Chapter 2
Approximations to Peano Curves: Algorithms and Software

In rallying every curve, every hill may be different than you thought. That makes it interesting.

Kimi Raikkonen

2.1 Space-Filling Curves and Reduction of Dimensionality

Due to the important role the space-filling curves play in the subsequent treatment it is appropriate to fix this term by some formal statement.

Definition 2.1. Single-valued continuous correspondence $y(x)$ mapping the unit interval $[0,1]$ on the x-axis onto the hypercube D from (1.2) is said to be a *Peano-type curve* or a *space-filling curve*.

Hereinafter we describe a particular type of space-filling curves emerging as the limit objects generated by the scheme from [132, 134, 135, 139, 140] succeeding the ideas from Hilbert [62]. Computable approximations to these curves are employed in the algorithms suggested in the rest of this book for solving multivariate Lipschitz global optimization problems.

We introduce now the *Curve construction scheme*.

Divide the hypercube D from (1.2) into 2^N equal hypercubes of the "first partition" by cutting D with N mutually orthogonal hyperplanes (each plain is parallel to one of the coordinate ones and passes through the middle points of the D edges orthogonal to this hyperplane); note that each of these subcubes has edge length equal to 2^{-1}. Use the index $z_1, 0 \leq z_1 \leq 2^N - 1$, to number all the subcubes obtained in the above partitioning; each particular subcube is, henceforth, designated $D(z_1)$ (for the sake of illustration case $N = 2$ is presented in Fig. 2.1; see the left picture). Then divide (in the above manner) each of the obtained first-partition cubes into 2^N second-partition subcubes numbered with the index

Y.D. Sergeyev et al., *Introduction to Global Optimization Exploiting Space-Filling Curves*, SpringerBriefs in Optimization, DOI 10.1007/978-1-4614-8042-6_2,

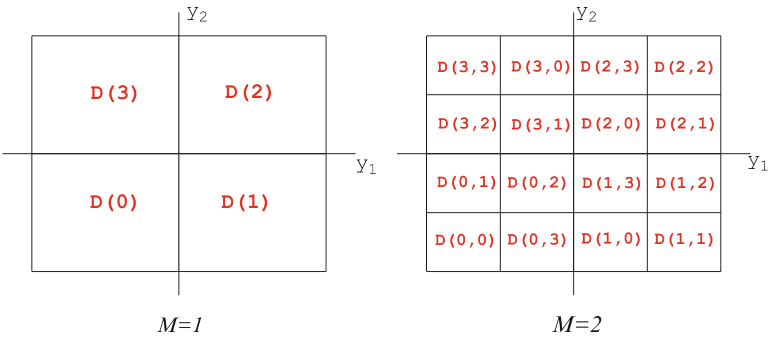

Fig. 2.1 Case $N = 2$. Subcubes of the first partition (*left picture*) and of the second partition (*right picture*) of the initial cube D

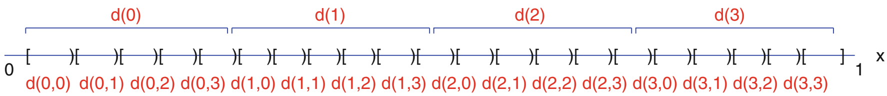

Fig. 2.2 Case $N = 2$. Subintervals $d(z_1)$ of the first partition and subintervals $d(z_1, z_2)$ of the second partition of the unit interval $[0,1]$ on the x-axis

z_2, $0 \le z_2 \le 2^N - 1$. Each particular subcube obtained by such a partitioning of $D(z_1)$ is designated $D(z_1, z_2)$ and it has edge length equal to 2^{-2}; see the right picture in Fig. 2.1.

Continuing this process, i.e., consequently cutting each hypercube of a current partition into 2^N subcubes of the subsequent partition (with a twice shorter edge length), yields hypercubes $D(z_1, \ldots, z_M)$ of any Mth partition with edge length equal to 2^{-M}. The total number of subcubes of the Mth partition is equal to 2^{MN} and

$$D \supset D(z_1) \supset D(z_1, z_2) \supset \ldots \supset D(z_1, \ldots, z_M), \tag{2.1.1}$$

where $0 \le z_j \le 2^N - 1, \quad 1 \le j \le M$.

Next, cut the interval [0,1] on the x-axis into 2^N equal parts; each particular part is designated $d(z_1)$, $0 \le z_1 \le 2^N - 1$: the numeration streams from left to right along the x-axis. Then, once again, cut each of the above parts into 2^N smaller (equal) parts, etc. Designate $d(z_1, \ldots, z_M)$, $0 \le z_j \le 2^N - 1$, $1 \le j \le M$, the subinterval of the Mth partition; the length of any such interval is equal to 2^{-MN}. Assume that each interval contains its left-end-point, but it does not contain its right-end-point; the only exception is for the case when the right-end-point is equal to unity, which corresponds to the relations $z_1 = z_2 = \ldots = z_M = 2^N - 1$. Obviously,

$$[0,1] \supset d(z_1) \supset d(z_1, z_2) \supset \ldots \supset d(z_1, \ldots, z_M)\,; \tag{2.1.2}$$

case $N = 2$ is illustrated by Fig. 2.2 (for $M = 1$ and $M = 2$).

Present the left-end-point v of the subinterval

$$d(z_1,\dots,z_M) = [v, v+2^{-MN}) \tag{2.1.3}$$

in the binary form

$$0 \le v = \sum_{i=1}^{MN} \alpha_i 2^{-i} < 1, \tag{2.1.4}$$

where $\alpha_1, \alpha_2, \dots, \alpha_{MN}$ are binary digits (i.e., $\alpha_i = 0$ or $\alpha_i = 1$). From (2.1.3), (2.1.4) and the already accepted condition that the numeration of the subintervals from (2.1.2) with any index z_j, $1 \le j \le M$, streams from left to right along the x-axis, this index is possible to present as

$$z_j = \sum_{i=0}^{N-1} \alpha_{jN-i} 2^i, \quad 1 \le j \le M\,. \tag{2.1.5}$$

In the sequel, the interval (2.1.3) will also be referred to as $d(M,v)$. The relations (2.1.4), (2.1.5) provide a basis for computing the parameters from one side of the identity

$$d(M,v) = d(z_1,\dots,z_M) \tag{2.1.6}$$

(i.e., M,v or $z_1,\dots,z_M$) being given the parameters from the other side of this identity (i.e., $z_1,\dots,z_M$ or M,v).

Now, establish a mutually single-valued correspondence between all the subintervals of any particular Mth partition and all the subcubes of the same Mth partition by accepting that $d(M,v)$ from (2.1.6) corresponds to $D(z_1,\dots,z_M)$ and vice versa. The above subcube will also be designated $D(M,v)$, i.e.,

$$D(M,v) = D(z_1,\dots,z_M), \tag{2.1.7}$$

where the indexes $z_1,\dots,z_M$ have the same values as in (2.1.6) and they could be computed through (2.1.3)–(2.1.5).

In accordance with (2.1.1) and (2.1.2), the introduced correspondence satisfies the

Condition 1. *$D(M+1,v') \subset D(M,v'')$ if and only if $d(M+1,v') \subset d(M,v'')$.*

We also require this correspondence to satisfy the following

Condition 2. *Two subintervals $d(M,v')$ and $d(M,v'')$ have a common end-point (this point may only be either v' or v'') if and only if the corresponding subcubes $D(M,v')$ and $D(M,v'')$ have a common face (i.e., these subcubes must be contiguous).*

Two linked systems of partitioning (i.e., the partitioning of the cube D from (1.2) and the partitioning of the unit interval $[0,1]$ on the x-axis) that meet the above two conditions provide the possibility for constructing the *evolvent curve* which may be employed in (1.7). Note that Condition 1 is already met, but Condition 2 has to be ensured by a special choice of numeration for the subcubes $D(z_1,\dots,z_M), M \geq 1$, which actually establishes the juxtaposition of the subcubes (2.1.7) to the subintervals (2.1.6). The particular scheme of such numeration suggested in [132, 134] will be introduced in the next section.

Theorem 2.1. *Let $y(x)$ be a correspondence defined by the assumption that for any $M \geq 1$ the image $y(x) \in D(M,v)$ if and only if the inverse image $x \in d(M,v)$. Then:*

1. *$y(x)$ is the single-valued continuous mapping of the unit interval $[0,1]$ onto the hypercube D from (1.2); hence, $y(x)$ is a space-filling curve.*
2. *If $F(y), y \in D$, is Lipschitzian with some constant L, then the univariate function $F(y(x)), x \in [0,1]$, satisfies Hölder conditions with the exponent N^{-1} and the coefficient $2L\sqrt{N+3}$, i.e.,*

$$|F(y(x')) - F(y(x''))| \leq 2L\sqrt{N+3}(|x'-x''|)^{1/N}, \qquad (2.1.8)$$
$$x', x'' \in [0,1]\,.$$

Proof. Any nested sequence (2.1.2) of intervals $d(M,v)$ from (2.1.3), (2.1.6) and the corresponding nested sequence (2.1.1) of subcubes $D(M,v)$ from (2.1.7) contract, respectively, to some point in $[0,1]$ and to some point in D with $M \to \infty$ because of the geometrically decreasing values 2^{-MN} and 2^{-M} which are, respectively, the length of $d(M,v)$ and the edge length of $D(M,v)$. Hence, the correspondence $y(x)$ really maps the x-intercept $[0,1]$ *onto* the hypercube D; the last property is due to the fact that for any $M \geq 1$ the union of all the subcubes $D(z_1,\dots,z_M)$ constitutes D.

The continuity of $y(x)$ is an obvious consequence of the second condition. Let $x', x'' \in [0,1]$ and $x' \neq x''$. Then there exists an integer $M \geq 1$ such that

$$2^{-(M+1)N} \leq |x'-x''| \leq 2^{-MN}\,. \qquad (2.1.9)$$

Therefore, either there is some interval $d(M,v)$ containing both points x', x'' or there are two intervals $d(M,v'), d(M,v'')$ having a common end-point and containing x' and x'' in the union. In the first case, $y', y'' \in D(M,v)$ and

$$|y_j(x') - y_j(x'')| \leq 2^{-M}, \quad 1 \leq j \leq N\,. \qquad (2.1.10)$$

In the second case, $y(x') \in D(M,v'), y(x'') \in D(M,v'')$, but the subcubes $D(M,v')$ and $D(M,v'')$ are contiguous due to Condition 2. This means that for some particular index k, $1 \leq k \leq N$,

$$|y_k(x') - y_k(x'')| \leq 2^{-(M-1)}\,; \qquad (2.1.11)$$

but for all integer values $j \neq k$, $1 \leq j \leq N$, the statement (2.1.10) is still true.

From (2.1.10) and (2.1.11),

$$
\begin{aligned}
||y(x') - y(x'')|| &= \left\{ \sum_{l=1}^{N} [y_l(x') - y_l(x'')]^2 \right\}^{1/2} \\
&\leq \left\{ (N-1)2^{-2M} + 2^{-2(M-1)} \right\}^{1/2} = 2^{-M}\sqrt{N+3},
\end{aligned}
$$

and, in consideration of (2.1.9), we derive the estimate

$$
||y(x') - y(x'')|| \leq 2\sqrt{N+3}(|x' - x''|)^{1/N}, \tag{2.1.12}
$$

whence it follows that the Euclidean distance between the points $y(x')$ and $y(x'')$ vanishes with $|x' - x''| \to 0$. Finally, employ (1.3), (2.1.12) to substantiate the relation (2.1.9) for the function $F(y(x)), x \in [0,1]$, which is the superposition of the Lipschitzian function $F(y), y \in D$, and the introduced space-filling curve $y(x)$.

$\square$

Once again, we recall that the space-filling curve (or Peano curve) $y(x)$ is defined as a limit object emerging in some sequential construction. Therefore, in practical application some appropriate approximations to $y(x)$ are to be used. Particular techniques for computing such approximations (with any preset accuracy) are suggested and substantiated in [46, 132, 134, 138]. Some of them are presented in the next section.

2.2 Approximations to Peano Curves

2.2.1 Partitions and Numerations

The left-end-points v of the subintervals $d(M,v)$ introduced in Sect. 2.1 are strictly ordered from the left to the right along the x-axis, which induces a strict order for the corresponding vectors $(z_1,\dots,z_M)$ from (2.1.6); vectors $(0,\dots,0)$ and $(2^N - 1,\dots,2^N - 1)$ are, respectively, the minimum and the maximum elements of this order.

