J. Symbolic Computation (1989) 7, 183–187

Symbolic Evaluation of Integrals Occurring in Accelerator Orbit Theory

B. AUTIN and J. BENGTSSON

CERN, Geneva, Switzerland

(*Received* 4 *February* 1987)

Definite integrals which appear in the perturbation theory of a particle's transverse oscillations and chromatic aberrations inside an accelerator are evaluated by symbolic computation. The symbolic program and the automatic FORTRAN coding of the generated functions are described. The results are checked by comparison with those obtained by direct numerical integration. It turns out that, once having established the FORTRAN function subprograms symbolically, their use for different parameters requires much less time than direct numerical integration.

1. Statement of the Problem

The transverse motion along a reference curve for a particle in an accelerator can be described by the two second-order differential equations (Courant & Snyder, 1958):

$$\frac{d^2x}{ds^2} + K_x(s)x = f_x(x, y),$$

$$\frac{d^2y}{ds^2} + K_y(s)y = f_y(x, y), \tag{1}$$

where s is the distance along the reference curve and $x = x(s)$ and $y = y(s)$ are the horizontal and vertical deviations from the reference curve. The restoring forces K_x and K_y are piecewise constant functions alternating between positive, negative and zero values with $K_x = -K_y$, and $f_x(x, y, s)$ and $f_y(x, y, s)$ are polynomials in x and y for a given interval of s. One way of solving these equations consists of applying a method of successive approximations, starting from the solutions of the (uncoupled) homogeneous equations

$$x = \sqrt{2J_x\beta_x(s)}\cos(\mu_x(s) + \varphi_x),$$

$$\mu_x(s) = \int_{s_0}^{s} ds'/\beta(s'), \tag{2}$$

where J_x and φ_x are constants given by the initial conditions, $\beta_x(s)$ the amplitude function and $\mu_x(s)$ the phase advance. We have a similar solution for the vertical plane.

The solutions obtained in this way can be composed by integrals of the following type (Autin & Bengtsson, 1988; Autin, 1987):

$$I_{s,c} = \int_0^L \beta_x^k(s)\beta_y^l(s)D^j(s) \begin{smallmatrix}\sin\\ \cos\end{smallmatrix} [m(\mu_x(s) + \mu_{x_1}) + n(\mu_y(s) + \mu_{y_1})]\, ds, \tag{3}$$

0747–7171/89/020183 + 05 \$03.00/0

where k and l are non-negative integers or half-integers, m and n are integers, $j = 0, 1, 2$, and μ_{x_1} and μ_{y_1} are constant. For $j = 0$ these integrals are called 'betatron' integrals.

It is clear that the solutions of (2) depend on the value of K_x and K_y. In detail we have:

1.1. $K_x = K_y = K = 0$

$$\sqrt{\beta(s)}\cos\mu(s) = \sqrt{\beta_1} - (\alpha_1/\sqrt{\beta_1})s, \tag{4a}$$

$$\sqrt{\beta(s)}\sin\mu(s) = s/\sqrt{\beta_1}, \tag{4b}$$

$$\beta(s) = \beta_1 - 2\alpha_1 s + \gamma_1 s^2, \tag{4c}$$

$$D(s) = D_1 + sD'_1, \tag{4d}$$

where α_1, β_1, γ_1, D_1 and D'_1 are parameters.

1.2. $K_x = -K_y = K > 0$

$$\sqrt{\beta_x(s)}\cos\mu_x(s) = \sqrt{\beta_{x_1}}\cos(\sqrt{K}s) - \frac{\alpha_{x_1}}{\sqrt{K\beta_{x_1}}}\sin(\sqrt{K}s), \tag{5a}$$

$$\sqrt{\beta_x(s)}\sin\mu_x(s) = \frac{1}{\sqrt{K\beta_{x_1}}}\sin(\sqrt{K}s), \tag{5b}$$

$$\beta_x(s) = \frac{1}{2}\left(\beta_{x_1} + \frac{\gamma_{x_1}}{K}\right) + \frac{1}{2}\left(\beta_{x_1} - \frac{\gamma_{x_1}}{K}\right)\cos(2\sqrt{K}s) - \frac{\alpha_{x_1}}{\sqrt{K}}\sin(2\sqrt{K}s), \tag{5c}$$

$$\sqrt{\beta_y(s)}\cos\mu_y(s) = \sqrt{\beta_{y_1}}\cosh(\sqrt{K}s) - \frac{\alpha_{y_1}}{\sqrt{K\beta_{y_1}}}\sinh(\sqrt{K}s), \tag{5d}$$

$$\sqrt{\beta_y(s)}\sin\mu_y(s) = \frac{\sinh(\sqrt{K}s)}{\sqrt{K\beta_{y_1}}}, \tag{5e}$$

$$\beta_y(s) = \frac{1}{2}\left(\beta_{y_1} - \frac{\gamma_{y_1}}{K}\right) + \frac{1}{2}\left(\beta_{y_1} - \frac{\gamma_{y_1}}{K}\right)\cosh(2\sqrt{K}s) - \frac{\alpha_{y_1}}{\sqrt{K}}\sinh(2\sqrt{K}s), \tag{5f}$$

