# Computer Algebra (2018-2019) Exercises 2
## Computing with algebraic extensions

**Deadline:** Monday 4 March, 4.00 pm.
**Total practical credit contribution:** 40%.

The exercises given here introduce you to a method of exact computation with expressions involving the roots of polynomials by means of an algebraic construction. Understanding the ideas introduced here is as much part of the exercises as the implementation (which involves a fairly small amount of code thanks to the powerful facilities offered by Axiom). As discussed in the introductory lecture, this exercise will show the power of the mathematical structures discussed.

The programming part requires you to write *simple* Axiom code; a list of built-in Axiom functions that will help you is given. This list is a suggestion and you might use other functions or only some of the ones mentioned; at any rate you should read the online documentation of any function that you do use.

As before, the code is given a number of lines of well laid out code (comments are not included but you must supply sensible comments for each function or face a penalty). The meaning of this is that an implementation has been carried out in that many lines. No great ingenuity was required for this. If you exceed the number of lines by more than 25% then you will lose half of your marks for the particular part. However the intention is not to scare you but rather to give you further guidance. Precede each of your functions with a description of the required input, the output and any assumptions made. For an example with comments, see Exercises 1. Your code must be clear, correct and reasonably efficient (which is guaranteed by the natural way of doing things).

There are also some pencil and paper exercises. These provide good practice for the kind of Mathematical reasoning with which you need to become familiar. There is no need to typeset your answers to these, clear handwritten answers are perfectly fine.

Remember that all supporting materials for this course can be found at:

http://www.inf.ed.ac.uk/teaching/courses/ca/

**Submission:** Submit your handwritten answers to the pencil and paper exercise as well as a printout of your implementation (I need the paper copy of your code in order to write comments to give you adequate feedback). Make sure you staple all your answer sheets in the top left hand corner. You must also submit your code electronically. Please put all your code in a file called `common.input`. Follow the instructions on the course web page on where to submit.

**Good Scholarly Practice:** Please remember the University requirement as regards all assessed work for credit. Details and advice about this can be found at:

http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct

and links from there. Note that, in particular, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).

**§1. Modular Arithmetic with Polynomials.** Consider a field $k$, an indeterminate $x$ over $k$ and a non-zero polynomial $p \in k[x]$. Recall from Exercises 1 that we can construct the ring $k[x]/(p)$ in which calculations are done modulo $p$. The analogy between this and the integer case $\mathbb{Z}_n$ is very close as the following exercise shows.

**Exercise 1.1** *Recall that a polynomial $p$ is said to be irreducible[1] in $k[x]$ if it is not a constant and whenever $p = uv$ then either $u$ or $v$ is a constant. Prove that $k[x]/(p)$ is a field if and only if $p$ is irreducible in $k[x]$. You may assume without proof that $k[x]/(p)$ is a commutative ring with identity. (Hint: the proof is analogous to the one that $\mathbb{Z}_n$ is a field if and only if $n$ is a prime.)* [*10 marks*]

*Assuming that $p$ is irreducible, comment briefly on how your proof suggests an algorithm for finding the multiplicative inverse of a non-zero element of $k[x]/(p)$. For the purpose of this part assume that we have an implementation of the basic operations on elements of $k$. (If this part is not completely straightforward then it is more than likely that your proof is wrong, you are just asked to make a simple observation.)* [*2 marks*]

**Note:** Take care to express things clearly, do not confuse members of $k[x]$ with members of $k[x]/(p)$. For example $x^3 + 2x - 1$ as an element of $k[x]$ has a well defined degree (it is 3) but as an element of $k[x]/(x^2 + 1)$ it is meaningless to talk about its "degree" since in fact it denotes an equivalence class of polynomials from $k[x]$. The proof part falls naturally into two tasks: (i) if $p$ is not irreducible then $k[x]/(p)$ is not a field (look at the proof for $Z_n$) and (ii) if $p$ is irreducible then $k[x]/(p)$ is indeed a field (remember that you may assume that $k[x]/(p)$ is always a commutative ring with identity, so what extra properties do you need to establish?). There will be a heavy penalty for answers that ignore this advice. Finally, note that the parentheses around $p$ in $k[x]/(p)$ are *not* optional[2].

