Tracking Structures

Etienne Forest
KEK
Tsukuba, Japan

Talk at Jefferson Laboratory
6/13/2003

The Secret of Modern Propaganda

- Its task is the highest creative art of putting complicated events and facts in a way simple enough to be understood by the man on the street.
- If propaganda is to succeed, it must know what it wants. It must keep a clear and firm goal in mind, and seek the appropriate means and methods to reach that goal.
- Propaganda is a means to an end. Its purpose is to lead the people to an understanding that will allow them to willingly and without internal resistance devote themselves to the tasks and goals of a superior leadership.
- Good propaganda does not need to lie, indeed it may not lie. It has no reason to fear the truth. It is a mistake to believe that people cannot take the truth. They can. It is only a matter of presenting the truth to people in a way that they will be able to understand. A propaganda that lies proves that it has a bad cause. It cannot be successful in the long run. A good propaganda will always come along that serves a good cause. But propaganda is still necessary if a good cause is to succeed. A good idea does not win simply because it is good. It must be presented properly if it is to win.

What is the complex idea here?

Write a tracking code primarily for single particle dynamics which will handle all possible “topologies” : common beamlines, pretzels, recirculators, dog-bones and beyond– in a way which mimics nature so that objects in real life have a parallel existence in the silicon world.

This code should admit in theory arbitrarily complex magnets and still computes uncompromisingly all the various quantities which interests us: radiation “integrals”, lattice functions and their nonlinear equivalents. Within its model(s) for the various magnets, it must be capable of computing ANYTHING correctly.

Question: Can this set of goals be reduced into a simple idea which can guide us or are we condemned to keep this huge set of requirements in our heads at all time?

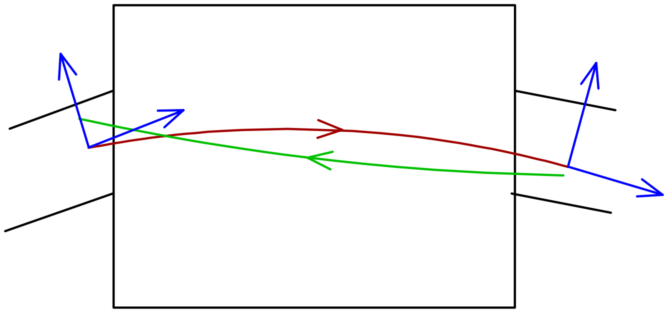
Answer: Thanks to existence of polymorphic types (Taylor-real*8), normal form theory and the existence of pointers in modern languages we know with absolute certainty that the above requirements can be reduced to a single simple idea from which everything else follows.

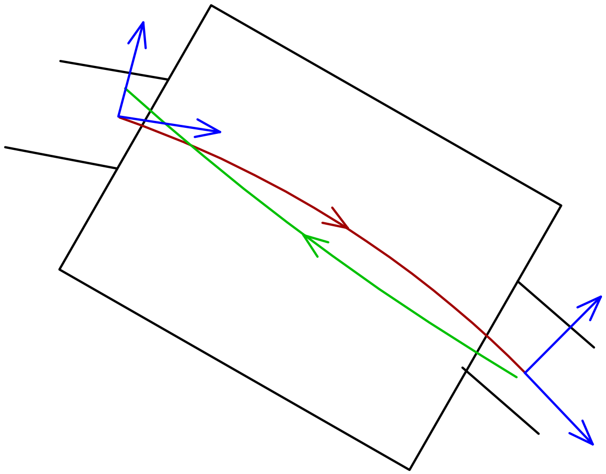
The failure to uncover this idea or, as in the case of the CLASSIC/MAD9 gang, to appreciate its importance, leads inexorably to failure as I predicted in 1995 and was later confirmed.

So what is that idea?

