ARE WE CAREFUL ENOUGH WHEN USING COMPUTER ALGEBRA?

J. M. Aguirregabiria, A. Hernández, and M. Rivas

Department Editor: Denis Donnelly
donnelly@siena.bitnet

Computer-algebra systems are gaining increasing acceptance in the everyday work of physicists and students of physics. There exist affordable programs that run on the cheapest computers, and even the hardware necessary for the most complex and expensive programs has been continuously decreasing in price. Nowadays, computing a rational integral by hand, or with the help of a table, is becoming nearly as rare as computing a logarithm by looking at a table. Furthermore, the widespread availability of computer-algebra programs may help to introduce new topics in graduate and undergraduate courses and teach standard topics in a fresh way.[1]

But, despite the advantage of computer algebra, we cannot avoid expressing some concerns about the way in which students are starting to use computer-algebra systems. When we see students using pocket calculators to compute $\sqrt{9}$ or cos 0, we know that, at least in these cases, they will get the right answers. Unfortunately, computer algebra, especially if it is used without extreme care, may not always give the right answer.

Introductory textbooks on numerical calculus usually discuss to some extent the different types of error that unavoidably arise in numerical computing. One might be tempted to assume that symbolic computing (as opposed to numerical computing) is "exact," whereas errors in numerical calculus occur because of its "approximate" nature. In fact, computer-algebra systems are complex, with room for bugs and subtleties. For obvious but not always the right reasons, manuals of most computer-algebra systems pay little attention to the limitations of the programs. Often left unstated is the fact that some functions or commands may give symbolic answers that either are incorrect or have a limited domain of validity. Moreover, although it is easy to teach algebraic properties to the computer, analysis is often subtle, and the abilities of computer-algebra systems are impressive but always limited and completely unintelligent.

Consequently, we think that obtaining incorrect or misleading results when using these systems may occur more often than when employing numerical codes. Therefore, we need to provide our students with instruction in the use of computer algebra.

The aim of this article is to present ideas and examples (most of them extracted from actual work in physics problems) that could be helpful in computer-algebra training. To discuss the examples, we shall be working with four of the most widely used computer-algebra systems: Derive,[2] MACSYMA,[3] Maple,[4], and *Mathematica*.[5] Although the first system is much smaller than the others, it is useful, inexpensive, and runs on the most humble personal computer. The last three are complete and complex programs that have versions for many computer types. However, a larger hardware configuration is needed for their use.

J. M. Aguirregabiria, A. Hernández, and M. Rivas are in Física Teorica, Universidad del País Vasco, Apdo. 644, 48080 *Bilbao, Spain.*

"True" bugs

In practice, all nontrivial programs have errors, and so it is not surprising that every computer-algebra system has bugs and occasionally gives erroneous results. In fact, these systems may have programming errors that give incorrect mathematical answers as well as bugs or limitations in the user interface. Moreover, it is not always easy to interrupt run-away calculations. Abrupt exits (due sometimes to poor management of stack and memory resources) sometimes give users little opportunity to save their work.

Although we have found bugs in every computer-algebra system that we have tried, we will not discuss them here, because they are program-dependent and change from version to version. We wish to discuss problems of a potentially more serious nature. They are problems that cannot be called "bugs," because they do not cause anomalous behavior against the explicit intentions of the developer. They are contrary to users' needs. In some cases, explicit design options lead to results that are outright erroneous. For instance, let us assume that a lazy student is asked if the function x^{-2} is integrable over the interval $(-1,1)$. If the student types the equivalent of $\int_{-1}^{1} dx/x^2$, the answer will be incorrectly given as -2 in Derive and in all but the latest version of *Mathematica*! Of course, a careless student might obtain by hand the same answer, by blindly applying Barrow's rule, but this is no excuse for the software's giving the wrong answer, just as the fact that some students are unable to compute, say, square roots, would not justify calculators' returning completely erroneous answers when computing them. Maple, which always tries to ascertain if a primitive is continuous, gives the right answer in the latest version, whereas it returned the integral unevaluated in previous versions. MACSYMA also recognizes that the integral is divergent.