Definition 2.2. The vector $(z''_1,\dots,z''_M), M \geq 1$, is said to *precede* the vector $(z'_1,\dots,z'_M)$ if either $z'_1 < z''_1$ or there exists an integer $k, 1 \leq k < M$, such that $z'_j = z''_j$, $1 \leq j \leq k$, and $z'_{k+1} < z''_{k+1}$. Two vectors $(z'_1,\dots,z'_M)$, $(z''_1,\dots,z''_M)$ and the corresponding subcubes $D(z'_1,\dots,z'_M)$, $D(z''_1,\dots,z''_M)$ are said to be *adjacent* if one of these vectors precedes the other and there is no third vector $(z_1,\dots,z_M)$ satisfying the relations

$$
(z'_1,\dots,z'_M) \prec (z_1,\dots,z_M) \prec (z''_1,\dots,z''_M)
$$

or

$$
(z''_1,\dots,z''_M) \prec (z_1,\dots,z_M) \prec (z'_1,\dots,z'_M)\,;
$$

here $\prec$ is the precedence sign.

From this definition, if two subintervals $d(M,v'), d(M,v'')$ have a common end-point, then the corresponding vectors $(z'_1,\dots,z'_M)$, $(z''_1,\dots,z''_M)$ from (2.1.6) have to be adjacent.

Therefore, Condition 2 from Sect. 2.1 is possible to interpret as the necessity for any two adjacent subcubes $D(z'_1,\dots,z'_M), D(z''_1,\dots,z''_M)$ to have a common face (i.e., to be contiguous).

Introduce the auxiliary hypercube

$$\Delta = \{y \in R^N : \, -2^{-1} \le y_j \le 3 \cdot 2^{-1}, \, 1 \le j \le N\} \tag{2.2.1}$$

and designate $\Delta(s), 0 \le s \le 2^N - 1$, the subcubes constituting the first partition of Δ. Due to the special choice of Δ set by (2.2.1), the central points $u(s)$ of the corresponding subcubes $\Delta(s)$ (in the sequel, these points are referred as *centers*) are N-dimensional binary vectors (each coordinate is presented by some binary digit).

Install the numeration of the above centers (and, hence, the numeration of the corresponding subcubes $\Delta(s)$) by the relations

$$u_j(s) = (\beta_j + \beta_{j-1}) \bmod 2, \quad 1 \le j \le N, \quad u_N(s) = \beta_{N-1}, \tag{2.2.2}$$

where $\beta_j, 0 \le j < N$, are the digits in the binary presentation of the number s:

$$s = \beta_{N-1} 2^{N-1} + \ldots + \beta_0 2^0 \,. \tag{2.2.3}$$

Theorem 2.2. *The numeration of subcubes $\Delta(s)$ set by the relations (2.2.2), (2.2.3) ensures that:*

1. *All the centers $u(s), 0 \le s \le 2^N - 1$, are different.*
2. *Any two centers $u(s), u(s+1), 0 \le s < 2^N - 1$, are different just in one coordinate.*
3.

$$u(0) = (0,\dots,0,0), \quad u(2^N - 1) = (0,\dots,0,1) \,. \tag{2.2.4}$$

Proof. **1.** Consider the first statement and assume the opposite, i.e., that the relations (2.2.2) juxtapose the same center to the different numbers s, s':

$$u(s) = u(s'), \quad s \ne s', \quad 0 \le s, s' \le 2^N - 1 \,; \tag{2.2.5}$$

where s is from (2.2.3) and s' is also given in the binary form

$$s' = \beta'_{N-1} 2^{N-1} + \ldots + \beta'_0 2^0 \,. \tag{2.2.6}$$

From (2.2.2), (2.2.3), (2.2.6) and the first equality in (2.2.5) follows that

$$\beta_{N-1} = \beta'_{N-1} \tag{2.2.7}$$

and

$$(\beta_j + \beta_{j-1}) \bmod 2 = (\beta'_j + \beta'_{j-1}) \bmod 2, \quad 1 \le j < N\,. \tag{2.2.8}$$

The relations (2.2.8) imply that for any integer j, $1 \le j < N$, either the equalities

$$\beta_j = \beta'_j, \quad \beta_{j-1} = \beta'_{j-1} \tag{2.2.9}$$

or the equalities

$$\beta_j = \rceil \beta'_j, \quad \beta_{j-1} = \rceil \beta'_{j-1} \tag{2.2.10}$$

have to be true; here $\rceil$ is the negation symbol inverting the value of the binary digit (i.e., $\rceil 0 = 1$, $\rceil 1 = 0$).

Suppose that (2.2.9) is true for any integer j in the range $1 \le j \le k < N$. This implies the validity of the relations

$$\beta_k = \beta'_k, \quad \beta_{k-1} = \beta'_{k-1}\,. \tag{2.2.11}$$

If $k+1 < N$, then the conditions (2.2.10) could not be met for $j = k+1$ because the equalities

$$\beta_{k+1} = \rceil \beta'_{k+1}, \quad \beta_k = \rceil \beta'_k$$

are in contradiction with (2.2.11). Case $k+1 = N$, in consideration of (2.2.7), (2.2.9), leads to the conclusion that $s = s'$ which is in contradiction with the assumption (2.2.5).

Another option, i.e., the supposition that (2.2.10) is true for any integer j in the range $1 \le j \le k < N$, being considered in the analogous way, also brings a contradiction.

2. Assume that the number s from (2.2.3) satisfies the inequalities $0 \le s < 2^N - 1$. Then there exists an integer k, $1 \le k \le N$, such that the binary form of the next number $s+1$ can be written as

$$s+1 = \beta_{N-1}2^{N-1} + \ldots + \beta_{N-k+1}2^{N-k+1} + \rceil \beta_{N-k}2^{N-k} + \ldots + \rceil \beta_0 2^0\,. \tag{2.2.12}$$

From (2.2.2), (2.2.3), (2.2.12) and the relations

$$(\beta_j + \beta_{j-1}) \bmod 2 = (\rceil \beta_j + \rceil \beta_{j-1}) \bmod 2, \quad 1 \le j < N, \tag{2.2.13}$$

follows that

$$u_j(s+1) = u_j(s), \quad 1 \le j \le N, \quad j \ne N-k+1\,. \tag{2.2.14}$$

If $k = 1$, then

$$u_N(s+1) = \rceil \beta_{N-1} = \rceil u_N(s)\;; \tag{2.2.15}$$

otherwise, i.e., if $1 < k \leq N$,

$$\begin{aligned} u_{N-k+1}(s+1) &= (\beta_{N-k+1} + \rceil \beta_{N-k}) \bmod 2 \\ &= \rceil (\beta_{N-k+1} + \beta_{N-k}) \bmod 2 = \rceil u_{N-k+1}(s)\;. \end{aligned} \tag{2.2.16}$$

Hence, the second statement of the Theorem is also proved.

3. The relations (2.2.4) directly follow from (2.2.2), (2.2.3); the value for $u(0)$ corresponds to $\beta_0 = \ldots = \beta_{N-1} = 0$ and the value for $u(2^N - 1)$ corresponds to $\beta_0 = \ldots = \beta_{N-1} = 1$. □

The suggested scheme (2.2.2), (2.2.3) is possible to use for numbering the subcubes $D(z_1), 0 \leq z_1 \leq 2^N - 1$, constituting the first partition of the hypercube D from (1.2). This will ensure that adjacent subcubes have a common face. To do so we introduce the linear mapping

$$g(y) = 2y + p, \quad y \in R^N, \tag{2.2.17}$$

with

$$p = (2^{-1}, \ldots, 2^{-1}) \in R^N, \tag{2.2.18}$$

meeting the condition

$$g(D) = \Delta \tag{2.2.19}$$

and assume that the subcube $D(z_1)$ has the number $z_1 = s$ if and only if $D(z_1)$ is the inverse image of $\Delta(s)$, i.e.,

$$g(D(z_1)) = \Delta(s)\;. \tag{2.2.20}$$

To employ the above approach for numbering the second partition subcubes $D(z_1, z_2)$, $0 \leq z_2 \leq 2^N - 1$, from $D(z_1), 0 \leq z_1 \leq 2^N - 1$, we introduce the linear mappings

$$g(z_1; y) = 2^2\{y - [u(z_1) - p]2^{-1}\} + p \tag{2.2.21}$$

meeting the conditions

$$g(z_1; D(z_1)) = \Delta, \quad 0 \leq z_1 \leq 2^N - 1,$$

and assume that the subcube $D(z_1,z_2)$ from $D(z_1)$ has the index $z_2 = s$ if and only if $D(z_1,z_2)$ is the inverse image of $\Delta(s)$, i.e.,

$$g(z_1;D(z_1,z_2)) = \Delta(s)\ . \tag{2.2.22}$$

Note that $u(z_1)$ from (2.2.21) is the center of the subcube $\Delta(z_1)$ juxtaposed to $D(z_1)$ by the relation (2.2.20).

The suggested numeration of the second-partition subcubes $D(z_1,z_2)$ with indexes z_2, $0 \le z_2 \le 2^N - 1$, by means of the scheme (2.2.21), (2.2.22) ensures contiguity of any two adjacent subcubes $D(z_1,z_2)$, $D(z_1,z_2+1)$, $0 \le z_2 < 2^N - 1$, from the same cube $D(z_1)$. But there is still a problem to be solved. Any two subcubes $D(z_1,2^N-1)$ and $D(z_1+1,0)$, $0 \le z_1 < 2^N - 1$, are adjacent too and, therefore, they should also have a common face. This means that there should be some special linkage in the numerations of elements $D(z_1,z_2)$, $0 \le z_2 \le 2^N - 1$, from different subcubes $D(z_1)$, $0 \le z_1 \le 2^N - 1$.

To provide the basis for such a linkage we, first, introduce a variety of numerations for the elements $\Delta(s), 0 \le s \le 2^N - 1$. As follows from (2.2.4), the numeration defined by the rules (2.2.2), (2.2.3) ensures that the initial center $u(0)$ is the zero-vector and that the centers $u(0)$ and $u(2^N-1)$ differ only in the Nth coordinate. The *permutation* of u_N and u_t in $u(s)$ resulting in the vector designated

$$u^t(s) = (u_1(s),\dots,u_{t-1}(s),u_N(s),u_{t+1}(s),\dots,u_{N-1}(s),u_t(s)),$$

$1 \le t \le N$, does not change the initial vector, i.e.,

$$u^t(0) = u(0), \quad 1 \le t \le N,$$

and moves the only nonzero coordinate in $u(2^N-1)$ to the t-th position, which means that $u^t(0)$ and $u^t(2^N-1)$ are different only in the t-th coordinate:

$$u_i^t(2^N-1) = \begin{cases} u_i^t(0), & i \ne t, \\ \rceil u_i^t(0), & i = t. \end{cases} \tag{2.2.23}$$

The *addition* of some binary vector $q \in R^N$ to the vectors $u^t(s)$, $0 \le s \le 2^N - 1$, resulting in the new vectors $u^{tq}(s)$ from

$$u_i^{tq}(s) = (u_i^t(s) + q_i)\mathrm{mod}\,2, \quad 1 \le i \le N, \tag{2.2.24}$$

ensures that the used vector q is the center of the initial subcube $\Delta(0)$, i.e.,

$$u^{tq}(0) = q\ .$$

Thereby, these two operations allow us to construct the numeration which assures that the initial subcube $\Delta(0)$ has the preset center q and the centers of the subcubes $\Delta(0)$ and $\Delta(2^N-1)$ are different only in the t-th coordinate, i.e., they satisfy the condition (2.2.23) for any preset integer t, $1 \le t \le N$.

Next, we introduce the integer function

$$l(s-1) = l(s) = \min\{j : 2 \le j \le N, \beta_{j-1} = 1\}, \tag{2.2.25}$$

where s, $2 \le s \le 2^N - 2$, is an *even* integer, β_{j-1} is from (2.2.3), and

$$l(0) = l(2^N - 1) = 1\ . \tag{2.2.26}$$

As it follows from (2.2.2), (2.2.3), (2.2.12), (2.2.14), and (2.2.25),

$$l(s) = N - k + 1\ ;$$

hence, $l(s)$ from (2.2.25) is the number of the only coordinate in which the centers $u(s)$ and $u(s-1)$ are different, i.e.,

$$u_i(s) = \begin{cases} \rceil u_i(s-1),\ i = l(s), \\ u_i(s-1),\quad i \ne l(s). \end{cases}$$

We assume that the centers of the initial subcube $D(z_1, 0)$ and of the last subcube $D(z_1, 2^N - 1)$ from the cube $D(z_1)$, $0 \le z_1 \le 2^N - 1$, should also be different just in one coordinate and that the function (2.2.25) determines the number of this coordinate.