The Factory Magnet Object

- Attached to the physical objects are two single particle propagators: forward and backward.
- The magnet and its propagators cannot depend on the beam line in which they are ultimately placed.
- Our beam line structure (layout) must permit the full Euclidean group acting on the local propagators: translations and rotations.





Conclusion: the alignment properties of the propagators immersed in the beam line structure (layout) will be inherited from the layout. And, conversely, in the single dynamics case, the layout will derive its propagators from the factory magnet object.

Simple Principles

- The two magnet propagators are attached to the magnet and depend on internal details irrelevant to our code. We access these details through physical knobs.
- The beam line structure must accommodate these magnets and permit misalignments which cannot depend on the internal details of the propagators.

At no point in time can we compromise on the above two points. These are sine qua non conditions. We must be unchangeable in our principle, hard as steel in our organization but supple and adaptable in our tactics and methods to achieve these goals.

Remarkable results:

If we insist during the development of the code to respect fanatically the two principles outline above, then thanks to the modern tools at our disposal since the early 1990s, this code will satisfy all the complex prerequisites listed before.

In particular, the separation of the beam line structure from the factory magnet object will impose a novel idea on us: a beam line is NOT a collection of magnet propagators. The concept that a beam line is a sequential list of magnet propagators has been known to be wrong since the early 90s and yet the misguided CLASSIC gang misprogrammed MAD9/CLASSIC when they ought to have known better. **It is mathematically and physically wrong.**

The beam line is actually a collection of discrete "s" variables: on the computer it is represented as a link list which contains Euclidean patches/misalignments and a pointer to the actual factory magnet propagators.

This will permit the correct simulation of recirculators, pretzel and other oddities. It will follow that, in a code equipped with the correct structures, insuring that magnets in common beam lines behave as one and the same magnet will be automatic rather than something imposed by hacker-style programming. In reality, in the silicon world, they will be the same object as in the real world.

Subsidiary remarkable results:

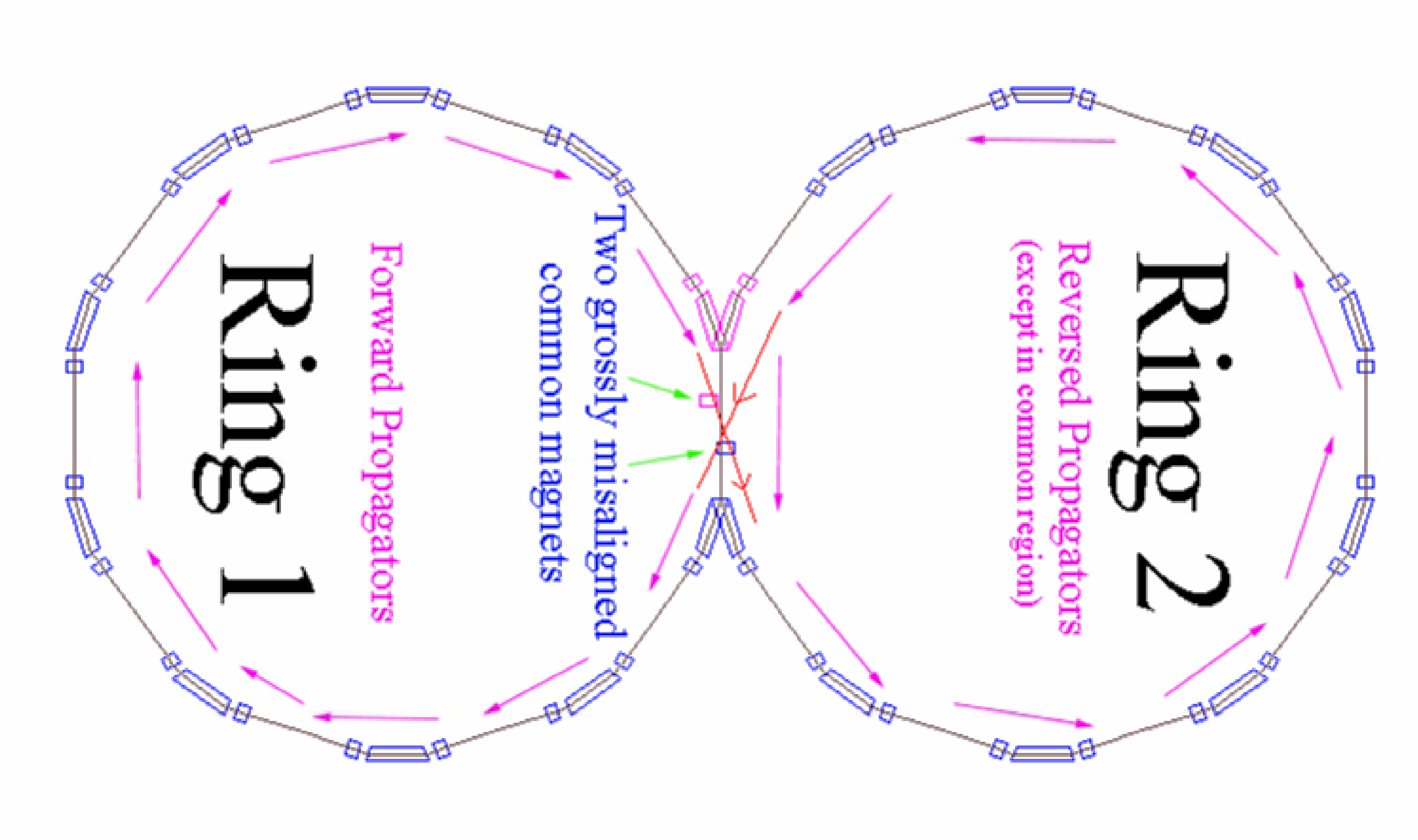
Although our set of simple principles does not address the issue of algorithms (computation of lattice functions, etc…), in fact, this issue is automatically handled. Or to be more precise, its handling is not relevant to code design. How can that be? Three words:

1) Automatic Differentiation (Berz's DA for example)
2) Operator Overloading/Polymorphism (Bengtsson's original idea, re-implemented in F95 by myself)
3) Normal Form theory on Taylor series maps and moments

These things, in modern languages, such as C++ in 1990, and now in F95 as well, insure that **WE NEED NOT** to worry about "algorithms" for our favorite objects (β,α,γ, and the rest)

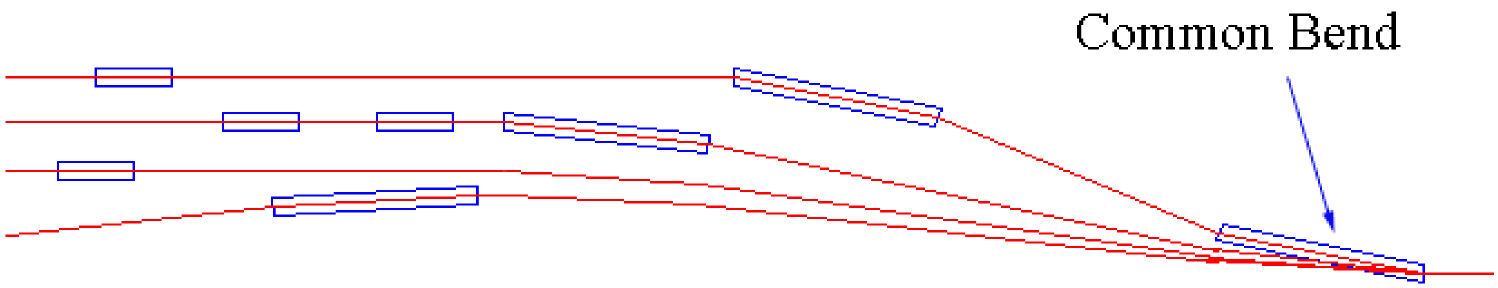
And yet the MAD9/CLASSIC effort worried about these things constantly, putting an emphasis on field representations and algorithms for the computations of lattice functions. In the end, CLASSIC is incapable of doing any recirculators! A totally predictable failure that prompted Bengtsson and myself to quit the Titanic before it left port!