$$D(s) = D_1\cos(\sqrt{K}s) + \frac{D'_1}{\sqrt{K}}\sin(\sqrt{K}s). \tag{5g}$$

1.3. $K_x = -K_y = K < 0$

$$\sqrt{\beta_x(s)}\cos\mu_x(s) = \sqrt{\beta_{x_1}}\cosh(\sqrt{|K|}s) - \frac{\alpha_{x_1}}{\sqrt{|K|\beta_{x_1}}}\sinh(\sqrt{|K|}s), \tag{6a}$$

$$\sqrt{\beta_x(s)}\sin\mu_x(s) = \frac{1}{\sqrt{|K|\beta_{x_1}}}\sinh(\sqrt{|K|}s), \tag{6b}$$

$$\beta_x(s) = \frac{1}{2}\left(\beta_{x_1} - \frac{\gamma_{x_1}}{|K|}\right) + \frac{1}{2}\left(\beta_{x_1} - \frac{\gamma_{x_1}}{|K|}\right)\cosh(2\sqrt{|K|}s) - \frac{\alpha_{x_1}}{\sqrt{|K|}}\sinh(2\sqrt{|K|}s), \tag{6c}$$

$$\sqrt{\beta_y(s)}\cos\mu_y(s) = \sqrt{\beta_{y_1}}\cos(\sqrt{|K|}s) - \frac{\alpha_{y_1}}{\sqrt{|K|\beta_{y_1}}}\sin(\sqrt{|K|}s), \tag{6d}$$

$$\sqrt{\beta_y(s)}\sin\mu_y(s) = \frac{1}{\sqrt{|K|\beta_{y_1}}}\sin(\sqrt{|K|}s), \tag{6e}$$

$$\beta_y(s) = \frac{1}{2}\left(\beta_{y_1} - \frac{\gamma_{y_1}}{|K|}\right) + \frac{1}{2}\left(\beta_{y_1} - \frac{\gamma_{y_1}}{|K|}\right)\cos(2\sqrt{|K|}s) - \frac{\alpha_{y_1}}{\sqrt{|K|}}\sin(2\sqrt{|K|}s), \tag{6f}$$

$$D(s) = D_1\cosh(\sqrt{|K|}s) + \frac{D'_1}{\sqrt{|K|}}\sinh(\sqrt{|K|}s), \tag{6g}$$

where α_{x_1}, β_{x_1}, γ_{x_1}, α_{y_1}, β_{y_1}, γ_{y_1}, D_1, D'_1 and K are constants.

2. Strategy for the Evaluation of $I_{s,c}$

A straightforward evaluation of the integrals (3) using the integration facilities in existing symbolic algebra systems like the INT operator in REDUCE (Hearn, 1983) is not feasible This is essentially due to the appearance of a trigonometric factor with compared arguments in the integrand. The problem may be solved by the following strategy.

We first expand the factor

$$\begin{smallmatrix}\sin\\ \cos\end{smallmatrix}[m(\mu_x(s)+\mu_{x_1})+n(\mu_y(s)+\mu_{y_1})]$$

into a polynomial in $\sin\mu_x(s)$, $\cos\mu_x(s)$, $\sin\mu_y(s)$, $\cos\mu_y(s)$ by using

$$\begin{aligned}\sin(a_1+\alpha_2) &= \sin\alpha_1\cos\alpha_2+\cos\alpha_1\sin\alpha_2,\\ \cos(a_1+\alpha_2) &= \cos\alpha_1\cos\alpha_2-\sin\alpha_1\sin\alpha_2,\end{aligned} \tag{7}$$

and by applying repeatedly, depending on m and n,

$$\begin{aligned}\sin(k\alpha) &= \sin\alpha\cos(k-1)\alpha+\cos\alpha\sin(k-1)\alpha,\\ \cos(k\alpha) &= \cos\alpha\cos(k-1)\alpha-\sin\alpha\sin(k-1)\alpha.\end{aligned} \tag{9}$$

Then we replace the products $\sqrt{\beta_x(s)}\cos\mu_x(s)$, etc. by their representations (4)–(6), respectively. It follows from these formulae that these substitutions also eliminate the factor $\beta_x^k(s)\beta_y^l(s)$ in the case $k = m/2$, $l = n/2$. For the other cases, relations (4c), (5c), (5f) and (6c), (6f) have to be used in addition. By this procedure, the integrand has been transformed into a polynomial in s if $K = 0$, or into a polynomial in trigonometric and hyperbolic functions of s if $K \neq 0$. It is further simplified by linearizing the trigonometric functions:

$$\begin{aligned}\sin^2\alpha &= \tfrac{1}{2}(1-\cos 2\alpha),\\ \cos^2\alpha &= \tfrac{1}{2}(1+\cos 2\alpha),\\ \sin\alpha\cos\alpha &= \tfrac{1}{2}\sin 2\alpha,\\ \sinh^2\alpha &= \tfrac{1}{2}(\cosh 2\alpha-1),\\ \cosh^2\alpha &= \tfrac{1}{2}(\cosh 2\alpha+1),\\ \sinh\alpha\cosh\alpha &= \tfrac{1}{2}\sinh 2\alpha,\end{aligned} \tag{9}$$

and becomes a sum of bilinear terms in trigonometric and hyperbolic functions of s.