**§2. Algebraic Extensions.** Let $x$ be an indeterminate over a field $k$ and suppose that $p \in k[x]$. Exercise 1.1 shows that $k[x]/(p)$ is a field precisely when $p$ is irreducible in $k[x]$. It is fairly straightforward to see that $k$ is a subfield of $k[x]/(p)$; the latter field is called an algebraic extension of $k$. Moreover note that the polynomial $p$ is guaranteed to have a root in $k[x]/(p)$, namely $x$ when viewed as an element of this field. This is simply because $(p \bmod p) = 0$. Put this way we don't seem to have achieved much, however this is far from the case: the real point is that we have found a field that contains $k$ as a subfield and has a root for $p$. In this regard it is easier to appreciate the result if we view $p$ as a polynomial in a new indeterminate $y$ over $k$ which is also an indeterminate over $k[x]/(p)$; we could use any new name[3]. We are then saying that the polynomial $p(y)$ that is irreducible in $k[y]$ (and hence has no root in $k$ unless it is linear, i.e., it has degree 1) has a root in the field $k[x]/(p)$ and this field extends $k$. It follows that $p(y)$ factorizes in $k[x]/(p)[y]$; now if an irreducible factor is not linear (so that it does not have a root in $k[x]/(p)$) then we can construct a root for it by the same process. It follows that after finitely many steps we can construct a field $k'$ that contains $k$ as a subfield and is such that $p(y)$ has a full set of roots in $k'$. The example of the complex numbers given below shows that sometimes we can stop after just one extension (in fact there we not only get the roots of $x^2+1$ but the roots of all polynomials even ones with complex coefficients—this is not obvious!). For the rest of this section we fix $p$ and assume that it is irreducible in $k[x]$.

It is worthwhile looking at an example. Consider $1 + x^2 \in \mathbb{R}[x]$. The relevant extension is then $\mathbb{R}[x]/(1 + x^2)$. How do we add and multiply elements in this field? Every element can be written uniquely as

$$a_0 + a_1 x.$$

This is because every remainder modulo $1 + x^2$ has degree strictly less than 2 (i.e., at most 1) and two different polynomials of degree at most 1 correspond to different remainders (since $1 + x^2$ cannot divide their difference).

Addition is obvious

$$(a_0 + a_1 x) + (b_0 + b_1 x) = (a_0 + b_0) + (a_1 + b_1)x.$$

As for multiplication we have

$$(a_0 + a_1 x)(b_0 + b_1 x) = a_0 b_0 + (a_1 b_0 + a_0 b_1)x + a_1 b_1 x^2.$$

[1] Note that if a polynomial is not irreducible then we call it exactly that rather than 'reducible' as might be expected!

[2] Technically $(p)$ denotes the ideal generated by $p$ in $k[x]$ and $k[x]/(p)$ the quotient ring of $k[x]$ by $(p)$. We will discuss these ideas later on in the course.

[3] The point is that $x$ is not an indeterminate over $k[x]/(p)$. An alternative is to replace $x$ by a different indeterminate $a$, say, in the construction and form the field $k[a]/(p(a))$ which is naturally isomorphic to $k[x]/(p(x))$.

| $+$ | $0$ | $1$ | $2$ | $x$ | $x+1$ | $x+2$ | $2x$ | $2x+1$ | $2x+2$ |
|---|---|---|---|---|---|---|---|---|---|
| $0$ | $0$ | $1$ | $2$ | $x$ | $x+1$ | $x+2$ | $2x$ | $2x+1$ | $2x+2$ |
| $1$ | $1$ | $2$ | $0$ | $x+1$ | $x+2$ | $x$ | $2x+1$ | $2x+2$ | $2x$ |
| $2$ | $2$ | $0$ | $1$ | $x+2$ | $x$ | $x+1$ | $2x+2$ | $2x$ | $2x+1$ |
| $x$ | $x$ | $x+1$ | $x+2$ | $2x$ | $2x+1$ | $2x+2$ | $0$ | $1$ | $2$ |
| $x+1$ | $x+1$ | $x+2$ | $x$ | $2x+1$ | $2x+2$ | $2x$ | $1$ | $2$ | $0$ |
| $x+2$ | $x+2$ | $x$ | $x+1$ | $2x+2$ | $2x$ | $2x+1$ | $2$ | $0$ | $1$ |
| $2x$ | $2x$ | $2x+1$ | $2x+2$ | $0$ | $1$ | $2$ | $x$ | $x+1$ | $x+2$ |
| $2x+1$ | $2x+1$ | $2x+2$ | $2x$ | $1$ | $2$ | $0$ | $x+1$ | $x+2$ | $x$ |
| $2x+2$ | $2x+2$ | $2x$ | $2x+1$ | $2$ | $0$ | $1$ | $x+2$ | $x$ | $x+1$ |

