Polynomials and Diophantine equations

F. A. Ramponi
Rev. 0.4, 2021-05-06

0.1 Polynomials and divisibility

0.1.1 Definitions

A *polynomial* is an expression like

$$a = \sum_{i=0}^{n} a_i \Box^i = a_n \Box^n + a_{n-1} \Box^{n-1} + \ldots + a_1 \Box + a_0, \tag{1}$$

where all the coefficients a_i are supposed to belong to some field $\mathbb{K}$ and $\Box$ is a placeholder. In other words, a polynomial is a *finite* sequence of elements of $\mathbb{K}$. The set of all the polynomials over the field $\mathbb{K}$ is denoted $\mathbb{K}[z]$.

The addition and the multiplication of two polynomials $a = \sum_{i=0}^{n} a_i \Box^i$ and $a = \sum_{j=0}^{m} b_j \Box^j$ are defined in the obvious way that you remember from elementary algebra ($a_i \Box^i \cdot b_j \Box^j = a_i b_j \Box^{i+j}$, and all the properties of a field apply).

If k is the maximum index in (1) such that $a_k \neq 0$, then k in is called the *degree* of the polynomial a and is denoted $\deg a$. In particular, a constant polynomial has degree 0; the only exception is the zero polynomial $a = 0$, to which some authors assign no degree at all; to restore some compatibility between formulas, I will let instead $\deg 0 := -\infty$. The sum of two polynomials a, b has degree equal to $\max\{\deg a, \deg b\}$ if the degrees differ; if $\deg a = \deg b = n$, the sum has degree $\leq n$ (depending of course on the leading coefficient[s]). The product of a, b has the degree $\deg ab = \deg a + \deg b\}$.

A *polynomial function* is a function $a : \mathbb{K} \to \mathbb{K}$ defined by

$$a(z) = \sum_{i=0}^{n} a_i z^i = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0.$$

Hence, to each polynomial there corresponds a polynomial function in the obvious way, *but the converse is in general false*: note, for instance, that in the field $\mathbb{B} = \{0, 1\}$ (the binary field, where the addition and multiplication are the XOR and the AND operators respectively), the polynomials $a = 1\Box$ and $b = 1\Box + 1\Box^2 + 1\Box^3$ yield the same polynomial function $f(z) = z$, and yet they are different polynomials. It is true, however, that if $\mathbb{K}$ is infinite ($\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$) polynomials and polynomial functions are in one-to-one correspondence, so we shall not worry about this issue anymore (from now on, z will act both as the placeholder of the polynomial and the variable of the corresponding function).

0.1.2 Divisibility and roots

A polynomial $b \in \mathbb{K}[z]$ is said to *divide* another polynomial $a \in \mathbb{K}[z]$ if there exists $q \in \mathbb{K}[z]$ such that $a = bq$. This is denoted $b|a$.

A number $\bar{z} \in \mathbb{K}$ is a *root* of a polynomial a if $a(\bar{z}) = 0$.

Theorem 0.1.1 *$\bar{z}$ is a root of a if and only if $(z-\bar{z})|a$.*

Theorem 0.1.2 *(fundamental theorem of algebra). Any polynomial $a \in \mathbb{C}[z]$ with $\deg a \geq 1$ admits a root $\bar{z} \in \mathbb{C}$.*

This is usually stated by saying that the field $\mathbb{C}$ is algebraically closed. Note, by contrast, that $\mathbb{R}$ is not algebraically closed: for instance the polynomial $z^2+1 \in \mathbb{R}[z]$ has no root in $\mathbb{R}$.

Corollary 0.1.1 *Any polynomial $a \in \mathbb{C}[z]$ with $\deg a = n \geq 1$ can be factorized in a unique way as*

$$a(z) = a_n(z-\bar{z}_1)(z-\bar{z}_2)\dots(z-\bar{z}_n). \tag{2}$$

The number a_n in (2) is the leading coefficient (i.e. the coefficient of the term with maximum exponent) in the common expression of $a(z) = a_n z^n + \dots + a_0$. The polynomial a is called *monic* if $a_n = 1$, so that a monic polynomial reads $a(z) = z^n + a_{n-1}z^{n-1} + \dots + a_0$.