Limited domains of validity

Computer-algebra systems not only sometimes give erroneous results, they also may give results that are only correct over a limited domain. Sometimes the assumptions

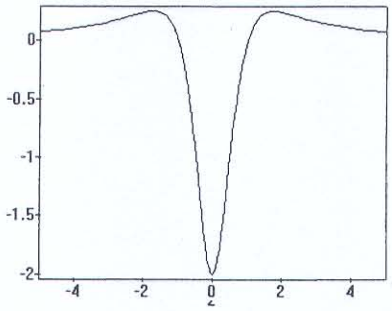
```
Potential
> V := 1/sqrt(r^2+(z-I)^2)+1/sqrt(r^2+(z+I)^2):

Electric field
> Er := -diff(V,r):
> Ez := -diff(V,z):

Laplace's equation
> simplify(diff(r*diff(V,r),r)/r+diff(V,z$2));
                                    0

Charge density
> limit(Ez,z=0,right)-limit(Ez,z=0,left);
                                    0

Plot Ez for r = 0
> plot(Re(subs(r=0,Ez)),z=-5..5);
```

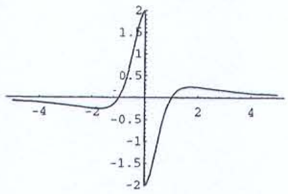



```
Total charge
> simplify(int(r*Er,z=-infinity..infinity));
                                    4

>
```

Figure 1. Maple "worksheet" shows results with a limited domain of validity for the charge density σ and total charge q of potential (1) and an erroneous plot for E_z *along the axis* $r=0$.

```
Potential
In[1]:=
V = 1/Sqrt[r^2+(z-I)^2]+1/Sqrt[r^2+(z+I)^2];
Electric field
In[2]:=
Er = -D[V,r] // Together ;
Ez = -D[V,z] // Together ;
Laplace's equation
In[4]:=
D[r D[V,r],r]/r+D[V,{z,2}] // Together
Out[4]=
0
Charge density
In[5]:=
Limit[Ez,z->0,Direction->1]-Limit[Ez,z->0,Direction->-1]
Out[5]=
0
Plot Ez for r = 0
In[6]:=
Plot[Ez /. r->0, {z,-5,5}, PlotRange->{-2,2}];
```




```
Total charge
In[7]:=
Integrate[r Er // Simplify,{z,-Infinity,Infinity}] //
        Simplify
Out[7]=
4
```

Figure 2. Mathematica *"notebook" shows the same limited domain of validity as in Fig. 1 for the charge density σ and total charge q* of potential (1), but produces a correct plot for E_z.

are not stated explicitly with the results. (In some cases, short notes about the corresponding functions appear in the instruction manual—but not always.)

Consider this example. Some time ago, a colleague asked us if the following function in cylindrical coordinates $(r,\ \varphi,\ z)$,

$$V=\frac{1}{8\pi\epsilon_0}\left(\frac{1}{\sqrt{r^2+(z-i)^2}}+\frac{1}{\sqrt{r^2+(z+i)^2}}\right), \qquad (1)$$

could be considered as an electrostatic potential. One can easily see by using any computer-algebra system that V satisfies Laplace's equation. (This fact is obvious because V is formally the potential of two point charges $q=1/2$ located at points $r=0,\ z=\pm i$.) So, it is rather natural to ask if the function can be considered as the electrostatic potential of a charge density with physical meaning. It is also apparent that V is real except perhaps in the disk $\{z=0,\ r^2\leqslant 1\}$, and a little thought shows that the discontinuities (i.e., the charge) must be located precisely in that disk.

