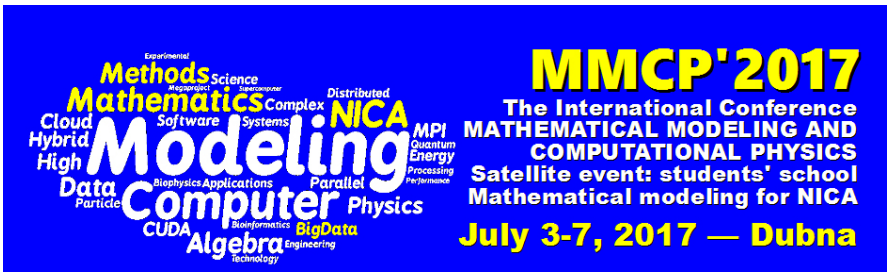

Contribution ID: **199** Type: **not specified**

CUNNINGHAM NUMBERS IN ACCELERATED MODULAR ARITHMETIC AND APPLICATIONS

Thursday, 6 July 2017 13:45 (30 minutes)

For certain modular algorithms, the modulus need not be a prime, and implementers are left with more freedom to choose moduli. One suggestion, made by Knuth [3], is to use integers of the form A similar choice is Numbers of these forms are Cunningham numbers for the special case of the base We discuss suitability of these types of moduli for standard computer algebraic modular algorithms. Different schemes of choice of these types of moduli are analyzed along with algorithms for conversion of arbitrary precision integers into the modular representation. In particular, this leads to division free linear time conversion to modular representation, and multiplication free reconstruction of the result from modular images. Experimental implementation of the described algorithms along with applications in multi-precision linear algebra and evaluation of rapidly convergent hypergeometric series are discussed. For very high bit lengths, over 10 000 bits, a modular representation using these Cunningham moduli outperforms both the standard binary representation as well as a modular representation using small prime moduli.

References

1. Garner H. The Residue Number System // IRE Transactions no Electronic Computers. Vol. 8. 1959. Pp. 140-147.

2. Geddes K.O., Czapor S.R., Labahn G. Algorithms for Computer Algebra (6th printing). –Boston: Kluwer Academic. 1992.

3. Knuth D.E. The Art of Computer Programming, Vol 2.

4. Křížek M., Luca F., Somer L. 17 Lectures on Fermat Numbers: From Number Theory to Geometry. –New York: Springer. 2001.

Primary author: Dr ZIMA, Eugene (Physics and Computer Science Department, Wilfrid Laurier University, Waterloo, Canada)

Presenter: Dr ZIMA, Eugene (Physics and Computer Science Department, Wilfrid Laurier University, Waterloo, Canada)

Session Classification: Computer Algebra and Quantum Computing with Applications (I)