Theorem 0.1.3 *(Euclid). Given any two polynomials $a, b \in \mathbb{C}[z]$ (or $\mathbb{R}[z]$), $b \neq 0$, there exist unique polynomials q, r such that*

$$a = bq + r, \qquad \deg r < \deg b.$$

The proof is based essentially on the properties of Euclid's division algorithm. Here q is called the *quotient* of a and b, and r is called the *remainder* of the division of a by b. Note that this is one of those cases in which $\deg 0 := -\infty$ (or anyway negative) becomes useful: indeed if $\deg b = 0$ (b is a nonzero constant, e.g. $b = 1$), then $r = 0$, and the definition $-\infty < 0$ maintains the claim of the theorem.

0.2 Greatest Common Divisor and Euclid's algorithm

0.2.1 Definitions

A *common divisor* of two polynomials a, b is a polynomial d such that $d|a$ and $d|b$.

A *greatest common divisor* (g.c.d. for short) of a, b is a common divisor d of a, b such that, if d' is another common divisor of a, b, then $d'|d$. Note that (unlike the well-known g.c.d. of two *numbers* $m, n \in \mathbb{N}$) the g.c.d. of two polynomials over $\mathbb{R}$ or $\mathbb{C}$ is never unique, due to the arbitrary choice of the leading coefficient: it becomes unique if we require it to be *monic*. We will denote the monic g.c.d. as $\gcd(a, b)$.

If $\gcd(a, b) = 1$ ("constant" polynomial of degree 0) the polynomials a, b are called *coprime*.

0.2.2 Euclid's algorithm

The existence of a g.c.d. of two polynomials a, b is guaranteed by the convergence properties of Euclid's algorithm to compute it.

Let $a_0 := a$ and $b_0 := b$ be two polynomials (e.g. in $\mathbb{R}[z]$) whose g.c.d. we want to compute; note that, without any loss of generality, we can assume $\deg b_0 \leq \deg a_0$. The algorithm proceeds as follows. We find quotient and remainder according to Euclid's division theorem:

$$a_0 = b_0 q_0 + r_0 \qquad (\deg r_0 < \deg b_0).$$

A key fact to notice here is that $\gcd(a_0, b_0) = \gcd(b_0, r_0)$. Indeed, if d is a common divisor of a_0, b_0, then it divides also $a_0 - b_0 q_0 = r_0$, and hence it is a common divisor of b_0, r_0; *vice versa*, if

d divides b_0, r_0 then it must divide also $b_0 q_0 + r_0 = a_0$, and hence it must be a common divisor of a_0, b_0. This is true for any common divisor p; it follows that it is true also for the unique monic g.c.d., that is $\gcd(a, b) = \gcd(a_0, b_0) = \gcd(b_0, r_0)$. Now, the entire point of Euclid's algorithm is that passing from a_0, b_0 to b_0, r_0 the respective degrees have decreased: $\deg b_0 \leq \deg a_0$ by assumption, and $\deg r_0 < \deg b_0$ by Euclid's theorem. Therefore finding $\gcd(b_0, r_0)$ is a smaller-scale problem with the same result; it makes sense to let $a_1 := b_0$ and $b_1 := r_0$ and proceed with the same iteration,

$$a_1 = b_1 q_1 + r_1 \qquad (\deg r_1 < \deg b_1),$$

thus obtaining the new problem $\gcd(a, b) = \gcd(b_0, r_0) = \gcd(a_1, b_1) = \gcd(b_1, r_1)$, where $\deg b_1 \leq \deg a_1$ and $\deg r_1 < \deg b_1$. We proceed again and again, letting each time $a_{k+1} := b_k$ and $b_{k+1} := r_k$:

$$\begin{aligned}
a_2 &= b_2 q_2 + r_2 \qquad (\deg r_2 < \deg b_2), \\
&\vdots \\
a_k &= b_k q_k + r_k \qquad (\deg r_k < \deg b_k), \\
a_{k+1} &= b_{k+1} q_{k+1} + r_{k+1} \qquad (\deg r_{k+1} < \deg b_{k+1}), \\
&\vdots
\end{aligned}$$

Since the degree of the remainder keeps decreasing, at some step n it will unavoidably happen that

$$\begin{aligned}
&\vdots \\
a_n &= b_n q_n + r_n \qquad (\deg r_n < \deg b_n), \\
a_{n+1} &= b_{n+1} q_{n+1}, \qquad \text{i.e. } r_{n+1} = 0.
\end{aligned}$$