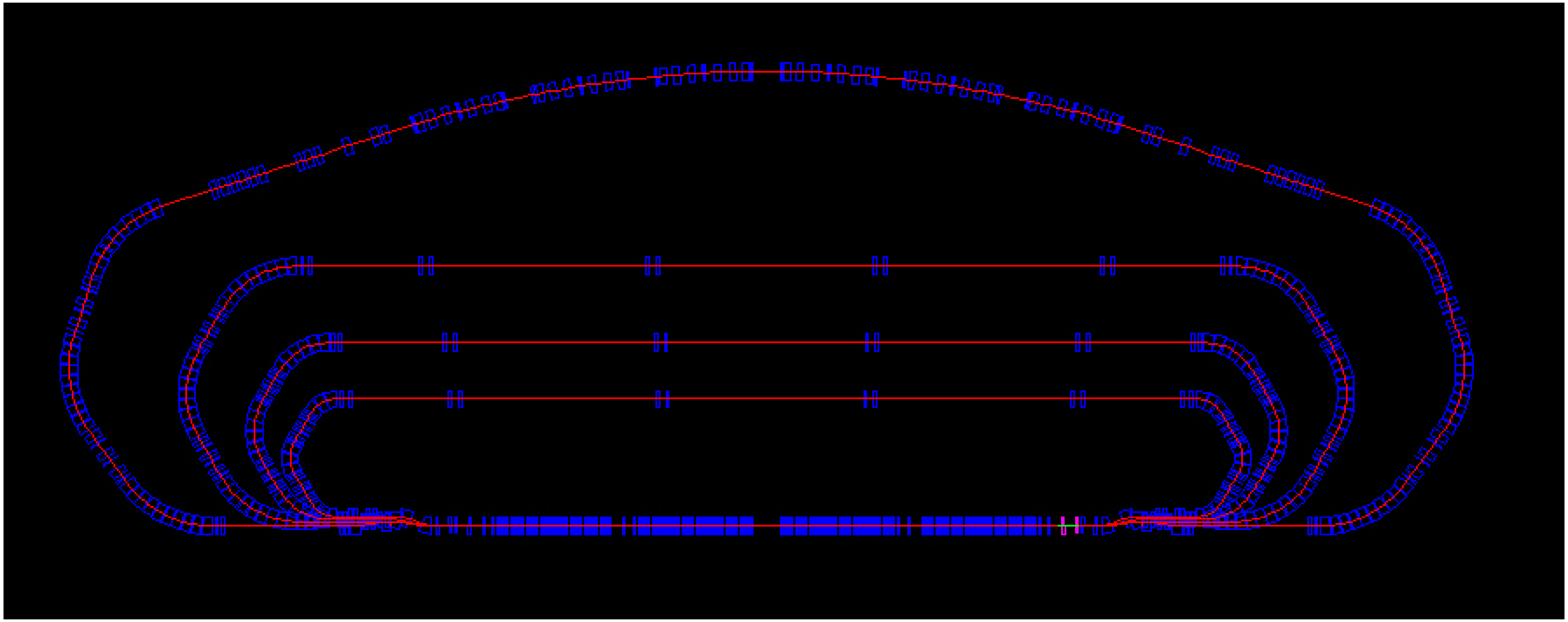
Examples of Structures



