

A Variant of Gerdt's Algorithm for Computing Involutive Bases

V. P. Gerdt*, Amir Hashemi†, Benyamin M.-Alizadeh†

* *Laboratory of Information Technologies
Joint Institute for Nuclear Research
Joliot-Curie 6, 141980 Dubna, Moscow region, Russia*

† *Department of Mathematical Sciences,
Isfahan University of Technology, Isfahan, 84156-83111, Iran*

In [1], the first author presented an efficient algorithm for computing involutive (and reduced Gröbner) bases. In this paper, we consider a modification of the algorithm which simplifies matters to understand it and to implement. We prove correctness and termination of the modified algorithm and also correctness of the used criteria. The proposed algorithm has been implemented in `Maple`. We present experimental comparison, via some examples, of performance of the modified algorithm with its original form described in [1] and implemented in `Maple` too. In doing so, we have taken care to provide uniform implementation details for the both algorithms.

Key words and phrases: Gröbner bases, Buchberger's criteria, involutive bases, Gerdt's algorithm.

1. Introduction

The most important algorithmic object in computational algebraic geometry is *Gröbner basis*. The notion of Gröbner basis was introduced and an algorithm of its construction was designed in 1965 by Buchberger in his Ph.D. thesis [2]. Later on [3], he discovered two criteria for detecting some unnecessary, and thus useless, reductions that made the Gröbner bases method a practical tool to solve a wide class of problems in polynomial ideals theory and in many other research areas of science and engineering [4].

In the 20s of the last century, French mathematician Janet [5] developed a constructive approach to analysis of certain systems of partial differential equations based on their completion to involution (cf. [6]). The Janet approach was generalized to arbitrary polynomially nonlinear systems by American mathematician Thomas [7]. Based on the related methods described in the book by Pommaret [8], Zharkov and Blinkov [9] introduced the notion of *involutive bases* in commutative algebra. The particular form of an involutive basis they used nowadays is called *Pommaret basis* [6, 10].

Gerdt and Blinkov [10] have proposed a more general concept of involutive bases for polynomial ideals and designed algorithmic methods for their construction. The underlying idea of the involutive approach is to translate the methods originating from Janet's results into polynomial ideals theory in order to construct involutive bases by combining algorithmic ideas of the theory of Gröbner bases and those in the theory of involutive differential systems. In doing so, Gerdt and Blinkov [10] introduced the concept of *involutive division*. Moreover, they have applied the involutive form of Buchberger's criteria. This led to a strong computational tool which was a serious alternative to the conventional Buchberger algorithm (note that any involutive basis is also a Gröbner basis). Apel and Hemmecke in [11] proposed two more criteria for detecting unnecessary reductions in involutive basis computations (see also [12]) which, in the aggregate with the criteria by Gerdt and Blinkov [10], are equivalent to Buchberger's criteria. The first author in [1] described an efficient algorithm to compute involutive and Gröbner bases using all these criteria. We refer to Seiler's

The contribution of the first author (V.P.G.) was partly supported by the grant 01-01-00200 from the Russian Foundation for Basic Research and by the grant 3810.2010.2 from the Ministry of Education and Science of the Russian Federation.

book [6] for a comprehensive study and application of involution to commutative algebra and geometric theory of partial differential equations.

In this paper, we propose a variant of Gerdt's algorithm [1] for constructing involutive bases. This algorithm seems to be simpler both to understand and to implement. We prove correctness for the new version of algorithm which includes the criteria and its termination. We have implemented this algorithm in `Maple` and we compare here its performance with our implementation of the original Gerdt's algorithm via some examples.

The structure of the paper is as follows. Section 2 contains the basic definitions and notations related to theory of involutive bases, and a short description of the algorithm for their computing proposed in its initial form [10]. In Section 3, we briefly present Gerdt's algorithm from [1] to compute involutive bases. Section 4 is devoted to description of the modified algorithm together with a proof of its correction and termination. In Section 5, we compare, by using some benchmarking examples, performance of the last algorithm with our implementation of Gerdt's algorithm done also in `Maple`.

2. Preliminaries

In this section, we recall some basic definitions and notations from the theory of involutive bases which are used in the paper, and briefly describe the initial algorithm proposed in [10] for computing such bases.

Let K be a field and $R = K[x_1, \dots, x_n]$ be the polynomial ring in the variables x_1, \dots, x_n over K . Below, we denote a monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in R$ by \mathbf{x}^α where $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ is a sequence of non-negative integers. A *monomial ordering* on R used in theory of Gröbner and involutive bases is a total order \prec on the set of all monomials satisfying the following properties:

- it is multiplicative; i.e., $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$ implies $\mathbf{x}^{\alpha+\gamma} \prec \mathbf{x}^{\beta+\gamma}$ for all $\alpha, \beta, \gamma \in \mathbb{N}^n$;
- the constant monomial is the smallest; i.e., $1 \prec \mathbf{x}^\alpha$ for all $\alpha \in \mathbb{N}^n$.

