# Data Structure

We introduce the notion of $D$-**grid** representation, where $D$ is the grid dimension, for the purpose of parametric studies. This formalism is a methodological effort and contribution to the field of parametric design and analysis.

In such a study a **system** is analyzed regarding several variables named **parameter**. Each parameter can vary in a discrete or continuous range of scalar values. If not, the modeling of the system can easily be reinterpreted in that way. For instance a (x,y,z) point parameter can be understood as 3 scalar parameters.

Complex multivariable systems often lead to complex and costly analysis process. In practice, it is almost impossible to know the **state** of the system for every state of inputs ; nor analytically ; nor numerically. It might be a question of pure feasibility but it might also be a question of time or money.

Thus, it is common to analyze the system in a **finite number of interesting states**, where the choice of those states, and the corresponding values of the input parameters, comes down to the knowledge of the designer.

Depending on the sharpness of the grid, the designer would be interested in the results at grid **nodes** or, possibly, to **interpolate** results to predict any state of the system, thus reducing its margin of error.

We model the set of combinations of input values as a $D$-grid. A $D$-grid is a $D$-dimensional grid generated by $D$ **finite and strictly ordered sets**, where each grid **dimension** corresponds to a **variable parameter** of the study. The finite number of states that a (scalar) parameter can have, and where the system is to be evaluated, are stacked in ascending order into a set.

We found that the proper comprehension of this object, as detailed below, leads to a solid and generic methodology to organize and conduct a parametric study. It structures the way results can be browsed and interpolated over the parametric space.

`Ivy` is a small `.NET` library written in `C#` that enable the creation and the manipulation of d-dimensional grids. It is composed of 2 dlls :

- `IvyCore.dll` - the core library
- `IvyGh.gha` - an interface for Grasshopper, a parametric design software

## Set

Let $S = \{s_1 < s_2 < \ ... < s_N\}$ be a **finite and strictly ordered set** of $\mathbb{R}$ :

$$S = \{s_i \in \mathbb{R}, i \in [1, N] \mid \forall(i, j) : i < j \Rightarrow s_i < s_j\}$$

We denote :

- $N$ the cardinality of $S$
- $\mathbb{S}$ the infinite set of such sets ($S \in \mathbb{S}$)

$S$ will indifferently be referred to as a $N$-*tuple* or simply a **tuple**.

## Grid

Let $G = S_1 \times S_2 \times \cdots \times S_D$ be a $D$-grid or simply a **grid**, defined as the **cartesian product** of $D$ finite and strictly ordered sets :

$$G = \prod_{j \in [1,D]} S_j \quad , \quad S_j = \{s_1^j < s_2^j < \ ... \ < s_{N_j}^j\} \in \mathbb{S}$$

We denote :

- $D$ the number of finite and strictly ordered sets
- $N_j$ the cardinality of $S_j \in \mathbb{S}$
- $s_i^j$ the $j^{th}$ element of the $i^{th}$ of $S_j$
- $I_j = [s_1^j, s_{N_j}^j] \in \mathbb{R}$ the interval in which $s^j$ evolves.

   While this grid structure lays out a convenient representation of a $D$-dimensional **discrete** parametric space, we introduce for each parameter the continuous interval in which it varies as we are also interested to interpolate the results for any state of inputs.

A $D$-grid can be seen as an **hyperrectangle**, also known as a $D$**-orthotope**. It's a particular kind of convex **polytope** of $\mathbb{R}^d$.

   This is a generalization from the easy-to-understand $\mathbb{R}^1$, $\mathbb{R}^2$ and $\mathbb{R}^3$ cases where :

   - a 1-grid is a segments divided into smaller segment called edge
   - a 2-grid is a rectangle divided into smaller rectangles called face
   - a 3-grid is a parallelepiped divided into smaller parallelepipeds called cell

   In $\mathbb{R}^d$, a **polytope** would indicate a geometric object of dimension $D$ whereas a **facet** would account for an object of dimension $N$.