The equality

$$\gcd(a, b) = \gcd(b_0, r_0) = \gcd(b_1, r_1) = \ldots = \gcd(b_k, r_k) = \ldots = \gcd(b_n, r_n) = \gcd(b_{n+1}, r_{n+1})$$

is preserved throughout the whole execution, but at the end we obtain $\gcd(b_{n+1}, r_{n+1}) = \gcd(b_{n+1}, 0) = b_{n+1}$: indeed every polynomial divides the zero polynomial. Hence the last non-zero remainder $r_n = b_{n+1}$ is the g.c.d. computed by Euclid's algorithm. Note that nothing, so far, ensures that the result is actually monic, for the leading coefficient of the result depends on the coefficients of the original polynomials a and b; however, monicity can be trivially enforced dividing by its leading coefficient. Here follows a quick and dirty recursive implementation in Python:

```
from numpy.polynomial.polynomial import polydiv, polysub, polymul

# a, b: vector-like objects (of floating point numbers)
# [1., 2., 3.] ~= the polynomial 1 + 2z + 3z^2
def Euclid(a, b):
   if len(b) == 1 and b[0] == 0.0:   # the zero polynomial
      return [c/a[-1] for c in a]    # enforce monic

   (q, r) = polydiv(a, b)
   return Euclid(b, r)
```

0.2.3 Extended Euclid's algorithm

Euclid's algorithm does not only allow to compute a g.c.d. of two polynomials a, b, but also to recover two polynomials x, y such that

$$ax + by = \gcd(a, b).$$

(This is a particular case of a *Diophantine equation.*) To see how to recover x and y, turn back to the n-th step of Euclid's algorithm:

$$\begin{aligned} &\vdots \\ a_n &= b_n q_n + r_n, \\ a_{n+1} &= b_{n+1} q_{n+1}. \end{aligned}$$

From the very last equation $a_{n+1} = b_{n+1}q_{n+1}$ we recover

$$\begin{aligned} \gcd &= b_{n+1} \\ &= b_{n+1} + \underbrace{a_{n+1} - b_{n+1}q_{n+1}}_{=0} \\ &= a_{n+1} \cdot \underbrace{1}_{:=x_{n+1}} + b_{n+1} \underbrace{(1 - q_{n+1})}_{:=y_{n+1}} \\ &= a_{n+1}x_{n+1} + b_{n+1}y_{n+1}. \end{aligned}$$

Let's backtrace to the n-th step:

$$\begin{aligned} \gcd &= a_{n+1}x_{n+1} + b_{n+1}y_{n+1} \\ &= b_n x_{n+1} + r_n y_{n+1} \\ &= b_n x_{n+1} + (a_n - b_n q_n) y_{n+1} \\ &= a_n \underbrace{y_{n+1}}_{:=x_n} + b_n \underbrace{(x_{n+1} - q_n y_{n+1})}_{:=y_n} \\ &= a_n x_n + b_n y_n, \end{aligned}$$

and to the previous one,

$$\begin{aligned} \gcd &= a_n x_n + b_n y_n \\ &= b_{n-1} x_n + (a_{n-1} - b_{n-1} q_{n-1}) y_n \\ &= a_{n-1} \underbrace{y_n}_{:=x_{n-1}} + b_{n-1} \underbrace{(x_n - q_{n-1} y_n)}_{:=y_{n-1}} \\ &= a_{n-1}x_{n-1} + b_{n-1}y_{n-1}, \end{aligned}$$

and so on and so forth. Tracing back and back in the recursion, i.e. letting

$$\begin{cases} x_{k-1} &= y_k, \\ y_{k-1} &= x_k - q_{k-1} y_k, \end{cases}$$

we get to the first equation, where we find

$$\gcd = a_0 x_0 + b_0 y_0 := ax + by,$$

and we have recovered the desired polynomials x, y. The subject is well explained in Graham, Knuth, Patashnik, *Concrete Mathematics* or Louridakis, *Real world algorithms*, both referring