A typical example of a such monomial ordering is the pure lexicographical ordering, denoted by \prec_{lex} . For two monomials $\mathbf{x}^\alpha, \mathbf{x}^\beta \in R$ $\mathbf{x}^\alpha \prec_{lex} \mathbf{x}^\beta$ if the left most nonzero entry of $\beta - \alpha$ is positive.

Let $I = \langle f_1, \dots, f_k \rangle$ be the ideal of R generated by the polynomials $f_1, \dots, f_k \in R$. Also let $f \in R$ and \prec be a monomial ordering on R . The *leading monomial* of f is the greatest monomial (with respect to \prec) appearing in f , and we denote it by $\text{LM}(f)$. If $F \subset R$ is a set of polynomials, we denote by $\text{LM}(F)$ the set $\{\text{LM}(f) \mid f \in F\}$. The *leading coefficient* of f , denoted by $\text{LC}(f)$, is the coefficient of $\text{LM}(f)$. The *leading term* of f is $\text{LT}(f) = \text{LC}(f)\text{LM}(f)$. The *leading term ideal* of I is defined to be

$$\text{LT}(I) = \langle \text{LT}(f) \mid f \in I \rangle.$$

A finite set $G = \{g_1, \dots, g_k\} \subset I$ is called a *Gröbner basis* of I w.r.t. \prec if $\text{LT}(I) = \langle \text{LT}(g_1), \dots, \text{LT}(g_k) \rangle$. For more details, we refer to [13], pages 213–214.

We recall below the definition of a special kind of involutive bases, namely, *Janet basis*. For this purpose, we describe first the notion of an *involutive division* [10] which is a restricted monomial division [1] that, together with a monomial ordering, determines the properties and the structure of an involutive basis. This makes the main difference between involutive bases and Gröbner bases. The idea is to partition the variables into two subsets of multiplicative and non-multiplicative variables, and only the multiplicative variables can be used in the divisibility relation.

Definition 1 (see [10]). An involutive division \mathcal{L} on the set of monomials of R is given, if for any finite set U of monomials and any $u \in U$, the set of variables is partitioned into the subsets of multiplicative $M_{\mathcal{L}}(u, U)$ and non-multiplicative $NM_{\mathcal{L}}(u, U)$ variables, and the following three conditions hold:

- $u, v \in U$, $u\mathcal{L}(u, U) \cap v\mathcal{L}(v, U) \neq \emptyset \implies u \in v\mathcal{L}(v, U)$ or $v \in u\mathcal{L}(u, U)$

- $v \in U, v \in u\mathcal{L}(u, U) \implies \mathcal{L}(v, U) \subset \mathcal{L}(u, U)$
- $u \in V$ and $V \subset U \implies \mathcal{L}(u, U) \subset \mathcal{L}(u, V)$

where $\mathcal{L}(u, U)$ denotes the monoid generated by the variables in $M_{\mathcal{L}}(u, U)$. If $v \in u\mathcal{L}(u, U)$ then u is called \mathcal{L} –(involutive) divisor of v and the involutive divisibility relation is denoted by $u|_{\mathcal{L}}v$. If v has no involutive divisors in a set U , then it is called \mathcal{L} –irreducible modulo U .

There are involutive divisions based on the classical partitions of variables suggested by Janet [5] and Thomas [7]. In this paper, we are interested in Janet division [10] only.

Example 1 (Janet division). Let U be a finite set of monomials and $u \in U$. The variable x_1 is multiplicative for u , if u has the maximum degree of x_1 in U . For $i > 1$, the variable x_i is multiplicative for u , if it has the maximum degree of x_i in the set $\{v \in U \mid \deg_j(v) = \deg_j(u), 1 \leq j < i\}$, where $\deg_j(v)$ denotes the degree of x_j in a monomial v .

Proposition 1 (see [10]). Janet division satisfies the conditions in Definition 1 and is noetherian.

Throughout this paper \mathcal{L} is assumed to be the Janet division. Now, we define an involutive basis.

Definition 2 (see [1]). Let $I \subset R$ be an ideal and \prec a monomial ordering on R . Let \mathcal{L} be an involutive division. A finite set $G \subset I$ is an (\mathcal{L} -)involutive basis for I if for all $f \in I$ there exists $g \in G$ such that $\text{LM}(g)|_{\mathcal{L}}\text{LM}(f)$.

From this definition and from that for Gröbner basis [2, 13] it follows that an involutive basis for an ideal is its Gröbner basis, but the converse is not always true.

Remark 1. By using an involutive division in the conventional division algorithm for polynomial rings, we obtain an involutive division algorithm. If F is a finite polynomial set and \mathcal{L} is an involutive division, then we use $\text{NF}_{\mathcal{L}}(f, F)$ to denote the remainder of f on involutive division by F .

The following theorem provides an algorithmic characterization of involutive basis for a given ideal which is an involutive analogue of the Buchberger characterization of Gröbner basis.

