Determinants of Modular Macaulay Matrices

S. Smith and M. Minimair
smithsab@shu.edu and manfred@minimair.org
Department of Mathematics and Computer Science
Seton Hall University
South Orange, New Jersey, USA

April 9, 2007

Introduction: Resultants of multi-variate polynomials are fundamental for solving systems of polynomial equations and have numerous applications. Thus they are being intensely studied.

Macaulay [3] showed that the projective resultant of a list of multi-variate polynomials is the quotient of two determinants. The numerator of this quotient is the determinant of a (sparse) matrix containing the coefficients of the polynomials, the so-called Macaulay matrix. The denominator is the determinant of a small minor of the Macaulay matrix. Current algorithms [1] for computing projective resultants [4] require computing the determinant of the Macaulay matrix in most cases. (In some degenerate cases they require computing coefficients of a characteristic polynomial.) Therefore it is essential for efficient resultant computations to efficiently compute determinants of Macaulay matrices.

Thus the goal of this undergraduate research is to efficiently compute determinants of Macaulay matrices. This undergraduate research is being conducted under the direction of M. Minimair and is being supported by NSF grant CCF-0430741.

Objective: The objective of this work is to efficiently compute determinants of modular Macaulay matrices via Gaussian elimination. By modular matrices we mean matrices whose entries are remainders modulo a given prime number. Such modular matrices commonly occur in symbolic computations in homomorphic images, driven by the Chinese Remainder Theorem.

Algorithm: Since Macaulay matrices are highly sparse, the efficiency of computing determinants via Gaussian elimination can be influenced through the pivoting strategy and the data structures used.

The pivoting strategy, chosen by us, minimizes the number of multiplications in each step of the Gaussian elimination. This minimization problem is efficiently solved by using a data structure described in [2] for storing the number of non-zero matrix entries per row.

Furthermore, we designed a particular data structure for sparse matrices in order to speed up the row operations. This data structure consists of the sparse matrix stored in a 2D array (as a dense matrix) and a linked list for each row. These linked lists contain the column indices of the non-zero entries in each corresponding row. The implementation of the linked lists is array-based and allows access, inserts and deletes in constant time.

Moreover, through experimentation we found out that our algorithm runs 10-30% faster on the transposes of the Macaulay matrices.

Results: The algorithm has been implemented in C++. Its running time has been compared to the running time of the dense determinant function of the software library NTL 5.4 [5]. Due to available computer memory, we were able to compute determinants of matrices of up to approximate size 8000×8000. We found that our implementation runs faster than NTL's determinant function for all Macaulay matrices, (naturally) except for ones for linear polynomials which are dense matrices. The largest speed-up recored is about 130.

Further Work: We are currently researching theoretical explanations for the speed-ups attained by some of the algorithmic features. For example, we are studying why using transposes of Macaulay matrices leads to a 10-30% speed-up. Furthemore, we observed that switching to dense Gaussian elimination after a certain number of sparse elimination steps may lead to a further substantial speed-up. We are currently investigating the optimal step for switching.

References

[1] J. Canny. Generalised characteristic polynomials. *J. Symbolic Computation*, 9:241–250, 1990.

[2] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, New York, 2003.

[3] F. S. Macaulay. *The algebraic theory of modular systems*. Cambridge Mathematical Library, 1916.

[4] M. Minimair. MR. http://minimair.org/MR.mpl, April 2003. Macaulay resultant package for Maple.

[5] V. Shoup. Ntl. http://shoup.net/ntl/.