# Arithmetic Operations with Polynomials on TI-92

*Dušan Pagon*, Faculty of Education, Maribor

## 1 Introduction

Giving for many years the courses of linear and higher algebra to our students - future secondary school teachers, we found out that they usually do not see the important connections between different algebraical structures they are taught. For example, working with polynomials in one variable they do not use the advantage of the fact that this is a linear space, though they have previously studied in details the standard vector spaces $\mathbb{R}^n$ and their linear transformations.

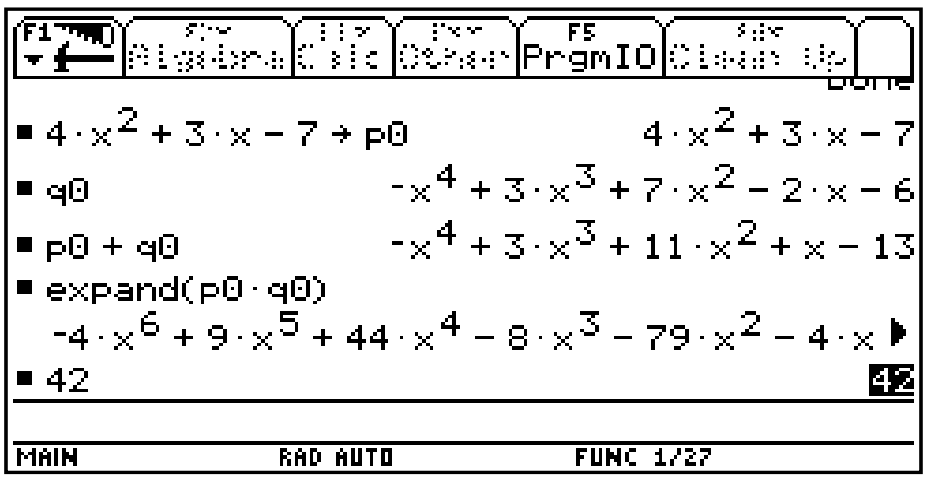
Modern computer algebra systems (CAS), from our point of view, offer new possibilities for establishing various links between different mathematical structures and we believe that this is one of the ways to consolidate the fragmented students knowledge. As a concrete example, in this paper we are dealing with vector notation of polynomials on the TI-92 calculators. We are also demonstrating how some of the most common operations on polynomials (arithmetic operations, greatest common divisor, evaluation on a set of points) are performed in this case.

## 2 The algebra of polynomials in one variable

Given an arbitrary ring $R$ (for instance the set of all integers $\mathbb{Z}$) and an abstract variable $x$ we define a polynomial $p$ in variable $x$ over the ring $R$ as a formal sum of the form $a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$. On the set $R[x]$ of all such polynomials the operations of addition and multiplication are introduced in the following way. For any two polynomials $p = \sum_{i=0}^{n} a_i x^i$ and $q = \sum_{i=0}^{m} b_i x^i$ we put $p + q = \sum_{i=0}^{n} (a_i + b_i) x^i$,
where in case of $n > m$ we assume $b_i = 0$ for all $i > m$ (and vice versa), and

$pq = \sum_{k=0}^{n+m} c_k x^k$, with $c_k = \sum_{i=0}^{k} a_i b_{k-i}$.

It is an easy exercise to check that in this way $R[x]$ itself becomes a ring, and even an algebra if the original ring $R$ is assumed to be a field. The CAS of TI-92 includes the standard polynomial arithmetics (for polynomials of one and several variables).



While the operation of polynomial addition is quite transparent and consists simply of a certain number $(\min\{m, n\} + 1)$ of additions in the main ring $R$, the rule for the polynomial multiplication is less obvious and performing it on TI-92 in a "black-box" manner students hardly get an impression of its complexity. Only looking carefully what happens with the coefficients of both polynomials during their multiplication:

$$
\begin{array}{cccccccc}
a_n & a_{n-1} & a_{n-2} & \dots & a_0 & & \times & b_m\, b_{m-1}\, \dots b_0 \\
\hline
a_nb_m & a_{n-1}b_m & a_{n-2}b_m & \dots & a_0b_m & & & \\
 & a_nb_{m-1} & a_{n-1}b_{m-1} & \dots & a_1b_{m-1} & a_0b_{m-1} & & \\
 & & a_nb_{m-2} & \dots & a_2b_{m-2} & a_1b_{m-2} & a_0b_{m-2} & \\
 & & \ddots & & \vdots & & \vdots & \\
 & & & & a_nb_0 & \dots & a_1b_0 & a_0b_0 \\
\hline
c_{m+n} & c_{m+n-1} & & \dots & & \dots & c_1 & c_0
\end{array}
$$

the students realize that in this case the calculator has to perform $(m + 1)(n + 1)$ multiplications and $mn$ additions of the coefficients of polynomials $p$ and $q$.

Students in Slovenia are, in accordance with our mathematical curriculum and through physics, quite early introduced to the concept of vectors in a standard linear space $\mathbb{R}^n$ with the operations of vector addition, multiplication by scalars and the dot product:

$(a_0, a_1, \dots a_n) + (b_0, b_1, \dots b_n) = (a_0 + b_0, a_1 + b_1, \dots a_n + b_n)$,

$t \cdot (a_0, a_1, \dots a_n) = (t \cdot a_0, t \cdot a_1, \dots t \cdot a_n)$, for all $t \in \mathbb{R}$,

$(a_0, a_1, \dots a_n) \cdot (b_0, b_1, \dots b_n) = a_0b_0 + a_1b_1 + \dots + a_nb_n \in \mathbb{R}$.

The parallel between the first two operations for polynomials and vectors is quite obvious. To examine the operation of multiplication in $R[x]$ let us first find a convenient notation for polynomials in a vector form. One of the possibilities is to connect the polynomial $p = \sum_{i=0}^{n} a_ix^i$ with the vector $v_p = (a_0, a_1, a_2, \dots a_n)$. It is not difficult for the students to find the pattern for this conversion on TI-92. On one hand, starting with the polynomial

$p = a_nx^n + a_{n-1}x^{n-1} + \dots + a_2x^2 + a_1x + a_0$

it is sufficient to evaluate at point 0 the sequence of $n + 1$ polynomials obtained by the recursive formula:

$p_0 = p$ and $p_i = \frac{p_{i-1} - p_{i-1}(0)}{x}$ for $1 \le i \le n$.

Then $v_p = (p_0(0),\ p_1(0), \dots p_n(0))$.

Similarly, starting with an arbitrary vector $v_p = (a_0, a_1, a_2, \dots a_n) \in R^n$ we can reconstruct the corresponding polynomial $p$ by the following steps:

$p_0 = a_n$

$p_1 = x \cdot p_0 + a_{n-1}$

$\vdots$

$p_n = x \cdot p_{n-1} + a_0 = \sum_{i=0}^{n} a_ix^i = p$

The conversion functions on TI-92 may look like

```
pove(pp,xx)
Func
Local aa,ii,qq,vv
pp|xx=0→aa
pp-aa→qq
[[aa]]→vv
While (qq|xx=π) ≠ 0
qq/xx→qq
qq|xx=0→aa
qq-aa→qq
augment(vv,[[aa]])→vv
EndWhile
EndFunc
```

```
vepo(vv,xx)
Func © Convert a vect to a polynom
Local ii,jj,mm
dim(mat▷list(vv))→ii
dotP([[1]],subMat(vv,1,ii,1,ii))→mm
For jj,ii-1,1,1
mm*xx+km(vv,jj)→mm
EndFor
expand(mm)
EndFunc
```

