A Brief overview of enhancements to MACSYMA's Poisson package

Bruce R. Miller*

September 7, 1995

1 Introduction

I will assume the reader is familiar with the current implementation of Macsyma's[1] POISSON package. Here, I briefly describe the changes I made to the package and whether they are expected to impact existing uses of POISSON.

- The first change was in tuning the code for modern lisps and for tuning the data structures, resulting in 10% speedup generally and a 2-3 times speedup in multiplication, for example. These changes should be compatible unless other packages are using the internals unbeknownst to me (which I think is unlikely, however).
- The multipliers of the angles in the trig arguments are packed into single integers. In the current implementation, the product $2\cos 8u \times \sin 8u$ should give $\sin 16u$, but if the encoding restricts the range to ± 15, the field for u overflows into the next field (say for v). The enhancement adds an extra 'overflow' bit to each field, and an error check to detect such overflows when multiplying Poisson series. Note, however, that the default parameters yield a smaller range for each argument multiplier resulting in a slight incompatibility.
- An initialization procedure has been added for `poisvars` (as well as the existing one for `poislim`). This assures that POISSON's internal parameters are set consistently whenever either variable is changed.

  Two extensions are provided by allowing `poislim` to be set to:

  - a list of integers representing the maximum multiplier for each angular variable, allowing a different range for each multiplier.

*Email:miller@cam.nist.gov

[1]The use of commercial products or their names implies no endorsement by NIST, the Dept. of Commerce or the U.S. Government.

  - False (in the sense of 'no limit') this allows the multipliers of the angles in trig arguments to be arbitrary macsyma expressions. In this case, the argument multipliers are represented by lists rather than packed integers.

  An alternative for initialization is the function `pois_setup` which takes alternating variables and their maximum multiplier.

  Another new variable is provided `pois_encode_liberalize` (default `False`): when true, it accepts sine and cosines of arguments which are not linear in `poisvars` with integral coefficients; the non-linear part is taken into the coefficients rather than signalling an error. The default value is compatible with existing usage.

- For convenience, the functions `poisplus` and `poistimes` were modified to accept any number of arguments.

- The '5 argument form' of `poissubst`: I understand what it computes, but frankly, I do not understand what it is supposed to represent mathematically. I finally convinced myself that `poissubst(a,b,c,d,n)` was *intended* to compute the equivalent of `taylor(subst(a+d,b,c),d,0,n)`, albeit more efficiently. I followed a suggestion by J. P. Fitch, noting that differentiation of a Poisson series is quite fast, we simply use Taylor's theorem. This is an incompatible change.

2 Suggestions

There are further changes to the package that would make it more robust and useful, particularly by making the use of poisson more transparent and integrated, but there appears to be no great interest in this facility in the user community. Is this wrong?