| $*$ | $0$ | $1$ | $2$ | $x$ | $x+1$ | $x+2$ | $2x$ | $2x+1$ | $2x+2$ |
|---|---|---|---|---|---|---|---|---|---|
| $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $1$ | $0$ | $1$ | $2$ | $x$ | $x+1$ | $x+2$ | $2x$ | $2x+1$ | $2x+2$ |
| $2$ | $0$ | $2$ | $1$ | $2x$ | $2x+2$ | $2x+1$ | $x$ | $x+2$ | $x+1$ |
| $x$ | $0$ | $x$ | $2x$ | $2$ | $x+2$ | $2x+2$ | $1$ | $x+1$ | $2x+1$ |
| $x+1$ | $0$ | $x+1$ | $2x+2$ | $x+2$ | $2x$ | $1$ | $2x+1$ | $2$ | $x$ |
| $x+2$ | $0$ | $x+2$ | $2x+1$ | $2x+2$ | $1$ | $x$ | $x+1$ | $2x$ | $2$ |
| $2x$ | $0$ | $2x$ | $x$ | $1$ | $2x+1$ | $x+1$ | $2$ | $2x+2$ | $x+2$ |
| $2x+1$ | $0$ | $2x+1$ | $x+2$ | $x+1$ | $2$ | $2x$ | $2x+2$ | $x$ | $1$ |
| $2x+2$ | $0$ | $2x+2$ | $x+1$ | $2x+1$ | $x$ | $2$ | $x+2$ | $1$ | $2x$ |

Figure 1: The addition and multiplication tables for $\mathbb{Z}_3[x]/(x^2+1)$.

But $x^2 = 1\cdot(x^2+1)-1$, i.e., $x^2 = -1 \pmod{1+x^2}$ so that

$$(a_0+a_1x)(b_0+b_1x) = (a_0b_0-a_1b_1)+(a_1b_0+a_0b_1)x \pmod{1+x^2}$$

which corresponds exactly to the multiplication of complex numbers with $x$ playing the rôle of $i=\sqrt{-1}$ (or of $-i$) and as expected $x$ mod $(1+x^2)$ (i.e., the image $x$ in $\mathbb{R}[x]/(1+x^2)$ ) is a root of $1+y^2$. Thus we have produced a purely algebraic construction of the complex numbers.

As another example, the polynomial $y^2+1 \in \mathbb{Z}_3[y]$ is irreducible (for otherwise it would have a linear factor and hence a root in $\mathbb{Z}_3$). We set $p = x^2+1$ and construct the field $\mathbb{Z}_3[x]/(p)$. The elements of this are represented uniquely by

$$ax+b, \quad \text{for } a,b \in \mathbb{Z}_3,$$

i.e., they are

$$0,1,2,x,x+1,x+2,2x,2x+1,2x+2.$$

Addition and multiplication are given by the same rules as for $\mathbb{R}[x]/(x^2+1)$ but of course we use coefficients from $\mathbb{Z}_3$. The tables are shown in Figure 1. We could present these tables in a way that hides the construction by consistently relabelling the elements, e.g., by

| $0$ | $1$ | $2$ | $x$ | $x+1$ | $x+2$ | $2x$ | $2x+1$ | $2x+2$ |
|---|---|---|---|---|---|---|---|---|
| $\updownarrow$ | $\updownarrow$ | $\updownarrow$ | $\updownarrow$ | $\updownarrow$ | $\updownarrow$ | $\updownarrow$ | $\updownarrow$ | $\updownarrow$ |
| $0$ | $1$ | $2$ | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ |

and then verify (laboriously) that they do indeed form a field. (The point of mentioning this is to help you appreciate the value of the construction; it guarantees for us that we have a field without further checking. Furthermore it would be quite hard to construct the tables just from scratch.) Of course, after relabelling, we would have essentially the same field which contains $\mathbb{Z}_3$ as a subfield and has a root for $y^2+1$, i.e., the element $a$. The only advantage of such a relabelling is that there is no longer the possibility of confusing $x$ as an element of $k[x]/(p)$ with $x$ as an