Theorem 1 (see [10]). Let $I \subset R$ be an ideal, \prec a monomial ordering on R , and \mathcal{L} an involutive division. Then a finite subset $G \subset I$ is an involutive basis for I if for each $f \in G$ and each $x \in \text{NM}_{\mathcal{L}}(\text{LM}(f), \text{LM}(G))$, we have $\text{NF}_{\mathcal{L}}(xf, G) = 0$.

Based on this theorem, one can design an algorithm to compute involutive bases. We recall here the `INVBASIS` algorithm from [10] (see also [1]) which computes a *minimal* involutive basis for an ideal. An involutive basis G is called minimal if for any other involutive basis \tilde{G} the inclusion $\text{LM}(G) \subseteq \text{LM}(\tilde{G})$ holds. A minimal involutive basis exists and being monic and *involutively autoreduced*, i.e. satisfying

$$(\forall g \in G) [g = \text{NF}_{\mathcal{L}}(g, G \setminus \{g\})]$$

is *unique* for a given ideal, a monomial ordering and a *constructive* involutive division (see [10] for the definition of constructivity).

Here, as it has been mentioned in [1], in comparison to the algorithm in [10] another selection strategy is used. By this strategy, a polynomial whose leading monomial has no proper divisor is chosen. However, this algorithm is not efficient in practice, it processes the repeated prolongations and does not use any criterion to avoid the unnecessary reductions. For this purpose, we consider in the next section, the Gerdt's algorithm from [1] as an improvement of `INVBASIS`.

Algorithm 1 INVBASIS

```

inputs
   $F$ , a set of polynomials;  $\mathcal{L}$ , an involutive division;  $\prec$ , a monomial ordering.
outputs
  a minimal involutive basis for  $\langle F \rangle$  w.r.t  $\mathcal{L}$  and  $\prec$ .
  Select  $f \in F$  with no proper divisor of  $\text{LM}(f)$  in  $\text{LM}(F)$ 
   $G := \{f\}$ ;
   $Q := F \setminus G$ ;
  while  $Q \neq \emptyset$  do
    Select and remove  $p \in Q$  with no proper divisor of  $\text{LM}(p)$  in  $\text{LM}(Q)$ ;
     $h := \text{NF}_{\mathcal{L}}(p, G)$ ;
    if  $h \neq 0$  then
      for  $g \in G$  which  $\text{LM}(g)$  is properly divisible by  $\text{LM}(h)$  do
         $Q := Q \cup \{g\}$ ;
         $G := G \setminus \{g\}$ ;
      end for
       $G := G \cup \{h\}$ ;
       $Q := Q \cup \{xg \mid g \in G, x \in \text{NM}_{\mathcal{L}}(\text{LM}(g), \text{LM}(G))\}$ ;
    end if
  end while
Return  $(G)$ 

```

3. Gerdt's algorithm

The Gerdt's algorithm [1] improves the above algorithm INVBASIS. It uses the involutive form of Buchberger's criteria to avoid unnecessary reductions and avoids the repeated processing of non-multiplicative prolongations. In order to explain the structure of this algorithm, we recall some more definitions and notations.

Definition 3 (see [1]). An *ancestor* of a polynomial $f \in F \subset R \setminus \{0\}$, denoted by $\text{anc}(f)$, is a polynomial $g \in F$ of the smallest $\deg(\text{LM}(g))$ among those satisfying $\text{LM}(f) = u\text{LM}(g)$ where u is either the unit monomial or a power product of non-multiplicative variables for $\text{LM}(g)$ and such that $\text{NF}_{\mathcal{L}}(f - ug, F \setminus \{f\}) = 0$ if $f \neq ug$.

This additional information for an element in a polynomial set is useful to avoid unnecessary reductions specially by applying the adapted Buchberger's criteria (see below).

Proposition 2 (see [11]). Let $I \subset R$ be an ideal and $G \subset I$ be a finite set. Let also \prec be a monomial ordering on R and \mathcal{L} an involutive division. Then G is an \mathcal{L} -involutive basis for I if for all $f \in G$ one of the following conditions holds (we use " \sqsubset " to denote proper conventional division):

- 1) for all $x \in \text{NM}_{\mathcal{L}}(\text{LM}(f), \text{LM}(G))$ the equality $\text{NF}_{\mathcal{L}}(xf, G) = 0$ holds;
- 2) there exists $g \in G$ with $\text{LM}(g) \mid_{\mathcal{L}} \text{LM}(f)$ satisfying one of the following:
 - (C_1) $\text{LM}(\text{anc}(f))\text{LM}(\text{anc}(g)) = \text{LM}(f)$;
 - (C_2) $\text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(g))) \sqsubset \text{LM}(f)$;
 - (C_3) there exists $t \in G$ such that:
 - $\text{lcm}(\text{LM}(t), \text{LM}(f)) \sqsubset \text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(g)))$;
 - $\text{lcm}(\text{LM}(t), \text{LM}(g)) \sqsubset \text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(g)))$;
 - (C_4) there exists $t \in G$ computed before $\text{anc}(f)$ and $y \in \text{NM}_{\mathcal{L}}(\text{LM}(t), \text{LM}(G))$ s.t.
 - $y\text{LM}(t) = \text{LM}(f)$;
 - $\text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(t))) \sqsubset \text{LM}(f)$.