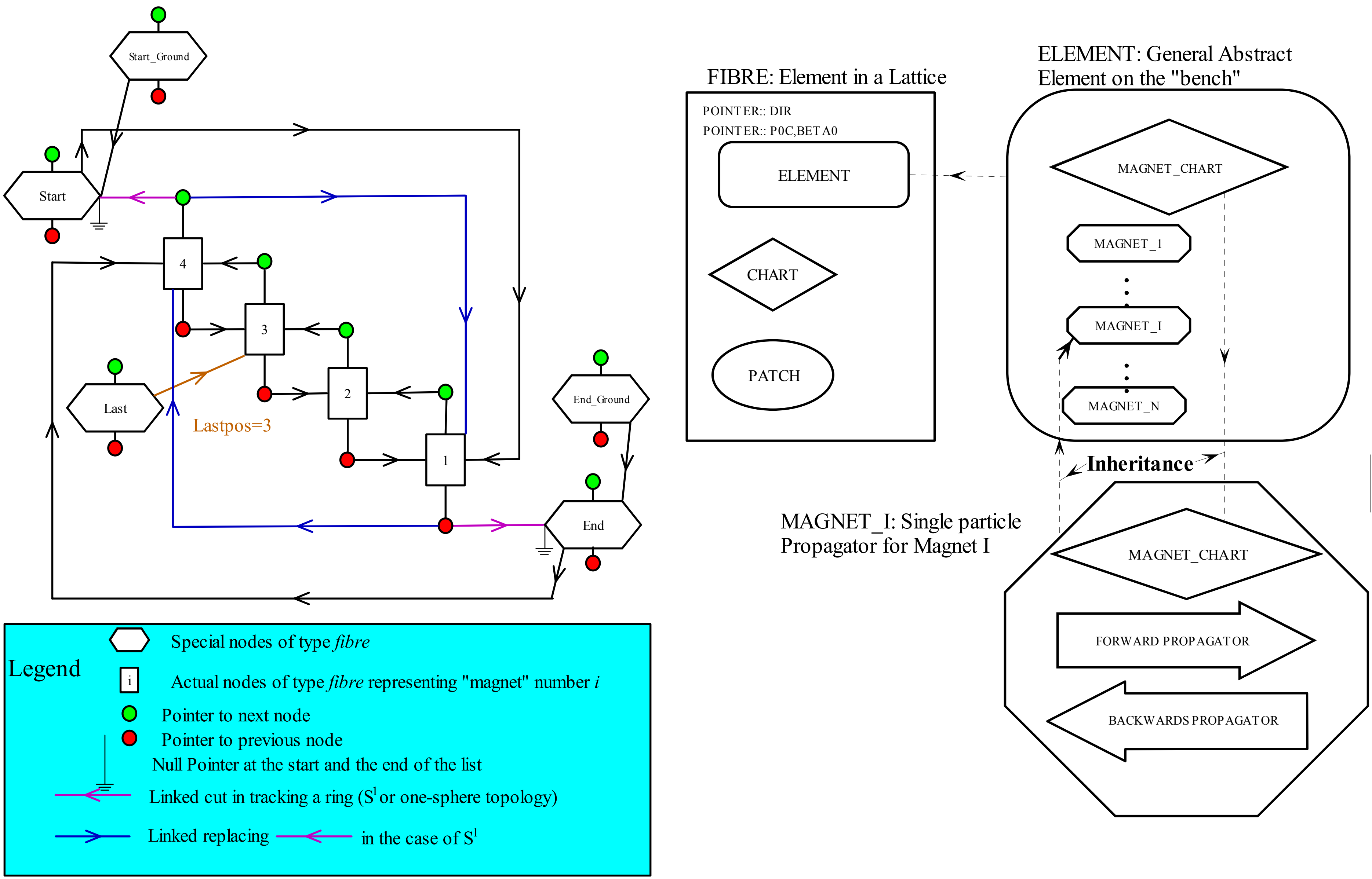
Two Rings or One double Ring