to the computation of the g.c.d. of *integers* (but the story is the same for polynomials). In particular Louridakis claims (for integers, not for polynomials... have to check) that the depth of the recursion is logarithmic in $\min\{\deg a, \deg b\}$, so there's no worry for stack overflows (there shouldn't be anyway even if it was linear, for the polynomials that a control engineer needs never have such a large degree). Some Python code:

```
def ExtendedEuclid(a, b):
   if len(b) == 1 and b[0] == 0.0:   # the zero polynomial
      return (a, [1.0], [0.0])

   (q, r) = polydiv(a, b)
   (d, x, y) = ExtendedEuclid(b, r)
   return (d, y,  polysub(x, polymul(q, y)) )   # (d, y, x - qy)

# for example:
print( ExtendedEuclid([6., 5., 1.], [2., 3., 1.]) )
# (array([ 4.,  2.]), array([ 1.]), array([-1.]))
```

Exercise: please test the code, refactor a bit, rework it in such a way that it returns the unique *monic* $\gcd(a, b)$ and the corresponding x, y. With the reworked version, whenever a and b are *coprime*, the algorithm should find x and y such that $ax + by = 1$.

0.3 Diophantine equations

0.3.1 Definition

Given three polynomials a, b, c the equation

$$ax + by = c, \tag{3}$$

in the unknowns x, y, is called a *Diophantine equation*, or sometimes a *Bezout identity*.

Theorem 0.3.1 *The Diophantine equation* (3) *has a solution if and only if* $\gcd(a, b)|c$.

Proof. Let $\gcd(a, b) = g$ and suppose that $g|c$: then $c = g\bar{c}$ for some polynomial $\bar{c}$. The extended Euclid's algorithm allows to compute polynomials $\bar{x}, \bar{y}$ such that $a\bar{x} + b\bar{y} = g$, but then multiplying by $\bar{c}$ we obtain a solution:

$$a \cdot \underbrace{\bar{x}\bar{c}}_{x} + b \cdot \underbrace{\bar{y}\bar{c}}_{y} = \underbrace{g\bar{c}}_{c}.$$

Suppose, on the other hand, that $g \nmid c$: however we choose x and y it still holds that $g|ax + by$, hence there is no possibility that $ax + by$ and c are the same polynomial. □

In particular, if a and b are coprime ($\gcd(a, b) = 1$), then (3) *always* admit a solution. However, with or without coprimality, if a solution exists it is never unique. Indeed it is immediate to check that if x_0, y_0 is a particular solution, then for any polynomial h

$$\begin{aligned} x &= x_0 + bh \\ y &= y_0 - ah \end{aligned} \tag{4}$$

is also a solution.

This degree of freedom plays in our favor, because we need to keep the degree of both x and y under control. For example, as an exercise you are invited to show that (4) implies that there

exists a solution with $\deg y < \deg a$.

Our typical situation in control applications will involve a, b coprime with $\deg a = n$, $\deg b \leq n-1$; they will be, respectively, the denominator and the numerator of the transfer function $W(z) = \frac{b(z)}{a(z)}$ of a plant:

- $\deg a = n$ means that the transfer function can be realized with a state-space model of dimension n. The realization is trivial in either reachability or observability canonical form: assuming, with no loss of generality, that the polynomial a is monic, the coefficients of $a(z)$ end up in the last row/column of the companion matrix A, and the coefficients of $b(z)$ end up in B or C, depending on the canonical form being chosen;
- the fact that $\deg b < \deg a$ means that the plant reacts with "at least one delay", i.e. that there is no direct connection between input and output (*strictly proper* transfer function $\Leftrightarrow$ no D in the state-space model; recall: the first sample of the impulse response is precisely D);
- the fact that a and b are coprime (no common factor, no "zero-pole cancellation" possible) means that such n-dimensional state-space model is *both* reachable *and* observable.

The method in the next section shows that, if c is an arbitrary polynomial with $\deg c = 2n-1$, then there exists also a solution x, y with $\deg x = n-1, \deg y \leq n-1$ (these will be respectively the denominator and numerator of the transfer function of a *controller*).