Now we can present Gerdt's algorithm for computing involutive bases in which we associate to each polynomial f , the triple $p = \{f, g, V\}$ where $f = \text{poly}(p)$ is the polynomial itself, $g = \text{anc}(p)$ is its ancestor and $V = \text{NM}(p)$ is the list of non-

multiplicative variables of f which have been already processed in the algorithm. If P is a set of triples, we denote by $\text{poly}(P)$ the set $\{\text{poly}(p) \mid p \in P\}$. If no confusion arises, we may refer to a triple p instead of $\text{poly}(p)$, and vice versa.

Algorithm 2 GERDT

inputs

F , a set of polynomials; \mathcal{L} , an involutive division; \prec , a monomial ordering

outputs

a minimal involutive basis for $\langle F \rangle$ w.r.t \mathcal{L} and \prec

Select $f \in F$ with no proper divisor of $\text{LM}(f)$ in $\text{LM}(F)$

$T := \{\{f, f, \emptyset\}\};$

$Q := \{\{q, q, \emptyset\} \mid q \in F \setminus \{f\}\};$

$Q := \text{HEADREDUCE}(Q, T, \mathcal{L}, \prec);$

while $Q \neq \emptyset$ **do**

Select and remove $p \in Q$ with no proper divisor of $\text{LM}(\text{poly}(p))$ in $\text{LM}(\text{poly}(Q));$

if $\text{poly}(p) = \text{anc}(p)$ **then**

for $q \in T$ with $\text{LM}(\text{poly}(p)) \sqsubset \text{LM}(\text{poly}(q))$ **do**

$Q := Q \cup \{q\};$

$T := T \setminus \{q\};$

end for

end if

$h := \text{TAINORMALFORM}(p, T, \mathcal{L}, \prec)$

$T := T \cup \{\{h, \text{anc}(p), \text{NM}(p)\}\};$

for $q \in T$ and $x \in \text{NM}_{\mathcal{L}}(\text{LM}(\text{poly}(q)), \text{LM}(\text{poly}(T))) \setminus \text{NM}(q)$ **do**

$Q := Q \cup \{\{x \text{ poly}(q), \text{anc}(q), \emptyset\}\};$

$\text{NM}(q) := \text{NM}(q) \cap \text{NM}_{\mathcal{L}}(\text{LM}(\text{poly}(q)), \text{LM}(\text{poly}(T))) \cup \{x\};$

end for

$Q := \text{HEADREDUCE}(Q, T, \mathcal{L}, \prec);$

end while

Return $(\text{poly}(T))$

This algorithm includes three subalgorithms HEADREDUCE, HEADNORMALFORM and TAINORMALFORM which we present below.

Algorithm 3 HEADREDUCE

inputs

Q and T , sets of triples; \mathcal{L} , an involutive division; \prec , a monomial ordering

outputs

The Q of triples whose polynomials are \mathcal{L} -head reduced modulo T

$S := Q$

$Q := \emptyset;$

while $S \neq \emptyset$ **do**

Select and remove $p \in S$

$h := \text{HEADNORMALFORM}(p, T, \mathcal{L});$

if $h \neq 0$ **then**

if $\text{LM}(\text{poly}(p)) \neq \text{LM}(h)$ **then**

$Q := Q \cup \{\{h, h, \emptyset\}\};$

else

$Q := Q \cup \{p\};$

end if

else

if $\text{LM}(\text{poly}(p)) = \text{LM}(\text{anc}(p))$ **then**

$S := S \setminus \{q \in S \mid \text{anc}(q) = \text{poly}(p)\};$

end if

end if

end while

Return (Q)

In the below subalgorithm HEADNORMALFORM the Boolean expression Criteria(p, g) is true if at least one of the four criteria in Proposition 2 holds for p and g .

Algorithm 4 HEADNORMALFORM

inputs
 T , a set of triples; p , a triple; \mathcal{L} , an involutive division; \prec , a monomial ordering

outputs
 \mathcal{L} -head normal form of $\text{poly}(p)$ modulo T

$h := \text{poly}(p)$;
 $G := \text{poly}(T)$;
if $\text{LM}(h)$ is \mathcal{L} -irreducible modulo G **then**
 Return (h)
else
 Select $g \in G$ with $\text{LM}(\text{poly}(g))|_{\mathcal{L}}\text{LM}(h)$;
 if $\text{LM}(h) \neq \text{LM}(\text{anc}(p))$ **then**
 if Criteria(p, g) **then**
 Return (0)
 end if
 else
 while $h \neq 0$ and $\text{LM}(h)$ is \mathcal{L} -reducible modulo G **do**
 Select $g \in G$ with $\text{LM}(g)|_{\mathcal{L}}\text{LM}(h)$;
 $h := h - g \frac{\text{LT}(h)}{\text{LT}(g)}$;
 end while
 end if
end if
Return (h)

