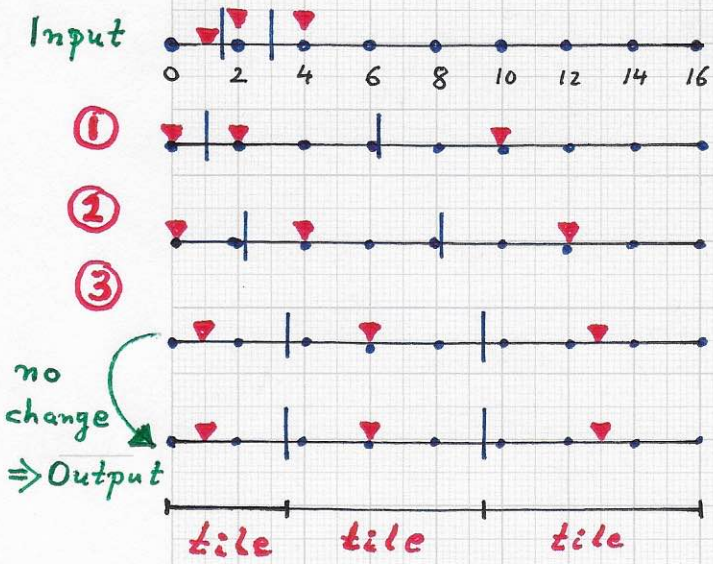
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

Simple 1D example for knot placement on the line:



• Given sites: 0, 2, 4, 6, 8, 10, 12, 14, 16

• Initial knots: 1, 2, 4

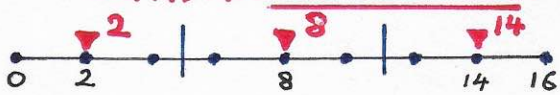
• Results of steps ①, ② and ③ – updating knots:

① 0, 2, 10
② 0, 4, 12
③ 1, 6, 13

Knots do not change after step ③. DONE.

• Output knots: 1, 6, 13

• The knots 1, 6 and 13 have 2, 3 and 4 associated sites, respectively. Thus, the result represents a LOCAL minimum. The GLOBAL min. is:



The progression of this optimization algorithm is shown in the left figure for a 1D scenario, where **nine** equidistantly spaced sites and **three** initial locations of **three knots** define the input. After only three update steps, the three knot locations no longer change, and the algorithm terminates. The result corresponds to a LOCAL MINIMUM of the used error function E. The fact that different numbers of original sites are associated with the knots (2, 3 and 4) indicate that the output does not represent a global minimum.

• **Note.** Many algorithms are known and used for the analysis of high-dimensional data organized in clusters. So-called "K-means algorithms" determine the K clusters and cluster centers, where the value of K is usually an input datum. The discussed knot placement method is a K-means clustering algorithm.

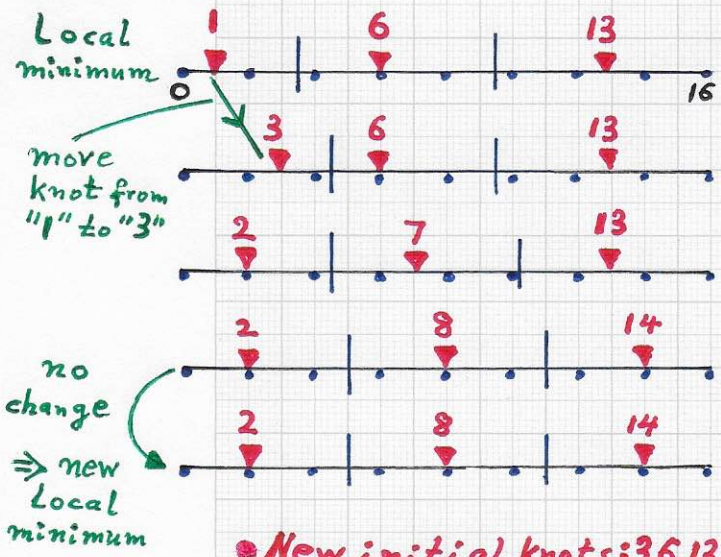
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

Heuristic method for improving the locally minimal/optimal knot placement: Move a knot with minimal number of associated sites towards a "neighbor knot" with (locally) maximal number of associated sites!