indeterminate over $k$. An alternative is to denote the elements of $k[x]/(p)$ by $[ax+b]$ but this becomes tiresome and one soon drops the brackets, taking care to be clear over which ring or field one is working in[4]. Using the simple notation we have established, we have that $x$ (as an element of $\mathbb{Z}_3[x]/(p)$) is a root of $y^2+1$. This means that $y-x$ divides $y^2+1$. Thus we have

$$y^2+1=(y-x)(ay+b)$$

for some $a,b\in\mathbb{Z}_3[x]/(p)$. Comparing leading coefficients of the two sides we see that $a=1$. Comparing constant terms we see that $-bx=1$ and so $b=-x^{-1}$. Now we can see from our multiplication table that $x^{-1}=2x$ (remember that all this is in $\mathbb{Z}_3[x]/(p)$). Thus we have

$$y^2+1=(y-x)(y-2x).$$

It is a good idea to confirm the claim above by multiplying the factors out, using the tables to simplify coefficients.

**Exercise 2.1** *As noted in Exercises 1 the polynomial $p=x^2+x+1$ is irreducible in $\mathbb{Z}_2[x]$ so that $\mathbb{Z}_2[x]/(p)$ is a field with four elements: 0, 1, $x$, $1+x$. Recall the addition and multiplication tables of this field from Exercises 1. Express the polynomial $y^2+y+1$ as a product of two linear polynomials in $\mathbb{Z}_2[x]/(p)[y]$, i.e., find two roots of the polynomial in the field $\mathbb{Z}_2[x]/(p)$. (Recall that $\alpha$ is a root of a polynomial $f(y)$ if and only if $y-\alpha$ divides $f(y)$.)* [4 marks]

*Note that $\mathbb{Z}_2[x]/(p)[y]$ denotes the ring of polynomials in $y$ with coefficients from $\mathbb{Z}_2[x]/(p)$.*

Let $x$, $y$ be indeterminates over a field $k$ and $f,g\in k[x]/(p)[y]$. Note that $f$, $g$ are polynomials in $y$ whose coefficients come from the field $k[x]/(p)$. In this case it makes sense to ask for the greatest common divisor of $f$, $g$. In order to appreciate the full force of what we do suppose we have an extension field $k'$ of $k$, i.e., $k\subseteq k'$ and the operations of $k$ (addition and multiplication) are those of $k'$. Suppose that there is an $\alpha\in k'$ that is a root of $p$. Note that we cannot have $\alpha\in k$ unless $\deg(p)=1$ since $\alpha$ is a root of $p$ if and only if $x-\alpha$ is a factor of $p$ and of course we have assumed that $p$ is irreducible in $k[x]$. (You might like to consider the situation $k=\mathbb{R}$, $p=x^2+1$, $k'=\mathbb{C}$ and $\alpha=\sqrt{-1}$.) Now consider the smallest subfield of $k'$ that contains both $k$ and $\alpha$; this is denoted by $k(\alpha)$. In fact it can be shown that $k(\alpha)$ consists of all elements of the form $a_1\alpha^{m-1}+a_2\alpha^{m-2}+\cdots+a_{m-1}$ where $a_i\in k$ for all $i$ and $m=\deg(p)$ (again you might like to consider the complex numbers as an example). It can be shown that $k(\alpha)$ is isomorphic as a field to $k[x]/(p)$, i.e., the two fields are really the same up to relabelling of elements. Moreover in this isomorphism we send $\alpha$ to $x$ as an element of $k[x]/(p)$. To be explicit, the isomorphism is:

$$\phi:k(\alpha)\to k[x]/(p)$$
$$a_1\alpha^{m-1}+a_2\alpha^{m-2}+\cdots+a_{m-1}\mapsto a_1x^{m-1}+a_2x^{m-2}+\cdots+a_{m-1}.$$