Algorithm 5 TAILNORMALFORM

inputs
 T , a set of triples; p , a triple, which is \mathcal{L} -head reduced modulo T ;
 \mathcal{L} , an involutive division; \prec , a monomial ordering

outputs
 \mathcal{L} -Normal form of $\text{poly}(p)$ modulo T

$h := \text{poly}(p)$;
 $G := \text{poly}(T)$;
while h has a term t which is \mathcal{L} -reducible modulo G **do**
 Select $g \in G$ with $\text{LM}(g)|_{\mathcal{L}}t$;
 $h := h - g \frac{t}{\text{LT}(g)}$;
end while
Return (h)

4. Modified algorithm

In this section, we present the following variant of Gerdt's algorithm for computing involutive bases. This variant seems to be simpler both to understand and to implement. We also present here a proof of its correctness and termination including the correctness proof for the used criteria.

Algorithm 6 VARGERDT

```

inputs
   $F$ , a set of polynomials;  $\mathcal{L}$ , an involutive division;  $\prec$ , a monomial ordering
outputs
  a minimal  $\mathcal{L}$ -involutive basis for  $\langle F \rangle$ 
  Select  $f \in F$  with no proper divisor of  $\text{LM}(f)$  in  $\text{LM}(F)$ 
   $T := \{\{f, f, \emptyset\}\}$ ;
   $Q := \{\{q, q, \emptyset\} \mid q \in F \setminus \{f\}\}$ ;
  while  $Q \neq \emptyset$  do
    Select and remove  $p \in Q$  with no proper divisor of  $\text{LM}(\text{poly}(p))$  in  $\text{LM}(\text{poly}(Q))$ ;
     $h := \text{NORMALFORM}(p, T, \mathcal{L}, \prec)$ ;
    if  $h = 0$  and  $\text{LM}(\text{poly}(p)) = \text{LM}(\text{anc}(p))$  then
       $Q := \{q \in Q \mid \text{anc}(q) \neq \text{poly}(p)\}$ ;
    else
      if  $\text{LM}(\text{poly}(p)) \neq \text{LM}(h)$  then
        for  $q \in T$  with  $\text{LM}(\text{poly}(h)) \sqsubset \text{LM}(\text{poly}(q))$  do
           $Q := Q \cup \{q\}$ ;
           $T := T \setminus \{q\}$ ;
        end for
         $T := T \cup \{\{h, h, \emptyset\}\}$ ;
      else
         $T := T \cup \{\{h, \text{anc}(p), \text{NM}(p)\}\}$ ;
      end if
      for  $q \in T$  and  $x \in \text{NM}_{\mathcal{L}}(\text{LM}(\text{poly}(q)), \text{LM}(\text{poly}(T))) \setminus \text{NM}(q)$  do
         $Q := Q \cup \{\{x \text{ poly}(q), \text{anc}(q), \emptyset\}\}$ ;
         $\text{NM}(q) := \text{NM}(q) \cap \text{NM}_{\mathcal{L}}(\text{LM}(\text{poly}(q)), \text{LM}(\text{poly}(T))) \cup \{x\}$ ;
      end for
    end if
  end while
  Return  $(\text{poly}(T))$ 

```

Algorithm 7 NORMALFORM

```

inputs
   $p$ , a triple;  $T$ , a set of triples;  $\mathcal{L}$ , an involutive division;  $\prec$ , a monomial ordering
outputs
   $\mathcal{L}$ -Normal form of  $p$  modulo  $T$ 
   $h := \text{poly}(p)$ ;
   $G := \text{poly}(T)$ ;
  while  $h$  has a term  $t$  which is  $\mathcal{L}$ -reducible modulo  $G$  do
    Select  $g \in G$  with  $\text{LM}(g) \mid_{\mathcal{L}} t$ ;
    if  $t = \text{LT}(\text{poly}(p))$  and  $\text{Criteria}(h, g)$  then
      Return  $(0)$ 
    else
       $h := h - g \frac{t}{\text{LT}(g)}$ ;
    end if
  end while
  Return  $(h)$ 

```

The subalgorithm NORMALFORM returns the \mathcal{L} -normal form of a polynomial w.r.t. a given set of polynomials (see below). Moreover, this subalgorithm detects some unnecessary reductions using the involutive form of Buchberger's criteria. It is worth noting that Gerdt and Yanovich [14] have shown that the use of C_3 and C_4 criteria (see Proposition 2) may notably slow down the computation of involutive bases. That is why, we use only C_1 and C_2 criteria. Below, we give a simple proof of correctness for these criteria.

