PSEUDOCODE

1- most relevant variables

number of training points [cases] → *numb_cases* [max. (v.1) → 60]

number of independent variables [inputs] → *numb_inputs* [max. (v.1) → 10]

number of combinations [inputs raised at different exponents] → *numb_comb* [$5 \cdot 10^6$ - 10^7]

input(i,j) → for i=1, *numb_cases*
for j=1, *numb_inputs*

output(i) → for i=1, *numb_cases*

combinatorics

x1 (i,m) → for i=1, *numb_cases*
for m=1, *numb_comb*

array storing the inputs being considered by each combination

elem_x1(i,m,n) → for i=1, *numb_cases*
for m=1, *numb_comb*
for n=1, *numb_elem* [max. number of inputs accounted within each combination]

array storing the exponents for each element in the given combination

exp_x1(i,m,n) → for i=1, *numb_cases*
for m=1, *numb_comb*
for n=1, *numb_elem*

regressions

reg_res (i,m) → for i=1, *numb_cases*
for m=1, *numb_comb*

pol_coef (i,m,n2) → for i=1, *numb_cases*
for m=1, *numb_comb*
for n2=1, *3*

decision algorithm

pol_error(i,m) → for i=1, *numb_cases*
for m=1, *numb_comb*

pol_error(i,m)=abs[100·(reg_res(i,m)-output(i))]/output(i)

mean_pol_error(m) → for m=1, *numb_comb*

mean_pol_error(m)=MEAN[pol_error(i,m)]for i=1, numb_cases

max_error_trend → max. *pol_error(i,m)* allowed in order to consider "m" right for this "i" [case]

min_numb_right → lowest proportion [out of *numb_cases*] of "right i" in order to consider a trend

2- combinatorics

the arrays *x1(i,m), elem_x1(m,n)* & *exp_x1(m,n)* get filled by accounting for all the possible combinations among the given inputs [with no repetition]

[input(1)^1.5, input(1), input(5)] ⊂ x1(360) ✓

elem_x1(360,1) = 1 [input(1)] / exp_x1(360, 1) = 1.5
elem_x1(360,2) = 1 [input(1)] / exp_x1(360, 2) = 1
elem_x1(360,3) = 5 [input(5)] / exp_x1(360, 3) = 1

[input(1)^1.5, input(1)^1.5] ⊂ x1(5000) X

3- regressions

second-degree polynomial regressions are performed for each couple [*x1(i,m),output(i)*]

pol_res (3,360) = 3.17
pol_res (3,360,1) = 0.0003 - pol_res (3,360,2) = 1500.37 - pol_res (3,360,3) = 4

4- decision algorithm

depending upon the given values [default configuration - recommendation: better don't modify them] for *max_error_trend* [5%] & *min_numb_right* [85%] there will be a pre-trend-determination process [any regression that, after being applied to all the given cases, verifies both conditions]

out of all the selected pre-trends, only the best one [lowest *mean_pol_error*] is given to the user as final result