(The non-trivial aspect is to show that this is a well defined map and that it is indeed an isomorphism.) The statement that $\phi$ is an isomorphism means that it is a bijection (or 1-1 and onto, if you are old fashioned like me) and for all $r,s\in k(\alpha)$ we have $\phi(r+s)=\phi(r)+\phi(s)$, $\phi(rs)=\phi(r)\phi(s)$. Moreover the inverse function $\phi^{-1}$ of $\phi$ (the inverse exists because $\phi$ is a bijection) has the same properties. Taken together, what these facts say is that $\phi$ relabels $k(\alpha)$ to $k[x]/(p)$ and $\phi^{-1}$ goes in the other direction. The critical point is that this a faithful process in algebraic terms, i.e., we get the same answer no matter whether we add and relabel or relabel and then add (similarly for multiplying). Hence any purely algebraic operations in one of the fields can just as well be carried out in the other (we relabel to get from the starting field to the other one, carry out the operations and then relabel in the reverse direction). Note that we are not saying that all mathematical properties for the elements of the fields are the same, only the purely algebraic ones. For example, consider $k=\mathbb{Q}$, $p=x^2-2$ and $k'=\mathbb{R}$, here we have two choices for $\alpha$, i.e., $\pm\sqrt{2}$. The three fields $\mathbb{Q}(\sqrt{2})$, $\mathbb{Q}(-\sqrt{2})$ and $\mathbb{Q}[x]/(x^2-2)$ are isomorphic but of course $\sqrt{2}$ is different from $-\sqrt{2}$ and

[4]Recall that the elements of $k[x]/(p)$ are equivalence classes. Generally we use $[a]$ to denote the equivalence class of $a$ under an equivalence relation but in practice we represent a class by any member of it.

satisfies properties that $-\sqrt{2}$ does not, e.g., $\sqrt{2} > 0$ but $-\sqrt{2} < 0$ (by the way this shows that we cannot define the order $<$ by purely algebraic means). The isomorphisms are expressed by

$$a + b\sqrt{2} \leftrightarrow a - b\sqrt{2} \leftrightarrow a + bx,$$

bearing in mind that every element of $\mathbb{Q}[x]/(x^2-2)$ can be represented uniquely in the form $a+bx$ for some $a, b \in \mathbb{Q}$.

Now let $f$, $g$, $p$ be as above and $\alpha$ a root of $p$ in an appropriate extension field of $k$. Since gcd's are defined purely algebraically, the gcd of $f(\alpha, y)$, $g(\alpha, y)$ in $k(\alpha)[y]$ can just as well be computed in $k[x]/(p)[y]$ (in the result we understand $x$ to stand for $\alpha$).

**Exercise 2.2** *Let $p$ be an irreducible polynomial of $k[x]$ of degree $m$. Prove that every element of $k[x]/(p)$ can be represented uniquely by an expression of the form $a_1x^{m-1} + a_2x^{m-2} + \cdots + a_{m-1}$. There are two things to prove here:* [*10 marks*]

1. *(Existence.) Every element of $k[x]/(p)$ can be represented as described. This just means that each equivalence class of $k[x]/(p)$ contains an element of the kind shown.*
2. *(Uniqeness.) Each equivalence class contains exactly one element of the given kind. As usual assume there are two such elements in a class and show they must be the same. So assume that $f = a_1x^{m-1} + a_2x^{m-2} + \cdots + a_{m-1}$ and $g = b_1x^{m-1} + b_2x^{m-2} + \cdots + b_{m-1}$ are in the same equivalence class and prove that $a_i = b_i$ for $1 \leq i \leq m-1$, i.e., that $f = g$ in $k[x]$. (In effect we are assuming that the elements of $k$ are represented by some normal form and this extends to the elements of $k[x]/(p)$ by the process described.)*

**§3. Intersections of Algebraic Curves.** Let $f \in k[x, y]$. The set of zeros of $f$ forms an *algebraic curve*, i.e., $\{(\alpha, \beta) \mid f(\alpha, \beta) = 0\}$. For example, $x^2+y^2-1 \in \mathbb{R}[x, y]$ defines a circle while $y - x^2 \in \mathbb{R}[x, y]$ defines a parabola. For this section we will consider polynomials with rational coefficients but take the curve they define over the complex numbers (we will follow common practice and say 'the curve $f$' rather than 'the curve defined by $f$'). In particular we will consider pairs of curves and look at their common points. Now given such a pair $f$, $g$ it can be shown that they have finitely many common points if and only if their greatest common divisor is a constant[5] (which we normalise to 1). This statement depends crucially on the fact that $\mathbb{C}$ is algebraically closed (i.e., every non-constant polynomial with coefficients from $\mathbb{C}$ has a root in $\mathbb{C}$). From now on we assume that $\gcd(f, g) = 1$ and focus on the question of finding the finitely many common points. Put

$$f = f_0(x)y^m + f_1(x)y^{m-1} + \cdots + f_m(x),$$
$$g = g_0(x)y^n + g_1(x)y^{n-1} + \cdots + g_n(x).$$