We also introduce a binary function

$$w_i(s+1) = w_i(s) = \begin{cases} \rceil u_i(s),\ i = 1, \\ u_i(s),\quad 2 \le i \le N, \end{cases} \tag{2.2.27}$$

where s is supposed to be the *odd* number and

$$w(0) = u(0)\ ; \tag{2.2.28}$$

as already mentioned $\rceil$ is a negation sign inverting the value of the binary digit. The binary vector $w(z_1)$ is to be used for determining the position of the center of the subcube $\Delta(0)$ employed in (2.2.22) with $z_2 = 0$, i.e., it will used to set the value of the vector q embedded into (2.2.24).

Now, we suggest carrying out the numbering of the subcubes $D(z_1, z_2)$, $0 \le z_2 \le 2^N - 1$, in the following way. From (2.2.25) to (2.2.28), compute the values $l = l(z_1)$, $w = w(z_1)$ for the given index z_1, $0 \le z_1 \le 2^N - 1$. Select

$$t = l(z_1), \quad q = w(z_1) \tag{2.2.29}$$

and, using the above described permutations and additions, determine the vectors $u^{tq}(s)$, $0 \le s \le 2^N - 1$; note that the vector function $u^{tq}(s)$ may be different for different values of z_1, $0 \le z_1 \le 2^N - 1$. Finally, we employ (2.2.22) to number the subcubes $D(z_1, z_2)$, $0 \le z_2 \le 2^N - 1$, under the condition that the index s of

Table 2.1 Vectors (β_0, β_1) with β_0, β_1 from (2.2.3), $u(s)$ from (2.2.2), $w(s)$ from (2.2.27), (2.2.28) as the functions of s, $0 \le s \le 3$ (case $N = 2$, $M = 2$)

s	(β_0, β_1)	(u_1, u_2)	l	(w_1, w_2)
0	(0,0)	(0,0)	1	(0,0)
1	(1,0)	(1,0)	2	(0,0)
2	(0,1)	(1,1)	2	(0,0)
3	(1,1)	(0,1)	1	(1,1)

Table 2.2 Centers $u^{tq}(s)$ from (2.2.24), (2.2.29) juxtaposed to the subcubes $\Delta(s)$ from (2.2.22) as the functions of z_1, $0 \le z_1 \le 3$

z_1	$u^{tq}(0)$	$u^{tq}(1)$	$u^{tq}(2)$	$u^{tq}(3)$
0	(0,0)	(0,1)	(1,1)	(1,0)
1	(0,0)	(1,0)	(1,1)	(0,1)
2	(0,0)	(1,0)	(1,1)	(0,1)
3	(1,1)	(1,0)	(0,0)	(0,1)

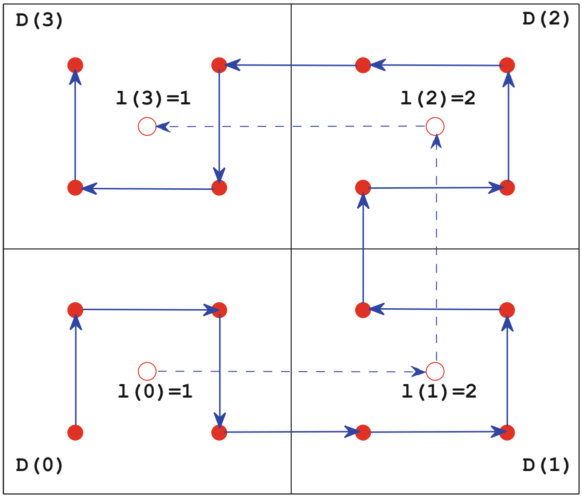

Fig. 2.3 Case $N = 2, M = 2$. *Solid-line arrows* link the centers (marked by the *red dots*) of the second-partition subsquares in the order of numbering providing contiguity of the adjacent squares. The *dotted-line arrows* link the centers of the first-partition subsquares

the subcube $\Delta(s)$ from the right-hand side of (2.2.22) has to be identical with the number of the center $u^{tq}(s)$ of this subcube generated for the given index z_1 in the above way.

Tables 2.1, 2.2 and Fig. 2.3 illustrate the role of the numbers l from (2.2.25), (2.2.26) and of the vectors w from (2.2.27), (2.2.28) in establishing the numeration of the second-partition subcubes $D(z_1, z_2)$ which were already pictured in Fig. 2.1 (case $N = 2, M = 2$). Table 2.1 presents the vectors (β_0, β_1) with β_0, β_1 from (2.2.3), $u(s)$ from (2.2.2), $w(s)$ from (2.2.27), (2.2.28), and the number $l(s)$ from (2.2.25), (2.2.26) as the functions of s, $0 \le s \le 3$. Table 2.2 contains the sets of the centers $u^{tq}(s)$, $0 \le s \le 3$, juxtaposed to the subcubes $\Delta(s)$ from (2.2.22) for the given index z_1, $0 \le z_1 \le 3$. These centers are computed from (2.2.24) under the conditions (2.2.29).

Circles in Fig. 2.3 mark the centers of the first-partition subcubes $D(z_1)$, $0 \le z_1 \le 3$. These centers are linked with the dotted-line arrows in the order of numeration. The corresponding centers $u(s)$ juxtaposed to the subcubes $\Delta(s)$ from (2.2.20) while numbering $D(z_1), 0 \le z_1 \le 3$, are given in the third column of Table 2.1.

Red dots in Fig. 2.3 mark the centers of the second-partition subcubes $D(z_1, z_2)$. Centers of the adjacent subcubes are linked with solid-line arrows streaming from the initial subcube $D(0,0)$ to the last subcube $D(3,3)$. Centers $u^{tq}(s)$ juxtaposed to the subcubes $\Delta(s)$ from (2.2.22) while numbering $D(z_1, z_2), 0 \le z_2 \le 3$, are given in Table 2.2 (each row of this table corresponds to some particular value of the first index z_1).

The picture allows us to clarify the role of the vector $w(z_1)$ from the last column of Table 2.1 which points the position for the center of the initial subcube $D(z_1, 0)$ from the next partition of the cube $D(z_1)$. As it is clear from the picture, the values $l(z_1)$ and $w(z_1)$ are coherent in such a way that the centers of the subcubes $D(z_1, 3)$ and $D(z_1 + 1, 0), 0 \le z_1 < 3$, are different just in one coordinate, though these adjacent subcubes belong to different cubes of the foregoing partition. The centers of such cubes are linked with thick arrows in Fig. 2.3.

Let us consider now how to link numerations in subsequent partitions. Note that the already considered two cases (numbering in the first partition and numbering in the second partition) were treated in somewhat different ways. In the first case we used the relations (2.2.17)–(2.2.20) juxtaposing the centers $u(s)$ from (2.2.2), (2.2.3) to the cubes $\Delta(s)$. In the second case we employed the relations (2.2.21), (2.2.22) and the cubes $\Delta(s)$ which juxtaposed the centers $u^{tq}(s)$ from (2.2.24) linked with the corresponding centers $u(z_1)$ due to (2.2.25)–(2.2.29); in fact, each center $u^{tq}(z_2)$ depends also on some value z_1 (we use the short notation $u^{tq}(s)$ just to compact the writing; this should not cause any confusion).

It is possible to unify both considered cases by introducing the linear mappings

$$g(z_1, \dots, z_M; y) = 2^{M+1} \left\{ y - \sum_{j=1}^{M} [u^{tq}(z_j) - p] 2^{-j} \right\} + p \tag{2.2.30}$$

and assuming that the subcube $D(z_1, \dots, z_M, z_{M+1})$ is characterized by the index z_{M+1} if and only if

$$g(z_1, \dots, z_M; D(z_1, \dots, z_M, z_{M+1})) = \Delta(s)\ ; \tag{2.2.31}$$

here the cube $\Delta(s)$ is the one having the center $u^{tq}(s)$ from (2.2.24) with

$$t = t(z_{M+1}) = \begin{cases} N, & M = 0, \\ l(z_M), & M > 0, \end{cases} \tag{2.2.32}$$

and

$$q = q(z_{M+1}) = \begin{cases} (0, \dots, 0) \in R^N, & M = 0, \\ w(z_M), & M > 0. \end{cases} \tag{2.2.33}$$

If $M = 1$, then the relations (2.2.30) and (2.2.32), (2.2.33) are, respectively, identical to the relations (2.2.21) and (2.2.29). If $M = 0$, which corresponds to the numeration in the first partition, then (2.2.30) is identical to (2.2.17) and application of (2.2.24) in conjunction with (2.2.32), (2.2.33) yields

$$u^{tq}(s) = u(s), \quad 0 \le s \le 2^N - 1 \, .$$

Thus, (2.2.30), (2.2.31) together with (2.2.24), (2.2.32), (2.2.33), and (2.2.25)–(2.2.28) combine the rules for numbering in the first and in the second partitions.

Moreover, it is possible to generalize this scheme for any $M > 1$. The only amendment needed is to accept that the rule (2.2.24) transforming $u(s)$ into $u^{tq}(s)$ has to be appended with similar transformation for the vector $w(s)$

$$w_i^{tq}(s) = (w_i^t(s) + q_i) \bmod 2, \quad 1 \le i \le N, \tag{2.2.34}$$

and with the following transformation for the integer $l(s)$

$$l^t(s) = \begin{cases} N, & l(s) = t, \\ t, & l(s) = N, \\ l(s), & l(s) \ne N \text{ and } l(s) \ne t, \end{cases} \tag{2.2.35}$$

where t is the pointer used in the permutations yielding $u^t(s)$ and $w^t(s)$.

It has to be clarified that all the values $u(z_M)$, $l(z_M)$, $w(z_M)$ embedded into the right-hand sides of the expressions (2.2.27), (2.2.32), (2.2.33) to produce the subsequent auxiliary values w, t, q for the numeration in the next partition are functions of the corresponding values u, l, w generated in the foregoing partition. Once again, we stress that $u^{tq}(z_{M+1})$, $w^{tq}(z_{M+1})$, and $l^t(z_{M+1})$ are dependent on $z_1, .., z_M$ if $M \ge 1$.

Theorem 2.3. *The introduced system of the linked numerations ensures the contiguity of any two adjacent subcubes from any Mth $(M \ge 1)$ partition of the cube D from (1.2); see [132].*

Proof. **1.** Consider any two adjacent subcubes $D(z_1)$ and $D(z_1 + 1)$, $0 \le z_1 < 2^N - 1$, of the first partition mapped by the correspondence (2.2.17) onto the auxiliary subcubes $\Delta(z_1)$ and $\Delta(z_1 + 1)$; see (2.2.20). As already proved in Theorem 2.2, the centers $u(z_1)$, $u(z_1 + 1)$, $0 \le z_1 < 2^N - 1$, of the subcubes $\Delta(z_1)$, $\Delta(z_1 + 1)$ are different just in one coordinate if they are numbered in accordance with the rules (2.2.2), (2.2.3). That is, the subcubes $\Delta(z_1)$, $\Delta(z_1 + 1)$ have to be contiguous and, therefore, the corresponding cubes $D(z_1)$, $D(z_1 + 1)$ are contiguous too.

Suppose that the Theorem is true for any adjacent subcubes of the k-th partition of the cube D, where $1 \le k \le M$. Then it is left to prove that it is also true for the adjacent subcubes of the $(M + 1)$st partition.

As long as for the given $z_1, 0 \le z_1 \le 2^N - 1$, the set of all the subcubes $D(z_1, z_2, \ldots, z_{M+1})$ constitutes the Mth partition of the cube $D(z_1)$, then, due to the assumption, all the adjacent subcubes $D(z_1, z_2, \ldots, z_{M+1})$ from $D(z_1)$ are contiguous. Thus, it is left to demonstrate that for any given z_1, $0 \le z_1 \le 2^N - 1$, the subcubes

$$D(z_1, 2^N-1, \dots, 2^N-1) \quad \text{and} \quad D(z_1+1, 0, \dots, 0) \tag{2.2.36}$$

of the $(M+1)$st partition are also contiguous.

