Construction of matrix for polyfit(c)

Vector of values for x0: $X0 := \text{stack}(2,10,20,35,60,100,150,200,300,400,500,600)$

$i := 0..30 \qquad X0_i := i \cdot 20$

Number of points in the appropriate vectors for x: $NrPts := 200$

```
$\begin{pmatrix} Xmatrix \\ Yvector \end{pmatrix} :=$
    $X \leftarrow 0$
    $Y \leftarrow 0$
    for $x_0 \in X0$
        $x_{end} \leftarrow X1(0.25, x_0)$
        for $x \in x_0, x_0 + \frac{x_{end} - x_0}{NrPts} .. x_{end}$
            $i \leftarrow rows(Y)$
            $X_{i,0} \leftarrow x$
            $X_{i,1} \leftarrow x_0$
            $Y_i \leftarrow \alpha(x, x_0)$
    return $\begin{pmatrix} X \\ Y \end{pmatrix}$
```

$MC := \text{polyfitc}(Xmatrix, Yvector, 5)$

MC =

	0	1	2
0	"Term"	"Coefficient"	"Std Error"
1	"Intercept"	0.298	NaN
2	"A"	-0	NaN
3	"B"	0.003	NaN
4	"AB"	$2.198\cdot10^{-6}$	NaN
5	"AA"	$-7.349\cdot10^{-9}$	NaN
6	"BB"	$-1.933\cdot10^{-5}$	NaN
7	"AAB"	$6.091\cdot10^{-11}$	NaN
8	"ABB"	$-9.127\cdot10^{-9}$	NaN
9	"AAA"	$-1.125\cdot10^{-13}$	NaN
10	"BBB"	$6.941\cdot10^{-8}$	NaN
11	"AABB"	$-1.49\cdot10^{-13}$	NaN
12	"AAAB"	0	NaN
13	"ABBB"	$1.622\cdot10^{-11}$	NaN
14	"AAAA"	0	NaN
15	"BBBB"	$-1.137\cdot10^{-10}$	NaN
16	"AAABB"	0	NaN
17	"AABBB"	0	NaN
18	"AAAAB"	0	NaN
19	"ABBBB"	$-1.037\cdot10^{-14}$	NaN
20	"AAAAA"	0	NaN
21	"BBBBB"	$6.896\cdot10^{-14}$	...

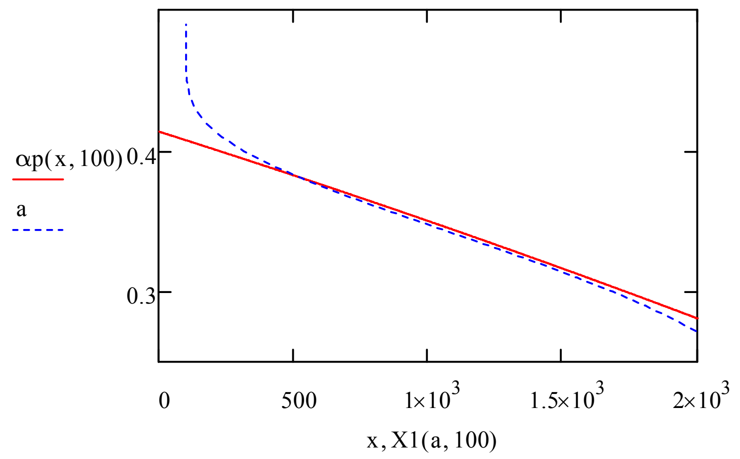
	0
0	0.298
1	-0
2	0.003

$\text{coeffs} := \text{submatrix}(MC, 1, \text{rows}(MC) - 1, 1, 1) =$

2	[illegible]
3	$2.198 \cdot 10^{-6}$
4	$-7.349 \cdot 10^{-9}$
5	$-1.933 \cdot 10^{-5}$
6	$6.091 \cdot 10^{-11}$
7	$-9.127 \cdot 10^{-9}$
8	$-1.125 \cdot 10^{-13}$
9	$6.941 \cdot 10^{-8}$
10	$-1.49 \cdot 10^{-13}$
11	0
12	$1.622 \cdot 10^{-11}$
13	0
14	$-1.137 \cdot 10^{-10}$
15	...

$$\alpha p(x, x0) := \text{coeffs}^T \cdot \text{stack}\left(1, x, x0, x \cdot x0, x^2, x0^2, x^2 \cdot x0, x \cdot x0^2, x^3, x0^3, x^2 \cdot x0^2, x^3 \cdot x0, x \cdot x0^3, x^4, x0^4, x^3 \cdot x0^2, x^2 \cdot x0^3, x^4 \cdot x0, x \cdot x0^4, x^5, x0\right.$$

$a := 0.25, 0.26 .. 0.5$



Thats a very bad fit, probably unusable. Increasing the number of points for x does not do anything better.