The surprise arises when computing the charge density in the disk by using

$$\sigma=\epsilon_0(\lim_{z\to 0+} E_z-\lim_{z\to 0-} E_z)=\epsilon_0\left(\lim_{z\to 0-}\frac{\partial V}{\partial z}-\lim_{z\to 0+}\frac{\partial V}{\partial z}\right) \qquad (2)$$

The four programs give the same answer: zero. One could be tempted to think that, if four independent programs give the same answer, it must be correct. In this case, however, even this wide agreement is not enough to avoid an erroneous conclusion. Let us assume that we are careful enough to plot E_z for $r=0$ and $-5<z<5$. If we use Maple, we could be completely convinced of the rightness of the null result, because doing the plot before simplifying E_z shows a continuous function (see Fig. 1). As shown in Fig. 2, if we use Derive or *Mathematica*, we see that E_z has a jump trough at $z=0$ (the same can be done by using MACSYMA, but two plots, one for $z<0$ and the other for $z>0$, seem necessary). This example is a clear indication that something is wrong with the symbolic result. In fact, it is not difficult to check by hand that

$$\sigma=\begin{cases} -\dfrac{1}{2\pi}(1-r^2)^{-3/2}, & \text{if } r<1; \\ 0, & \text{otherwise.}\end{cases} \qquad (3)$$

So, the answer for σ was not exactly wrong, but rather it was only valid for $r>1$, though no indication of this range was given by any of the programs. Probably the four programs were assuming $\sqrt{(r^2-1)^2}=r^2-1$, which is not always true. This assumption also explains Maple's erroneous plot; apparently the program uses this symbolic simplification before evaluating the numerical result.

Derive has the nice possibility of declaring the domain of a variable. It is then possible to choose $0<r<1$. But if this is done, the program exhausts all memory and returns, after a long time, with no result. It seems that Derive is using l'Hospital's rule in this case to compute limits. The difficult point is the calculation of $\lim_{z\to 0\pm} z/R$, where

$$R=\sqrt{\sqrt{z^4+2z^2(r^2+1)+(r^2-1)^2}+z^2+r^2-1}. \quad (4)$$

Applying l'Hospital's rule once, one obtains

$$\lim_{z\to 0\pm}\frac{z}{R}=\lim_{z\to 0\pm}\frac{1}{\partial R/\partial z}=\frac{1-r^2}{2}\lim_{z\to 0\pm}\frac{R}{z}. \quad (5)$$

The desired result is essentially contained in the above expression, but a blind repeated application of l'Hospital's rule will obviously lead to an endless loop. *Mathematica* returns $\mp\infty\sqrt{\sqrt{(r^2-1)^2}-1+r^2}$ for $\lim_{z\to 0\pm} R/z$, which at least suggests that something is happening. Surprisingly, *Mathematica* gives 0 for $\lim_{z\to 0\pm} z/R$. Maple's latest version has an **assume** command that does not change the behavior described before. Maple always gives 0 for $\lim_{z\to 0\pm} z/R$, because it simplifies $\sqrt{a^2}=a$, even if it is said to assume $a<0$. (The answer to $\lim_{z\to 0\pm} R/z$ is ∞ sign$(\sqrt{r^2-1})$.) MACSYMA asks if r^2-1 is positive or negative, but always returns 0 for $\lim_{z\to 0\pm} z/R$, and leaves $\lim_{z\to 0\pm} R/z$ unevaluated if it is told that $\sqrt{(r^2-1)^2}+r^2-1$ is zero.

The charge density in (3) diverges as $r\to 1$, and it is easy to check that (3) gives an infinite charge for the disk. So, one expects an additional infinite charge located on the disk edge because the potential is regular outside the disk. Although now it is clear that (3) does not represent a realistic electrostatic potential, one can still compute the total charge by applying Gauss's theorem to an infinite cylinder of radius r around the axis OZ:

$$q=\epsilon_0\int_{-\infty}^{\infty}2\pi r E_r\,dz=-2\pi\epsilon_0 r\int_{-\infty}^{\infty}\frac{\partial V}{\partial r}\,dz. \quad (6)$$

Derive and MACSYMA are not able to calculate (6) without further help,[6] and Maple and *Mathematica* give the same result, 1, with no indication that this answer is valid only for $r>1$.