Switching beam lines

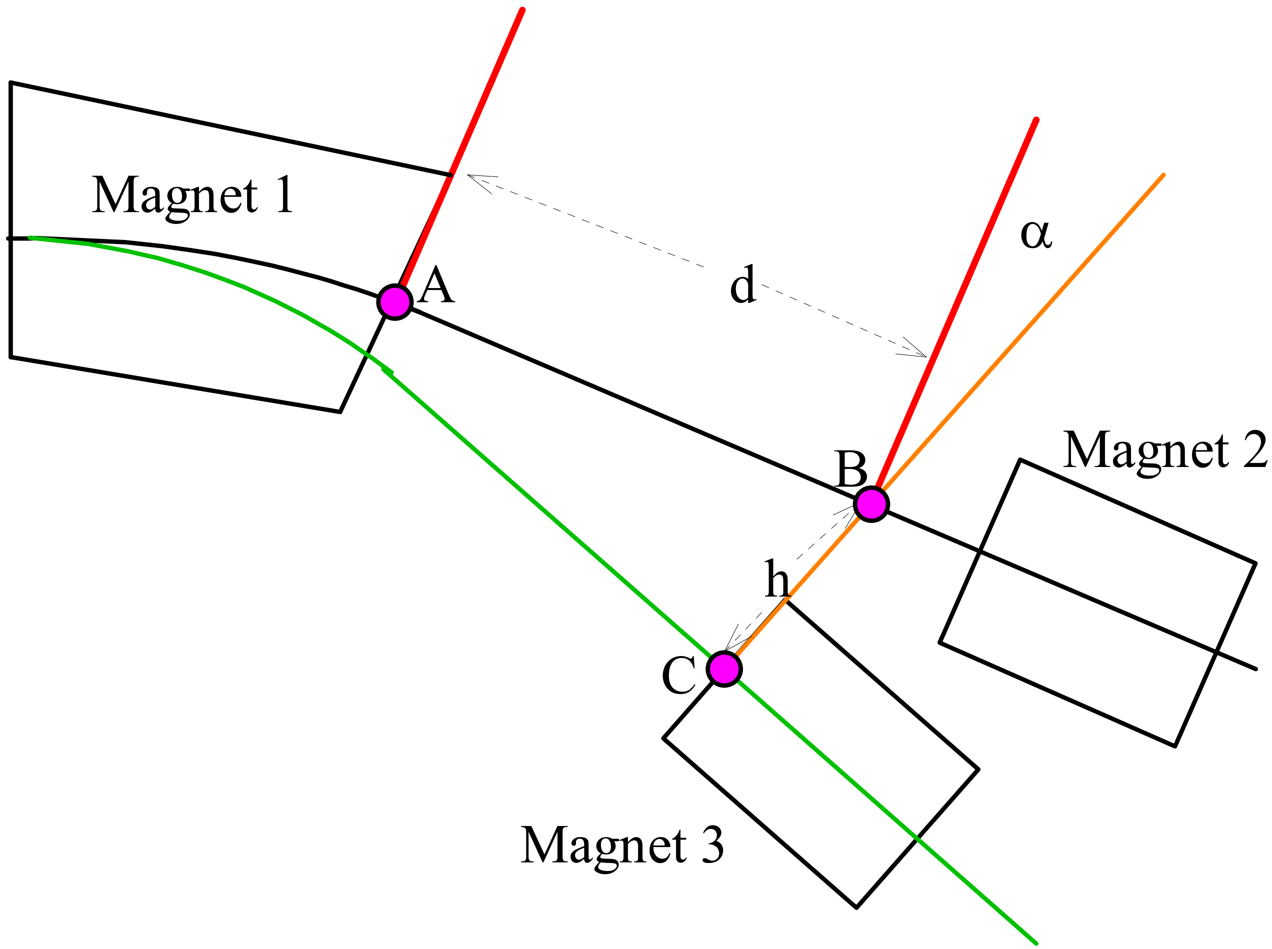