3. Structure of the Generated Fortran Code

Our purpose is to elaborate a program which performs the integration symbolically and generates the FORTRAN code for the corresponding integral.

For each integral with given j, k, l, m, n, a FORTRAN function subprogram is generated. The parameters defined in (4)–(6) are listed in a COMMON block. Due to the large number of integrals that have to be evaluated we define five global functions with two parameters like.

DIP(NB, TYPE),
QUD(NB, TYPE),
SXT(NB, TYPE),
OCT(NB, TYPE),
CHR(NB, TYPE),

where NB is an integer referring to a subset of logically connected integrals. TYPE is also an integer taking the values $-1, 0, +1$ depending on $K < 0$, $K = 0$, or $K > 0$.

4. Implementation of the Symbolic Program

The symbolic program has been implemented in REDUCE. It is composed of an integration procedure (INTS) followed by a FORTRAN coding of the integral (FCODE). The integration procedure is called with given values of j, k, l, m, n.

4.1. DESCRIPTION OF THE SYMBOLIC PROGRAM

The symbolic integration (INTS) resumes the logical steps defined in section 2, namely:

(1) $\beta_x(s)$, $\beta_y(s)$, $\mu_x(s)$ and $\mu_y(s)$ are defined.
(2) Rules (7) and (8) are applied.
(3) Relations (4), (5), (6) are defined and applied according to TYPE.
(4) Substitution of $\varphi = \sqrt{Ks}$ if TYPE = 2, 3.
(5) Rules (9) are applied.
(6) Integration by INT and substitution of limits 0, L.

Note that step (4) was introduced in order to overcome certain difficulties with the LET command in REDUCE. The LET rule could not be applied to all occurrences without this substitution.

The FORTRAN code mainly contains WRITE-statements for the definition of the function and of the COMMON-block containing the parameters and for the declaration of the variables. REDUCE automatically divides the expression into multiple FORTRAN statements if the number of lines for one statement exceeds a specified number.

Finally, the main program consists of a series of calls to the two procedures preceded by definitions of the constants in each case. When the program is executed the result is stored in two files containing the intermediate results during the integration on one hand, and the FORTRAN functions on the other hand.

4.2. COMPARISON WITH NUMERICAL INTEGRATION

To check the symbolic calculations we have compared the results deduced from the symbolic calculation with a numerical integration using Romberg's method (Dahlqvist & Björk, 1979).

Table 1. CPU time (ms).

Integral	$K_x = -K_y = K > 0$			$K = 0$		
j, k, l, m, n	Optimized code	Non-optimized code	Numerical integration	Optimized code	Non-optimized code	Numerical integration
0, 3/2, 0, 1, 0	1	3	6	1	2	19
0, 1/2, 1, 1, 0	2	3	94	2	1	15
0, 3/2, 0, 3, 0	1	3	214	1	2	18
0, 1/2, 1, 1, −2	4	9	190	1	3	14
0, 1/2, 1, 1, 2	4	9	198	1	3	15

Table 2. CPU time (ms).

Integral	$K_x = -K_y = K < 0$			$K = 0$		
j, k, l, m, n	Optimized code	Non-optimized code	Numerical integration	Optimized code	Non-optimized code	Numerical integration
0, 2, 0, 2, 0	2	4	181	2	1	30
0, 1, 1, 2, 0	4	6	180	1	2	28
0, 2, 0, 4, 0	2	5	191	1	1	28
0, 1, 1, 2, −2	6	18	178	1	4	24
0, 1, 1, 2, 2	6	18	191	2	5	27
0, 1, 1, 0, 2	4	6	184	1	1	28
0, 0, 2, 0, 2	2	4	181	1	2	28
0, 0, 2, 0, 4	3	6	191	1	2	28

We show in Tables 1 and 2 the CPU time in ms needed to calculate the integral using a symbolically generated FORTRAN function with optimized code, non-optimized code and a numerical integration with an accuracy of twelve decimals. The calculations have been done on a μVAX using double precision.

It follows from these tables that the effort in investing into symbolic computation is of paramount reward when the problem is repetitive; in our case, once the methodology had been fixed for one integral, it was just a matter of routine to produce more than one hundred integrals of similar nature, containing about 5000 lines of FORTRAN code.

References

Autin, B. (1987). Non-linear betatron oscillations. In: *Physics of Particle Accelerators, AIP Conf. Proc. 153*, Vol. 2, pp. 288–347. New York.

Autin, B., Bengtsson, J. (1988). Application of symbolic computation to the search of complicated primitives: the example of the 'betatron' integrals. *Comput. Phys. Comm.* **48**, 181–195.

Courant, E. D., Snyder, H. S. (1958). Theory of alternating gradient synchrotron. *Ann. Phys.* **3**, 1–48.

Dahlqvist, G., Björk, A. (1979). *Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall.

Hearn, A. C. (1983). *REDUCE User's Manual*, Version 3.2. Santa Monica, CA: Rand Corporation.