To find the electrostatic field created by a cylindrical charge density $\sigma \propto z \sin\phi$ by direct application of Coulomb's law, we needed some time ago to calculate the following integral:[7]

$$\int_{-\pi}^{\pi}\frac{\cos\varphi(\cos\varphi-x)\,d\varphi}{(\cos\varphi-x)^2+(\sin\varphi-y)^2} =\begin{cases}\pi, & \text{if } x^2+y^2<1;\\ -\pi(x^2-y^2)/(x^2+y^2)^2, & \text{if } x^2+y^2>1.\end{cases} \quad (7)$$

Although the answer is not difficult to obtain by using complex variable techniques, it is even easier to tell the computer to do it for you. At that time, Derive was unable to solve the integral correctly, and *Mathematica* returned only the last value (its latest version is no longer able to return any result in this case), but our knowledge of the physical problem told us that the result should depend on the sign of x^2+y^2-1. In this case, Maple gave us the right answer. MACSYMA and the current version of Derive also give the correct result.

Another kind of difficulty arises because computer-algebra systems are not able to use all the mathematical rules they know in the right places, as in the following example:

$$\lim_{\epsilon\to 0}\frac{\epsilon}{\sin^2\varphi+(1-\epsilon)\cos^2\varphi-1}=-\sec^2\varphi. \quad (8)$$

Derive gets the right trivial answer of this fake limit, but MACSYMA, Maple, and *Mathematica* (the big systems!) give 0 because they fail to recognize the most elementary trigonometric identity. If you try an expansion in powers of ϵ, the last three programs give a completely absurd series in powers of $\epsilon/(\cos^2\varphi+\sin^2\varphi-1)$. Of course, this is a rather artificial example and it is sufficient to simplify the function before applying the limit to get the right answer. But the same problem arises in the following example,

$$\lim_{\epsilon\to 0}\frac{\epsilon}{\sqrt{1+\epsilon}\sin^2\varphi+\sqrt{1-\epsilon}\cos^2\varphi-1} =\frac{2}{1-2\cos^2\varphi}, \quad (9)$$

and more involved expressions. Moreover, the trouble can become really serious if this kind of difficulty happens in the middle of a very complex computation, where intermediate results are not even seen by the user.

Helping your system

Another problem that sometimes arises is that students erroneously assume that computer-algebra systems always know more than they do about how to compute some expressions. For instance, if a computer-algebra system returns an integral unevaluated, a student could be tempted to think that there is no exact solution in terms of known functions, that solving the integral is exceedingly complicated, or that very sophisticated methods are needed to evaluate it. However, even though it is true that computer-algebra systems use powerful and general algorithms, sometimes a more elementary approach proves useful.

The following example was posed to us by a colleague. He wanted to check the value of the following integral appearing in a paper[8] that dealt with the evaluation of a class of diagrams in many-body theory:

$$\int\frac{x}{\sqrt{a+bx^2}}\log\frac{c+dx^2+x\sqrt{a+bx^2}}{c+dx^2-x\sqrt{a+bx^2}}\,dx. \quad (10)$$

Maple was not able to evaluate this integral on its own (and the same happens with the other systems), but if it is instructed to do the elementary integration by parts, step by step, the (rather involved) result can be readily found (see Fig. 3).

Intermediate expression
```
> sq := sqrt(a+b*x^2):
```
The integrand
```
> ii := x/sq*ln((c+d*x^2+x*sq)/(c+d*x^2-x*sq)):
```
Maple cannot compute it without help
```
> int(ii,x);
```
$$\int \frac{x \ln\left(\frac{c+d\,x^2+x\sqrt{a+b\,x^2}}{c+d\,x^2-x\sqrt{a+b\,x^2}}\right)}{\sqrt{a+b\,x^2}}\,dx$$

Instruct it performing integration by parts
```
> u := int(x/sq,x):

> v := ln((c+d*x^2+x*sq)/(c+d*x^2-x*sq)):

> dv := simplify(diff(v,x)):

> r := u*v-int(u*dv,x):
```
Check the result
```
> normal(diff(r,x)-ii);
```
$$0$$
```
>
```

Figure 3. Integration by parts helps Maple to compute the integral (10).

