P-adic Reconstruction of Rational Numbers

Paul S. Wang*
Kent State University
Department of Mathematical Sciences
Kent
Ohio U.S.A.
44242

M.J.T. Guy
University of Cambridge
Computer Laboratory
Corn Exchange Street
Cambridge, England
CB2 3QG

J.H. Davenport
Emmanuel College
Cambridge
England
CB2 3AP

1. Introduction

In a recent paper, Wang [1981] introduces a p-adic algorithm for the construction of partial fraction decompositions. This differs from the usual p-adic algorithms for factorisation or the computation of greatest common divisors ([Wang, 1978], [Wang, 1980], [Moore & Norman, 1981]) in that the p-adic image is used to reconstruct rational numbers, rather than integers.

In order to generate these rational numbers, Wang [1981] gives an algorithm RATCONVERT (p. 216) to deduce the rational u/v such that $uv^{-1} \equiv c \pmod m$, with $|u|, |v| \leq \sqrt{(m/2)}$. In this paper we prove that RATCONVERT always reconstructs such a number, if it exists. As Wang points out, the representation, if it exists, is unique. Such a number need not exist: e.g. modulo 9 we have that 0/1 = 0/2 = 0, 1/1 = 2/2 = 1, 2/1 = 2, 1/2 = 5, -1/1 = -2/2 = 8, -2/1 = 7, -1/2 = 4, leaving 3 and 6 without any such representation.

2. The Algorithm

Wang's algorithm is (we adapt the notation slightly):

Algorithm RATCONVERT(c,m)

[1] u:= (1, 0, m), v:= (0, 1, c)

[2] While $\sqrt{(m/2)} \leq v_3$ do

[3] {q:= $\lfloor u_3/v_3 \rfloor$, r:= u-qv, u:= v, v:= r}

[4] If $|v_2| \geq \sqrt{(m/2)}$ then error()

[5] Return (v_3, v_2)

[Wang ensures that v_2 is positive, but we do not need this complication]

Throughout the iteration, the following equations hold:

$$u_1 m + u_2 c = u_3,$$
$$v_1 m + v_2 c = v_3.$$

If the algorithm terminates with a solution, then it clearly satisfies

$$v_3/v_2 \equiv c \pmod m.$$

We have to prove that any solution to $a/b \equiv c$ satisfying the size constraint $|a|, |b| \leq \sqrt{(m/2)}$ will be generated by this algorithm.

3. Continued Fractions.

The algorithm can be viewed another way: as the calculation of approximations to c/m, since

$$-v_1/v_2 = c/m - v_3/(mv_2).$$

The numbers q computed in step [3] are, in fact, the successive terms in the continued fraction approximation to c/m (see [Hardy & Wright, 1979] Theorem 161 or Knuth [1981] pp. 339-343):

$$\frac{c}{m} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cdots}}}.$$

After one iteration of step [3], we have $u_1 = 0$, $u_2 = 1$ and $-u_1/u_2$ is the first approximation (0) to c/m, and $v_1 = 1$, $v_2 = q_1$, and $-v_1/v_2$ is the second approximation ($1/q_1$) to c/m. The definition of v_1, v_2 in terms of the previous values of u_1, u_2, v_1, v_2 and q is then (after allowing for our different sign convention) just the standard iteration $P_{n+1} = qP_n + P_{n-1}$ for computing a convergent in terms of its two immediate predecessors. Hence we have that the sequence of $-v_1/v_2$ is the sequence of convergents to c/m.

Conversely, suppose that $a/b \equiv c \pmod m$, with $|a|, |b| \leq \sqrt{(m/2)}$. Then $a+dm = bc$ for some integer d, so that

$$\frac{d}{b} - \frac{c}{m} = \frac{a}{bm} .$$

Hence d/b is an approximation to c/m, and, since $|a|, |b| \leq \sqrt{(m/2)}$, the error is

$$\left|\frac{a}{bm}\right| \leq \frac{\sqrt{(m/2)}}{bm} \leq \frac{1}{2b\sqrt{(m/2)}} \leq \frac{1}{2b^2} .$$

We can now appeal to Hardy & Wright [1979] Theorem 184 (p. 153): If

$$\left|\frac{p}{q} - x\right| \leq \frac{1}{2q^2}$$

then p/q is a convergent [to x]. Hence the triple (d, b, a) is indeed generated by the algorithm, and so the algorithm is guaranteed to find a rational in the appropriate range which corresponds to c mod m, if such a value exists.

4. References

Hardy & Wright,1979
Hardy,G.H. & Wright,E.M., An Introduction to the Theory of Numbers (5th. ed.). Clarendon Press, Oxford, 1979.

Knuth,1981a
Knuth,D.E., The Art of Computer Programming, Vol. II, Semi-numerical Algorithms. Second Edition, Addison-Wesley, 1981.

Moore & Norman,1981
Moore,P.M.A. & Norman,A.C., Implementing a Polynomial Factorization and GCD Package. Proceedings of the 1981 ACM Symposium on Symbolic and Algebraic Computation, ACM Inc., New York, 1981, pp. 109-116.

Wang,1978
Wang,P.S., An Improved Multivariable Polynomial Factorising Algorithm. Math. Comp. 32(1978) pp. 1215-1231. Zbl. 388.10035.

Wang,1980
Wang,P.S., The EEZ-GCD Algorithm. SIGSAM Bulletin 14(1980) 2 pp. 50-60. Zbl. 445.68026.

Wang,1981
Wang,P.S., A p-adic Algorithm for Univariate Partial Fractions. Proceedings of the 1981 ACM Symposium on Symbolic and Algebraic Computation, ACM Inc., New York, 1981, pp. 212-217.

*Research supported in part by grants from the National Science Foundation (NSF-MCS 80-02414) and the Department of Energy (DE AS02 ER7602075-A010).

Running REDUCE and Lisp/360 interactively under a Time Sharing Option

Zdeněk Kalina
Czechoslovak Academy of Sciences
Astronomical Institute
6, Budečská, 120 23 Prague

The users of algebraic systems are acquainted with many advantages of using diaglog versions of these systems.

We have in use the algebraic system REDUCE of Professor A. C. Hearn and a Time Sharing Option system in our computing centre. Therefore, we decided to adopt Lisp/360 to make possible the use of a dialog version of REDUCE.

The aim was taken more widely and we made a user oriented Conversational Symbolic Manipulation System /COSYMS/. COSYMS is based on Lisp/360, so called Monitor, and the algebraic system REDUCE. The Monitor controls the initial and repeated activations of the Lisp interpreter. Lisp involved a special Command Analyser and a new I/O section. REDUCE is generated with interactive options.

Those who are interested may obtain more detailed information from the author.