        ERROR(`non-removable singularity`)
    fi;
    r := ccquad(fproc,left,right,eps,487,
                        err,nofun,c);
    if r = FAIL then
        ERROR(`singularity in or near
                interval of integration`)
    fi;

    # Calculate relative error tolerance;
```

```
    #   if too small, use absolute error.
    tol := max(eps*abs(r), 0.001*eps);
    if err <= tol then RETURN( r ) else
        ERROR(`failed to converge`)
    fi
fi
end:
```

Figure 1. Maple code for the control routine.

3. General Series Approximations

The main tool for handling integrand singularities is a facility for generating very general (non-Taylor) series expansions of expressions. The function for this purpose has the same argument syntax as the **taylor** function:

$$\textbf{series}(expr,\ var{=}value,\ ord).$$

If there exists a Laurent series expansion of *expr* about the point $var = value$, with finite principal part, then this will be the result returned. More generally, the series returned might involve algebraic singularities, logarithmic singularities, or exponential singularities such as $\exp(-1/x)$. (Series involving the latter singularity must be treated as one-sided series expansions for $x > 0$.)

As long as the non-polynomial functions introduced into the series expansion can be bounded by a power of $var-value$, it is possible to generate a series with a correct order term $O((var-value)^p)$ for some p. For example, for all $x > 0$ we have:

$$\ln(x) < x$$

$$\exp(-\frac{1}{x}) < 1$$

$$\exp(-\frac{1}{x^2}) < 1.$$

Some examples of general series follow.

```
series(sqrt(sin(x)), x=0, 4);
```

$$x^{1/2} - \frac{1}{12}\,x^{5/2} + O(x^{9/2})$$

```
series(ln(1-cos(2*x)), x=0, 8);
```

$$\ln(2) + 2\ln(x) - \frac{1}{3}x^2 - \frac{1}{90}x^4 - \frac{2}{2835}x^6 + O(x^8)$$

```
series(1/(1-x*exp(-cos(x)/x)), x=0, 4);
```

$$1 + \exp(-\frac{1}{x})\,x + (\frac{1}{2}\exp(-\frac{1}{x}) + \exp(-\frac{1}{x})^2)\,x^2 +$$
$$(\frac{1}{8}\exp(-\frac{1}{x}) + \exp(-\frac{1}{x})^2 + \exp(-\frac{1}{x})^3)\,x^3 + O(x^4)$$

Consider briefly the case of exponential singularities. Let the exponential subexpression be $\exp(g(x))$ and let the point of expansion be $x = 0$. If $g(x)$ has a Taylor series expansion then $\exp(g(x))$ has a Taylor series expansion. Otherwise, if $g(x)$ has a Laurent series expansion with finite principal part then the following technique is used to generate a general series. If the Laurent expansion is

$$g(x) = c_{-k}\,x^{-k} + \cdots + c_{-1}\,x^{-1} + c_0 + c_1\,x + \cdots$$

then

$$\exp(g(x)) = \exp(-\frac{1}{x^k})^{-c_{-k}} \cdots \exp(-\frac{1}{x})^{-c_{-1}}$$
$$\exp(c_0 + c_1\,x + \cdots).$$

The last exponential term here has a regular Taylor series expansion. To complete the general series expansion for the original expression containing $\exp(g(x))$ as a subexpression, an ordinary Taylor series expansion is performed after substituting each non-regular exponential term

$$\exp(-\frac{1}{x^i})$$

by a unique symbol independent of x.

4. Transforming to an Analytic Integrand

If an integrand is found to be non-analytic at a point of the interval of integration, the first technique tried is to look for a transformation that will remove the singularity. Three types of transformations are attempted:

(i) subtracting off the singularity;

(ii) an algebraic transformation of variables;

(iii) a non-algebraic transformation of variables.

In all three cases, the fundamental tool is the general series expansion discussed in the preceding section. As was seen in section 2, we may assume that the singularity is at the left end-point of the interval. This property is important because of the "one-sided" validity of some of the general series expansions being generated.

Let f be the integrand, x be the variable of integration, x_0 be the singular point, and s be the general series expansion of f about x_0.

For case (i), the method used is to test each term in the expansion s to determine which terms are regular (i.e., have a Taylor series expansion at x_0). If the number of non-regular terms is less than half the number of terms in s, then make the conjecture that the expression $f - q$ might be analytic at x_0, where q denotes the sum of the non-regular terms. Test this conjecture, and if it is true then the integration of $f - q$ can proceed normally. In some cases, the non-regular part q will be integrable by the general integrator; otherwise, it will passed on to the general techniques discussed in subsequent sections.

Example 1.

Consider the problem of integrating over the interval [0, 1] the function

$$f = \ln(1 - \cos(2x)).$$

The general series expansion of f at $x = 0$ is of the form:

$$\ln(2) + 2\ln(x) - \frac{1}{3}x^2 - \frac{1}{90}x^4 - \frac{2}{2835}x^6 + O(x^8).$$

The non-regular part is

$$q = 2\ln(x).$$

The new expression

$$f - q = \ln(1 - \cos(2x)) - 2\ln(x)$$

is analytic on the interval [0, 1]. Thus it can be integrated by the numerical method, yielding the value 0.5797067686 (computing to 10 digits of accuracy). Integrating q is easy because it has the indefinite integral

$$2\ x\ \ln(x) - 2\ x;$$

its definite integral is therefore −2. Finally, summing the two values, we obtain the value −1.420293231 for the definite integral of f. □

Case (ii) outlined above is the case of an algebraic transformation of variables. This method comes into effect whenever there are fractional powers of $x - x_0$ appearing in the series expansion s, whether or not there are other non-regular functions appearing in the expansion. The idea is to transform away algebraic singularities, and if other singularities remain they will be handled on a second pass. The specific method used is to compute the least common multiple n of all denominators of the fractional powers, and then to apply the change of variables:

$$t = (x - x_0)^{1/n}$$

Example 2.

Consider the problem of integrating over the interval [0, 2] the function

$$f = \mathrm{sqrt}(\sin(x)).$$

The general series expansion of f at $x = 0$ is of the form:

$$x^{1/2} - \frac{1}{12}\,x^{5/2} + O(x^{9/2}).$$

Applying the change of variables $t = x^{1/2}$ yields the new integrand

$$2\ t\ \mathrm{sqrt}(\sin(t^2))$$

to be integrated over the interval [0, sqrt(2)]. This new integrand is analytic and therefore it can be integrated by the numerical method, yielding the value 1.620723408 (computing to 10 digits of accuracy). □

Case (iii) outlined above is the case of a non-algebraic transformation of variables. When this method comes into effect, the general series expansion s of the integrand f at $x = x_0$ contains no fractional powers but it contains other non-regular functions. Once again, the method is based on looking at the terms in the expansion s. For each non-regular function $u(x)$ appearing in s, attempt to find a variable transformation $t = u(x)$ that yields an analytic integrand. There are two steps that must succeed before the variable transformation is accepted:

(a) the **solve** function must succeed in solving $t = u(x)$ for x, yielding the substitution for x: $x = w(t)$;

(b) the new integrand $g = f(w(t))\, w'(t)$ must be analytic at $t{=}u(x_0)$.

Of course, the value $u(x_0)$ is computed via the **limit** function, if necessary. Also, this value may be infinite in which case a $t = 1/t$ substitution is used. If these two steps succeed then the integration continues using the transformed integrand.

The algorithm for these transformations is presented in Figures 2a and 2b. It is invoked by the routine **singular** presented in the next section.

