Modeling of Non-Trivial Data-Structures with a Generic Scientific Simulation Environment

René Heinzl, Philipp Schwaha,
Carlos Giani, and Siegfried Selberherr
Institute for Microelectronics, TU Wien, Vienna, Austria
{heinzl|schwaha|selberherr}@iue.tuwien.ac.at

Abstract

Scientific computing requires efficient specification and handling of structure generation and associated modeling tasks. A generic scientific simulation environment was developed to ease these tasks by homogeneous specification of data structures in a dimensionally and topologically neutral way. The concept of fiber bundles is used to separate data structure issues from quantity management. To explain the introduced mechanisms the fiber notation of the Möbius stripe and the Hopf fibration are given with our environment.

1 Introduction

The approach taken with our generic scientific simulation environment (GSSE [Heinzl et al., 2006a]) extends concept based programming of the STL to arbitrary dimensions similar to the grid algorithms library (GrAL [Berti, 2000]). In addition, an efficient notation for data structures and mathematical concepts is obtained. The main difference to GrAL is the introduction of the concept of fiber bundles [Butler & Bryson, 1992], which separates the mechanism of application design into base space and fiber space properties. The base space is modeled by a CW-complex [Benger, 2004], whereas the fiber space is modeled by a generic data accessor mechanism, similar to the cursor and property map concept [Abrahams et al., 2003].

1.1 Related Work

In the following some work related to our approach is briefly presented.

The **Grid Algorithms Library, GrAL** [Berti, 2000] was one of the first contributions to the unification of data structures of arbitrary dimension for the field of scientific computing. A common interface for grids with a dimensionally and topologically independent way of access and traversal was designed.

The **Fiber Library** [Benger, 2004] implements several mechanisms for base and fiber space properties. The base space is modeled by a CW-complex. For the fiber space several mechanisms are offered to handle various data models with a minimum of data specification.

For the **GSSE** we have developed a consistent data structure interface based on algebraic topology and order theory. With this interface specification we can make use of several libraries, e.g., GrAL. Another important advantage is that the GSSE is mainly built for a library centric application design, which means that application design is greatly supported.

1.2 Theory of Fiber Bundles

In this section we briefly overview the concept of fiber bundles [Benger, 2004] as description for data structures of various dimensions and topological properties.

Let E, B be topological spaces and $\pi : E \to B$ a continuous map. Then (E, B, π) is called a **fiber bundle**, if there exists a space F such that the union of the inverse images of π of a neighborhood $U_b \subset B$ of each point $b \in B$ is homeomorphic to $U_b \times F$, whereby the homeomorphism ϕ has to be such that the projection pr_1 of $U_b \times F$ yields U_b again and the following diagram commutes:

$$\begin{tikzcd} \pi^{-1}(U_b) \arrow[r, "\phi"] \arrow[d, "\pi"'] & U_b \times F \arrow[dl, "pr_1"] \\ U_b & \end{tikzcd}$$

Figure 1: Structure of fiber bundles.

We have introduced [Heinzl et al., 2006b] a common interface for various data structures of arbitrary dimension. The advantages of our approach are similar to the cursor and property map but differ in several details as given in Table 1. The similarity is that both approaches can be implemented independently. The main difference is that the fiber bundle approach equips the fiber space with more structure, e.g., storage of more than one value corresponding to the traversal position, and preservation of neighborhoods. These features are especially useful in the area of scientific computing, where different data sets have to be managed, e.g., multiple scalar or vector values on vertices, faces, or cells.

	cursor and property map	fiber bundles
isomorphic base space	no	yes
traversal possibilities	STL iteration	cell complex
traversal base space	yes	yes
traversal fiber space	no	yes
data access	single data	topological space
fiber space slices	no	yes

Table 1: Cursor and property map compared to the GSSE approach.

As can be seen the concept of the cursor and property map can be extended to the fiber bundle approach with additional properties and mechanisms.

