Measurements and comparisons of SMP

Michael B. Monogan
Gaston H. Gonnet
Bruce W. Char

Research Report CS-85-47
December 1985

Measurements and comparisons of SMP

Michael B. Monagan
Gaston H. Gonnet
Bruce W. Char

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

ABSTRACT

In the April, 1985 issue of *Communications of the Association for Computing Machinery,* Stephen Wolfram wrote an article entitled "Symbolic Mathematical Computation". This report includes the text of a letter to the editor of *CACM* in response to some of the claims made in the article about SMP, along with the details of various measurements and tests performed in support of the statements made in the letter.

This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada and Digital Equipment Corporation, and in part by the Academic Development Fund of the University of Waterloo. The opinions expressed herein are those of the authors and not necessarily those of any grantor.

The Letter

The April/85 issue of *Communications of the ACM* contains an article entitled "Symbolic Mathematical Computation" by Stephen Wolfram. As developers of a symbolic manipulation system, Maple, we had hoped that such an article would generally promote interest in computer algebra among our colleagues. We are writing this letter because instead we have perceived by comments from colleagues and those made in electronic news/discussion groups, that this article has misguided people, and has damaged the credibility of work in the symbolic computation area.

First of all it should be noted that in general, this article, though not treated editorially as an advertisement, reads as a sales pitch of a product, with its assertions that other computer algebra systems are less "general" or "efficient". The article never acknowledges that many features of SMP's design are implemented in several other systems (mathematical extensibility, interactive usage, portability to mainframes and workstations, two dimensional output of mathematical expressions, on-going incorporation of mathematical knowledge into the system, to name a few).

On the first page we read: "The most advanced and complete mathematics system at present available is probably SMP." We are concerned that the prestige of CACM will lend itself to the acceptance of this claim, leading to a generally held opinion that shortcomings of SMP are "state-of-the-art" for all computer algebra systems. The following examples, taken from SMP version 1.5.0 (the latest version we had as of November, 1985, running on a Vax 11/780 with Berkeley Unix) indicate our concern with the proposition: "if SMP is 'the most advanced and complete mathematics system', what can a user expect from the other systems" ?

```
/* In the following examples, #I[...] labels input statements, while
    #O[...] labels the lines printed by the system in response to its input */

/* "B" is the function which specifies that exact integers of arbitrary
precision instead of floating point numbers, will be used in the calculation. */

#I[1]::  B[ (x+1)^80 ]
                  80
#O[1]:*  (x + (1))


#I[2]::  Ex[ % ]
         /* The lengthy result from expanding the above expression,
         not reproduced here, contains *negative* coefficients */
```

```
#I[1]::  p : B[ (x+1)^100 ]
                  100
#O[1]:*  (x + (1))

#I[2]::  Ex[ p ]
         /* Appears to be in an infinite loop (exceeded 2000 CPU seconds) */
```

```
#I[1]::  Pgcd[ (x+1)^2, (x+1)*(x-1), x ]

#O[1]:   (1 - x) (1 + x)
        /* Wrong, the polynomial gcd is obviously  1 + x  */
```

```
#I[1]::  Fac[ B[x^2-1] ]

#O[1]:   (-1 + x) (1 + x)

#I[2]::  Fac[ B[x^2-y^2] ]

          2         2
#O[2]:*  y  (-1) + x

        /* SMP apparently thinks that  x^2-y^2  does not factor */
```

```
       /*A numerical Bessel function computation*/
#I[1]::  N[BesJ[0,2.7],40]

8878 Bus error
/* This is a Unix system error.  It means that SMP ''crashed'' . */
/* The N function is used here to compute a 40 Digit floating point approx. */
```

```
#I[1]::  x/0

         x
#O[1]:   -
         0

/* Although SMP allows such an expression to exist, this leads to bugs */
/* For example, taking the limit of the above as x goes to 2 we get */

#I[2]::  Lim[%,x,2]

#O[2]:   2
```

```
#I[1]::  374^3*x - 374*374*374*x
                                 3
#O[1]:   -5.23136*^7x + 374  x

#I[2]::  B[%]

#O[2]:*  x (-52313624) + x (52313624)
        /* SMP did NOT do the coefficient arithmetic in a sum */
```

```
#I[1]::  d : x^2-1-(x+1)*(x-1)

                                      2
#O[1]:   -1 - (-1 + x) (1 + x) + x

/* In SMP, little attempt is made to recognize zeroes.  This leads
to wrong or meaningless answers being returned, as well as system failures
as the following examples illustrate.  Other systems make use a normal form
for polynomials and rational functions to recognize zeroes. */

#I[2]::  Pgcd[ d, x^2+x+1, x ]

#O[2]:   1 + (1 - (1 - x) (1 + x)) (2 - (1 - x) (1 + x))

#I[3]::  Ex[ % ]

             2    4
#O[3]:   1 + x  + x
        /* Note that the degree of both input polynomials is less than 4 */

#I[4]::  Lim[ 1/d,x,0 ]

          1
#O[4]:*  ----
            3
        0 x

#I[5]::  Int[ 1/d,x ]

SMP INTERNAL ERROR
Numerical overflow
```

```
#I[1]::  Coef[ x,(x+y)*(x+z) ]

#O[1]:   0
        /* Mathematically, the coefficient in x is not 0 . */
```

```
#I[1]::  Sol[ {x+y=1, 2*x+2*y=2}, {x,y} ]

#O[1]:   {{}}

        /* There are, in fact, an infinite number of solutions to this */
        /* trivial system of linear equations which we obtain by doing */

#I[2]::  Sol[ {x+y=1}, {x,y} ]

#O[2]:   {x -> 1 - y}
```

```
#I[1]::  Int[ -1/x^2, {x,-1,1} ]

#O[1]:   2
/* Wrong.  The integrand is clearly negative over the integration range */
/* SMP makes the error of assuming continuity over the range of integration */
```

```
#I[1]::  Simtran[{{3,5,7},{11,13,17},{19,23,29}}]

          0   0   0
#O[1]:   {{-},{-},{-}}
          0   0   0
        /* Simtran supposedly computes the similarity transformation matrix. */
        /* The correct result has three non-zero vector entries. */
```