Lemma 1. *Let $I \subset R$ be an ideal, \prec a monomial ordering on R and \mathcal{L} an involutive division. Let $T \subset I$ be the last computed basis during a computation of an involutive basis for I , and $f \in I$. Then $\text{NF}_{\mathcal{L}}(f, T) = 0$ if there exists $q \in T$ with $\text{LM}(q)|_{\mathcal{L}}\text{LM}(f)$ satisfying one of the following conditions:*

- (C_1) $\text{LM}(\text{anc}(f))\text{LM}(\text{anc}(q)) = \text{LM}(f)$
 (C_2) $\text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(q))) \sqsubset \text{LM}(f)$.

Proof. Suppose that (f, q) satisfies C_1 . Then, for $\text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(q)))$ two cases are possible: If it is a proper divisor of $\text{LM}(\text{anc}(f))\text{LM}(\text{anc}(q)) = \text{LM}(f)$, there exists a monomial $s \neq 1$ such that $\text{LM}(f) = \text{slcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(q)))$. Let $f = u \cdot \text{anc}(f)$, $q = v \cdot \text{anc}(q)$ and $\text{LT}(f) = t \cdot \text{LT}(q)$ for some monomials u and v and some term t . Thus, we can write $f - tq = u \cdot \text{anc}(f) - tv \cdot \text{anc}(q) = s(u' \cdot \text{anc}(f) - v' \cdot \text{anc}(q))$ where $u = su'$ and $vt = sv'$ for some monomials u', v' . Since, $\text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(q)))$ is a proper divisor of $\text{LM}(f)$, then $u' \cdot \text{anc}(f) - v' \cdot \text{anc}(q)$ has been computed before f (see the selection strategy for choosing polynomials in the algorithm), and therefore, it has a standard representation w.r.t. T . This implies that $f - tq$ has also a standard representation w.r.t. T , and $\text{NF}_{\mathcal{L}}(f, T) = 0$. As the second case, if $\text{lcm}(\text{LM}(\text{anc}(f)), \text{LM}(\text{anc}(q)))$ is equal to $\text{LM}(\text{anc}(f))\text{LM}(\text{anc}(q))$, then by the Buchberger's first criterion, we can conclude that $u' \cdot \text{anc}(f) - v' \cdot \text{anc}(q)$ has a standard representation w.r.t. T , and this proves the assertion like the first case.

Now, if (f, q) satisfies C_2 , the proof is similar to that for the above first case. \square

Just as above, for two polynomials p and q , the Boolean expression $\text{Criteria}(p, q)$ is true if they satisfy at least one of the criteria C_1 or C_2 , and false otherwise.

Theorem 2. *The VARGERDT algorithm computes a minimal involutive basis for the ideal generated by the input polynomial set.*

Proof. Let $F \subset R$ be a finite set of polynomials taken as the input of VARGERDT algorithm. Let \prec be a monomial ordering on R and \mathcal{L} be a constructive noetherian involutive division [10]. Let G be a polynomial set that is the output of the algorithm for F . Since (by the structure of the algorithm) for each $f \in G$ and each $x \in \text{NM}_{\mathcal{L}}(\text{LM}(f), \text{LM}(G))$, we examine xf (we either compute its \mathcal{L} -normal form w.r.t. G or we discard it by the criteria), it follows that $\text{NF}_{\mathcal{L}}(xf, G) = 0$. Therefore, G is an involutive basis for $\langle F \rangle$ by Theorem 1. Moreover, this basis is minimal by the definition of this concept due to the structure of the VARGERDT algorithm and constructivity of \mathcal{L} .

The termination of VARGERDT algorithm follows from the noetherianity of \mathcal{L} , see Proposition 1. \square

Remark 2. The algorithm VARGERDT as well as Gerdt's algorithm not necessarily outputs an involutively tail autoreduced basis. If the output is G , then, in accordance to the definition that follows Theorem 6 of Section 2, the tail autoreduction is provided by the command

$$(\forall g \in G) [g := \text{NF}_{\mathcal{L}}(g, G \setminus \{g\})].$$

5. Experiments and Results

We have implemented both algorithms VARGERDT and GERDT in `Maple 14`¹. For an efficient implementation of GERDT algorithm, we refer to [15]. It is worth noting that, in this paper, we are willing to compare the structure and behavior of VARGERDT and GERDT algorithms on the same platform. Therefore, we do not compare our implementations with [15].

¹The `Maple` code of our programs and examples is available on the Web page <http://invo.jinr.ru/>

To compare the behavior of these algorithms, we used some well-known examples from the the collection [16]. These examples have been widely used for verification and comparison of different software packages for construction of Grobner bases. In our benchmarking we applied the following selection strategy for the polynomials in Q . An element at the initialization step to be inserted in set Q is that with has the lowest leading monomial with respect to the degree-reverse-lexicographic ordering. The same strategy is used for selected elements from Q in the main loop. In the case of several such polynomials, that one is selected whose ancestor has been examined earliest.