```
# The expression f in the variable x has a singularity at x = a.
# The general series expansion of f at x=a is ser.

transform := proc (f, x, a, ser)
local inds, q, s, powinds, base, n, t;

inds := map(proc(e,x) if has(e,x) and e<>x then e fi end,
                                                indets(ser), x);
# Check if the singularity can be "subtracted off".
if type(ser, `+`) then
    q := map(proc(e,x) if not type(e,polynom,x) then e fi end,
                                                [op(ser)], x);
    if nops(q) < nops(ser)/2 then
        q := convert(q, `+`);
        s := traperror( taylor(f-q, x=a) );
        if type(s, taylor) then RETURN(`subtract off`,q) fi
    fi
fi;
# Separate from inds the fractional powers of x-a.
powinds := map(proc(e) if type(e,`^`) and type(op(2,e),fraction)
                                                then e fi end, inds);
inds := inds minus powinds;
if nops(powinds) > 0 then
    # Apply an algebraic transformation.
    base := op(1, powinds[1]);
    n := ilcm( op(map(proc(e) op(2,op(2,e)) end, powinds)) );
    s := solve({t=base^(1/n)}, {x});
    if s <> NULL and nops([s]) = 1 then
        RETURN( subs(t=x, op(2,s[1])), base^(1/n) )
    fi
fi;
# Search for a non-algebraic transformation.
findtransform(f, x, a, inds)
end:
```

Figure 2a. Algorithm for finding transformations.