⇒ Move knot with value 1 towards knot with value 6! (See previous page.) ⇒ Move left-most knot with value 1 to location 3:



• New initial knots: 3, 6, 13

• Perform the iterative knot placement algorithm again. (Last page!)

• Output knots: 2, 8, 14

⇒ This knot placement represents a global optimum. Each knot has three associated sites, and error function $E_2 = 0$.

McMahon and Franke suggested another algorithm for knot placement (in their paper referenced on p. 18, 8/20/22). The main rationale underlying this alternative method for knot placement is the goal of viewing all given original data (site and associated value) as relevant, equally important. This objective translates to the following optimization criterion:

Knots are placed well when the numbers of original sites associated with each knot are the same, i.e., these numbers should minimally deviate from each other. Thus, the alternative error function to be minimized is the SUM OF SQUARED DIFFERENCES BETWEEN THE NUMBER OF SITES ASSOCIATED WITH A KNOT AND THE IDEAL AVERAGE NUMBER OF SITES THAT SHOULD BE ASSOCIATED WITH THE KNOT. Thus, this error function is $E_2 = \sum_{i=1}^{N} (n_i - n/N)^2$, where n_i is the number of sites belonging to $*_i$.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

General strategy for knot placement ensuring that a nearly constant number of sites is eventually associated with each knot:

- The goal of minimizing error E_2 is integrated into a more general design of the presented algorithm for minimizing error E.

The main steps are:

1) Create an initial placement of N knots, e.g., using the initial algorithm for updating knot locations to minimize error E.

2) In this placement, identify a knot (or multiple knots) with the minimal number of sites associated with it (them).

3) Move the identified knot(s) in the direction of a "neighbor knot" with the largest number of sites associated with it.

```
/* For example, a knot */
/* has those other knots */
/* as its neighbors when */
/* their tiles – in an un- */
/* derlying Voronoi dia- */
/* gram of all knots – are */
/* neighbor tiles of the */
/* knot to be moved. */
```

• • •

It is instructive to calculate the values of the two error functions E and E_2 – that are minimized in tandem. First, we compute the error values for the knot set $\{1, 6, 13\}$, i.e., the values for the first local minimum reached:

$$E = \sum_{j=1}^{9} \min_i (x_j - \bar{x}_i)^2$$
$$= (0-1)^2 + (2-1)^2 + (4-6)^2 + (6-6)^2 + (8-6)^2 + (10-13)^2 + (12-13)^2 + (14-13)^2 + (16-13)^2$$
$$= 1+1+4+4+9+1+1+9 = 30.$$

$$E_2 = \sum_{i=1}^{3} (n_i - 9/3)^2$$
$$= (2-3)^2 + (3-3)^2 + (4-3)^2 = 2.$$

After moving the knot with location $x=1$ to location $x=3$ and performing the knot placement algorithm again, we obtain the new knot set $\{2, 8, 14\}$. The new error values are

$$E = (0-2)^2 + (2-2)^2 + (4-2)^2 + (6-8)^2 + (8-8)^2 + (10-8)^2 + (12-14)^2 + (14-14)^2 + (16-14)^2 = 24.$$

$$E_2 = (3-3)^2 + (3-3)^2 + (3-3)^2 = 0.$$

In this simple example, a global optimum is obtained quickly: Each knot has three associated sites (optimal) and is the centroid of its three associated sites (optimal).

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• • •

```
/* One must use a "proper" */
/* value when moving a    */
/* knot a_i towards a     */
/* neighbor knot b. For   */
/* example the maxi-      */
/* mally allowed distance */
/* of movement could be   */
/* d_max = ||b - a_i|| / 2.  */
/* Further, in later sta- */
/* ges of knot movements, */
/* one should reduce the  */
/* allowed distance more  */
/* and more.              */
```

4) After movement of the knot(s), perform the update algorithm again for minimizing error E until a new local minimum is reached.

5) Calculate the values of the errors E and E_2.