0.4 Sylvester's equation

The Diophantine equation can be solved in a straightforward way through a system of linear equations involving a so-called *Sylvester* matrix. Since I cannot find a standard reference where the matrix and the equation are written unambiguously, I will proceed by means of an example where $\deg a = n = 3$ and $\deg b = n-1 = 2$. Let

$$\begin{aligned} a(z) &= \alpha_3 z^3 + \alpha_2 z^2 + \alpha_1 z + \alpha_0, \\ b(z) &= \beta_2 z^2 + \beta_1 z + \beta_0, \\ x(z) &= x_2 z^2 + x_1 z + x_0, \\ y(z) &= y_2 z^2 + y_1 z + y_0, \end{aligned}$$

where a, b are known polynomials ($\alpha_3 \neq 0$ and $\beta_2 \neq 0$), and x, y are unknown ones. We ask if the Diophantine equation

$$ax + by = c$$

admits solution for an *arbitrary* polynomial c with degree $2n-1 = 5$. ($c(z) = c_5 z^5 + c_4 z^4 + c_3 z^3 + c_2 z^2 + c_1 z + c_0$, where $c_5 \neq 0$). We already know that, *in general*, a solution exists if and only if a, b are coprime, but here we are asking something more: the degree of x and y should be at most $n-1 = 2$. Indeed the solution exists and is unique. Let's work out the computation:

$$\begin{aligned} &(\alpha_3 z^3 + \alpha_2 z^2 + \alpha_1 z + \alpha_0)(x_2 z^2 + x_1 z + x_0) + (\beta_2 z^2 + \beta_1 z + \beta_0)(y_2 z^2 + y_1 z + y_0) \\ &\quad = c_5 z^5 + c_4 z^4 + c_3 z^3 + c_2 z^2 + c_1 z + c_0; \\ &(\alpha_3 x_2) z^5 + (\alpha_3 x_1 + \alpha_2 x_2 + \beta_2 y_2) z^4 + (\alpha_3 x_0 + \alpha_2 x_1 + \alpha_1 x_2 + \beta_2 y_1 + \beta_1 y_2) z^3 \\ &\quad + (\alpha_2 x_0 + \alpha_1 x_1 + \alpha_0 x_2 + \beta_2 y_0 + \beta_1 y_1 + \beta_0 y_2) z^2 + (\alpha_1 x_0 + \alpha_0 x_1 + \beta_1 y_0 + \beta_0 y_1) z + (\alpha_0 x_0 + \beta_0 y_0) \\ &\quad = c_5 z^5 + c_4 z^4 + c_3 z^3 + c_2 z^2 + c_1 z + c_0; \end{aligned}$$

equating the coefficients of corresponding powers of z in the second equation, we obtain

$$\left[\begin{array}{ccc|ccc} \alpha_3 & 0 & 0 & 0 & 0 & 0 \\ \alpha_2 & \alpha_3 & 0 & \beta_2 & 0 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \beta_1 & \beta_2 & 0 \\ \alpha_0 & \alpha_1 & \alpha_2 & \beta_0 & \beta_1 & \beta_2 \\ 0 & \alpha_0 & \alpha_1 & 0 & \beta_0 & \beta_1 \\ 0 & 0 & \alpha_0 & 0 & 0 & \beta_0 \end{array}\right] \left[\begin{array}{c} x_2 \\ x_1 \\ x_0 \\ \hline y_2 \\ y_1 \\ y_0 \end{array}\right] = \left[\begin{array}{c} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{array}\right]. \tag{5}$$

Now, the problem *$ax + by = c$ admits a solution for arbitrary c* if and only if the matrix in (5) is nonsingular, i.e. if it has nonzero determinant. This is equivalent to the fact that a, b are coprime.

You are invited to check that $\alpha_3 \neq 0$ implies $x_2 \neq 0$, so that $\deg x = 2$, and in particular if $\alpha_3 = 1$ and $c_5 = 1$, i.e. if both a and c are monic, then x is monic too.

Since $\alpha_3 \neq 0$, the matrix in (5) is nonsingular (check the determinant!) if and only if such is nonsingular the following one, obtained deleting the first row and the first column:

$$S = \left[\begin{array}{cc|ccc} \alpha_3 & 0 & \beta_2 & 0 & 0 \\ \alpha_2 & \alpha_3 & \beta_1 & \beta_2 & 0 \\ \alpha_1 & \alpha_2 & \beta_0 & \beta_1 & \beta_2 \\ \alpha_0 & \alpha_1 & 0 & \beta_0 & \beta_1 \\ 0 & \alpha_0 & 0 & 0 & \beta_0 \end{array}\right]. \tag{6}$$