Now assuming that $\gcd(f_0, g_0) = 1$ it can be shown that the $x$-coordinates of the common points of the curves defined by $f$, $g$ are precisely the roots of

$$r = \mathrm{Res}_y(f, g),$$

where $\mathrm{Res}_y$ denotes the *resultant* of its two polynomial arguments with respect to $y$. Axiom has a built-in function `resultant` that can be used to compute this; we will define this very useful function when studying the modular gcd algorithm for polynomials. For a simple example take

$$f = x^2 + y^2 - 1,$$
$$g = x - y,$$

i.e., a circle of radius 1 and a straight line through the origin with a gradient of $\pi/4$ (or 45°). These satisfy the two assumptions above and

$$r = \mathrm{Res}_y(x^2 + y^2 - 1, x - y) = 2x^2 - 1.$$

[5] If the gcd is not a constant then it gives us a common sub-curve of the curves given by $f$, $g$. Dividing this out of $f$ and $g$ we now obtain the remaining curves defined by $f$, $g$ which have finitely many points in common.

(Check this in Axiom by the command `resultant(f,g,y)`. Alternatively you can declare $f$, $g$ to have type `UP(y,UP(x,INT))` and then use `resultant(f,g)`.) This agrees with the result we obtain if we just substitute $y = x$ in $f$; of course the method we have outlined is more powerful since it applies to situations where neither polynomial is linear in an indeterminate.

Returning to the general situation, we say that two curves $f$, $g$ are in *good position* if for each possible $x$-coordinate value the curves $f$, $g$ have at most one common point, i.e., for all $\alpha \in \mathbb{C}$ there is at most one $\beta \in \mathbb{C}$ such that $f(\alpha, \beta) = g(\alpha, \beta) = 0$. This just states that above each point on the $x$-axis there is at most one point of intersection of the two curves. Given an $x$-coordinate $\alpha$ of a common point of $f$, $g$ we can find the $y$-coordinate(s) by considering the system

$$f(\alpha, y) = 0,$$
$$g(\alpha, y) = 0.$$

The common solutions to these are precisely the roots of

$$h_\alpha = \gcd(f(\alpha, y), g(\alpha, y) = 0.$$

This is justified by the following simple result.

**Lemma 3.1** *Let $p, q \in k[y]$ and assume that they are not both zero. Then the common roots of $p$ and $q$ (in any field that contains all the coefficients of $p$ and $q$ not necessarily just $k$) are precisely the roots of* $\gcd(p, q)$.

**Proof** It is clear that any root of $\gcd(f, g)$ must be a root of both $f$ and $g$ since $\gcd(f, g)$ divides both polynomials.

Conversely, recall that there are polynomials $u$, $v$ in $k[y]$ such that $\gcd(f, g) = uf + vg$. It follows that any common root of $f$, $g$ is also a root of $\gcd(f, g)$ □

Now $f$, $g$ are in good position if and only if for all such $\alpha$ the corresponding polynomial $h_\alpha$ has exactly one root. This final condition holds if and only if we have

$$h_\alpha = \gamma(y - \beta)^e$$

for some $\gamma, \beta \in \mathbb{C}$ and integer $e > 0$, or in other words if and only if the square free part[6] of $h_\alpha$ has degree 1 (note that since we ensure that our gcd's are monic we will always have $\gamma = 1$). In such a case we can obtain the $y$-coordinate explicitly in terms of $\alpha$.

These considerations are all very well but how do we calculate the gcd and the square free part (bearing in mind that $\alpha$ need not be a rational number)? In fact how do we even obtain $\alpha$? The answer is that we do not at any stage need numerical values: the relevant $\alpha$'s are the roots of

$$r = \mathrm{Res}_y(f, g).$$

Let us factorize this as

$$r = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t},$$

where each $p_i$ is an irreducible polynomial (in $\mathbb{Q}[x]$). Thus we obtain the $x$-coordinates of all the common points by looking at the roots of each $p_i$. If we focus on one of these and let $\alpha$ be a root then we already know how to calculate $h_\alpha$: it is simply the gcd of $f$, $g$ when viewed as elements of $\mathbb{Q}[x]/(p_i)[y]$. The square free part is now obtained by working in the same algebraic structure, Axiom supplies the function `squareFreePart` for this purpose. Here is a short example session of its usage:

[6] The square free part of a polynomial is obtained by keeping all the factors but removing any repetitions. For a polynomial $p \in k[y]$ the square free part is given by $p/\gcd(p, p')$ where $p'$ is the derivative of $p$ w.r.t. $y$. We will prove this when studying roots of univariate polynomials.