The results are shown in the following tables where computation was performed on a personal computer with 2.33 GHz, 2×Intel(R) Xeon(R) Quad core, 16 GB RAM and 64 bites under the Linux operating system. The computation was done over \mathbb{Q} and the monomial ordering is always the degree-reverse-lexicographical one.

The time (resp. memo., reds., C_1 and C_2) column shows the consumed CPU time in seconds (resp. amount of megabytes of memory used, number of reductions to zero, the number of polynomials removed by C_1 and C_2 criteria) by the corresponding algorithm. The seventh column indicates the number of polynomials eliminated by *Rewritten* criterion: in VARGERDT algorithm, as well as in GERDT algorithm, when the polynomial part of a triple $\{f, g, V\}$ reduces to zero, and $f = g$, then we can discard any polynomial whose ancestor is f . We call this criterion the *Rewritten* criterion. The last column shows the largest degree of intermediate polynomials treated during the computation of involutive bases.

A comparison of the timing columns in the above tables and our test for some other examples shows that VARGERDT is more efficient and stable than GERDT. Indeed, the main difference between these algorithms is that in GERDT algorithm, we do first the head reduction of all non-multiplicative prolongations (which have not been examined yet) and then we choose one of them to complete its involutive reduction to the full normal form, while in VARGERDT we choose a polynomial (among the non-multiplicative prolongations), and we compute its full normal form.

In Gerdt's algorithm the head involutive reduction of the whole set of non-multiplicative prolongation is done in order to provide a platform for matching a good selection strategy. We refer to paper [17] for details on heuristically good selection strategies for the Janet division. The choice of such selection strategy as well as the use of proper data structures for fast search of involutive divisor [1] plays a key role in providing a high computational efficiency of the involutive algorithms. In this paper we do not consider these important aspects of the practical construction of involutive bases and compare two algorithms experimentally for their simplest and identical for the both algorithms implementation.

Acknowledgements.

The research presented in the paper was completed during the stay of the second author (A. Hashemi) at the Joint Institute for Nuclear Research in Dubna, Russia. He would like to thank Professor V. P. Gerdt for the invitation, hospitality and support. The contribution of the first author (V. P. Gerdt) was partially supported by the grant 01-01-00200 from the Russian Foundation for Basic Research and by the grant 3810.2010.2 from the Ministry of Education and Science of the Russian Federation.

References

1. *Gerdt V. P.* Involutive Algorithms for Computing Gröbner Bases // Computational Commutative and Non-Commutative Algebraic Geometry. — Amsterdam: IOS, 2005. — Vol. 196 of *NATO Sci. Ser. III Comput. Syst. Sci.* — Pp. 199–225.
2. *Buchberger B.* Ein Algorithms zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal: Ph.D. thesis / Universität Innsbruck. — 1965.

Table 1

Noon4	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	14.29	819.2	56	6	19	0	10
GERDT	14.92	821.4	50	6	19	0	9

Cyclic5	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	12.15	701.8	77	36	9	18	9
GERDT	12.80	686.6	83	40	5	0	8

Sturmfels	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	64.29	4152.6	91	43	212	0	6
GERDT	80.72	5103.7	80	42	218	0	5

Katsura5	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	97.18	9636.8	47	22	1	0	12
GERDT	54.50	4731.8	47	22	1	0	10

Eco7	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	102.62	6066.8	132	51	48	20	6
GERDT	80.21	4249.7	124	46	40	0	4

Liu	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	2.65	155.1	18	6	3	0	0
GERDT	3.81	236.0	18	6	3	0	0

Lichtblau	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	111.21	22302.9	25	0	9	1	11
GERDT	> 8 hours	?	?	?	?	?	?

Katsura6	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	213.59	16124.3	128	44	3	0	8
GERDT	4227.32	1632783.6	128	44	3	0	7

Cyclic6	time	memo.	reds.	C_1	C_2	R	deg.
VARGERDT	1313.88	222625.1	542	169	7	47	11
GERDT	2571.84	655916.5	476	152	18	0	10