```
findtransform := proc (f, x, x0, inds)
local i, t, inv_sbst, sbst, newf, t0, s;

for i to nops(inds) do
    inv_sbst := t=inds[i];
    sbst := solve({inv_sbst}, {x});
    if sbst <> NULL and nops([sbst]) = 1 then
        newf := subs(sbst, f) * diff(op(2,sbst[1]), t);
        t0 := traperror( limit(inds[i], x = x0) );
        if type(evalf(t0), numeric) then
            s := traperror( taylor(newf, t=t0) );
            if type(s, taylor) then
                RETURN( subs(t=x, op(2,sbst[1])), inds[i] ) fi
        elif has(t0, infinity) then
            s := traperror( taylor(subs(t=1/t, newf), t=0) );
            if type(s, taylor) then
                RETURN( subs(t=x, op(2,sbst[1])), inds[i] ) fi
        fi
    fi
od;
FAIL
end:
```

Figure 2b. Algorithm for finding non-algebraic transformations.

5. The Case of a Singular Integrand

The control routine presented in Figure 1 of section 2 dispatches the problem to **singular** if a singularity is encountered. This singularity-handling algorithm is presented in Figure 3.

The singularity is known to be at the left endpoint a. The general series expansion of the integrand at the singular point is generated and the algorithm of section 4 is invoked to search for a transformation. The transformation may be the `subtract off` type or it may be a substitution of variables. In either case, the indicated transformation is performed. The case where no transformation is found (case FAIL) is discussed below.

```
singular := proc (f, x, a, b, eps)
local ord, s, t, newf, newa, newb;

# Compute general series expansion of f.
ord := trunc( evalf(-ln(eps)) );
s := series(f, x=a, ord);
if type(s, laurent) then
    ERROR(`integrand has a pole at`, a)
fi;