One can easily introduce the notion of a **normalized grid**, leading to represent it as a **hypercube** of length 1, which is a **regular polytope**.

## Node

Let $g_{i^1 i^2 \ \dots \ i^D} = (s_{i^1}, s_{i^2}, ..., s_{i^D})$ be a $D$-node or simply a **node** or **vertex** of a $D$-grid. It can be described as a tuple where :

$$g_{i^1 i^2 \ \dots \ i^D} = (s_{i^1}, s_{i^2}, ..., s_{i^D}) \ \mid \ s_{i^j} \in S_j \ , \ i^j \in [1, N_j]$$

Here we use the notation $s_{i^j}$ instead of $s_i^j$ because it is more convenient. The total number of nodes in the grid, which is also the cardinality of $G$, is given by :

$$N = card(G) = N_1 \times N_1 \times \cdots \times N_D$$

## Labeling

As suggested by the employed notation, nodes can simply be identified by stacking together the indexes of the values taken by each parameter. Thus, a node has a unique tuple "address" in the grid :

$$i^1 i^2 \ \cdots \ i^D \Leftrightarrow (i^1, i^2, \cdots, i^D)$$

## Numbering

However, there is exactly $N!$ different ways to order a finite set of $N$ nodes :

$$i^j \in [1, N^j] \longrightarrow \sigma_k^j(i) \in [1, N^j] \ , \ j \in [1, D]$$

Where $\sigma_k^j$ is the $k^{th}$ permutation function (in a total of $N^j!$) over the set $S_j$.

Fore sure, not all ways of numbering this nodes are made equals in terms of convenience, especially when it comes to construct grid's cells from their corner nodes.

The proposed numbering is the following :

| nř | $i^1$ | $i^1$ | $i^2$ | $\cdots$ | $i^D$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | $\cdots$ | 1 |
| 2 | 2 | 1 | 1 | $\cdots$ | 1 |
| 3 | 3 | 1 | 1 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $N_0$ | $N_0$ | 1 | 1 | $\cdots$ | 1 |
| +1 | 1 | 2 | 1 | $\cdots$ | 1 |
| +2 | 1 | 2 | 1 | $\cdots$ | 1 |
| +3 | 1 | 2 | 1 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $2 * N_0$ | $N_0$ | 2 | 1 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $(N_1 - 1) * N_0$ | $N_0$ | $N_1 - 1$ | 1 | $\cdots$ | 1 |
| +1 | 1 | $N_1$ | 1 | $\cdots$ | 1 |
| +2 | 2 | $N_1$ | 1 | $\cdots$ | 1 |
| +3 | 3 | $N_1$ | 1 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $N_1 * N_0$ | $N_0$ | $N_1$ | 1 | $\cdots$ | 1 |
| +1 | 1 | 1 | 2 | $\cdots$ | 1 |
| +2 | 2 | 1 | 2 | $\cdots$ | 1 |
| +3 | 3 | 1 | 2 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $2 * (N_1 * N_0)$ | $N_0$ | 1 | 2 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $N_2 * N_1 * N_0$ | $N_0$ | $N_1$ | $N_2$ | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $N - N_0$ | $N_0$ | $N_1$ | $N_2$ | $\cdots$ | $N_D$ |
| +1 | 1 | $N_1$ | $N_2$ | $\cdots$ | $N_D$ |
| +2 | 2 | $N_1$ | $N_2$ | $\cdots$ | $N_D$ |
| +3 | 3 | $N_1$ | $N_2$ | $\cdots$ | $N_D$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | $\vdots$ |
| $N$ | $N_0$ | $N_1$ | $N_2$ | $\cdots$ | $N_D$ |

## Cell

A $D$-grid can be decomposed in a group of $D$-cells called **cell**. Grids and cells are objects of the same topolgy. A cell can bee seen as the smallest possible grid in terms of vertices.

Let a **cell** be an element of the grid. It can be seen as a d-tuple of the form :

A d-cell is itself a **hyper rectangle** also known as a **d-orthotope**.

as the cartesian product of a finite number of such set.