3. *Buchberger B.* A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner-bases // Symbolic and algebraic computation (EUROSAM '79, Internat. Sympos., Marseille, 1979). — Berlin: Springer, 1979. — Vol. 72 of *Lecture Notes in Comput. Sci.* — Pp. 3–21.
4. Gröbner Bases and Applications / Ed. by B. Buchberger, F. Winkler. — Cambridge: Cambridge University Press, 1998. — Vol. 251 of *London Mathematical Society Lecture Note Series.* — ISBN 0-521-63298-6, Pp. viii+552. — Papers from the Conference on 33 Years of Gröbner Bases held at the University of Linz, Linz, February 2–4, 1998.
5. *Janet M.* Les systèmes d'équations aux dérivées partielles // *J. Math. Pures Appl.* — 1920. — Vol. 3. — Pp. 65–151. — ISSN 0021-7824.
6. *Seiler W. M.* Involution. — Berlin: Springer-Verlag, 2010. — Vol. 24 of *Algorithms and Computation in Mathematics.* — ISBN 978-3-642-01286-0, Pp. xxii+650. — <http://dx.doi.org/10.1007/978-3-642-01287-7>. — The formal theory of differential equations and its applications in computer algebra.
7. *Thomas J.* Differential Systems. — New York: American Mathematical Society, 1937. — Pp. xx+363.
8. *Pommaret J.-F.* Systems of Partial Differential Equations and Lie Pseudogroups. — New York: Gordon & Breach Science Publishers, 1978. — Vol. 14 of *Mathematics and its Applications.* — ISBN 0-677-00270-*, Pp. xiv+411. — With a preface by André Lichnerowicz.
9. *Zharkov A. Y., Blinkov Y. A.* Involution Approach to Investigating Polynomial Systems // *Math. Comput. Simulation.* — 1996. — Vol. 42, No 4-6. — Pp. 323–332. — ISSN 0378-4754. — [http://dx.doi.org/10.1016/S0378-4754\(96\)00006-7](http://dx.doi.org/10.1016/S0378-4754(96)00006-7). — Symbolic computation, new trends and developments (Lille, 1993).
10. *Gerdt V. P., Blinkov Y. A.* Involutive Bases of Polynomial Ideals // *Math. Comput. Simulation.* — 1998. — Vol. 45, No 5-6. — Pp. 519–541. — ISSN 0378-4754. — [http://dx.doi.org/10.1016/S0378-4754\(97\)00127-4](http://dx.doi.org/10.1016/S0378-4754(97)00127-4). — Simplification of systems of algebraic and differential equations with applications.
11. *Apel J., Hemmecke R.* Detecting Unnecessary Reductions in an Involutive Basis Computation // *J. Symbolic Comput.* — 2005. — Vol. 40, No 4-5. — Pp. 1131–1149. — ISSN 0747-7171. — <http://dx.doi.org/10.1016/j.jsc.2004.04.004>.
12. *Gerdt V. P.* On an Algorithmic Optimization in the Computation of Involutive Bases // *Programming and Computer Software.* — 2002. — Vol. 28, No 2. — Pp. 62–65. — ISSN 0132-3474. — <http://dx.doi.org/10.1023/A:1014816631983>.
13. *Becker T., Weispfenning V.* Gröbner Bases. — New York: Springer-Verlag, 1993. — Vol. 141 of *Graduate Texts in Mathematics.* — ISBN 0-387-97971-9, Pp. xxii+574. — A computational approach to commutative algebra, In cooperation with Heinz Kredel.
14. *Gerdt V. P., Yanovich D. A.* Effectiveness of Involutive Criteria in Computation of Polynomial Janet Bases // *Programming and Computer Software.* — 2006. — Vol. 32, No 3. — Pp. 134–138. — ISSN 0132-3474. — <http://dx.doi.org/10.1134/S0361768806030030>.
15. The Maple package "Janet": I. Polynomial Systems and II. Linear Partial Differential Equations / Y. A. Blinkov, V. P. Gerdt, C. F. Cid et al. // *Computer Algebra in Scientific Computing (CASC 2003)*, V. G. Ganzha, E. W. Mayr, and E. V. Vorozhtsov, eds. — Institute of Informatics, Technical University of Munich, Garching, 2003. — Pp. 31–54.
16. *Bini D., Mourrain B.* Polynomial Test Suite. — <http://www-sop.inria.fr/saga/POL/>.
17. *Gerdt V. P., Blinkov Y. A.* On Selection of Nonmultiplicative Prolongations in Computation of Janet Bases // *Programming and Computer Software.* — 2007. — Vol. 33, No 3. — Pp. 147–153. — ISSN 0132-3474. — <http://dx.doi.org/10.1134/S0361768807030048>.

УДК 512.7+519

Вариант алгоритма Гердта для вычисления инволютивных базисов**В. П. Гердт^{*}, А. Хашеми[†], Б. М.-Ализадех[†]**

^{} Лаборатория информационных технологий
Объединённый институт ядерных исследований
ул. Жолио-Кюри д.6, Дубна, Московская область, 141980, Россия*

*[†] Отделение математических наук
Исфаханский технологический университет
84156-83111, Исфахан, Иран*

В работе [1] первого автора был представлен эффективный алгоритм вычисления инволютивных базисов и приведенных базисов Гребнера. В данной работе мы рассмотрим модификацию указанного алгоритма, которая упрощает его понимание и облегчает программную реализацию. Мы докажем корректность модифицированного алгоритма и используемых вместе с использованными в нем критериями и его оканчиваемость. Предложенный алгоритм реализован на языке Maple. Путем вычислительных экспериментов с рядом полиномиальных систем мы сопоставим времена счета модифицированного алгоритма с его оригинальной формой, описанной в [1] и также реализованной на языке Maple. При этом детали оба алгоритма были реализованы схожим образом, чтобы обеспечить корректность сравнения их вычислительной эффективности.

Ключевые слова: базисы Гребнера, критерии Бухбергера, инволютивные базисы, алгоритм Гердта.