In accordance with (2.2.30), (2.2.31), the point

$$y(z_1, \dots, z_M) = \sum_{j=1}^{M} [u^{tq}(z_j) - p] 2^{-j} \tag{2.2.37}$$

belongs to all the subcubes $D(z_1, \dots, z_M, z_{M+1})$, $0 \le z_{M+1} \le 2^N - 1$, from $D(z_1, \dots, z_M)$. Therefore, in the sequel, the point (2.2.37) is to be referred to as the *center of the subcube* $D(z_1, \dots, z_M)$. Then, the necessary and sufficient condition for the cubes from (2.2.36) to be contiguous could be stated as the existence of a number l, $1 \le l \le N$, such that the centers of these cubes satisfy the requirement

$$\left| y_i(z_1, 2^N-1, \dots, 2^N-1) - y_i(z_1+1, 0, \dots, 0) \right| = \\ = \begin{cases} 0, & i \ne l, \\ 2^{-(M+1)}, & i = l; \end{cases} \tag{2.2.38}$$

i.e., the centers of the cubes from (2.2.36) have to be different just in one, l-th, coordinate and the absolute difference in this coordinate has to be equal to the edge length for the $(M+1)$st partition subcube. We proceed with computing the estimate for the left-hand side of (2.2.38) for the accepted system of numeration.

2. Introduce the notations $u(z_1, \dots, z_M; z_{M+1})$, $w(z_1, \dots, z_M; z_{M+1})$ for the vectors $u^{tq}(z_{M+1})$, $w^{tq}(z_{M+1})$ corresponding to the particular subcube $D(z_1, \dots, z_M, z_{M+1})$ from the cube $D(z_1, \dots, z_M)$.

Suppose that $z_1 = 2k - 1$, $1 \le k \le 2^{N-1} - 1$, i.e., z_1 is the *odd* number and $z_1 < 2^N - 1$, and consider the sequence of indexes $z_1, z_2, \dots; z_j = 2^N - 1$, $j \ge 2$.

First, we study the sequence of numbers $t(z_j)$, $j \ge 1$, corresponding to the introduced sequence of indexes. From (2.2.32),

$$t(z_1) = N \tag{2.2.39}$$

and, as it follows from (2.2.25),

$$l(z_1) = l(z_1+1) > 1\ . \tag{2.2.40}$$

Now, from (2.2.35), (2.2.39), (2.2.40), we derive that $t(z_2) = l(z_1)$.

In accordance with (2.2.26), $l(z_2) = l(2^N - 1) = 1$; hence, due to (2.2.32), (2.2.35), (2.2.40), we get the value $t(z_3) = 1$.

Reproduction of the above reasoning for $z_3 = 2^N - 1$ and $z_4 = 2^N - 1$ yields the estimates $t(z_4) = N$, $t(z_5) = 1$; and by inductive inference, finally, we obtain the dependence

$$t(z_j) = \begin{cases} 1, & j = 2\nu + 1,\ \nu \geq 1, \\ l(z_1), & j = 2, \\ N, & j = 1,\ j = 2\nu,\ \nu \geq 2. \end{cases} \tag{2.2.41}$$

From (2.2.33), $q(z_1) = (0, \dots, 0)$ and, with account of (2.2.24), (2.2.33), (2.2.34) and (2.2.41), we derive the relations

$$u^{tq}(z_1) = u(z_1), \quad q(z_2) = w^{tq}(z_1) = w(z_1)\,. \tag{2.2.42}$$

Now, it is possible to analyze the second-partition subcubes from $D(z_1)$. From (2.2.4), (2.2.24), (2.2.41), (2.2.42) follows that

$$u_i(z_1; 2^N - 1) = \begin{cases} w_i(z_1), & i \neq l(z_1), \\ \rceil w_i(z_1), & i = l(z_1), \end{cases} \quad 1 \leq i \leq N, \tag{2.2.43}$$

whence, in consideration of (2.2.27), (2.2.40),

$$u_i(z_1; 2^N - 1) = \begin{cases} u_i(z_1), & i \neq 1,\ i \neq l(z_1), \\ \rceil u_i(z_1), & i = 1,\ i = l(z_1), \end{cases} \quad 1 \leq i \leq N. \tag{2.2.44}$$

In the analogous way, from (2.2.4), (2.2.27), (2.2.33), (2.2.34), (2.2.40), and (2.2.41), obtain

$$q_i(z_3) = w_i(z_1; 2^N - 1) = \begin{cases} u_i(z_1), & i \neq l(z_1), \\ \rceil u_i(z_1), & i = l(z_1), \end{cases} \quad 1 \leq i \leq N. \tag{2.2.45}$$

Next, from (2.2.4), (2.2.24), (2.2.40)–(2.2.45), establish the identity

$$u(z_1, 2^N - 1; 2^N - 1) = u(z_1; 2^N - 1) \tag{2.2.46}$$

and, due to (2.2.27), (2.2.34), derive the relation

$$w_i(z_1, 2^N - 1; 2^N - 1) =$$

$$= \begin{cases} u_i(z_1), & l(z_i) \neq i \neq 1,\ l(z_1) \neq i \neq N,\ l(z_1) = i = N, \\ \rceil u_i(z_1), & l(z_i) \neq i = 1,\ l(z_1) \neq i = N,\ l(z_1) = i \neq N, \end{cases}$$

for $1 \leq i \leq N$, which, due to (2.2.33), represents also the vector $q(z_4)$. By repetition of the above discourse for $z_4 = 2^N - 1$, obtain the identities

$$u(z_1, 2^N-1, 2^N-1; 2^N-1) = u(z_1; 2^N-1),$$
$$w(z_1, 2^N-1, 2^N-1; 2^N-1) = w(z_1; 2^N-1), \tag{2.2.47}$$

whence, due to (2.2.41) and (2.2.45), follows

$$t(z_5) = t(z_3) = 1, \quad q(z_5) = q(z_3) = w(z_1; 2^N-1) .$$

This means that each subsequent repetition of the above discourse will just add one more parameter (equal to 2^N-1) into the left-hand side of (2.2.47).

Therefore, for any $M > 1$

$$u(z_1, 2^N-1, \ldots, 2^N-1; 2^N-1) = u(z_1; 2^N-1),$$

which being substituted into (2.2.37) yields

$$y(z_1, \ldots, z_M, z_{M+1}) = y(z_1, 2^N-1, \ldots, 2^N-1) =$$
$$= \frac{1}{2}\{u(z_1) + (1-2^{-M})u(z_1; 2^N-1) - (2-2^{-M})p\}. \tag{2.2.48}$$

Proceed to the numbering of subcubes from $D(z_1+1)$ where z_1+1 is the even number ($2 \le z_1+1 \le 2^N-2$) and consider the sequence of indexes $z_1+1, z_2, \ldots$ under the condition that $z_j = 0$, $j \ge 2$.

From (2.2.27),

$$w(z_1+1) = w(z_1) \tag{2.2.49}$$

and, in accordance with (2.2.32), (2.2.33),

$$t(z_1+1) = N, \quad q(z_1+1) = (0, \ldots, 0) .$$

Therefore, from (2.2.24), (2.2.34)

$$u^{tq}(z_1+1) = u(z_1+1), \quad q(z_2) = w(z_1) .$$

For $z_2 = 0$, from (2.2.4), (2.2.24), (2.2.34), obtain that

$$u^t(0) = w^t(0) = (0, \ldots, 0), \quad 1 \le t \le N, \tag{2.2.50}$$
$$u(z_1+1; 0) = w(z_1), \quad q(z_3) = w(z_1+1; 0) = w(z_1) . \tag{2.2.51}$$

One more iteration (for $z_3 = 0$) results in similar relations

$$u(z_1+1, 0; 0) = w(z_1), \quad q(z_4) = w(z_1+1, 0; 0) = w(z_1),$$

which means that the successive application of (2.2.24), (2.2.34), in consideration of (2.2.49)–(2.2.51), ensures the validity of

$$u(z_1+1,0,\dots,0;0) = w(z_1) \tag{2.2.52}$$

for any $M > 1$. By plugging (2.2.52) into (2.2.37), obtain

$$\begin{aligned} y(z_1+1,z_2,\dots,z_{M+1}) &= y(z_1+1,0,\dots,0) = \\ &= \frac{1}{2}\{u(z_1+1)+(1-2^{-M})w(z_1)-(2-2^{-M})p\}. \end{aligned} \tag{2.2.53}$$

Finally from (2.2.48) and (2.2.53), we derive the estimate

$$\begin{aligned} \delta_i &= |y_i(z_1,2^N-1,\dots,2^N-1)-y_i(z_1+1,0,\dots,0)| = \\ &= \frac{1}{2}|u_i(z_1)-u_i(z_1+1)+u_i(z_1;2^N-1)- \\ &\quad -w_i(z_1)+2^{-M}[w_i(z_1)-u_i(z_1;2^N-1)]|. \end{aligned}$$

From the comment following the definition (2.2.25) and from (2.2.43),

$$u_i(z_1+1)=u_i(z_1), \quad u_i(z_1;2^N-1)=w_i(z_1), \quad i \neq l(z_1)\;.$$

Therefore, $\delta_i = 0$ if $i \neq l(z_1)$. Consider the case when $i = l(z_1)$. In accordance with (2.2.27),

$$w_l(z_1) = u_l(z_1)$$

and, in consideration of (2.2.25) and (2.2.44),

$$u_l(z_1+1) = \rceil u_l(z_1) = u_l(z_1;2^N-1),$$

which means that $\delta_l = 2^{-(M+1)}$. So, the relations (2.2.38) are validated for the odd number z_1, $1 < z_1 < 2^N-1$, with $l = l(z_1)$.

3. Suppose that $z_1 = 2k$, $1 \le k \le 2^{N-1}-1$, i.e., $z_1 > 0$ is the *even* integer and consider the sequence of indexes $z_1,z_2,\dots;z_j = 2^N-1$, $j \ge 2$ (note that (2.2.41) is valid for the elements of this sequence).

In consideration of the linking between $u(s)$ and $u(s-1)$ introduced by (2.2.25) for the case of the even integer s and due to (2.2.27), derive

$$w_i(z_1) = w_i(z_1-1) = \begin{cases} \rceil u_i(z_1-1), & i=1, \\ u_i(z_1-1), & i \neq 1, \end{cases}$$

$$q_i(z_2) = w_i(z_1) = \begin{cases} \rceil u_i(z_1), & i=1,\ i=l(z_1), \\ u_i(z_1), & i\neq 1,\ i\neq l(z_1), \end{cases} \qquad 1 \le i \le N. \tag{2.2.54}$$

From $t(z_2) = l(z_1) > 1$ and (2.2.4), (2.2.24), (2.2.54),

$$u_i(z_1; 2^N - 1) = \begin{cases} \rceil w_i(z_1), \; i = l(z_1), \\ w_i(z_1), \quad i \neq l(z_1), \end{cases} = \begin{cases} \rceil u_i(z_1), \; i = 1, \\ u_i(z_1), \quad i \neq 1, \end{cases} \qquad 1 \le i \le N, \tag{2.2.55}$$

for $1 \le i \le N$, and due to (2.2.27), (2.2.34),

$$q(z_3) = w(z_1; 2^N - 1) = u(z_1) \, . \tag{2.2.56}$$

By analogy, and in consideration of (2.2.56), obtain

$$t(z_3) = 1, \quad q(z_3) = u(z_1), \tag{2.2.57}$$

$$u(z_1, 2^N - 1; 2^N - 1) = u(z_1; 2^N - 1), \tag{2.2.58}$$

$$q_i(z_4) = w_i(z_1, 2^N - 1; 2^N - 1) = \begin{cases} \rceil u_i(z_1), \; i = 1, \; i = N, \\ u_i(z_1), \quad i \neq 1, \; i \neq N, \end{cases}$$

for $1 \le i \le N$. One more iteration yields $t(z_4) = N$,

$$\begin{aligned} &u(z_1, 2^N - 1, 2^N - 1; 2^N - 1) = u(z_1; 2^N - 1), \\ &q(z_5) = w(z_1, 2^N - 1, 2^N - 1; 2^N - 1) = u(z_1) \, . \end{aligned} \tag{2.2.59}$$

Next, due to (2.2.59), we have the relations

$$t(z_5) = 1, \quad q(z_5) = u(z_1),$$

$$u(z_1, 2^N - 1, 2^N - 1, 2^N - 1; 2^N - 1) = u(z_1; 2^N - 1),$$

which reproduce the state of discourse presented by (2.2.57), (2.2.58). Therefore, for any $M > 1$

$$u(z_1, 2^N - 1, \ldots, 2^N - 1; 2^N - 1) = u(z_1; 2^N - 1),$$

where $u(z_1; 2^N - 1)$ is from (2.2.55). Hence, the equality (2.2.48) is valid also for the even number $z_1 > 0$.