$P := \emptyset$;
$R := \{\text{irreducible factors of } \mathrm{Res}_y(f,g)\}$;
**for each** $r$ **in** $R$ **do**
  $G :=$ square free part$(\gcd(f,g))$;  (all computed in $\mathbb{Q}[x]/(r)[y]$)
  **if** $\deg(G,y) \neq 1$ **then**
    `ERROR("Curves in bad position")`
  **fi**;
  $Y :=$ (solution of $G = 0$ in $y$);
  $P := P \cup \{[x, Y, r]\}$
**od**

Figure 2: Algorithm for finding common points of algebraic curves.

```
(1) -> f:=(x^2*y+y^2-1)^3
f:=(x^2*y+y^2-1)^3
(1) ->
         6     2 5      4      4     6     2  3        4      2     2
   (1)  y  + 3x y  + (3x  - 3)y  + (x  - 6x )y  + (- 3x  + 3)y  + 3x y - 1
                                                      Type: Polynomial(Integer)



(2) -> a:=rootOf(2*x^2-1)
a:=rootOf(2*x^2-1)
(2) ->
   (2)  x
                                                          Type: AlgebraicNumber
(3) -> squareFreePart(eval(f,x=a))
squareFreePart(eval(f,x=a))
(3) ->

         2   1
   (3)  y  + - y - 1
             2
                                          Type: Polynomial(AlgebraicNumber)
```

Note that $x^2$ in $x^2y + y^2 - 1$ has been replaced by $1/2$ since $x^2 = 1/2$ in $\mathbb{Q}[x]/(2x^2 - 1)$. Do not confuse `rootOf` with root finding methods, the assignment `a:=rootOf(2*x^2-1)` tells Axiom that `a` is to be treated as `x` viewed as a member a member of $\mathbb{Q}[x]/(2x^2 - 1)$

Returning to our simple example we see that the resultant $2x^2 - 1$ is irreducible so that we need to find the gcd of $x^2 + y^2 - 1$, $x - y$ in $\mathbb{Q}[x]/(2x^2 - 1)[y]$ and this is just $y - x$. What this tells us is that the common points of $f$, $g$ are precisely all $(a, a)$ where $a$ is a root of $2a^2 - 1$ which is what we would expect.

Let us change our example a little by taking

$$f = x^2 + y^2 - 1,$$
$$g = x^2 - y^2,$$

i.e., a circle of radius 1 and two straight lines through the origin (given by $x - y$ and $x + y$). It is clear that these curves are not in good position. Let us see this fact by following the method developed so far: for the resultant we have

$$r = \mathrm{Res}_y(f,g) = (2x^2 - 1)^2.$$

Thus the $x$-coordinates of the common points are given by $2x^2 - 1$ as before. This time the gcd (which is square free) is $y^2 - 1/2$, i.e., it is not linear. We still have useful information: the common points of $f$, $g$ are a subset of $\{(a, b) \mid 2a^2 - 1 = 0, b^2 - 1/2 = 0\}$, i.e., they are amongst the four points $(\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$. In this case all of these points are common to the two curves but this is not always so.

Let us now gather all the observations made above into an algorithm as shown in Figure 2. In the algorithm we understand $[x, Y, r]$ to stand for all pairs $(\alpha, Y(\alpha))$ where $\alpha$ is a root of $r$ (remember that $Y$ is a polynomial in $x$).

Write a function `common` with the following specification:

**Declaration:** `common(f,g,x,y)`. *[14 marks]*

**Parameters:** `f,g` polynomials in indeterminates `x`, `y` with rational coefficients,
`x,y` symbols.

**Output:** An error is signalled if any of the following happen: (i) the leading coefficients of $f$, $g$ (when these are viewed as polynomials in $y$) have a common root; (ii) the curves $f$, $g$ have infinitely many points in common or (iii) the curves are not in good position.

Assuming the above three conditions do not apply the result is a *list* of the (finitely many) common points of the curves defined by $f$, $g$. Each point is given as $[x, Y(x), P(x)]$ where $Y$, $P$ are polynomials and $P$ is irreducible. If the curves do not have any common points then the empty list is returned.