Of course all systems have their share of shortcomings, bugs, peculiarities, etc. Yet we feel that computer users will infer from the CACM article and the above examples that such untrustworthy results are the norm in all "computer mathematics" systems. This disappoints us deeply, as it does not fairly reflect the progress made in the field over the past twenty years.

On the third page we read: "They [referring to MACSYMA and REDUCE] are comparatively slow and able to handle comparatively small symbolic expressions, ...". We think we cannot allow this line to appear unchallenged. While it is certainly true that computer mathematics systems have their particular strengths and weaknesses, we find that SMP is weak at a very fundamental level, namely that of basic arithmetic. Again, we present examples. For readers who would like to see the exact input submitted to each of the systems (not presented here for lack of space) we refer them to Computer Science Technical Report No. CS 8547 available at the departmental address below.

Our sample problems were run on Dec Vax machines at Waterloo. SMP and Maple were tested on watdaisy, a Vax 11/780 running Berkley Unix BSD 4.1. Macsyma and Reduce were tested on watmum, a Vax 11/785 running Berkley Unix 4.2. In order to make a fair comparison, we have converted 11/785 times to their 11/780 equivalents (A Vax 11/785 runs 50% faster than a Vax 11/780). For the systems implemented in C (SMP and Maple) we measured the total memory (space) used, by application of Unix's *ps* command during the computations. The space figures reported include the system code and initial data space (1910 Kilo-Bytes for SMP, 163 Kilo-Bytes for Maple). Times reported are in CPU seconds.

Please note that the problems submitted are almost trivial examples. The other systems in our table are quite capable of handling much larger problems without running out of space. We perceive that SMP's problems arise from two reasons. First, some algorithms are clearly inefficient, particularly space inefficient. Secondly, unlike the other systems, no garbage collection took place during these computations. Garbage collection only occurred after the computation had completed (or had ran out of space). This apparent inability to perform garbage collection during the computation must severely limit the size of many computations than can be completed successfully using SMP.

Problem	SMP 1.5.0		Maple 4.0		Reduce 3.1	Macsyma 308
1	15.6	5,012Kb	0.3	171Kb	0.7	0.15
1a	>60	(*)	0.6	195Kb	1.4	0.3
2	25.8	4,309Kb	1.1	203Kb	3.6	9.0
2a	>60	(*)	3.2	283Kb	9.3	22.5
3	28	3,501Kb	4.8	460Kb	5.4	6.6
3a	>71	(*)	11.4	717Kb	29.8	31.1
3b	>71	(*)	9.9	438Kb	7.2	4.5
3c	>71	(*)	20.0	606Kb	24.1	16.8
4	>119	(*)	1.9	218Kb	1.5	15.0
5	>75	(*)	2.1	285Kb	0.8	0.8
5a	>71	(*)	23.8	554Kb	8.1	1.7
5b	>300	(*)	1.6	221Kb	0.7	0.5
6	>71	(*)	0.7	201Kb	2.1	0.7
7	>90	(*)	2.2	228Kb	5.7	15.0

(*) SMP ran out of space (exceeded 7.2 Mega-Bytes of virtual storage)