This is called *Sylvester matrix*. More in general, we have the following

Theorem 0.4.1 *Let*

$$\begin{aligned} a(z) &= \alpha_n z^n + \alpha_{n-1} z^{n-1} + \ldots + \alpha_1 z + \alpha_0, \\ b(z) &= \beta_m z^m + \beta_{m-1} z^{m-1} + \ldots + \beta_1 z + \beta_0; \end{aligned}$$

then a, b are coprime if and only if $\det S \neq 0$*, where*

$$S = \left[\begin{array}{ccc|ccc} \alpha_n & & & \beta_m & & \\ \alpha_{n-1} & \ddots & & \vdots & \beta_m & \\ \vdots & \ddots & \alpha_n & \beta_0 & \vdots & \ddots \\ \alpha_0 & \ddots & \alpha_{n-1} & & \beta_0 & \ddots \\ & \ddots & \vdots & & & \ddots \\ & & \alpha_0 & & & \end{array}\right. \left.\begin{array}{cc} & \\ & \\ & \beta_m \\ \ddots & \vdots \\ \ddots & \beta_0 \end{array}\right]. \tag{7}$$

Pay attention: S in (7) is a $(n+m) \times (n+m)$ matrix; the left "$\alpha_n \ldots \alpha_0$ part" of the matrix has exactly m columns (the degree of b), and the right "$\beta_m \ldots \beta_0$ part" has exactly n columns (the degree of a). Compare with (6), where $n = 3$ and $m = n - 1 = 2$.

Proof. The polynomials $a(z)$ and $b(z)$ are not coprime if and only if they have at least a common root:

$$\begin{aligned} a(z) &= (z - \bar{z})(\bar{\alpha}_{n-1} z^{n-1} + \ldots + \bar{\alpha}_1 z + \bar{\alpha}_0), \\ b(z) &= (z - \bar{z})(\bar{\beta}_{m-1} z^{m-1} + \ldots + \bar{\beta}_1 z + \bar{\beta}_0). \end{aligned}$$

It follows that

$$\begin{aligned}
&(z-\bar{z})(\bar{\alpha}_{n-1}z^{n-1}+\ldots+\bar{\alpha}_1 z+\bar{\alpha}_0)(\bar{\beta}_{m-1}z^{m-1}+\ldots+\bar{\beta}_1 z+\bar{\beta}_0)\\
&\qquad = a(z)(\bar{\beta}_{m-1}z^{m-1}+\ldots+\bar{\beta}_1 z+\bar{\beta}_0) = b(z)(\bar{\alpha}_{n-1}z^{n-1}+\ldots+\bar{\alpha}_1 z+\bar{\alpha}_0),\\
&a(z)(\bar{\beta}_{m-1}z^{m-1}+\ldots+\bar{\beta}_1 z+\bar{\beta}_0) - b(z)(\bar{\alpha}_{n-1}z^{n-1}+\ldots+\bar{\alpha}_1 z+\bar{\alpha}_0) = 0;
\end{aligned}$$

then, equating coefficients, we obtain

$$\left[\begin{array}{ccc|ccc}
\alpha_n & & & \beta_m & & \\
\alpha_{n-1} & \ddots & & \vdots & \beta_m & \\
\vdots & \ddots & \alpha_n & \beta_0 & \vdots & \ddots \\
\alpha_0 & \ddots & \alpha_{n-1} & & \beta_0 & \ddots \\
 & \ddots & \vdots & & & \ddots \\
 & & \alpha_0 & & &
\end{array}\right.
\left.\begin{array}{cc}
 & \beta_m \\
 & \vdots \\
 & \beta_0 \\
\ddots & \\
\ddots & \beta_m \\
\ddots & \vdots \\
 & \beta_0
\end{array}\right]
\left[\begin{array}{c}
\bar{\beta}_{m-1} \\
\vdots \\
\bar{\beta}_0 \\
\hline
-\bar{\alpha}_{n-1} \\
-\bar{\alpha}_{n-2} \\
\vdots \\
-\bar{\alpha}_0
\end{array}\right] = 0$$

for a suitable choice of non-zero coefficients $\bar{\beta}_{m-1},\ldots,\bar{\beta}_0,\bar{\alpha}_{n-1},\ldots,\bar{\alpha}_0$: this can happen if and only if $\det S = 0$. □