

Permanent City Research Online URL: http://openaccess.city.ac.uk/734/

Copyright & reuse
City University London has developed City Research Online so that its users may access the research outputs of City University London's staff. Copyright © and Moral Rights for this paper are retained by the individual author(s) and/ or other copyright holders. All material in City Research Online is checked for eligibility for copyright before being made available in the live archive. URLs from City Research Online may be freely distributed and linked to from other web pages.

Versions of research
The version in City Research Online may differ from the final published version. Users are advised to check the Permanent City Research Online URL above for the status of the paper.

Enquiries
If you have any enquiries about any aspect of City Research Online, or if you wish to make contact with the author(s) of this paper, please email the team at publications@city.ac.uk.
A Symbolic-Numeric Software Package for the Computation of the GCD of Several Polynomials

Dimitrios Christou¹, Nicos Karcanias¹ and Marilena Mitrouli²

¹ Control Engineering Research Centre, School of Engineering and Mathematical Sciences, City University, Northampton Square, EC1V 0HB, London, U.K.
² Department of Mathematics, University of Athens, Panepistemiopolis 15784, Athens, Greece.
dchrist@math.uoa.gr, N.Karcanias@city.ac.uk, mmitroul@math.uoa.gr

Abstract. This survey is intended to present a package of algorithms for the computation of exact or approximate GCDs of sets of several polynomials and the evaluation of the quality of the produced solutions. These algorithms are designed to operate in symbolic-numeric computational environments. The key of their effectiveness is the appropriate selection of the right type of operations (symbolic or numeric) for the individual parts of the algorithms. Symbolic processing is used to improve on the conditioning of the input data and handle an ill-conditioned subproblem and numeric tools are used in accelerating certain parts of an algorithm. A sort description of the basic algorithms of the package is presented by using the symbolic-numeric programming code of Maple.

Introduction

The interaction of different type of computations (symbolic-numerical) is challenging for the development of new algorithms and has become an interesting area of research in the last decade. The presented software package for the computation of the GCD of several polynomials includes the following algorithms:

1. The ERES algorithm, which is a matrix based algorithm for the computation of the GCD of polynomials [2, 9].
2. The ResultantMatrix algorithm, which constructs a Resultant type matrix for more than two polynomials [5].
4. The Strength algorithm, which evaluates the quality of an approximate GCD [7].

The algorithms have been implemented and thoroughly tested in the programming environment of Maple and we shall present them in a matrix based formulation by using the notation of the programming code of Maple and routines of the LinearAlgebra package, which is included in Maple 8 and later versions.
Description and implementation of the algorithms

The Hybrid ERES algorithm

The main algorithm that we use for the computation of the GCD is the ERES algorithm [2,9].

Hybrid_Eres_GCD := proc ( Pmn :: set(polynom), et::float )

INPUT : et = Tolerance for the termination criterion of the algorithm.

OUTPUT : gcd = Vector of the GCD coefficients.

Pm := BasisMatrix( Pmn );
P := convert( Pm, rational );
k := 1;
While k > 0 do # MAIN ITERATIVE PROCEDURE
    r, q := Dimensions( P );
    di := RowDegree( P, i=1..r );
    RowReorder( P );

    If max( di ) = min( di ) then
        Pf := convert( P, float );
        Normalize( Pf );
        rho, sigma, tol, w := PSVD1( Pf, et )
        If rho=1 then # TERMINATION CRITERION
            If hastype( Pm, float ) then
                gcd := w; # GCD FROM SVD
            else
                gcd := P[1,1..q]; # GCD FROM THE MATRIX
            end if;
            break;
        end if;
    end if;

    Scale( P );
    GaussianElimination( P );
    Shifting( P );
    Shrink( P );

end do;
gcd;
end proc;
Description of subroutines:

- **BasisMatrix(P_{mn})**: Creates the initial $m \times n + 1$ basis matrix of the given set $P_{mn}$ from the coefficients of its polynomials.
- **RowDegree(P, i=1..r)**: Specifies the degree $d_i$, $i = 1..r$ of each polynomial row of the matrix $P$.
- **RowReorder(P)**: Reorders matrix $P$ such that $d_{i-1} \leq d_i$ for all $i = 2..r$.
- **Normalize(Pf)**: Normalizes the rows of Pf using the Euclidean norm.
- **Scale(P)**: Scales the matrix $P$ such that $P[1,1] > P[i,1]$ for all $i = 1..r$.
- **Shifting(P)**: Applies the shifting operation on every row of $P$.
- **Shrink(P)**: Deletes the zero rows and columns of $P$.

The PSVD1 algorithm: The PSVD1 algorithm is based on the methodology of the partial singular value decomposition [12, 13]. It is actually a variation of the classical singular value decomposition method [4, 1], especially developed for the efficient computation of the unique singular value and its right singular vector of an approximate $\varepsilon_t$-rank 1 matrix. The PSVD1 algorithm is a quick and effective tool for the detection of an approximate rank 1 matrix. It can increase the performance of other methods, such as the ERES method, and can be implemented easily in any software programming environment.

PSVD1 := proc(A::Matrix, theta::float)

INPUT: $A$ = $m \times n$ matrix with real floating-point data.
theta = a small positive bound such that
sigma_1<=...<=sigma_{k-1}<=theta<=sigma_{k}
where $k = \min(m,n)$.

OUTPUT: rho = Indicator of the rank of the matrix.
sigma = Maximum singular value.
w = Respective right singular vector.
tol = Recommended tolerance for a potential rank-1 matrix.

The Strength of an approximate GCD

The quality of a given approximate GCD known as the strength of the approximate GCD is evaluated by the following algorithm. A rigorous definition of the approximate GCD has been given recently [7] that allows the computation of the strength of approximation and sets up a framework for computing the optimal approximate GCD. This approach is based on recent results on the representation of the GCD of many polynomials in terms of the factorisation of the generalised resultant and a Toeplitz matrix representation of the GCD [5].
The procedure Strength

\[
\text{minstrength} := \text{proc( Pmn::set(polynom), GCD::polynom )}
\]

INPUT: Pmn = set of m univariate polynomials.
\hspace{1cm} GCD = a polynomial which is given as a GCD from a GCD algorithm.

OUTPUT: St = strength of the given GCD.

The strength of a given approximation can be computed by the routine : Optimization[Minimize] in Maple.

References

5. Fatouros, S., Karcanias, N.: Resultant properties of the GCD of many polynomials and a factorisation representation of GCD. Int. Journ. Control 76 No. 16 (2003), 1666–1683