Problem		Description
1	Integer division	divide 3^5000 by 3^30
1a		divide 3^10000 by 3^30
2	Rational addition	sum(1/k,k=1..200)
2a		sum(1/k,k=1..400)
3	Polynomial multiplication	s := sum(k*x^k,k=1..64); expand(s^2);
3a		s := sum(k*x^k,k=1..128); expand(s^2);
3b	Let p := 1+x+y+z*y+z.	expand p^5 then expand that squared
3c		expand p^6 then expand that squared
4	Polynomial division	divide expand(p^6) by expand(p^3)
5	Polynomial factorization	factor expand((x^2+y^3)^3)
5a		factor x^6 - y^6
5b		factor x^4 - 3^20
6	Polynomial Gcd	Let f=(107*x+53)^7 and g=(109*x+59)^7 Compute the gcd of expand(f) and expand(g)
7	integrate	x * exp(a*x) * sin(b*x) with respect to x

We do not see why SMP can be claimed the "most advanced and complete mathematics system" if it has relative difficulty on the problems above. Nor do we see why other systems can be generally dismissed as "comparatively slow" or only able to handle "comparatively small expressions" relative to SMP. Indeed for problems 3, 3a, 3b and 3c, SMP was slower, and used much more space than the other systems despite the fact that it was using hardware floating point arithmetic instead of software-supported "exact" arithmetic.

In the same paragraph we read: "(Two other similar systems, SCRATCHPAD [4] and ALTRAN[1], have never been widely distributed.)". We believe that this is simply untrue. Altran, at its time, was as widely distributed as the most popular symbolic algebra system, if not itself the most popular.

On the same page we find the comment: "SMP was probably the first system designed from the start to be a general computer mathematics language". We find this surprising as it contradicts what we remember from some of Wolfram's own presentations of SMP as a system designed to satisfy the needs of physicists. Furthermore, such a statement belittles the work of others designing general-purpose "computer mathematics systems" which have been available freely or commercially over the past 15 years.

Only the author or the people responsible for the SMP system can attempt to undo the harm caused by the CACM article. We sincerely hope they rectify this either by substantially improving the system or by retracting these claims.

(signed)

Michael Monagan
Gaston Gonnet
Bruce Char

Symbolic Computation Group
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada.
{decvax,ihnp4}!watmath!watmum!watmaple

Appendix A – Test input for performance problems 1-7

Maple input for problem 1

```
a := 3^10:
b := a^3:
a := a^500:
st := time():
a / b:
time() - st;
quit
```

MACSYMA input for problem 1

```
a : 3^10$
b : a^3$
a : a^500$
showtime : true$
a / b$
quit();
```

REDUCE input for problem 1

```
a := 3^10$
b := a^3$
a := a^500$
on time$
a / b$
bye
```

SMP input for problem 1

```
a : B[3^10];
b : a^3;
a : a^500;
a / b;
N[#T[4]]
```

Maple input for problem 2

```
a := 1/k $ k=1..200 :
st := time():
# We cannot use Maple's sum function here,  sum(1/k,k=1..200)  returns
# Psi(201)-Psi(1)  which, although correct, is not what we wanted to test.
convert( [a], `+` ):
time() - st;
quit
```

MACSYMA input for problem 2

```
showtime : true$
sum(1/k,k,1,200)$
quit();
```

REDUCE input for problem 2

```
on time;
for k := 1 : 200 sum 1/k $
bye
```

SMP input for problem 2

```
Sum[ B[1/k], {k,1,200} ]
N[#T[1]]
```

Maple input for problem 2a

```
a := 1/k $ k=1..400 :
st := time():
# We cannot use Maple's sum function here,  sum(1/k,k=1..200)  returns
# Psi(401)-Psi(1)  which, although correct, is not what we wanted to test.
convert( [a], `+` ):
time() - st;
quit
```

MACSYMA input for problem 2a

```
showtime : true$
sum(1/k,k,1,400)$
quit();
```

REDUCE input for problem 2a

```
on time;
for k := 1 : 400 sum 1/k $
bye
```

SMP input for problem 2a

```
Sum[ B[1/k], {k,1,200} ]
N[#T[1]]
```

Maple input for problem 3

```
a := sum(k*x^k,k=1..64) :
st := time():
expand( a^2 ):
time() - st;
quit
```

MACSYMA input for problem 3

```
p : rat(sum(k*x^k,k,1,64))$
showtime : true$
p^2 $
quit();
```

REDUCE input for problem 3

```
a := for k := 1 : 64 sum k*x^k $
on time;
a^2 $
bye
```

SMP input for problem 3

```
p : Sum[k*x^k,{k,1,64}];
Ex[p^2]
```

Maple input for problem 3a

```
a := k*x^k $ k=1..128 :
# We cannot use Maple's sum command here because it returns
# a rational function result which is not what we wanted here.
a := convert([a],`+`) :
st := time():
expand( a^2 ):
time() - st;
quit
```