To evaluate the power dissipated by the Joule effect due to the current induced inside a conducting sphere by a point charge moving in its neighborhood, we needed to compute an integral in the form:[9]

$$\int_0^{\pi} E^2 \sin\,\theta d\theta, \tag{11}$$

where

$$E=\frac{1}{\sin\,\theta}\left[\frac{a\,\cos^2\theta+b\,\cos\,\theta+c}{(1-2r\,\cos\,\theta+r^2)^{5/2}}-\cos\,\theta\right] \tag{12}$$

is a component of the induced electric field. Derive, Maple, and *Mathematica* were unable to compute (11). It is easy to see that, by using $z=\sqrt{1-2r\,\cos\,\theta+r^2}$ as the integration variable, one obtains a rational integral that can then be evaluated. After some flags have been carefully set, MACSYMA is able to compute (11), but the work needed to obtain the final compact result is greater than that involved in instructing the program to use the transformation just discussed.

Although summing the series,

$$\sum_{n=1}^{\infty}\frac{(2n+1)(n+1)b^n}{n}=\frac{3b}{1-b}+\frac{2b}{(1-b)^2}-\log(1-b) \tag{13}$$

is not too difficult, our computer-algebra systems failed to obtain the result. Nevertheless, by instructing Maple to first expand the general term, the program was able to find the sum. (The latest version of *Mathematica* is able to compute (13) with no help.)

The preceding examples clearly show that elementary calculation techniques must still be taught to students, because a program may not always apply them as necessary.

Also, some calculations seem never ending or exhaust computer resources prematurely. In some cases, a little help may allow the program to complete the task or complete it faster.

Assume that we want to compare different power expansions of $f(\theta)$,

$$g(\theta)=\sum_{k=0}^{n}\frac{f^{(k)}(\theta_0)}{k!}(\theta-\theta_0)^k, \tag{14}$$

by using the L^2 norm:

$$\|f-g\|^2\equiv\int_0^{2\pi}(f(\theta)-g(\theta))^2 d\theta. \tag{15}$$

In[1]:=
```
<< Utilities`ShowTime`
```
Direct integration

In[2]:=
```
i1[f_] := Integrate[f,{x,0,2Pi}]
0. Second
```
First expand the integrand, then integrate each term

In[3]:=
```
i2[f_] := Map[Integrate[#,{x,0,2Pi}]&,Expand[f]]
0. Second
```
The integrand

In[4]:=
```
f = Sin[x]+3Cos[2x];
f = (f-Normal[Series[f,{x,1,4}]])^2;
0. Second
6.32 Second
```
Use both methods

In[6]:=
```
r1 = i1[f] ;
5559.89 Second
```
In[7]:=
```
r2 = i2[f] ;
104.2 Second
```
The results must agree, of course

In[8]:=
```
r1-r2 // Expand // Simplify
18.78 Second
```
Out[8]=
```
0
```

Figure 4. Telling Mathematica *to use linearity of integration speeds up the calculation of (15).*

By choosing $f=\sin(\theta)+3\cos(2\theta)$, $\theta_0=1$, and $n=4$, we see in Fig. 4 that a direct application of (15) in *Mathematica* takes fifty times longer than instructing the program to expand the integrand and apply the linearity of integration. Furthermore, this speed-up grows dramatically with n. Again, the computer-algebra system was not "intelligent" enough. On the contrary, MACSYMA computes the unexpanded integral faster. (For the expanded integral it seems to use complex-variable techniques.) Derive and Maple are very fast in both cases. These different behaviors also show that knowledge of the weaknesses and strengths of different computer-algebra systems may save time by helping the user to choose the most appropriate system in each case.