From (2.2.4), (2.2.28), for any $t, 1 \le t \le N$, follows

$$u^t(0) = w^t(0) = (0, \ldots, 0) \, .$$

These zero-vectors, being substituted into (2.2.24), (2.2.34), produce

$$u(z_1+1;0) = w(z_1+1),$$
$$q(z_3) = w(z_1+1;0) = w(z_1+1),$$
$$u(z_1+1,0;0) = w(z_1+1),$$
$$q(z_4) = w(z_1+1,0;0) = w(z_1+1),$$

consequently, for any $M > 1$:

$$u(z_1+1,0,\dots,0;0) = w(z_1+1), \tag{2.2.60}$$

where, in accordance with (2.2.27)

$$w_i(z_1+1) = \begin{cases} \rceil u_i(z_1+1), \ i=1, \\ u_i(z_1+1), \quad i \neq 1, \end{cases} \qquad 1 \le i \le N. \tag{2.2.61}$$

From (2.2.2), (2.2.3)

$$u_i(z_1+1) = \begin{cases} \rceil u_i(z_1), \ i=1, \\ u_i(z_1), \quad i \neq 1, \end{cases} \qquad 1 \le i \le N. \tag{2.2.62}$$

Recall in this occasion that z_1 is the even integer. Therefore, due to (2.2.61), we obtain that $w(z_1+1) = u(z_1)$. The last equality, in conjunction with (2.2.60) and (2.2.37), implies

$$\begin{aligned} y(z_1+1,z_2,\dots,z_{M+1}) &= y(z_1+1,0,\dots,0) = \\ &= \frac{1}{2}\{u(z_1+1) + (1-2^{-M})u(z_1) - (2-2^{-M})p\}. \end{aligned} \tag{2.2.63}$$

Now, from (2.2.48) and (2.2.63) follows the validity of (2.2.38) also for even indexes $z_1 > 0$ because, due to (2.2.55), (2.2.62),

$$u(z_1;2^N-1) = u(z_1+1)$$

and the vectors $u(z_1;2^N-1)$ and $u(z_1)$ are different only in the first coordinate (i.e., $l=1$); see (2.2.55).

4. Suppose that $z_1 = 0$ and consider the sequence of indexes $z_1, z_2, \dots;\ z_j = 2^N - 1,\ j \ge 2$. In this case, from (2.2.26), (2.2.32) and (2.2.35) follows the relation for the parameter t in the operation of permutation

$$t(z_j) = \begin{cases} 1, \ \ j = 2\nu, \ \nu \ge 1, \\ N, \ j = 2\nu+1, \ \nu \ge 0. \end{cases} \tag{2.2.64}$$

From (2.2.24), (2.2.28), (2.2.33), (2.2.34) and (2.2.64),

$$t(z_4) = t(z_2) = 1, \quad q(z_4) = q(z_2) = w(0),$$
$$u(0, 2^N - 1, 2^N - 1; 2^N - 1) = u(0, 2^N - 1; 2^N - 1) =$$
$$= u(0; 2^N - 1) = u(1),$$

i.e., the case $j = 4$ is the reproduction of the state of discourse at $j = 2$. Therefore for any $M > 1$:

$$u(0, 2^N - 1, \ldots, 2^N - 1; 2^N - 1) = u(0; 2^N - 1) = u(1)\,; \tag{2.2.65}$$

and formula (2.2.48) is true also for $z_1 = 0$.

Next consider the sequence of indexes $z_1 + 1, z_2, \ldots = 1, 0, 0, \ldots; z_j = 0,\ j \geq 2$ (note that (2.2.41) is true for the elements of this sequence with $l(1) = 2$).

At $z_1 = 0$, in consideration of (2.2.27),

$$u(z_1 + 1) = u(1), \quad q(z_2) = w(1) = u(0)\,. \tag{2.2.66}$$

In accordance with (2.2.24), (2.2.34),

$$u(1, 0, \ldots, 0; 0) = u(1; 0) = w(1),$$
$$w(1, 0, \ldots, 0; 0) = w(1; 0) = w(1), \tag{2.2.67}$$

where (2.2.67) is similar to (2.2.52). Therefore, formula (2.2.53) is true also for $z_1 = 0$. Thus, from (2.2.48), (2.2.53) and (2.2.2)–(2.2.4), (2.2.65), (2.2.66) follows the validity of (2.2.38) at $z_1 = 0$ with $l = 1$. $\square$

2.2.2 Types of Approximations and Their Analysis

Consider the space-filling curve $y(x)$ introduced in Theorem 2.1. This curve, continuously mapping the unit interval [0,1] onto the hypercube D from (1.2), was defined by establishing a correspondence between the subintervals $d(z_1, \ldots, z_M)$ from (2.1.3)–(2.1.6) and the subcubes $D(z_1, \ldots, z_M)$ of each Mth partition ($M = 1, 2, \ldots$) and assuming that the inclusion $x \in d(z_1, \ldots, z_M)$ induces the inclusion $y(x) \in D(z_1, \ldots, z_M)$. Therefore, for any preset accuracy ε, $0 < \varepsilon < 1$, it is possible to select a large integer $M > 1$ such that the deviation of any point $y(x), x \in d(z_1, \ldots, z_M)$, from the center $y(z_1, \ldots, z_M)$ of the hypercube $D(z_1, \ldots, z_M)$ introduced in (2.2.37) will not exceed ε (in each coordinate) because

$$|y_j(x) - y_j(z_1, \ldots, z_M)| \leq 2^{-(M+1)} \leq \varepsilon, \quad 1 \leq j \leq N\,.$$

This allows us to outline the following scheme for computing the approximation $y(z_1,\dots,z_M)$ for any point $y(x)$, $x \in [0,1]$, with the preset accuracy ε, $0<\varepsilon<1$:

1. Select the integer $M \geq -(\ln\varepsilon/\ln 2+1)$.
2. Detect the interval $d(M,v)$ containing the inverse image x, i.e., $x \in d(M,v) = [v, v+2^{-MN}]$ and estimate the indexes $z_1,\dots,z_M$ from (2.1.4), (2.1.5).
3. Compute the center $y(z_1,\dots,z_M)$ from (2.2.37). This last operation is executed by sequential estimation of the centers $u^{tq}(z_j)$, $1 \leq j \leq M$, from (2.2.24) with t from (2.2.32), (2.2.35) and q from (2.2.33), (2.2.34).

In all the above numerical examples the curve $y(x)$ was approximated by (2.2.37) at $N=2$, $M=10$.

Remark 2.1. The centers (2.2.37) constitute a uniform orthogonal net of 2^{MN} nodes in the hypercube D with mesh width equal to 2^{-M}. Therefore, all the points $x \in d(z_1,\dots,z_M)$ have the same image $y(z_1,\dots,z_M)$. But in some applications it is preferable to use a one-to-one continuous correspondence $l_M(y)$ approximating Peano curve $y(x)$ with the same accuracy as is ensured by the implementation of (2.2.37).

A piecewise-linear curve of this type is now described; it maps the interval [0,1] *into* (not onto) the cube D, but it covers the net constituted by the centers (2.2.37).

Establish the numeration of all the intervals (2.1.3) constituting the Mth partition of the interval $[0,1]$ by subscripts in increasing order of the coordinate:

$$d(z_1,\dots,z_M) = [v_i, v_i+2^{-MN}), \quad 0 \leq i \leq 2^{MN}-1\ .$$

Next, assume that the center $y(z_1,\dots,z_M)$ of the hypercube $D(z_1,\dots,z_M)$ is assigned the same number (the superscript) as the number of the subinterval $d(z_1,\dots,z_M)$ corresponding to this subcube, i.e.,

$$y^i = y(z_1,\dots,z_M), \quad 0 \leq i \leq 2^{MN}-1\ .$$

This numeration ensures that any two centers y^i, y^{i+1}, $0 \leq i < 2^{MN}-1$, correspond to the contiguous hypercubes (see Condition 2 from Sect. 2.1), which means that they are different just in one coordinate.

Consider the following curve $l(x) = l_M(x)$ mapping the unit interval $[0,1]$ into the hypercube D from (1.2):

$$l(x) = y^i + (y^{i+1}-y^i)[(w(x)-v_i)/(v_{i+1}-v_i)], \tag{2.2.68}$$

where the index i is from the conditions

$$v_i \leq w(x) \leq v_{i+1},$$

and

$$w(x) = x(1-2^{-MN}), \quad 0 \leq x \leq 1\ . \tag{2.2.69}$$

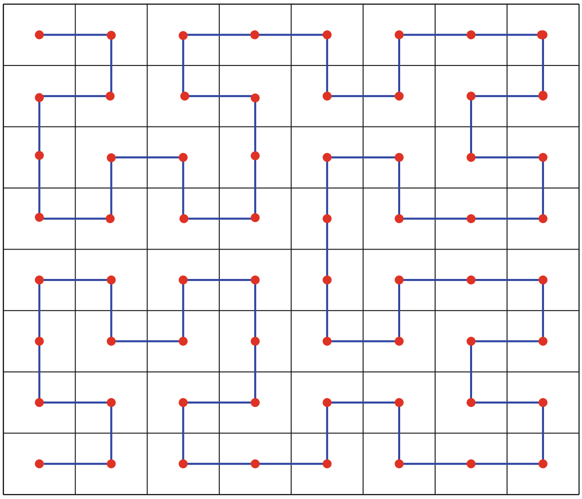

Fig. 2.4 Image of the interval $[0,1]$ generated by Peano-like piecewise-linear evolvent $l(x)$ from (2.2.68) at $N=2, M=3$; *red dots* correspond to the centers of the third-partition subsquares from (2.2.37) got through by the curve $l(x)$ in the order of established numeration

The image of any particular subinterval

$$[v_i(1-2^{-MN})^{-1}, v_{i+1}(1-2^{-MN})^{-1}], \quad 0 \le i < 2^{MN}-1, \tag{2.2.70}$$

generated by this curve is the linear segment connecting the nodes y^i, y^{i+1} and, thus, $l(x), 0 \le x \le 1$, is the piecewise-linear curve running through the centers $y_i, 0 \le i \le 2^{MN}-1$ in the order of the established numeration. The curve $l(x) = l_M(x)$ henceforth to be referred to as a *Peano-like piecewise-linear evolvent* because it approximates the Peano curve $y(x)$ from Theorem 2.1 with accuracy not worse than 2^{-M} in each coordinate; note that M is the parameter of the family of curves (2.2.68) as long as it determines the number and the positions of the nodes (2.2.37) used in the construction of $l(x)$. For the sake of illustration, Fig. 2.4 presents the image of the interval $[0,1]$ generated by $l(x)$ at $N=2, M=3$ (the corresponding centers $y^i, 0 \le i \le 63$, are marked by red dots).

Remark 2.2. The expression (2.2.68), (2.2.69) allow us to determine the point $l(x)$ for any given $x \in [0,1]$ by, first, estimating the difference

$$\Delta = (w(x)-v_i)/(v_{i+1}-v_i) = 2^{MN}(x-v_i)-x$$

and then employing (2.2.37) to compute the centers $y^i = y(z_1,\dots,z_M)$, y^{i+1} of the two adjacent subcubes of the Mth partition corresponding to the intervals $[v_i, v_{i+1}) = d(z_1,\dots,z_M)$ and $[v_{i+1}, v_{i+2})$; note that the index i is defined by the condition

$$w(x) = x(1-2^{-MN}) \in d(z_1,\dots,z_M)\,.$$

The scheme for computing centers $y(z_1,\dots,z_M)$ from (2.2.37) was already discussed in the previous subsection. As long as the adjacent centers y^i and y^{i+1} are different just in one coordinate, it is sufficient to compute only the center $y^i = y(z_1,\dots,z_M)$ and the number $\nu = \nu(z_1,\dots,z_M)$ of this coordinate. Then for any k, $1 \le k \le N$,

$$l_k(x) = y_k(z_1,\dots,z_M)+$$
$$[u_k^{tq}(z_M) - 2^{-1}]2^{-(M-1)} \times \begin{cases} 0, & k \ne \nu, \\ \Delta, & k = \nu,\ z_M = 2^N - 1, \\ -\Delta, & k = \nu,\ z_M \ne 2^N - 1, \end{cases}$$

where $u^{tq}(z_M)$ is from (2.2.37). Now, it is left to outline the scheme for computing the number ν.