MACSYMA input for problem 3a

```
p : rat(sum(k*x^k,k,1,128))$
showtime : true$
p^2 $
quit();
```

REDUCE input for problem 3a

```
a := for k := 1 : 128 sum k*x^k $
on time;
a^2 $
bye
```

SMP input for problem 3a

```
p : Sum[k*x^k,{k,1,128}];
Ex[p^2]
```

Maple input for problem 3b

```
p := expand((1+x+y+z+y*z)^5):
st := time():
expand(p^2):
time()-st;
quit
```

MACSYMA input for problem 3b

```
p : rat( (1+x+y+z+y*z) )$
p : p^5 $
showtime : true$
p^2 $
quit();
```

REDUCE input for problem 3b

```
p := (1+x+y+z+y*z)^5$
on time$
p^2 $
bye
```

SMP input for problem 3b

```
p : Ex[ (1+x+y+z+y*z)^5 ];
Ex[p^2]
```

Maple input for problem 3c

```
p := expand((1+x+y+z+y*z)^6):
st := time():
expand(p^2):
time()-st;
quit
```

MACSYMA input for problem 3c

```
p : rat( (1+x+y+z+y*z) )$
p : p^6 $
showtime : true$
p^2 $
quit();
```

REDUCE input for problem 3c

```
p := (1+x+y+z+y*z)^6$
on time$
p^2 $
bye
```

SMP input for problem 3c

```
p : Ex[ (1+x+y+z+y*z)^6 ];
Ex[p^2]
```

Maple input for problem 4

```
p := 1+x+y+z+y*z:
a := expand(p^6):
d := expand(p^3):
st := time():
divide(a,d,q);
time()-st;
quit
```

MACSYMA input for problem 4

```
p : rat(1+x+y+z+y*z)$
a : rat(p^6)$
d : rat(p^3)$
showtime : true$
a/d$
quit();
```

REDUCE input for problem 4

```
p := 1+x+y+z+y*z$
a := p^6$
d := p^3$
on time$
a/d$
bye
```

SMP input for problem 4

```
p : 1+x+y+z+y*z;
a : Ex[p^6];
d : Ex[p^3];
Pquo[a,d,x];
```

Maple input for problem 5

```
p := expand((x^2+y^3)^3):
st := time():
factor(p):
time() - st;
quit
```

MACSYMA input for problem 5

```
p : expand((x^2+y^3)^3)$
showtime : true$
factor(p)$
quit();
```

REDUCE input for problem 5

```
p := (x^2+y^3)^3$
on factor$
x^3-1$
on time$
p$
bye
```

SMP input for problem 5

```
p : Ex[ B[x^2+y^3]^3 ];
Fac[p]
```

Maple input for problem 5a

```
st := time():
factor(x^6-y^6):
time()-st;
quit
```

MACSYMA input for problem 5a

```
showtime : true$
factor(x^6-y^6)$
quit();
```

REDUCE input for problem 5a

```
on factor$
x^3-1$
on time$
x^6-y^6$
bye
```

SMP input for problem 5a

```
Fac[x^6-y^6]
```

Maple input for problem 5b

```
st := time():
factor(x^4-3^20):
time()-st;
quit
```

MACSYMA input for problem 5b

```
showtime : true$
factor(x^4-3^20)$
quit();
```

REDUCE input for problem 5b

```
on factor$
x^3-1$
on time$
x^4-3^20$
bye
```

SMP input for problem 5b

```
Fac[x^4-3^20]
```

Maple input for problem 6

```
a := expand((107*x+53)^7):
b := expand((109*x+59)^7):
st := time():
gcd(a,b):
time() - st;
quit
```

MACSYMA input for problem 6

```
a : expand((107*x+53)^7)$
b : expand((109*x+59)^7)$
showtime : true$
gcd(a,b)$
quit();
```

REDUCE input for problem 6

```
a := (107*x+53)^7$
b := (109*x+59)^7$
on time;
gcd(a,b)$
bye
```

SMP input for problem 6

```
a : Ex[ B[107*x+53]^7 ];
b : Ex[ B[109*x+59]^7 ];
Pgcd[a,b,x]
```

Maple input for problem 7

```
st := time():
int( x*exp(a*x)*sin(b*x), x ):
time() - st;
quit
```

MACSYMA input for problem 7

```
showtime : true;
integrate( x*exp(a*x)*sin(b*x), x )$
quit();
```

REDUCE input for problem 7

```
on time$
int( x*exp(a*x)*sin(b*x), x )$
bye
```

SMP input for problem 7

```
Int[ x*Exp[a*x]*Sin[b*x], x ]
```