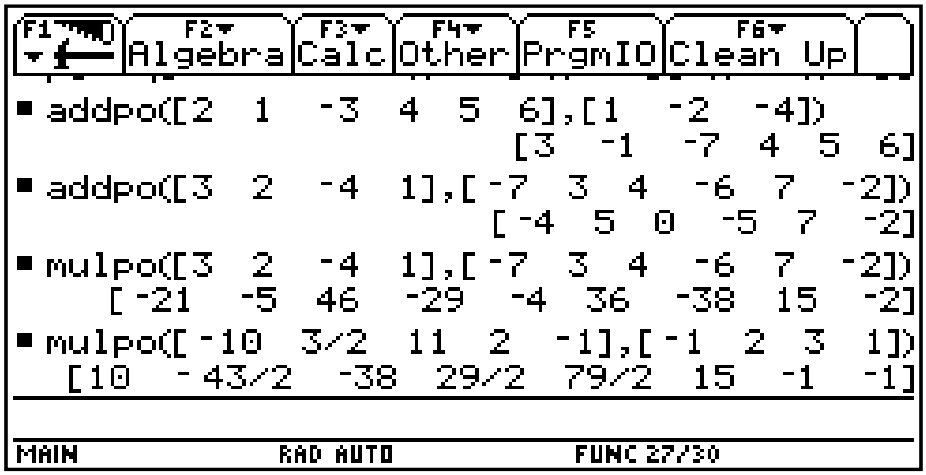
As the size of the obtained vector $v_p$ is equal to the degree of the related polynomial $p$ increased by 1, the students are not able to use directly the + command for the pair of vectors $(v_p,\ v_q)$ if polynomials $p$ and $q$ have different degrees. It is first necessary to complete the shorter of the two vectors by the appropriate number of zeros:

$$m < n \Rightarrow (a_0, a_1, \dots a_n) + (b_0, b_1, \dots b_m) = (a_0, a_1, \dots a_n) + (b_0, b_1, \dots b_m, \underbrace{0, 0, \dots 0}_{n-m}) =$$

$$= (a_0 + b_0, a_1 + b_1, \dots a_m + b_m, 0, 0, \dots 0).$$

In a similar way we can program TI-92 to perform the multiplication of two polynomials, written in the above vector form:

$$\begin{array}{cccccccc}
(a_0, & a_1, & a_2, & \dots & a_n) & \times & (b_0,\ b_1,\ \dots b_m) & \\
\hline
(a_0b_0, & a_1b_0, & a_2b_0, & \dots & a_nb_0) & & & \\
(0, & a_0b_1, & a_1b_1, & \dots & a_{n-1}b_1, & a_nb_1) & & \\
(0, & 0, & a_0b_2, & \dots & a_{n-2}b_2, & a_{n-1}b_2, & a_nb_2) & \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & & \\
(0, & 0, & \dots & 0, & a_0b_m, & \dots & a_{n-1}b_m, & a_nb_m) \\
\hline
(c_0, & c_1, & & \dots & & \dots & c_{n+m-1}, & c_{n+m})
\end{array}$$



In this way we need $m+1$ multiplications of vectors, whose size varies from $n+1$ to $n+m+1$, by scalars and $m$ additions of the resulting vectors.

# 3 Evaluation of polynomials and fast multiplication

Many times we are interested in the value $p(x_0)$ of a polynomial $p = \sum_{i=0}^{n} a_i x^i$ at the specific point $x_0 \in R$. It is well known that the optimal way to find this value by just $n$ multiplications and $n$ additions in the underlying ring $R$ is the so called Horner algorithm:

| | $a_n$ | $a_{n-1}$ | $a_{n-2}$ | ... | $a_2$ | $a_1$ | $a_0$ |
|---|---|---|---|---|---|---|---|
| | | $a_n x_0$ | $c_{n-2}x_0$ | ... | $c_2 x_0$ | $c_1 x_0$ | $c_0 x_0$ |
| $x_0$ | $a_n$ | $c_{n-2}$ | $c_{n-3}$ | ... | $c_1$ | $c_0$ | $p(x_0)$ |

In this way we also obtain the coefficients of the quotient $q(x) = \sum_{i=0}^{n-1} c_i x^i$ of the division of polynomial $p$ by binomial $x - x_0$.

As $c_i = \begin{cases} a_n, & \text{if } i = n-1 \\ a_{i+1} + c_{i+1} b, & \text{for } i < n-1 \end{cases}$ and $p(b) = a_0 + c_0 b$ the Horner algorithm can be implemented on TI-92 by using a single loop:

```
hornal(vv,pp)
Func
Local ii,kk,rr
dim(mat▷list((vv))→kk
subMat(vv,1,kk,1,kk)→rr
For ii,kk-1,1,-1
augment(rr,subMat(vv,1,ii,1,ii)+pp*subMat(rr,1,kk-ii,1,kk-ii))→rr
EndFor
EndFunc
```

For instance, the command hornal($[3,2,-4,5,1],-2$) results in $[1,3,-10,22,-41]$, therefore the value of the polynomial $x^4+5x^3-4x^2+2x+3$ at $x=-2$ is $-41$.