1.3 Homogeneous Interface for Data Structures

We briefly introduce parts of the interface specification for data structures. A full reference is given in [Heinzl et al., 2006b]. A formal concise definition of data structures can therewith be derived as presented in Table 2.

data structure	cell dimension	cell topology	complex topology
array/vector	0	simplex	global
SLL/stream	0	simplex	local(2)
graph	1	simplex	local
grid	2,3,4,..	cuboid	global
mesh	2,3,4,..	simplex	local

Table 2: Data structure classification scheme based on the dimension of cells, the cell topology, and the complex topology.

Examples of higher dimensional complexes for C++ with the GSSE notation are given next.

```
cell_type<3, simplex>         cell_s; // tetrahedron
cell_type<4, cuboid>          cell_c; // hypercube
complex_t<cell_s, local>    > base_s; //{1}
complex_t<cell_c, global>   > base_s; //{2}
```

Here {1} describes an unstructured tetrahedral mesh and {2} describes a cell complex based on hypercubes. In the following the poset structure and a possible rendering of a four-dimensional simplex are presented. The poset structure demonstrates the traversal capabilities of the GSSE data structure interface.

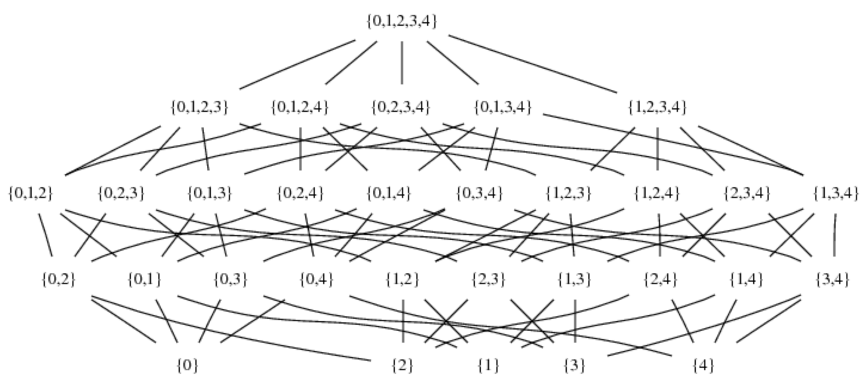

Figure 2: Poset of a four-dimensional simplex.

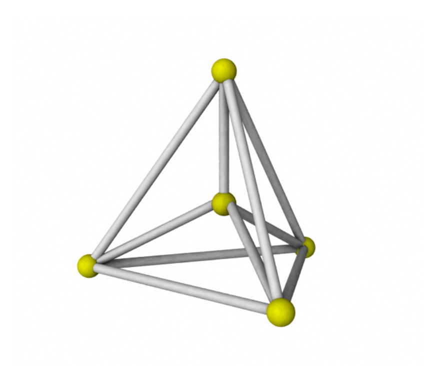

Figure 3: Rendering of a four-dimensional simplex.

The poset of the four-dimensional cube is omitted due to the large number of facets. Only one possible render image is given next:

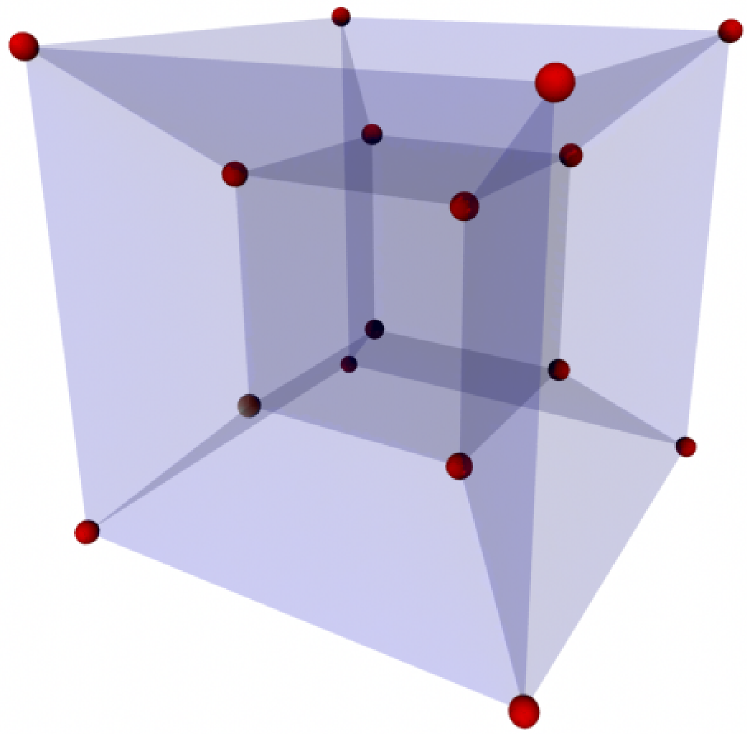

Figure 4: Rendering of a four-dimensional cube.

2 Non-Trivial Data Structures

Next to the already presented data structures which can be specified by CW-complex notation, several other non-trivial data structures exist. The embedded language of GSSE was extended to directly support the specification of several properties of data structures based on the structure of the fiber bundle approach with the corresponding base and fiber space. The topological structure of the corresponding space is given at compile-time, whereas the number of elements is given at run-time.

2.1 Möbius Stripe

The Möbius strip is a famous example of a non-trivial fiber bundle with $B = \mathbb{S}^1$ and $F = \mathbb{R}^1$ in an interval $I = [0, 1]$. The total space E can only be written locally as the product of B and F. The trivial counterpart is the infinite cylinder $\mathbb{S}^1 \times \mathbb{R}^1$, where the total space can be written globally. The modeling with the GSSE is presented in the following code snippet with our modeling language embedded within C++.

```
// compile time
//
base_space<S<1> >             base_s;
fiber_space<I<1> >            fiber_s;
total_space<base_s, fiber_s>  total_s;

// run time
//
base_s    bs(100);
fiber_s   fs(0,1,f(bs));
```

The run-time part specifies the number of discretization points for the circle, whereas the fiber is specified with the interval length and the `f(bs)` which is a simple function object describing the functional dependence of the interval of the fiber space. The following picture depicts a simple OpenGL rendering.

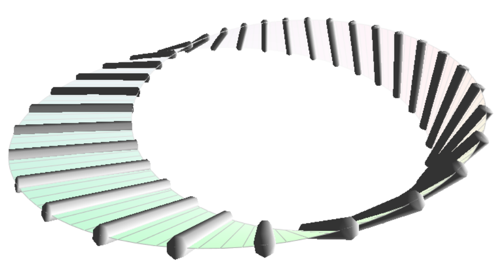

Figure 5: Visualization of the Möbius stripe based on the GSSE specification.

3 Hopf Fibration

Another example of an efficient fiber bundle notation is the Hopf fibration. The base space is modeled by S^2 (the hull of a sphere), the fiber space by S^1, and therewith the total space is modeled by a S^3 (hypersphere). The Hopf fibration can be expressed very efficiently within the GSSE.

```
// compile time
//
base_space<S<2> >               base_s;
fiber_space<S<1> >              fiber_s;
total_space<base_s, fiber_s>  total_s;

// run time
//
base_s     bs(100,100);
fiber_s    fs(100);
```

The number of discretization points is 100 in this case, which is feasible for visualization. The visualization of the Hopf fibration [Lyons, 2003] is more complex than the Möbius stripe due to the higher dimensional base space and fiber space. We use a stereographic projection where the S^3 can be seen in the three-dimensional space. The S^1 fibers are therewith projected as circles in $\mathbb{R}^3$ and depicted in Figure 6 where Figure 7 presents two possible projections.