Represent the sequence $z_1,\dots,z_M$ as $z_1,\dots,z_\mu,z_{\mu+1},\dots,z_M$ where $1 \le \mu \le M$ and $z_\mu \ne 2^N - 1$, $z_{\mu+1} = \dots = z_M = 2^N - 1$; note that the case $z_1 = \dots = z_M = 2^N - 1$ is impossible because the center $y(2^N - 1,\dots,2^N - 1)$ does not coincide with the node y^q, $q = 2^{MN} - 1$. As it follows from the construction of $y(x)$, the centers

$$y(z_1,\dots,z_\mu,2^N - 1,\dots,2^N - 1)$$

and

$$y(z_1,\dots,z_{\mu-1},z_\mu + 1,0,\dots,0)$$

corresponding to the adjacent subcubes are different in the same coordinate as the auxiliary centers

$$u(z_1,\dots,z_{\mu-1};z_\mu) \quad \text{and} \quad u(z_1,\dots,z_{\mu-1},z_\mu + 1)\,;$$

see the notations introduced in the second clause from the proof of Theorem 2.3. Therefore, if z_μ is the *odd* number, then, in accordance with (2.2.25),

$$\nu(z_1,\dots,z_M) = l(z_1,\dots,z_{\mu-1};z_\mu)\,.$$

If z_μ is *even*, then from (2.2.26), (2.2.32), (2.2.62), and the *permutation* rule,

$$\nu(z_1,\dots,z_M) = \begin{cases} 1, & t \ne N, \\ N, & t = 1, \end{cases}$$

where $t = l(z_1,\dots,z_{\mu-2};z_{\mu-1})$ if $\mu > 1$ and $t = N$ if $\mu = 1$.

Theorem 2.4. *If the function $g(y), y \in D$, is Lipschitzian with the constant L, then the one-dimensional function $g(l(x)), x \in [0,1]$, satisfies the uniform Hölder conditions (2.1.9).*

Proof. Suppose that $l(x) = l_M(x)$, $M > 1$, and let $x',x'' \in [0,1]$, $x' \ne x''$. If there exists an integer $n < M$ meeting the conditions

$$2^{-(n+1)M} \le |x' - x''| \le 2^{-nN}, \tag{2.2.71}$$

which are similar to (2.1.9), then justification of the relations (2.1.9) is just a reproduction of the corresponding discourse from the proof of Theorem 2.1.

Suppose that the conditions (2.2.71) are met at $n \geq M$. If the points x', x'' are from the same interval (2.2.70) and the corresponding images $l(x')$, $l(x'')$ belong to the same linear segment connecting the nodes y^i, y^{i+1}, which are different just in one coordinate, then from (2.2.68), (2.2.69), and (2.2.71),

$$||l(x') - l(x'')|| = 2^{MN}||y^i - y^{i+1}||(1-2^{-MN})|x' - x''| \leq$$
$$\leq 2^{M(N-1)}2^{-nN} = 2 \cdot 2^{-(n+1)}2^{(M-n)(N-1)} \leq 2(|x' - x''|)^{1/N} \tag{2.2.72}$$

because

$$||y^i - y^{i+1}|| = 2^{-M} .$$

If the points $l(x')$, $l(x'')$ belong to two different linear segments linked at the common end-point y^{i+1}, then

$$||l(x') - l(x'')|| \leq ||l(x') - y^i|| + ||y^{i+1} - l(x'')|| =$$
$$= ||y^{i+1} - y^i|| \left| 1 - (w(x') - v_i)2^{MN} \right| + ||y^{i+2} - y^{i+1}|| \left| w(x'') - v_{i+1} \right| =$$
$$= 2^{M(N-1)}(|w(x') - v_{i+1}| + |w(x'') - v_{i+1}|) \leq$$
$$\leq 2 \cdot 2^{M(N-1)}|x' - x''| < 2^{M(N-1)}2^{-nN} < 2(|x' - x''|)^{1/N},$$

which is equivalent to (2.2.72). Therefore, in consideration of the function $g(y)$, $y \in D$, being Lipschitzian, we obtain the relation

$$||g(l(x')) - g(l(x''))|| \leq 2L\sqrt{N+3}(|x' - x''|)^{1/N}, \quad x', x'' \in [0,1] . \tag{2.2.73}$$

The last statement proves the validity of (2.1.9). □

Let us study now Peano curves in comparison with spirals and TV evolvents.

The Peano-like curve $l(x)$, $x \in [0,1]$, covers the grid

$$H(M,N) = \{y^i; 0 \leq i \leq 2^{MN} - 1\} \tag{2.2.74}$$

having, as already mentioned, a mesh width equal to 2^{-M}. It should be stressed that the most important feature of the evolvent $l(x)$ is not in its piecewise linearity, covering the grid (2.2.74). The most important property is presented by the relation (2.2.72) which is similar to the relation (2.1.12) for the Peano curve $y(x)$. This property ensures the boundedness for the first divided differences of the function $F(l(x))$, $x \in [0,1]$, corresponding to the Lipschitzian function $F(y)$, $y \in D$; see (2.2.73).

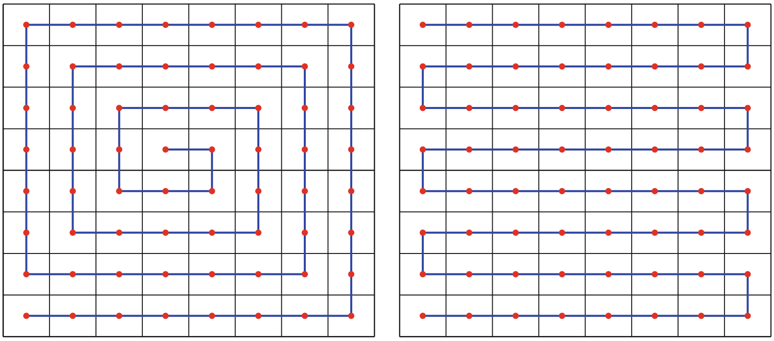

Fig. 2.5 Piecewise-linear curves covering the set (2.2.75) at $N = 2$, $M = 3$: spiral (the *left* picture) and TV evolvent (the *right* picture); nodes of the set (2.2.75) are marked by the *red dots*

We confine our consideration to the class of piecewise-linear curves and characterize the complexity of any particular curve from this family by the number of linear segments it is built of (each linear segment is assumed to be parallel to one of the coordinate axes). Spiral and TV evolvents (the images of the unit interval $[0,1]$ generated by these curves are given in Fig. 2.5; case $N = 2$, $M = 3$) are clearly from this family and they are much simpler than the Peano-like curve $l(x)$; see Fig. 2.4 (in both figures the nodes of the grid (2.2.75) are marked with red dots). For example, the TV evolvent is possible to present in the parametric form $t(x)$, $0 \le x \le 1$, by the following coordinate functions

$$
\begin{aligned}
t_1(x) &= (-1)^{q+1}2^{-1}\{2^{-M} - 1 + |\delta| - \delta\},\\
t_2(x) &= 2^{-1}\{(1+2q)2^{-M} - 1 + |\delta| - \delta\},
\end{aligned}
\tag{2.2.75}
$$

where $q = \lfloor k \rfloor$, $k = x(2^M - 2^{-M})$, $\delta = k - q - 1 + 2^{-M}$. The curve (2.2.75) is defined for $N = 2$. It is obviously much simpler than $l(x)$ and it is possible to generalize this scheme for $N > 2$.

But these simple piecewise-linear curves have essential drawback emerging from the very fact that the image of the unit interval $[0,1]$ generated by such a curve contains some linear segments covering a large number of nodes from (2.2.74). Let us focus on this feature assuming that there exists at least one linear segment covering 2^M nodes (which is exactly the case for the TV evolvent).

Total length of all the segments constituting the image of the interval $[0,1]$ generated by the curve $s(x) = s_M(x)$, $0 \le x \le 1$, is equal $2^{M(N-1)} - 2^{-M}$. Suppose that $s(x')$ and $s(x'')$ are, respectively, the initial and the end points of the above linear segment containing 2^M nodes of the net (2.2.75). Then

$$
||s(x') - s(x'')|| = 1 - 2^{-M} > 2^{-1},
$$

$$
|x' - x''| = (2^M - 1)/(2^{MN} - 1) < 2^{-M(N-1)},
$$

whence it follows

$$||s(x') - s(x'')|| > 2^{M(1-1/N)-1}(|x' - x''|)^{1/N} .$$

This means that there does not exist any coefficient that ensures the validity of a relation similar to (2.1.12), (2.2.72) and does not dependent on M.

The Peano curve $y(x)$ is defined as the limit object and, therefore, only approximations to this curve are applicable in the actual computing. The piecewise linear evolvent $l(x) = l_M(x)$ suggested above covers all the nodes of the grid $H(M,N)$ from (2.2.74) and, thus, it allows us to ensure the required accuracy in analyzing multidimensional problems by solving their one-dimensional images produced by the implementation of $l(x)$. But this evolvent has some deficiencies.

The first one is due to the fact that the grid $H(M+\nu,N)$ with mesh width equal to $2^{-(M+\nu)}$ does not contain the nodes of the less finer grid $H(M,N)$. Therefore, in general, the point $l_M(x')$ may not be covered by the curve $l_{M+\nu}(x)$, $0 \le x \le 1$, and, hence, the outcomes already obtained while computing the values $F(l_M(x))$ will not be of any use if the demand for greater accuracy necessitates switching to the curve $l_{M+\nu}(x)$, $\nu \ge 1$. This difficulty is possible to overcome by setting the parameter M equal to a substantially larger value than seems to be sufficient at the beginning of the search.

Another problem arises from the fact that $l(x)$ is a one-to-one correspondence between the unit interval $[0,1]$ and the set $\{l(x) :\ 0 \le x \le 1\} \subset D$ though the Peano curve $y(x)$ has a different property: the point $y \in D = \{y(x) :\ 0 \le x \le 1\}$ could have several inverse images in $[0,1]$ (but not more than 2^N). That is, the points $y \in D$ could be characterized by their *multiplicity* with respect to the correspondence $y(x)$. This is due to the fact that though each point $x \in [0,1]$ is contained just in one subinterval of any Mth partition, some subcubes corresponding to several different subintervals of the same Mth partition (e.g., all the subcubes of the first partition) could have a common vertex. Therefore, some different inverse images $x', x'' \in [0,1]$, $x' \ne x''$, could have the same image, i.e., $y(x') = y(x'')$.

This multiplicity of points $y \in D$ with respect to the correspondence $y(x)$ is the fundamental property reflecting the essence of the dimensionality notion: the segment $[0,1]$ and the cube D are sets of equal cardinality and the first one could be mapped onto the other by some single-valued mapping, but if this mapping is continuous, then it could not be univalent (i.e., it could not be a one-to-one correspondence), and the dimensionality N of the hypercube D determines the bound from above (2^N) for the maximal possible multiplicity of $y(x)$.

Therefore, the global minimizer y^* of the function $F(y)$ over D could have several inverse images x_i^*, $1 \le i \le m$, i.e., $y^* = y(x_i^*)$, $1 \le i \le m$, which are the global minimizers of the function $F(y(x))$ over $[0,1]$.

To overcome the above deficiencies of $l(x)$, we suggest one more evolvent $n(x) = n_M(x)$ mapping some uniform grid in the interval $[0,1]$ onto the grid $P(M,N)$ in the hypercube D from (1.2) having mesh width equal to 2^{-M} (in each coordinate) and meeting the condition

$$P(M,N) \subset P(M+1,N) . \tag{2.2.76}$$

The evolvent $n(x)$ approximates the Peano curve $y(x)$ and its points in D own the property of multiplicity; each node of the grid $P(M,N)$ could have several (but not more than 2^N) inverses in the interval $[0,1]$.

Construction of n(x). Assume that the set of nodes in $P(M,N)$ coincides with the set of vertices of the hypercubes $D(z_1,\ldots,z_M)$ of the Mth partition. Then the mesh width for such a grid is equal to 2^{-M} and the total number of all the nodes in $P(M,N)$ is $(2^M+1)^N$. As long as the vertices of the Mth partition hypercubes are also the vertices of some hypercubes of any subsequent partition $M+\nu$, $\nu \geq 1$, then the inclusion (2.2.77) is valid for the suggested grid.