# Search for a transformation of f.
t := transform(f,x,a,convert(s,polynom));

if t = FAIL then
    # Integrate the general series.
    intseries(f, x, a, b, eps, s)
elif t[1] = `subtract off` then
    control(f - t[2], x, a, b, eps/2)
    +  control(t[2], x, a, b, eps/2)
else
    newf := subs(x=t[1],f) * diff(t[1],x);
    newa := limit(t[2], x=a);
    newb := limit(t[2], x=b);
    control(newf, x, newa, newb, eps)
fi
end:
```

Figure 3. Singularity-handling routine.

5.1. Direct integration of the general series

Although the method of transformations is quite powerful, it will not always succeed in transforming to an analytic integrand. A method that has been found to be very successful is to use the general series approximation as a representation of the integrand, and to directly integrate this general series over its interval of accuracy. The remainder of the interval can then be handled by the numerical integration method. It is crucial that we move as far as possible away from the singularity before resuming, because the efficiency of the numerical integration method is affected by the nearness of singularities. The routine **intseries** (not presented here) determines an interval over which integration of the general series will be accurate and then integrates the series. It then invokes the numerical integration method for the remaining part of the interval.

The success of this technique relies on a powerful symbolic integrator to handle many of the singular functions that may arise. For general series involving $\ln(x)$, the indefinite integral of $\ln(x)\,x^k$ is an elementary function. For general series involving $\exp(-1/x)$, the indefinite integral of $\exp(-1/x)\,x^k$ can be expressed in terms of the exponential integral $\mathrm{Ei}(x)$. Similarly, series involving the singular function $\exp(-1/x^2)$ may lead to both the exponential integral and the error function $\mathrm{erf}(x)$.

Example 3.

Consider the problem of integrating over the interval $[0, \infty]$ the function

$$f = \frac{\exp(v - \frac{v^2}{2})}{1 + \frac{1}{2}\exp(v)}.$$

First, the interval would be split into $[0, 1]$ and $[1, \infty]$. For the finite interval, the numerical integration method is applied directly and it yields the value 0.7580564829 (computing to 10 digits of accuracy). For the infinite interval, the change of variables $v = 1/x$ transforms the problem into integrating over the interval $[0, 1]$ the new integrand:

$$g = \frac{\exp(\frac{1}{x} - \frac{1}{2\,x^2})}{(1 + \frac{1}{2}\exp(\frac{1}{x}))\,x^2}.$$

The first few terms of the general series expansion of g at $x = 0$ are:

$$\begin{aligned} s = {} & 2\exp(-\frac{1}{x^2})^{1/2}/x^2 - 4\exp(-\frac{1}{x^2})^{1/2}\exp(-\frac{1}{x})/x^2 \\ & + 8\exp(-\frac{1}{x^2})^{1/2}\exp(-\frac{1}{x})^2/x^2 \\ & - 16\exp(-\frac{1}{x^2})^{1/2}\exp(-\frac{1}{x})^3/x^2 + \cdots \end{aligned}$$

Maple's symbolic integrator determines that the indefinite integral of the first term of s is

$$- \mathrm{sqrt}(2\pi)\ \mathrm{erf}(\frac{1}{\mathrm{sqrt}(2)\ x});$$

the indefinite integral of the second term of s is

$$2\ \mathrm{sqrt}(2\pi)\ \exp(\frac{1}{2})\ \mathrm{erf}(\frac{1}{\mathrm{sqrt}(2)\ x} + \frac{1}{\mathrm{sqrt}(2)});$$

and similarly the integral of each term is computed. If one computes the definite integral over [0, 0.25], say, of the successive terms of s, one finds that the successive values are:

.000158776061, −.000004738616, .000000146186,

−.000000004620, .000000000149, −.000000000005

Clearly, the series representation is rapidly converging on this interval. Summing up these values yields the value 0.000154179155 for the integral of g over [0, 0.25]. For the remaining interval [0.25, 1], the numerical integration method is applied to integrate g, yielding the value 0.547306237063 . Summing these two values together with the previous result for the finite interval, we have obtained the desired integral of f over $[0, \infty]$ to be 1.305516899 (to 10 digits of accuracy). □

The technique of integrating a general series expansion appears to be a viable method for handling a singular integrand, given a symbolic integrator that handles a large class of special functions. More study should be carried out on methods to determine the optimal interval over which the general series is sufficiently accurate. For example, suppose we postulate (based on the order term for the series) that the general series expansion will be accurate on an interval $[a, a + r]$, where a is the point of singularity. Then the actual accuracy on the second half of this interval, $[a + r/2, a + r]$, could be tested by using an ordinary Taylor series expansion about the point $a + r$. It will be fruitful to determine an optimal choice for r because the further we can move away from the singularity, the more efficient will be the numerical integration method on the remaining interval of integration.

6. Concluding Remarks

We have presented some techniques that can form the basis for a fully automated numerical integration code in a symbolic computation system. A concept of general series expansions (non-Taylor, non-Laurent series) was used as the basis of algorithms to identify possible transformations that might remove the singularity. For the case where no transformation was successful, a method was outlined for integrating the general series expansion beyond the singular point, relying on the power of the system's symbolic integrator.

There are several areas of further study indicated by this investigation. For the case of an analytic integrand on a finite interval, various other numerical integration methods could be considered. An interface from the symbolic system to Fortran codes might be the best solution when the interval is finite, there are no singularities present, and ordinary precision is acceptable.

For the case of a singular integrand when no transformation is found to remove the singularity, further study of error estimation for general series expansions is warranted. As was noted in section 5, it is desirable to integrate the general series expansion as far beyond the singular point as possible, because the nearness of the singularity affects the efficiency of the numerical integration method which is used to complete the integration on the remaining part of the interval.

Finally, it is necessary to develop a scheme to automatically handle cases where the singularity cannot be removed and the symbolic integrator is unable to express the integral of the general series expansion. One possible technique for the latter case would be to transform back to an infinite range and to use some method of estimating the "tail" of the integrand that can be ignored for the accuracy that is desired. Symbolic manipulation techniques should be useful for this purpose. See [Fat81] for some ideas on this case.

7. References

Cha83.
B.W. Char, K.O. Geddes, W.M. Gentleman, and G.H. Gonnet, The design of Maple: A compact, portable, and powerful computer algebra system, *Proceedings of Eurocal '83*, pp. 101-115 (April, 1983). Springer-Verlag Lecture Notes in Computer Science no. 162.

Cha85.
B.W. Char, K.O. Geddes, G.H. Gonnet, and S.M. Watt, *Maple User's Guide,* WATCOM Publications Ltd., Waterloo, Ontario, Canada (1985). Incorporates the Maple tutorial "First Leaves", and "Maple Reference Manual, 4th edition".

Don75.
Elise de Doncker and Robert Piessens, A bibliography on automatic integration, *Report TW 26*, Katholieke Universiteit Leuven, (August 1975).

Don75a.
Elise de Doncker and Robert Piessens, Automatic computation of integrals with singular integrand, over a finite or an infinite range, *Report TW 22*, Katholieke Universiteit Leuven, (June 1975).

Fat81.
R.J. Fateman, Computer algebra and numerical integration, pp. 228-232 in *Proc. 1981 ACM Symposium on Symbolic and Algebraic Computation*, ed. P.S. Wang, ACM (August 1981).

Ged79.
K.O. Geddes, Remark on Algorithm 424, *ACM Trans. Math Software* **5**(2) p. 240 (June 1979).

Gen72.
W.M. Gentleman, Algorithm 424: Clenshaw-Curtis Quadrature, *Comm. ACM* **15**(5) pp. 353-355 (May 1972).

Gro82.
Numerical Algorithms Group, *Subroutine Library,* NAG USA Inc, 1131 Warren Avenue, Downers Grove, Ill. 60515 (1982).

IMS79.
IMSL, Inc., *IMSL Library, Edition 7,* International Mathematical and Statistical Libraries, Inc., Houston, Texas (1979).

Ris69.
R.H. Risch, The problem of integration in finite terms, *Trans. Am. Math. Soc.* **139** pp. 167-189 (1969).

Wan71.
P.S. Wang, *Evaluation of Definite Integrals by Symbolic Manipulation,* M.I.T., Cambridge, Mass. (1971). (Ph.D thesis).