In some cases, rather than explicitly defining certain quantities, it is better to make use of properties that directly

```
Direct definition of the set of integrals
In[1]:=
    i0[k_,p_,q_] := 1/Pi Integrate[
                          z^(2 k)/
                          ((a^2 z^2+b^2)^p (c^2 z^2+d^2)^q),
                                 {z,-Infinity,Infinity}]
Properties of the family
In[2]:=
    i[0 ,1 ,0 ] = 1/(a b) ;
In[3]:=
    i[0 ,0 ,1 ] = 1/(c d) ;
In[4]:=
    i[0 ,p_,0 ]:= i[0,p,0] =
                  D[i[0  ,p-1,0  ],b]/(2 b(1-p))
In[5]:=
    i[0 ,0 ,q_]:= i[0,0,q] =
                  D[i[0  ,0  ,q-1],d]/(2 d(1-q))
In[6]:=
    i[0 ,p_,q_]:= i[0,p,q] =
                  (a^2 i[0,p,q-1]-c^2 i[0,p-1,q])/
                  (a^2 d^2-b^2 c^2)
In[7]:=
    i[k_,0 ,q_]:= i[k,0,q] =
                  D[i[k-1,0  ,q-1],c]/(2 c(1-q))
In[8]:=
    i[k_,p_,q_]:= i[k,p,q] =
                  (i[k-1,p-1,q]-b^2 i[k-1,p,q])/a^2
Compute an integral by using both methods
In[9]:=
    (t1,r1) = i0[1,2,3] // Together  // Timing ;
In[10]:=
    (t2,r2) = i[1,2,3]  // Together // Timing ;
Relative computing time
In[11]:=
    t1/t2
Out[11]=
    11.5829
Check that both results agree
In[12]:=
    r1-r2 // Together // PowerExpand /.
             {Sqrt[x_^2] :> x, Sqrt[y_^(-2)] :> 1/y}
Out[12]=
    0
```

Figure 5. Teaching Mathematica *some properties of integrals in (16) leads to a far more effective way to compute them.*

influence the calculation to be performed. Several years ago, for instance, in order to calculate the electromagnetic angular momentum radiated by a system of two interacting point charges, we needed to compute a set of integrals of the form:[10]

$$I_{kpq}=\int_{-\infty}^{\infty}\frac{z^{2k}}{(a^2z^2+b^2)^p(c^2z^2+d^2)^q}\,dz,\quad (0\leqslant k<p+q). \tag{16}$$

Our old computer-algebra system[11] was unable to compute these integrals except in the most trivial cases. A little thought convinced us that all of the required integrals can be derived by differentiation and algebraic operations from the elementary result

$$I_{010}=\int_{-\infty}^{\infty}\frac{z^{2k}}{(a^2z^2+b^2)}\,dz=\frac{\pi}{ab}\,. \tag{17}$$

This observation gave a fast solution to our problem. Our computer-algebra systems are now able to compute such integrals, but the same trick speeds up calculations in a spectacular way. As can be seen in Fig. 5, *Mathematica* needs more than ten times longer to compute I_{123} by evaluating the integral in (16) than by using the properties of the family of integrals. So, these types of tricks could prove valuable if many such integrals must be computed. In *Mathematica*, one does not have to deal with annoying terms in the form of $\sqrt{a^2}$ and $\sqrt{b^{-2}}$. In the other three computer-algebra systems, the performance gain is very similar, but this approach may prove less useful, because the definition of the relation among integrals is not as natural as in *Mathematica*, and a short, but more traditional programming job is necessary.

Final comments

Some issues discussed in this paper are not exclusively related to computer-algebra but are also applicable to calculations performed by hand or by using tables. In fact, sometimes there are more possibilities of error if a careless student uses tables instead of a computer-algebra system. For instance, computing

$$\int_0^{\infty}\frac{x^2dx}{(1+x^4)} \tag{18}$$

by using Barrow's rule with the following primitive that can be found in well-known tables:[12]

$$\int\frac{x^2dx}{(1+x^4)}=\frac{1}{4\sqrt{2}}\left\{\log\frac{x^2-\sqrt{2}\,x+1}{x^2+\sqrt{2}\,x+1}+2\tan^{-1}\frac{\sqrt{2}x}{1-x^2}\right\}, \tag{19}$$

might lead to the conclusion that (18) is null, though it is an obviously ridiculous result for the integral of a positive function. The reason is, of course, that the primitive is not continuous over the interval. In this case, however, our four computer-algebra systems give the right answer, perhaps because they use the following continuous primitive,

$$\int\frac{x^2dx}{(1+x^4)}=\frac{1}{4\sqrt{2}}\left\{\log\frac{x^2-\sqrt{2}x+1}{x^2+\sqrt{2}x+1}+2[\tan^{-1}(\sqrt{2}x+1)+\tan^{-1}(\sqrt{2}x-1)]\right\}, \tag{20}$$

which differs from (19) by a function that is only piecewise constant.