Every polynomial of degree $n$ is uniquely determined by its values on a set containing $n+1$ points, $S = \{x_0, x_1, \ldots x_n\}$. Thus we have another possibility to adjoin a vector $w_{p,S} = (p(x_0), p(x_1), \ldots p(x_n))$ to every polynomial $p$. This mapping is 1-1 and obviously

$w_{p,S} + w_{q,S} = w_{p+q,S}$, $w_{p,S} \odot w_{q,S} = w_{p\cdot q,S}$.

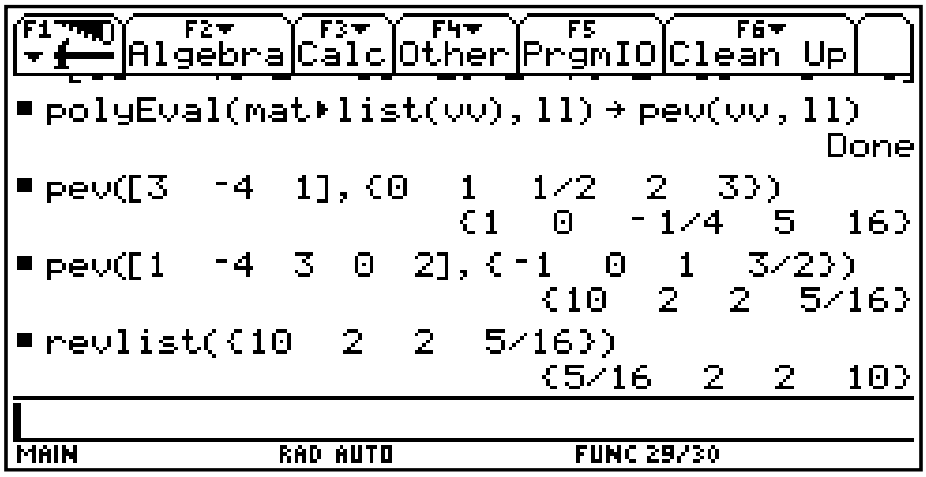
To be able to reconstruct the sum and the product of the original polynomials we must include in the set $S$ at least $\max\{\deg p, \deg q\} + 1$ points in the first case and at least $\deg p + \deg q + 1$ points in the second case.

By $\odot$ we have denoted the "point by point" multiplication of two vectors, for which on the TI-92 calculator the symbol .* is used.

The transformation of a polynomial $p$ into the vector $w_{p,S}$ is simple, as TI-92 library contains the **mat▷list** and **polyEval** commands: the first of them converts a vector

into a list and the second one evaluates the polynomial with coefficients $\{a_n, a_{n-1}, \dots a_0\}$ on the set of points $\{x_0, x_1, \dots x_m\}$. The only thing that we, according to our previous notation, have to do, is to define a function **revList** which will reverse the order of the coefficients of our polynomial.

```
revlist(ll)
Func
Local ii,mm
right(ll,1)→mm
For ii,2,dim(ll)
augment(mm,left(right(ll,ii),1))→mm
EndFor
EndFunc
```