Lets try it with the older function "regress"

$MV := \text{regress}(Xmatrix, Yvector, 3)$

$MV =$

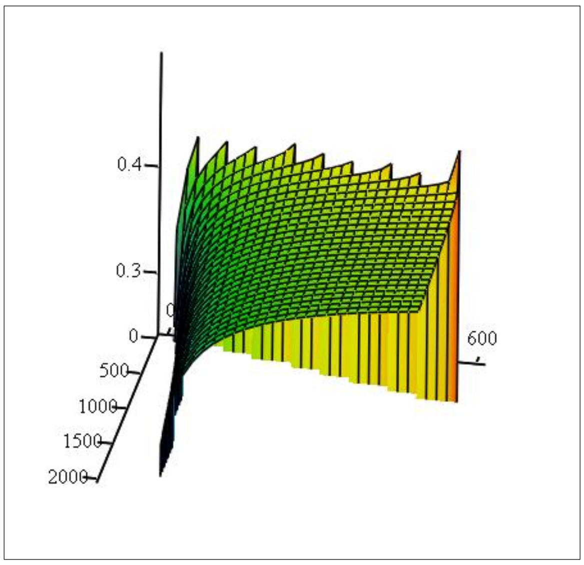
	0
0	3
1	3
2	3
3	$-1.625\cdot10^{-10}$
4	$2.061\cdot10^{-9}$
5	$-2.329\cdot10^{-6}$
6	0.001
7	$1.519\cdot10^{-7}$
8	$5.794\cdot10^{-13}$
9	0.31
10	$-3.974\cdot10^{-5}$
11	$-2.094\cdot10^{-10}$
12	$-1.098\cdot10^{-15}$

The coefficients begin with index 3 and are identical to the output of polyfitc.
The order of the coeffs is very strange (you may lookup the appropriate quicksheet) but we will get the very same function as above.

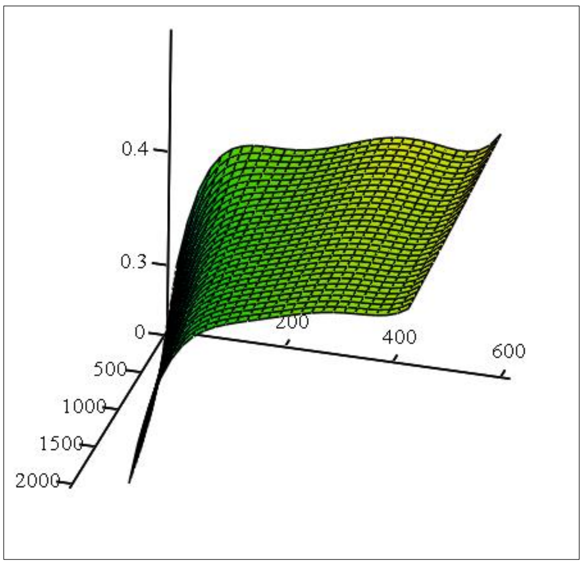
Lets do a quick hack and look at the 3D-surfaces. As many combinations of x/x0 do not evaluate and throw an error we define auxiliary functions to cope with. Unfortunately 3D-plots won't accept NaN so in case auf an error we set the return value to someithing outside of the plot area (-10) but we will see this as nasty vertical planes.

the "correct" function $\alpha1(x, x0) := -10 \ \text{ on error } \alpha(x, x0)$

x should go from 0 to 2000, x0 from 3 to 600, α is set from 0.25 to 0.5



$\alpha1$



αp

Hmmm, function α does not look so bad behaved, so I would had expected a better fit.
I am not sure if I had setup everything right as I do not have not much experience with those numerical approximations.

I won't do it but one thing you could try is doing a polynomial fit of higher degree than 3, but I'm not sure if this would help.