Note that each of 2^N vertices on any hypercube $D(z_1,\ldots,z_M)$ of the Mth partition is simultaneously the vertex of just one hypercube $D(z_1,\ldots,z_M,z_{M+1})$ from $D(z_1,\ldots,z_M)$. Denote $P(z_1,\ldots,z_{M+1})$ the common vertex of the hypercubes

$$D(z_1,\ldots,z_M,z_{M+1}) \subset D(z_1,\ldots,z_M)\,. \tag{2.2.77}$$

Due to (2.2.37), the center of the hypercube from the left-hand part of (2.2.77) and the center of the hypercube from the right-hand part of (2.2.77) are linked by the relation

$$y(z_1,\ldots,z_{M+1}) = y(z_1,\ldots,z_M) + (u^{tq}(z_{M+1}) - 2^{-1})2^{-(M+1)},$$

whence it follows that

$$n(z_1,\ldots,z_{M+1}) = y(z_1,\ldots,z_M) + (u^{tq}(z_{M+1}) - 2^{-1})2^{-M}, \tag{2.2.78}$$

and varying z_{M+1} from 0 to 2^N-1 results in computing from (2.2.78) all the 2^N vertices of the hypercube $D(z_1,\ldots,z_M)$.

Formula (2.2.78) establishes the single-valued correspondence between $2^{(M+1)N}$ intervals $d(z_1,\ldots,z_{M+1})$ of the Mth partition of $[0,1]$ and $(2^M+1)^N$ nodes $n(z_1,\ldots,z_{M+1})$ of the grid $P(M,N)$; this correspondence is obviously not a univalent (not one-to-one) correspondence.

Number all the intervals $d(z_1,\ldots,z_{M+1})$ from left to right with subscript i, $0 \leq i \leq 2^{(M+1)N}-1$ and denote v_i, v_{i+1} the end-points of the ith interval. Next, introduce the numeration of the centers $y^i = y(z_1,\ldots,z_{M+1})$ from (2.2.37) assuming that the center corresponding to the hypercube $D(z_1,\ldots,z_{M+1})$ is assigned the same number i as the number of the interval $d(z_1,\ldots,z_{M+1}) = [v_i,v_{i+1})$. Thus, we have defined the one-to-one correspondence of the nodes

$$v_i, \quad 0 \leq i \leq 2^{(M+1)N}-1, \tag{2.2.79}$$

constituting a uniform grid in the interval $[0,1]$ and of the centers y^i which, in accordance with (2.2.78), generates the one-to-one correspondence of the end-points v_i from (2.2.79) and of the nodes $p \in P(M,N)$.

Note that if the centers y^i and y^{i+1} are from the same hypercube of the Mth partition, then these centers (and consequently the corresponding points v_i and v_{i+1})

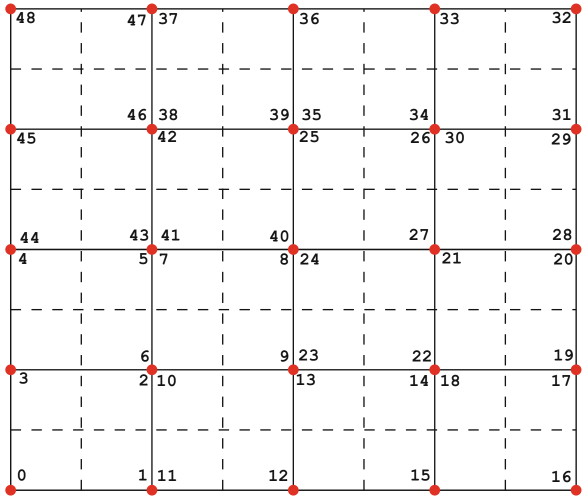

Fig. 2.6 Nodes of the grid $P(2,2)$ (marked by the *red dots*). Integers around the nodes indicate the numbers of the points from the uniform grid (2.2.80) mapped onto the corresponding nodes by $n(x)$

are juxtaposed with some different nodes from $P(M,N)$. Therefore, the node p may be juxtaposed to the points v_i, v_{i+1} if and only if the corresponding centers y^i and y^{i+1} are from some different (but adjacent) subcubes of the Mth partition. As long as the number of subcubes in the Mth partition of D is equal to 2^{MN}, then there are exactly $2^{MN}-1$ pairs v_i, v_{i+1} juxtaposed with the same node from $P(M,N)$ (in general, this node is different for different pairs v_i, v_{i+1} of the above type).

To ensure that any two *vicinal* nodes in $[0,1]$ are juxtaposed with different nodes from $P(M,N)$ we substitute each of the above pairs v_i, v_{i+1} with just one node in $[0,1]$. Next, we rearrange the collocation of nodes in $[0,1]$ to keep up the uniformity of the grid.

To do so we construct the uniform grid in the interval $[0,1]$ with the nodes

$$h_j, \quad 0 \le j \le 2^{(M+1)N} - 2^{MN} = q, \tag{2.2.80}$$

where $h_0 = 0$ and $h_q = 1$, and juxtapose to the node h_j of the grid (2.2.80) the node v_i of the grid (2.2.79), where

$$i = j + \lfloor (j-1)/(2^N - 1) \rfloor \,. \tag{2.2.81}$$

Next, we assume that the node h_j is juxtaposed with the node of the grid $P(M,N)$ generated by (2.2.78) for the center y^i with i from (2.2.81). This mapping of the uniform grid (2.2.81) in the interval $[0,1]$ onto the grid $P(M,N)$ in the hypercube D will be referred to as *Non-Univalent Peano-like Evolvent* (NUPE, for short) and designated $n(x) = n_M(x)$.

For the sake of illustration, Fig. 2.6 presents the nodes of the grid $P(2,2)$ (marked by the red dots) and each node is assigned with the numbers j of the points h_j from (2.2.80) mapped onto this node of the grid $P(2,2)$. These numbers are plotted around the relevant nodes.

The inverse images h_j corresponding to a given node $p \in P(M,N)$ could be computed in the following way. Let U be the set of 2^N different binary vectors $u \in R^N$. Then, in accordance with (2.2.78),

$$Y(p) = \{p - (u - 2^{-1})2^{-M} : \; u \in U\}, \quad p \in P(M,N),$$

is the set of all centers of the $(M+1)$st partition subcubes from D generating the same given node $p \in P(M,N)$. If the center $y \in Y(p)$ is assigned the number i, i.e., if $y = y^i$, then it corresponds to the node v_i of the grid (2.2.79), and being given this number i it is possible to compute the number $j = i - \lfloor i/2^N \rfloor$ of the corresponding node h_j from the grid (2.2.80). Different nodes h_j obtained as the result of these computations performed for all centers $y \in Y(p)$ constitute the set of all inverse images for the node $p \in P(M,N)$ with respect to $n(x)$. From (2.1.4), (2.1.5), follows that the point v_i corresponding to the center $y^i = y(z_1, \ldots, z_{M+1})$ can be estimated from the expression

$$v_i = \sum_{j=1}^{M+1} z_j 2^{-jN},$$

where the numbers $z_1, \ldots, z_{M+1}$ are to be computed by presenting the corresponding vector $y(z_1, \ldots, z_{M+1})$ in the form (2.2.37) and then analyzing the right-hand part of this expression.

2.3 Standard Routines for Computing Approximations to Peano Curves

This section presents the software package for computing images and inverse images for the suggested approximations to Peano curves (centers of the Mth partition hypercubes, piecewise-linear evolvents, and non-univalent evolvents). The package originally was written in FORTRAN by R.Strongin and later rewritten in C++ by V.Gergel. This last version is given here.

The principal function of the package is *mapd* which computes the image of the point x from $[0,1]$ and places this image into the array y. The required accuracy is set by selecting the number M of the corresponding partition which is to be assigned as the value of the function parameter m. The dimensionality N of the hypercube D from (1.2) is to be indicated by the parameter n. Notations m and n have the above meaning in the description of all the functions presented in this section. As long as the expansion (2.1.4) requires MN binary digits for representation of x, it has to be mentioned that there is the constraint $MN < \Gamma$ where the value of Γ is the number of digits in the mantissa and, therefore, depends on the computer that is used for the implementation. To select the particular evolvent it is necessary to assign some appropriate integer value to the function parameter *key*: 1 corresponds to the approximation by centers of the Mth partition hypercubes, 2 corresponds to the approximation by the piecewise-linear evolvent $l_M(x)$, and 3 corresponds to the non-univalent evolvent $n_M(x)$.

The function *mapd* uses the auxiliary function *node* which computes the vector $u(s)$ (designated *iu*) similar to the one from (2.2.2), (2.2.3) (parameter *is* corresponds to s). But the particular software realization we consider uses the scheme in which $|u_i(s)| = 1$, $1 \leq i \leq N$, i.e., $u(s)$ is not a binary vector (zero values of the coordinates from (2.2.2), (2.2.3) are replaced with -1 and, consequently, the negation operation is replaced by the invertor changing the sign of the coordinates). It should also be mentioned that $iu[0]$ and $iu[2^N - 1]$ are different in the first coordinate, but not in the Nth one as in (2.2.4).

This function also computes the vector $w(s)$ (designated *iv*) and the integer $l(s)$, respectively, from (2.2.27), (2.2.28) and (2.2.25), (2.2.26); it is assumed that $|w_i(s)| = 1$, $1 \leq i \leq N$.

The inverse images h_j from (2.2.80) for the given point $p \in P(M,N)$ generated by the non-univalent evolvent $n_M(x)$ are computed by the function *invmad*. It has to be mentioned that, instead of the term "inverse image," the shorter term "preimage" is used in the comments in the bodies of the functions. The computed coordinates of the inverse images are placed into the array *xp*. The size of this array is set by the parameter *kp*; the number of the actually computed inverse images is reflected by the value of the parameter *kxx*. The parameter *incr* provides the possibility for the user to stipulate the condition that the difference in the location of any two computed vicinal inverse images should be not less than the distance prescribed by the value of *incr*; the required distance in *incr* is recorded as the number of the sequential nodes of the grid (2.2.80) parting the vicinal inverse images.

The function *invmad* uses the auxiliary function *xyd* which computes the left-end points v_i (designated *xx*) from (2.2.79) corresponding to the centers y^i (placed into the array *y*) employed in (2.2.68). This function is also possible to use as the means for computing approximations to some inverse images of the points $y \in D$ with respect to the Peano curve $y(x)$. In this case, the function *xyd*, first, computes the nearest center $y^i = y(z_1, \ldots, z_M)$ to the given point $y \in D$ and then estimates the left-end point v_i of the interval $d(M, v_i) = d(z_1, \ldots, z_M)$. This point is the approximation (with accuracy not worse than 2^{-MN}) to one of the inverse images of the given point y.

Routines

File "map.h"

```
/* map modules */
#ifndef _MAP
#define _MAP
void mapd ( double, int, float *, int, int ); /* map x to y */
void invmad ( int, double *, int , int *, float *, int , int );
       /* map y to x */
#endif
```

File "x_to_y.c"

```
#include <math.h>
```

```
#include "map.h"
int n1,nexp,l,iq,iu[10],iv[10];
void mapd( double x, int m, float y[], int n, int key ) {
/* mapping y(x) : 1 - center, 2 - line, 3 - node */
double d,mne,dd,dr;
float p, r;
int iw[11], it, is, i, j, k;
void node ( int );
p=0.0;
n1=n-1;
for ( nexp=1,i=0; i<n; nexp*=2,i++ ); /* nexp=2**n */
d=x;
r=0.5;
it=0;
dr=nexp;
for ( mne=1,i=0; i<m; mne*=dr,i++ ); /* mne=dr**m */
for ( i=0; i<n; i++ ) {
   iw[i]=1; y[i]=0.0;
}
if ( key == 2 ) {
      d=d*(1.0-1.0/mne); k=0;
   } else
   if ( key > 2 ) {
      dr=mne/nexp;
      dr=dr-fmod(dr,1.0);
      dd=mne-dr;
      dr=d*dd;
      dd=dr-fmod(dr,1.0);
      dr=dd+(dd-1.0)/(nexp-1.0);
      dd=dr-fmod(dr,1.0);
      d=dd*(1./mne);
}
for ( j=0; j<m; j++ ) {
   iq=0;
   if ( x == 1.0 ) {
      is=nexp-1; d=0.0;
      } else {
      d=d*nexp;
      is=d;
      d=d-is;
   }
   i=is;
   node(i);
   i=iu[0];
   iu[0]=iu[it];
   iu[it]=i;
   i=iv[0];
   iv[0]=iv[it];
   iv[it]=i;
   if ( l == 0 )
      l=it;
      else if ( l == it ) l=0;
   if ( (iq>0)||((iq==0)&&(is==0)) ) k=l;
      else if ( iq<0 ) k = ( it==n1 ) ? 0 : n1;
```

```
    r=r*0.5;
    it=l;
    for ( i=0; i<n; i++ ) {
        iu[i]=iu[i]*iw[i];
        iw[i]=-iv[i]*iw[i];
        p=r*iu[i];
        p=p+y[i];
        y[i]=p;
    }
}
if ( key == 2 ) {
        if ( is==(nexp-1) ) i=-1;
        else i=1;
        p=2*i*iu[k]*r*d;
        p=y[k]-p;
        y[k]=p;
    } else if ( key == 3 ) {
        for ( i=0; i<n; i++ ) {
        p=r*iu[i];
        p=p+y[i];
        y[i]=p;
        }
    }
}
void node ( int is ) {
/* calculate iu=u[s], iv=v[s], l=l[s] by is=s */
int n,i,j,k1,k2,iff;
n=n1+1;
if ( is == 0 ) {
        l=n1;
        for ( i=0; i<n; i++ ) {
        iu[i]=-1; iv[i]=-1;
        }
    } else if ( is == (nexp-1) ) {
        l=n1; iu[0]=1; iv[0]=1;
        for ( i=1; i<n; i++ ) {
        iu[i]=-1; iv[i]=-1;
        }
        iv[n1]=1;
    } else {
        iff=nexp;
        k1=-1;
        for ( i=0; i<n; i++ ) {
        iff=iff/2;
        if ( is >= iff ) {
                if ( (is==iff)&&(is != 1) ) { l=i; iq=-1; }
                is=is-iff;
                k2=1;
            }
            else {
                k2=-1;
                if ( (is==(iff-1))&&(is!= 0) ) { l=i; iq=1; }
        }
        j=-k1*k2;
```

```
        iv[i]=j;
        iu[i]=j;
        k1=k2;
        }
        iv[l]=iv[l]*iq;
        iv[n1]=-iv[n1];
}
}
```