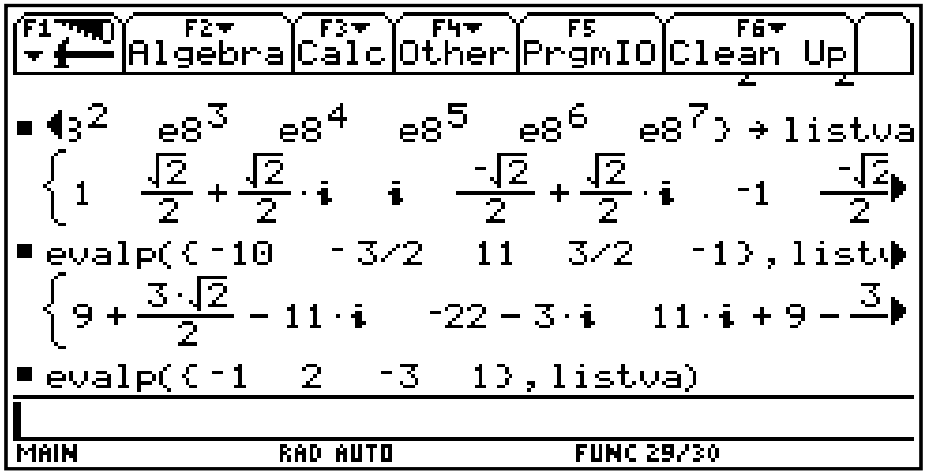

The inverse problem, of reconstruction of the polynomial if we know enough of his values, is solved by Lagrange interpolation formula. According to it $p = \sum_{i=0}^{n} t_i \prod_{j \neq i} \frac{x - x_i}{x_i - x_j}$ is the only polynomial of degree $n$ which values on the set of points $\{x_0, x_1, \dots x_n\}$ are $\{t_0, t_1, \dots t_n\}$.

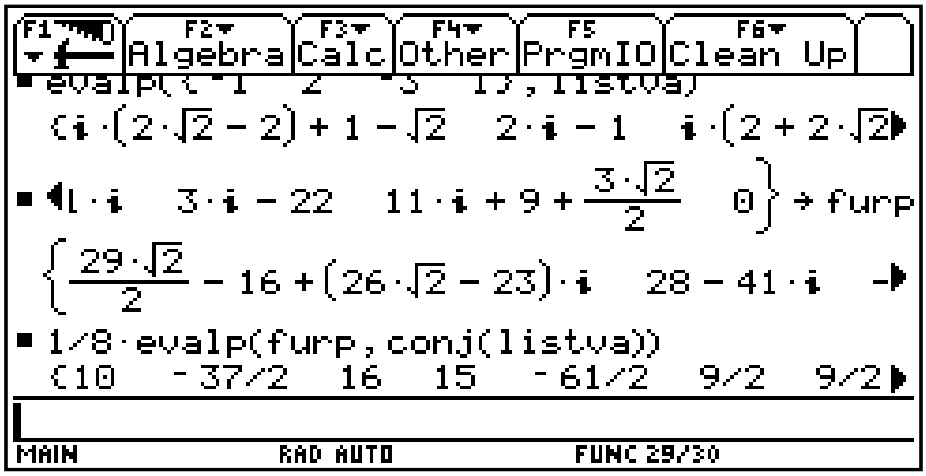
It looks quite surprising that for evaluation of a polynomial of degree $n$ on the set of $n$ points we do not need $n^2$ multiplications in the underlying field, as it would result from the repeated use of Horner algorithm, but just $\mathrm{O}(n\log_2{}^2 n)$. The idea for this fast procedure is in the Cooley's and Tuckey's algorithm for fast multiplication of the polynomials with the discrete Fourier transformation [2].

Suppose that our ring $R$ contains the $n^{th}$ root of unit $\omega^n = 1$ and $p \in R[x]$ is a polynomial of degree less than $n$. Than the linear mapping $\mathcal{F}_\omega(p) = \begin{pmatrix} p(1) \\ p(\omega) \\ p(\omega^2) \\ \vdots \\ p(\omega^{n-1}) \end{pmatrix}$ has a matrix

$$F_\omega = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2n-2} & \dots & \omega^{(n-1)^2} \end{pmatrix}$$

with Vandermonde determinant, so it is a 1-1 mapping between the space of polynomials of degree less than $n$ and the space of $n$-tupples of elements from $R$. It is also easy to see that the inverse transformation equals to $\frac{1}{n}\mathcal{F}_{\omega^{-1}}$. As we have already mentioned, the corresponding multiplication of the images in $R^n$ is "point by point" and for any two polynomials of degree less that $\frac{n}{2}$ we have $\mathcal{F}_\omega(pq) = \mathcal{F}_\omega(p) \cdot \mathcal{F}_\omega(q)$. Finally we obtain the required product of both polynomials as $pq = \frac{1}{n}\mathcal{F}_\omega^{-1}(\mathcal{F}_\omega(p) \cdot \mathcal{F}_\omega(q))$. We demonstrate this rule by the following example.




