

$i := 0..555$ $Q_{i,0} := i$ $Q_{i,1} := \sin\left(\frac{i}{50} \cdot \pi\right)$ $\text{rows}(Q) = 556$

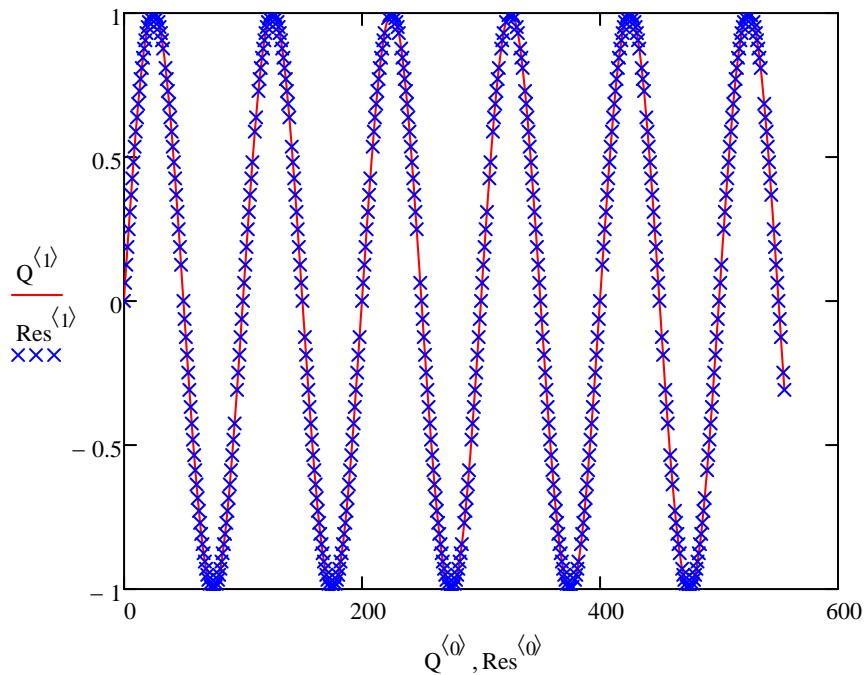
<pre> Rint(X) := "Find highest 2k fitting in data set" n ← rows(X) k ← 0 while n > 2^k k ← k + 1 sort[round[runif[n - 2^(k-1), X_{0,0}, X_{n-1,0}]]] </pre>	<pre> Isy(X) := "Check for double indices" q ← 0 for i ∈ 1..rows(X) - 1 q ← q + 1 if X_{i-1} = X_i q </pre>
--	---

```

KP2filter(Y) := "Create a vector of random integers within array boundaries and trims data array"
n ← rows(Y)
D ← Rint(Y)
while Isy(D) > 0
  D ← Rint(Y)
trim(Y,D)

```

$\text{Res} := \text{KP2filter}(Q)$ $\text{rows}(\text{Res}) = 512$



Ä = ■

creating data set

Functions:

Rint generates a vector of random Integers.

Isy checks for duplicates in the set generated by Rint.

original data array is trimmed by surplus points.

It shall return a 2^k array.

Works in cases where the number of data points is close to 2^k .

However, the larger the data array the less likely my function is to find a solution.

E.g. data set containing 555 elements. It puts out a 512 element vector - fine.

Change number of data elements to say, 999, it has troubles finding a solution. Whereas the next 2k element size of 1033 yields 1024.