File “y_to_x.c”

```
#include <math.h>
#include "map.h"
static void xyd ( double *, int, float *, int ); /* get a
preimage */
static void numbr ( int *iss);
extern int n1,nexp,l,iq,iu[10],iv[10];
double del;
void
invmad ( int m, double xp[], int kp, int *kxx, float p[], int n,
int incr ) {
/*
preimages calculation
- m - map level (number of partitioning)
- xp - preimages to be calculated
- kp - number of preimages that may be calculated (size of xp)
- kxx - number of preimages being calculated
- p - image for which preimages are calculated
- n - dimension of image (size of p)
- incr - minimum number of map nodes that must be between
preimages
*/
double mne,d1,dd,x,dr;
float r,d,u[10],y[10];
int i,k,kx,nexp;
void xyd ( double *, int, float *, int );
kx=0;
kp--;
for ( nexp=1,i=0; i<n; i++ ) { nexp*=2; u[i]=-1.0; }
dr=nexp;
for ( mne=1, r=0.5, i=0; i<m; i++ ) { mne*=dr; r*=0.5; }
dr=mne/nexp;
dr=dr-fmod(dr,1.0);
del=1./(mne-dr);
d1=del*(incr+0.5);
for ( kx=-1; kx<kp; ) {
    for ( i=0; i<n; i++ ) { /* label 2 */
        d=p[i];
        y[i]=d-r*u[i];
    }
for ( i=0; (i<n) && (fabs(y[i]) < 0.5) ; i++ );
    if ( i>=n ) {
        xyd(&x,m,y,n);
        dr=x*mne;
```

```
        dd=dr-fmod(dr,1.0);
        dr=dd/nexp;
        dd=dd-dr+fmod(dr,1.0);
        x=dd*del;
        if ( kx>kp ) break;
        k=kx++; /* label 9 */
        if ( kx == 0 ) xp[0]=x;
        else {
            while ( k>=0 ) {
                dr=fabs(x-xp[k]); /* label 11 */
                if ( dr<=d1 ) {
                    for ( kx-- ; k<kx; k++,xp[k]=xp[k+1] );
                    goto m6;
                } else
                    if ( x <= xp[k] ) {
                        xp[k+1]=xp[k]; k--;
                    } else break;
            }
            xp[k+1]=x;
        }
    }
    m6 : for ( i=n-1; (i>=0)&&(u[i]=(u[i]<=0.0) ? 1 : -1)<0; i--
);
    if ( i<0 ) break;
}
*kxx=++kx;
}
void xyd ( double *xx, int m, float y[], int n){
/* calculate preimage xx for the nearest m-level center of y */
/* (xx - left boundary point of m-level interval) */
double x,r1;
float r;
int iw[10];
int i,j,it,is;
void numbr ( int * );
n1=n-1;
for ( nexp=1,i=0; i<n; i++ ) { nexp*=2; iw[i]=1; }
r=0.5;
r1=1.0;
x=0.0;
it=0;
for ( j=0; j<m; j++ ) {
    r*=0.5;
    for ( i=0; i<n; i++ ) {
    iu[i] = ( y[i]<0 ) ? -1 : 1;
    y[i]-=r*iu[i];
    iu[i]*=iw[i];
    }
    i=iu[0];
    iu[0]=iu[it];
    iu[it]=i;
    numbr ( &is );
    i=iv[0];
    iv[0]=iv[it];
```

```
    iv[it]=i;
    for ( i=0; i<n; i++ )
    iw[i]=-iw[i]*iv[i];
    if ( l == 0 ) l=it;
    else if ( l == it ) l=0;
    it=l;
    r1=r1/nexp;
    x+=r1*is;
}
*xx=x;
}
void numbr ( int *iss) {
/* calculate s(u)=is,l(u)=l,v(u)=iv by u=iu */
int i,n,is,iff,k1,k2,l1;
n=n1+1;
iff=nexp;
is=0;
k1=-1;
for ( i=0; i<n; i++ ) {
    iff=iff/2;
    k2=-k1*iu[i];
    iv[i]=iu[i];
    k1=k2;
    if ( k2<0 ) l1=i;
    else { is+=iff; l=i; }
}
if ( is == 0 ) l=n1;
else {
    iv[n1]=-iv[n1];
    if ( is == (nexp-1) ) l=n1;
    else if ( l1 == n1 ) iv[l]=-iv[l]; else l=l1;
}
*iss=is;
}
```

Example 2.1. We consider in Fig. 2.7 an approximation of the Peano curve, the piecewise-linear evolvent (obtained with key=2), in dimension $N = 2$ and with level $M = 3$. The points 1, 2 and 3 have, respectively, coordinates in the interval $[0, 1]$ and

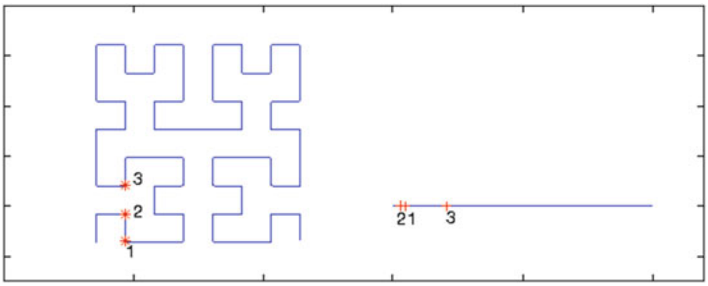

Fig. 2.7 Peano-like piecewise-linear evolvent, $N = 2$, $M = 3$

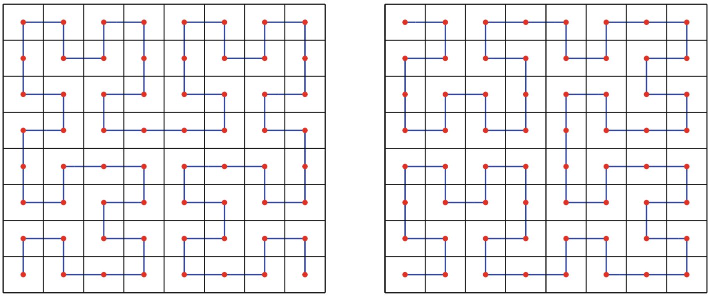

Fig. 2.8 Several rotated Peano curves can help to construct one-dimensional schemes better representing the information about vicinity of the points in the multidimensional domain

in the domain $D = \{y \in R^2 : -2^{-1} \leq y_j \leq 2^{-1},\ \ j = 1,2\}$:

$$1 : (0.0470) \qquad (-0.3125, -0.4326)$$
$$2 : (0.0320) \qquad (-0.3125, -0.3145)$$
$$3 : (0.2060) \qquad (-0.3125, -0.1848)$$

It can be seen that in the domain D the point 2 is equidistant from the points 1 and 3 whereas at the interval $[0,1]$ the point 2 is significantly more distant from the point 3 than from the point 1.

It has been recently shown (see [6]) that a simultaneous usage in one-dimensional reduction of several curves rotated with respect to the search domain (see Fig. 2.8) gives the possibility to have a better representation of the information about vicinity of the points in the multidimensional domain.

Example 2.2. The main function suggested here presents the test illustrating the way in which the functions are to be called. First, it computes the image $P_M(0.5)$ at $M = 10, N = 2$ and estimates all the inverse images for the obtained image; then it generates the images for all three estimated inverse images demonstrating that in all three cases the same image $(0,0)$ is obtained. Second, it computes the image $P_M(0.55)$ at $M = 14, N = 3$ and regenerates eight inverse images which is followed by generation of eight images (all the same).

File "test.c"

```
#include <conio.h>
#include <stdio.h>
#include "map.h"
#define KEY 3
main() {
double x, xp[32], d, del;
```

```
int i,j,n,m,kp,kx;
float y[10];
/* Initialization */
clrscr();
n=2;
m=5;
y[0] = 0.0;
y[1] = 0.0;
printf("\t\tTesting the map modules\n\n");
/* parameters input */
printf("Input dimension (2-5) - ");
scanf("%d",&n);
printf("Input map level (n*m<Γ) - ");
scanf("%d",&m);
printf("Input a preimage value - ");
scanf("%lf",&x);
/* calculation and output */
printf("\n\t\tCalculation results\n\n");
printf("Preimage = %lf, Dimension = %d, Map level = %d\n",x,n,m);
/* image calculation */
mapd ( x, m, y, n, KEY );
printf("Image (");
for ( i=0; i<n; i++ )
   printf(" %f%c",y[i],(i==n-1)?' ':',');
printf(" )\n\n");
/* back calculation preimages of image */
printf("Input number of preimages that may be calculated - ");
scanf("%d",&kp);
invmad ( m, xp, kp, &kx, y, n, 1 );
printf("\nPreimages \n\n");
for ( i=0; i<kx; i++ )
   printf(" %30.25f\n",xp[i]);
printf("\n");
/* testing preimages that are calculated */
for ( d=1.0, i=0; i<n; i++, d /= 2.0 );
for ( del=1.0, i=0; i<m; i++, del *= d );
for ( i=0; i<kx; i++ ) {
   mapd ( ( (xp[i]==1.0) ? xp[i] : xp[i] + 0.5 * del ), m, y, n,
KEY );
   printf("Image for %d preimage = (",i);
   for ( j=0; j<n; j++ )
      printf(" %f%c",y[j],(j==n-1)?' ':',');
   printf(" )\n\n");
}
getch();
}
```

Results of the test

```
       Testing the map modules
Input dimension (2--5) - 2
Input map level (n*m<Γ) - 10
Input a preimage value - 0.5
       Calculation results
```

```
Preimage = 0.500000, Dimension = 2, Map level = 10
Image ( 0.000000, 0.000000 )
Input number of preimages that may be calculated - 4
Preimages
   0.166666666666666574000000
   0.500000000000000000000000
   0.833333333333332593000000
Image for 0 preimage = ( 0.000000, 0.000000 )
Image for 1 preimage = ( 0.000000, 0.000000 )
Image for 2 preimage = ( 0.000000, 0.000000 )
      Testing the map modules
Input dimension (2--5) - 3
Input map level (n*m<Γ) - 14
Input a preimage value - 0.55
      Calculation results
Preimage = 0.550000, Dimension = 3, Map level = 14
Image ( 0.099976, 0.099976, -0.100098 )
Input number of preimages that may be calculated - 8
Preimages
   0.549999998870147566000000
   0.549999998932513234000000
   0.549999998942907142000000
   0.549999999005272810000000
   0.549999999914767512000000
   0.549999999925161420000000
   0.549999999987527088000000
   0.549999999997920996000000
Image for 0 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 1 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 2 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 3 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 4 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 5 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 6 preimage = ( 0.099976, 0.099976, -0.100098 )
Image for 7 preimage = ( 0.099976, 0.099976, -0.100098 )
```

Springer

http://www.springer.com/978-1-4614-8041-9

Introduction to Global Optimization Exploiting
Space-Filling Curves
Sergeyev, Y.D.; Strongin, R.G.; Lera, D.
2013, X, 125 p. 32 illus., 30 illus. in color., Softcover
ISBN: 978-1-4614-8041-9