To obtain a real advantage from this method, it is necessary to compute the Fourier image of every polynomial quickly, for instance, by splitting it into two equal parts. Suppose that $n = 2^s,\ s \in \mathbb{N}$. Then the remainders at the divisions $p = q_0(x^{\frac{n}{2}} - 1) + r_0 = q_1(x^{\frac{n}{2}} + 1) + r_1$ can be obtained by simple addition or substraction of the coefficients:

$$r_0 = \sum_{j=0}^{\frac{n}{2}-1}(a_j + a_{j+\frac{n}{2}})x^j,\ \ r_1 = \sum_{j=0}^{\frac{n}{2}-1}(a_j - a_{j+\frac{n}{2}})x^j$$

As $\omega$ is the $n^{th}$ root of unit, the values of polynomial $p$ on its powers can be found by evaluation of polynomials $r_0,\ r_1^*$ which have twice smaller degree:

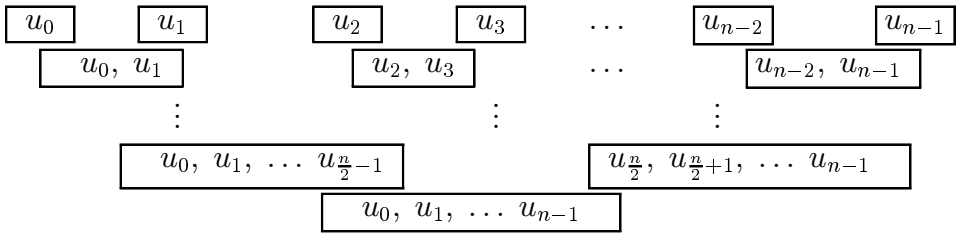
$p(\omega^{2i}) = r_0((\omega^2)^i),\ \ p(\omega^{2i+1}) = r_1(\omega \cdot \omega^{2i}) = r_1^*((\omega^2)^i)$

On the next step, instead of these values, we can find the values:

$r_{0,0}((\omega^4)^i)\ \ r_{0,1}^*((\omega^4)^i)\ \ \ \ r_{1,0}((\omega^4)^i)\ \ r_{1,1}^*((\omega^4)^i)$, and so on.

Now we can look for a solution of the same problem in a more general case, when $x_0, x_1, \ldots x_n \in F$ are arbitrary elements of some field $F$. Then $\mathcal{A} : p \to \begin{pmatrix} p(x_0) \\ p(x_1) \\ \vdots \\ p(x_n) \end{pmatrix}$ is an algebra isomorphism of $F[x]/\langle \prod_{i=0}^{n} (x - x_i) \rangle$ onto $F^{n+1}$. We again assume that $n+1$ is a power of 2 ($n = 2^s - 1$) and for each pair $(i, j)$, $0 \le i \le s$, $0 \le j < 2^{s-i}$ we construct a polynomial $q_{i,j} = \prod_{k=0}^{2^i - 1} (x - x_{j \cdot 2^i + k})$,

using the recursion: $q_{i+1,j} = q_{i,2j} q_{i,2j+1}$. Then $q_{i,j}(x_l) = 0$ if and only if $x_l$ is an element of the $j^{th}$ group (from the left) on the $i^{th}$ level (from the top) of the three




The values $p(x_i)$ are found in reverse order: $p$ is divided by $q_{s-1,0}$ and $q_{s-1,1}$. Then, every remainder is divided by the appropriate pair of polynomials from the next level. Finally, after division by linear polynomials $q_{0,j} = x - x_j$ the values $p(x_i)$ are what we are left with.

For instance, we want to evaluate

$r_{0,0} = x^7 + \frac{5}{2} x^6 + 2 x^5 - \frac{4}{3} x^4 + 3 x^3 - 2 x^2 - \frac{3}{2} x + 1$

on the set of points $v = \left[-2,\ -1,\ -\frac{1}{2},\ \frac{1}{3},\ \frac{3}{4},\ \frac{3}{2},\ 3,\ 4\right]$.