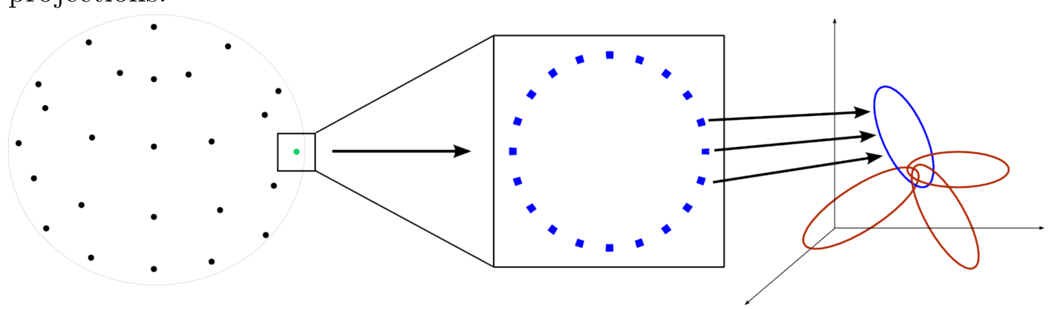

Figure 6: Projection sequence from $S^2 \rightarrow \mathbb{R}^3$ (left), $S^1 \rightarrow \mathbb{R}^4$ (middle), and finally into $\mathbb{R}^3$.

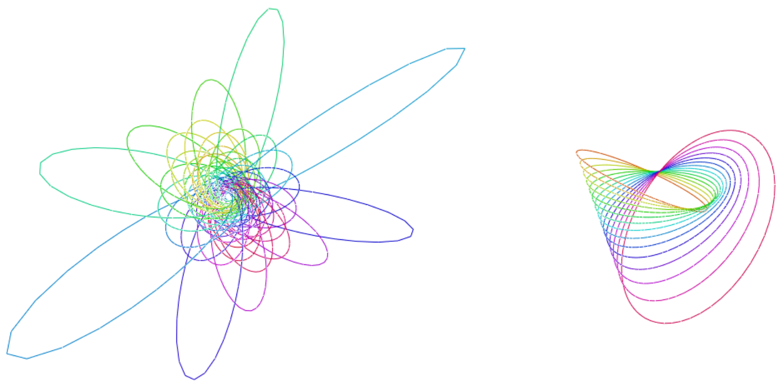

Figure 7: Left: a possbile stereographic projections of a discretized Hopf fibration. Right: visualization of a single circle of the base space.

4 Conclusions

The theory of fiber bundles separates the properties of the base space and the fiber space greatly. By formal specification of a common data structure interface based on algebraic topology a wide variety of generic environments can be used interoperably. GSSE supports the efficient notation of fiber bundles and accomplishes therewith a powerful mechanism to specify non-trivial data structures efficiently.

References

[Abrahams et al., 2003] Abrahams, D., Siek, J., & Witt, T. (2003). *New Iterator Concepts*. Technical Report N1477 03-0060, ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++.

[Benger, 2004] Benger, W. (2004). *Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model*. PhD thesis, Freie Universität Berlin.

[Berti, 2000] Berti, G. (2000). *Generic Software Components for Scientific Computing*. PhD thesis, Technische Universität Cottbus.

[Butler & Bryson, 1992] Butler, D. M. & Bryson, S. (1992). Vector Bundle Classes From Powerful Tool for Scientific Visualization. *Computers in Physics*, 6, 576–584.

[Heinzl et al., 2006a] Heinzl, R., Spevak, M., Schwaha, P., & Grasser, T. (2006a). A High Performance Generic Scientific Simulation Environment. In *Proc. of the PARA Conf.* (pp.$\tilde{6}$1). Umea, Sweden.

[Heinzl et al., 2006b] Heinzl, R., Spevak, M., Schwaha, P., & Selberherr, S. (2006b). A Generic Topology Library. In *Library Centric Sofware Design, OOPSLA* (pp. 85–93). Portland, OR, USA.

[Lyons, 2003] Lyons, D. (2003). An Elementary Introduction to the Hopf Fibration. *Mathematics Magazine*, 76(2), 87–98.