Nonetheless, since computer algebra allows very complex calculations to be attempted, which are often performed in an automatic or semi-automatic way, there is more place for errors. For instance, it is not rare for there to be many intermediate results that are not even seen by the user, or which are so complex that there is no hope to check them.

We do not blame computer-algebra systems. After all, we use them all the time! But we think that one must first learn to use them and, then, be very careful. We recommend the following check list of remedies that always prove useful for verifying calculations performed by hand or by a computer-algebra system:

- Use common sense and your previous knowledge of the problem to check if the result is plausible.
- Perform the calculation in more than one way.
- Check particular cases. Sometimes, exact cases are known, or one can evaluate some of them numerically. Graphics are also helpful in this respect.
- Use more than one computer-algebra system. This is useful not only for checking the validity of results, but also

because each system has its own strong points and performs some calculations better than the other systems. Note that this recommendation is equivalent to having a colleague check your calculations or apply techniques that you have not yet mastered. (Unfortunately colleagues and additional computer-algebra systems are not always available.)

One more thing, if you find a true error in a computer-algebra system, please inform the publisher. If the error can be corrected in the next version of the program, you will help other people save valuable time.

Acknowledgments

This work has been partially supported by the University of Basque Country under contract UPV/EHU 172.310 EA046/92.

References

1. See for instance: D. M. Cook, Computers in Physics **4**, 2, 197 (1990); D. M. Cook, Computers in Physics **4**, 3, 308 (1990); M. Horbatsch, Computers in Physics **4**, 6, 656 (1990); M. MacDonald, Eur. J. Phys. **12**, 10 (1991); P. Tam, Computers in Physics **5**, 4, 438 (1991); R. E. Crandall, Computers in Physics **5**, 6, 576 (1991); D. M. Cook, R. Dubisch, G. Sowell, P. Tam, and D. Donnelly, Computers in Physics **6**, 4, 411 (1992); **6**, 5, 530 (1992).
2. Derive is a trademark of and available from Soft Warehouse, 3615 Harding Avenue, Suite 505, Honolulu, Hawaii 96816-3735. We use version XM 2.56.
3. MACSYMA is a trademark of and available from MACSYMA Inc., 20 Academy Street, Arlington, MA 02174. We use version PC 417.125 delta 2.
4. Maple is a trademark of and available from Waterloo Maple Software, 160 Columbia Street, Unit 2, Waterloo, Ontario N2L 3L3, Canada. We use version V Release 2.
5. *Mathematica* is a trademark of and available from Wolfram Research, Inc., 100 Trade Center Drive, Champaign, IL 61820-7237. We use version 2.2.
6. When we write that a system was unable to compute an expression, we mean that we entered it into the program in the most straightforward way and that it returned unevaluated or, in some cases, that we decided to abort the calculation after it had been running for a long time (typically for over 1 h) in our 486DX-2 (66 MHz) personal computer with 16 Mbytes of RAM.
7. J. M. Aguirregabiria, A. Hernández, and M. Rivas, Am. J. Phys. **60**, 138 (1992).
8. R. Cenni and P. Saracco, Nucl. Phys. **A487**, 279 (1988).
9. J. M. Aguirregabiria, A. Hernández, and M. Rivas, *Velocity fields inside a conducting sphere near a slowly moving charge.* To appear in Am. J. Phys. (1993).
10. J. M. Aguirregabiria and J. M. Etxebarria, J. Math. Phys. **29**, 1832 (1988).
11. muMath is a trademark of Soft Warehouse, Licensed by Microsoft Corp. We use version 4.12. It is now nearly obsolete and has been substituted by Derive, but up to the most recent version of *Mathematica* it remained our only computer-algebra system able to compute (13) without help.
12. I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series and Products* (Academic Press, Orlando, 1980), Integral 2.132.3.