$$q_{1,0} = x^2 + 3x + 2$$

$$q_{1,1} = x^2 + \frac{1}{6} x - \frac{1}{6}$$

$$q_{1,2} = x^2 - \frac{9}{4} x + \frac{9}{8}$$

$$q_{1,3} = x^2 - 7x + 12$$

$$q_{2,0} = x^4 + \frac{19}{6} x^3 + \frac{7}{3} x^2 - \frac{1}{6} x - \frac{1}{3}$$

$$q_{2,1} = x^4 - \frac{37}{4} x^3 + \frac{231}{8} x^2 - \frac{279}{8} x + \frac{27}{2}$$

$$r_{1,0} = -\frac{121}{162} + \frac{5077}{324} x^3 + \frac{1669}{162} x^2 - \frac{577}{324} x$$

$$r_{1,1} = -\frac{779245}{128} + \frac{1696463}{768} x^3 - \frac{5289365}{512} x^2 + \frac{7487265}{512} x$$

$r_{2,0} = \frac{218}{3} + 77\,x$

$r_{2,1} = \frac{6239}{11664} - \frac{5261}{11664}\,x$

$r_{2,2} = -\frac{467611}{8192} + \frac{315273}{4096}\,x$

$r_{2,3} = -67669 + \frac{72116}{3}\,x$

$r_{3,0} = -\frac{244}{3}$

$r_{3,1} = -\frac{13}{3}$

$r_{3,2} = \frac{73}{96}$

$r_{3,3} = \frac{841}{2187}$

$r_{3,4} = \frac{10597}{16384}$

$r_{3,5} = \frac{467}{8}$

$r_{3,6} = 4447$

$r_{3,7} = \frac{85457}{3}$

# 4 Division with remainder. Euclidian algorithm

We shall conclude our paper with an algorithm for division of the polynomials, quite similar to the Horner algorithm. Let $p(x) = q(x) \cdot (x^k - b_{k-1}x^{k-1} - \ldots - b_1 x - b_0) + r$, where $r = \sum_{i=0}^{k-1} d_i x^i$. Then the coefficients of the quotient $q$ and of the remainder $r$ are calculated in the following table.

| | | | $a_n$ | $a_{n-1}$ | $a_{n-2}$ | $\ldots$ | $a_k$ | $a_{k-1}$ | $\ldots$ | $a_0$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $a_n b_{k-1}$ | $a_n b_{k-2} + c_{n-k-1} b_{k-1}$ | $\ldots$ | $\sum_{i=0}^{k-1} c_{k-i} b_i$ | $\sum_{i=0}^{k-2} c_{k-i-1} b_i$ | $\ldots$ | $c_0 b_0$ |
| $b_0$ | $\ldots$ | $b_{k-1}$ | $a_n$ | $c_{n-k-1}$ | $c_{n-k-2}$ | $\ldots$ | $c_0$ | $d_{k-1}$ | $\ldots$ | $d_0$ |

This means that the coefficients $b_i$ are each time moved one position to the right, into the "multiplication zone". After this we multiply them with the corresponding coefficients $c_i$, that we have already found, and finally, add the obtained products to the coefficient $a_i$.

Example. Let's divide the polynomial

$p = x^{10} - 3x^9 + x^8 - 2x^7 + 2x^6 - 3x^5 + x^4 - x^3 + x^2 - 2x + 1$ with the polynomial

$q = x^4 - 10x^2 + 1$.

| | | | | 1 | $-3$ | 1 | $-2$ | 2 | $-3$ | 1 | $-1$ | 1 | $-2$ | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 10 | $-30$ | 109 | $-317$ | 1099 | $-3168$ | 10889 | 320 | $-1100$ |
| $-1$ | 0 | 10 | 0 | 1 | $-3$ | 11 | $-32$ | 111 | $-320$ | 1100 | $-3169$ | 10890 | 318 | $-1099$ |

Thus the remainder is:

$r = -3169x^3 + 10890x^2 + 318x - 1099$.

We leave to the reader as an exercise the implementation of the last algorithm on TI-92.

References.


[1] Cormen T. H., Leiserson C. E., Rivest R. L. *Introduction to Algorithms*, MIT Press 1990.

[2] Gathen J., Gerhard J. *Modern Computer Algebra*, Cambridge University Press 1999.

[3] Graham R. L., Knuth D. E., Patashnik O. *Concrete Mathematics*, Addison-Wesley 1994.

[4] Milutinović U. *Dijeljenje polinoma.* In: Matematika **15** (1986), 4, 28-37.