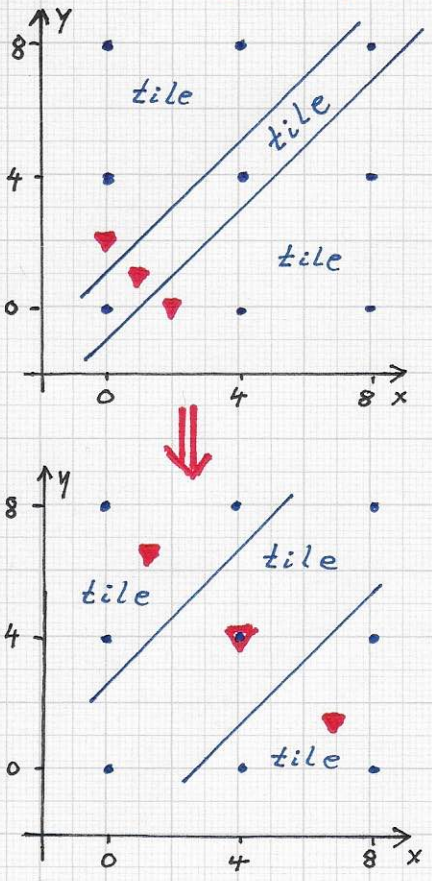
6) Has E_2 decreased in value?

i) yes: perform the procedure for knot movement again.

ii) no: reduce the value of allowed distance for movement; "un-do" the result just obtained and "re-do" the movement procedure with the reduced movement value.

• • •

We consider a few simple 2D examples to better understand the behavior of the E-minimizing and more general E_2-minimizing methods. First, we describe the knot placement update algorithm for $N = 3$ knots and $n = 9$ data sites: {•} given sites, {▼} knots



Here, the 9 sites originate from an "invisible" underlying uniform rectilinear grid. Initially, the 3 knots are clustered in the lower left corner. Each knot has three associated sites in its tile. The knots' locations are updated by moving each knot to the centroid of the sites in its tile. In this case, the algorithm terminates after just one update step. Further, E_2 is optimal since each tile contains the optimal number of sites: $9/3 = 3$. $\Rightarrow E_2 = 0$. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

• • •

7) Repeat these steps using ever decreasing step sizes for knot movement UNTIL

i) E_2's value has decreased
OR
ii) E's value has decreased
OR
iii) a knot placement is computed that was already generated previously.

8) Use the current knot placement as an initial placement, and perform all steps - starting with step 2) again.

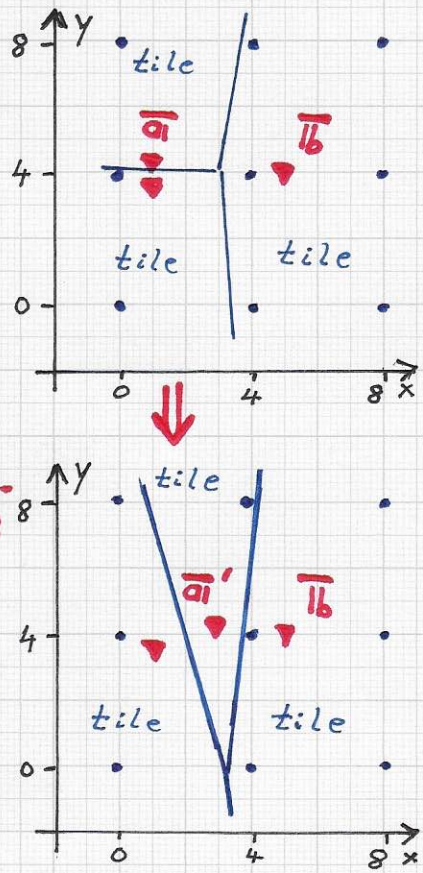
BUT: Apply the movement procedure in the "from-high-to-low manner":

Move the knot(s) with identified maximal number of associated sites in the direction of a "neighbor knot" with the smallest number of sites associated with it.

9) Termination: The algorithm terminates when the value of E_2 can no longer be lowered.

∴

Second, we sketch a simple example for the E_2-minimizing approach, focusing on the movement of a knot with minimal number of sites to a neighborknot with (locally) maximal number of sites. Again, we use $N = 3$ knots and $n = 9$ sites:



Numbers of sites in tiles:
- before move: 1, 2, 6
- after move: 1, 3, 5

Viewing this state as an initial or intermediate state and having to perform knot movement in the next step, one must move a_i towards b_i, since their tiles have the minimal and maximal numbers of sites, respectively. Point a_i becomes point a_i'.

⇒ The value of E_2 has improved:
- before move: $E_2 = (1-3)^2 + (2-3)^2 + (6-3)^2 = 14$
- after move: $E_2 = (1-3)^2 + (3-3)^2 + (5-3)^2 = 8$

• • •