# Chapter 1

# Mesh-like data layouts

`pyop3` was created to provide a richer abstraction than PyOP2 for describing stencil-like operations over unstructured meshes. Most of the innovation in `pyop3` stems from its novel data model. Data structures associated with a mesh are created using more information about the mesh topology. This lays the groundwork for a much more expressive DSL since more of the semantics are captured/represented.

## 1.1 Data layouts for the finite element method

### 1.1.1 Conflicting DMPlex and PyOP2 abstractions

#### Representing data layouts with DMPlex

DMPlex represents a mesh as a set of points where the points are divided into *strata* (cells, edges, vertices, etc). These points are connected in a graph (Hasse diagram) and a rich set of queries can be used to determine the right adjacencies needed for things like the finite element method.

In order to associated data with these mesh points, a typical PETSc application will construct a PETSc

. These are simple CSR-like (?) data structures that encode a data layout by associating a particular number of DoFs with each mesh point. Sections are a powerful tool for describing data layouts but they have a number of limitations:

- Sections are fully ragged. They only store DoF information per point in a completely unstructured way and are incapable of knowing, say, that every cell in the mesh stores exactly one DoF. This can prohibit the compiler from making certain optimisations (e.g. loop unrolling) that it would have been able to do were it to know of a constant loop extent. Additionally, this variable size increases memory pressure as redundant arrays of constant sizes need to be streamed through memory.

- DoFs per point are treated as a flat array. This means that shape information is lost for, say, vector-valued functions.

With PETSc/DMPlex, the P3 DoF layout would be represented as shown in Figure **??**.

**Data layouts in PyOP2**

PyOP2 takes a very different approach to describing data layouts to DMPlex. Firstly, it has no conception of what a mesh is and it deals solely with *sets* and *mappings between sets*. The rich query language provided by DMPlex is therefore unavailable and the task of determining the right adjacency maps is passed to the user.

### 1.1.2 What `pyop3` does