**Number of lines:** 18.

**Remarks:** You may assume that the first two actual parameters given in any call will be of the correct type but must check the last three.

The pseudocode of Figure 2 presents the result as a set rather than a list. The Axiom implementation could use sets but lists are a little simpler to handle. (You create a set, e.g., by issuing `set [1,2,3]`.) Since lists give us the functionality we need (there are never any repeated entries) we might as well use them and save a bit of syntax.

**Useful Axiom functions:** `gcd`, `degree`, `leadingCoefficient`, `rootOf`, `resultant`, `squareFreePart`, `eval`, `error`.

**Examples:**

```
(11) -> f:= x^2+y^2-1
f:= x^2+y^2-1
(11) ->
          2    2
   (11)  y  + x  - 1
                                                    Type: Polynomial(Integer)
(12) -> g:=x*y-1
g:=x*y-1
(12) ->
   (12)  x y - 1
                                                    Type: Polynomial(Integer)
(13) -> common(f,g,x,y)
common(f,g,x,y)
          3      4    2
   [[x,- x  + x,x  - x  + 1]]
                                                              Type: Void
(14) -> g:=x-y
```

```
g:=x-y
(14) ->
   (14)  - y + x
                                                        Type: Polynomial(Integer)
(15) -> common(f,g,x,y)
common(f,g,x,y)
           2
   [[x,x,2x  - 1]]
                                                                    Type: Void
(16) -> g:=x^2-y^2
g:=x^2-y^2
(16) ->
            2    2
   (16)  - y  + x
                                                        Type: Polynomial(Integer)
(17) -> common(f,g,x,y)
common(f,g,x,y)
 17) ->
   Error signalled from user code:
      Curves in bad position
(17) -> f:=y^3-2*x*y+3*x-1
f:=y^3-2*x*y+3*x-1
(17) ->
          3
   (17)  y  - 2x y + 3x - 1
                                                        Type: Polynomial(Integer)
(18) -> g:=x*y^2+x^2+y+1
g:=x*y^2+x^2+y+1
(18) ->
            2        2
   (18)  x y  + y + x  + 1
                                                        Type: Polynomial(Integer)
(19) -> common(f,g,x,y)
common(f,g,x,y)
   [ ->
              5      4      3      2
          - 18x  - 54x  - 81x  - 75x  - 7x - 23
     [x, ------------------------------------,
                           25
        6     5      4     3      2
       9x  + 9x  + 24x  - 6x  + 16x  - 8x + 2]
     ]
                                                                    Type: Void
(20) -> f:=y^2-x
f:=y^2-x
(20) ->
          2
   (20)  y  - x
                                                        Type: Polynomial(Integer)
(21) -> g:=y+1
g:=y+1
(21) ->
   (21)  y + 1
                                                        Type: Polynomial(Integer)
```

```
(22) -> common(f,g,x,y)
common(f,g,x,y)
   [[x,- 1,x - 1]]
                                                                  Type: Void
(23) -> f:=x*y-x-y+1
f:=x*y-x-y+1
(23) ->
   (23)  (x - 1)y - x + 1
                                                  Type: Polynomial(Integer)
(24) -> g:=y-1
g:=y-1
(24) ->
   (24)  y - 1
                                                  Type: Polynomial(Integer)
(25) -> common(f,g,x,y)
common(f,g,x,y)
 25) ->
   Error signalled from user code:
      The curves have infinitely many common points
(25) -> f:=x+y+1
f:=x+y+1
(25) ->
   (25)  y + x + 1
                                                  Type: Polynomial(Integer)
(26) -> g:=x-y-1
g:=x-y-1
(26) ->
   (26)  - y + x - 1
                                                  Type: Polynomial(Integer)
(27) -> common(f,g,x,y)
common(f,g,x,y)
   [[x,- 1,x]]
                                                                  Type: Void
```

You can download this test session from the web pages, the file is called `commonTests.input`. Of course you should test your implementation further.

We note that it is possible to get around the assumptions made for the algorithm. This is achieved by an appropriate change of coordinates, i.e., a substitution of the form

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

where the matrix is invertible. In fact almost all such transformations will work (this can be made precise in a technical sense).

Finally we note that the algorithm given here is not very efficient. However is is fine for many examples and the ideas can be developed further to obtain a much more efficient algorithm (details available from KK).

Kyriakos Kalorkoti, February 2019