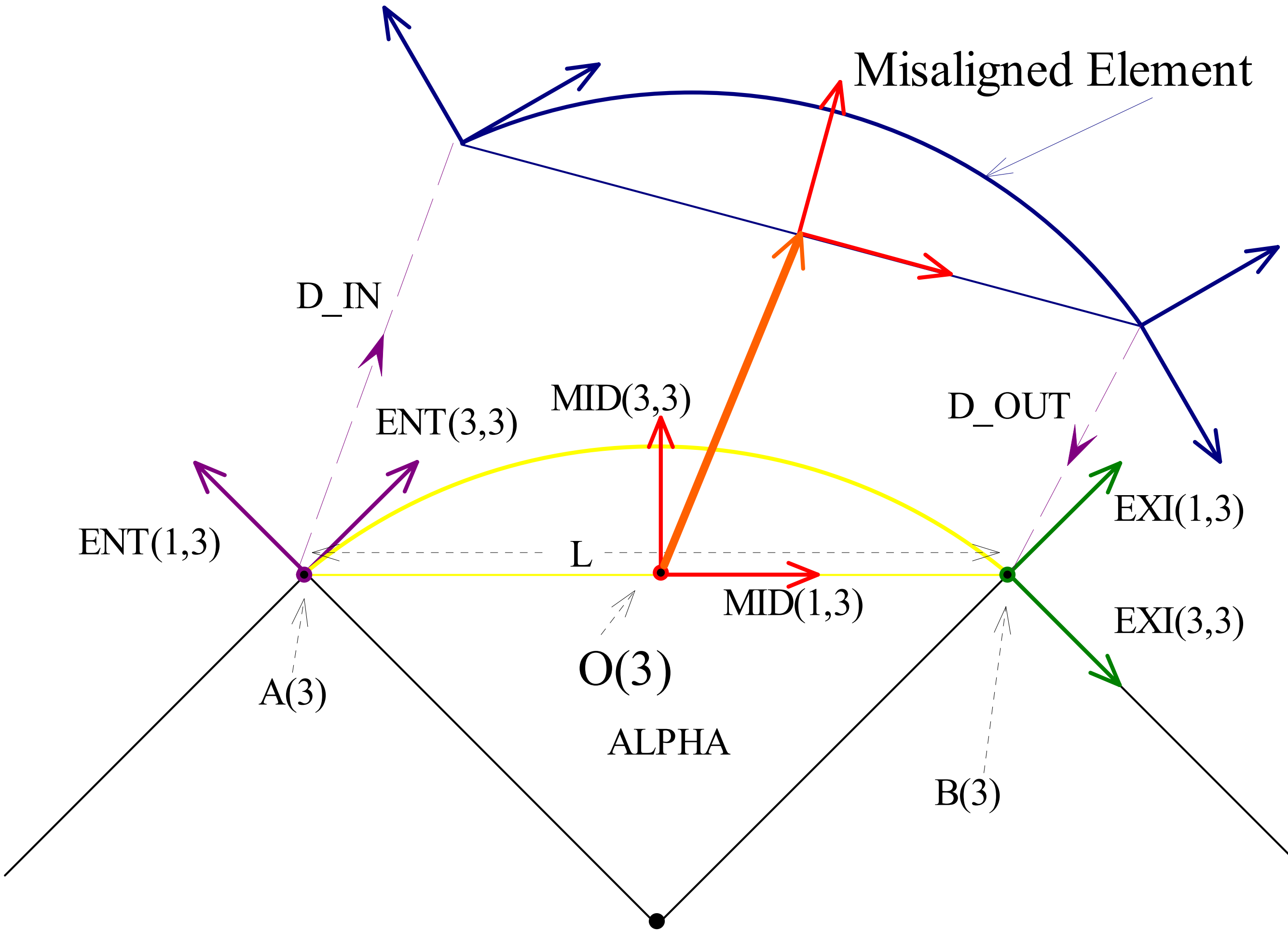
Layout Structure and Fibre



Patch

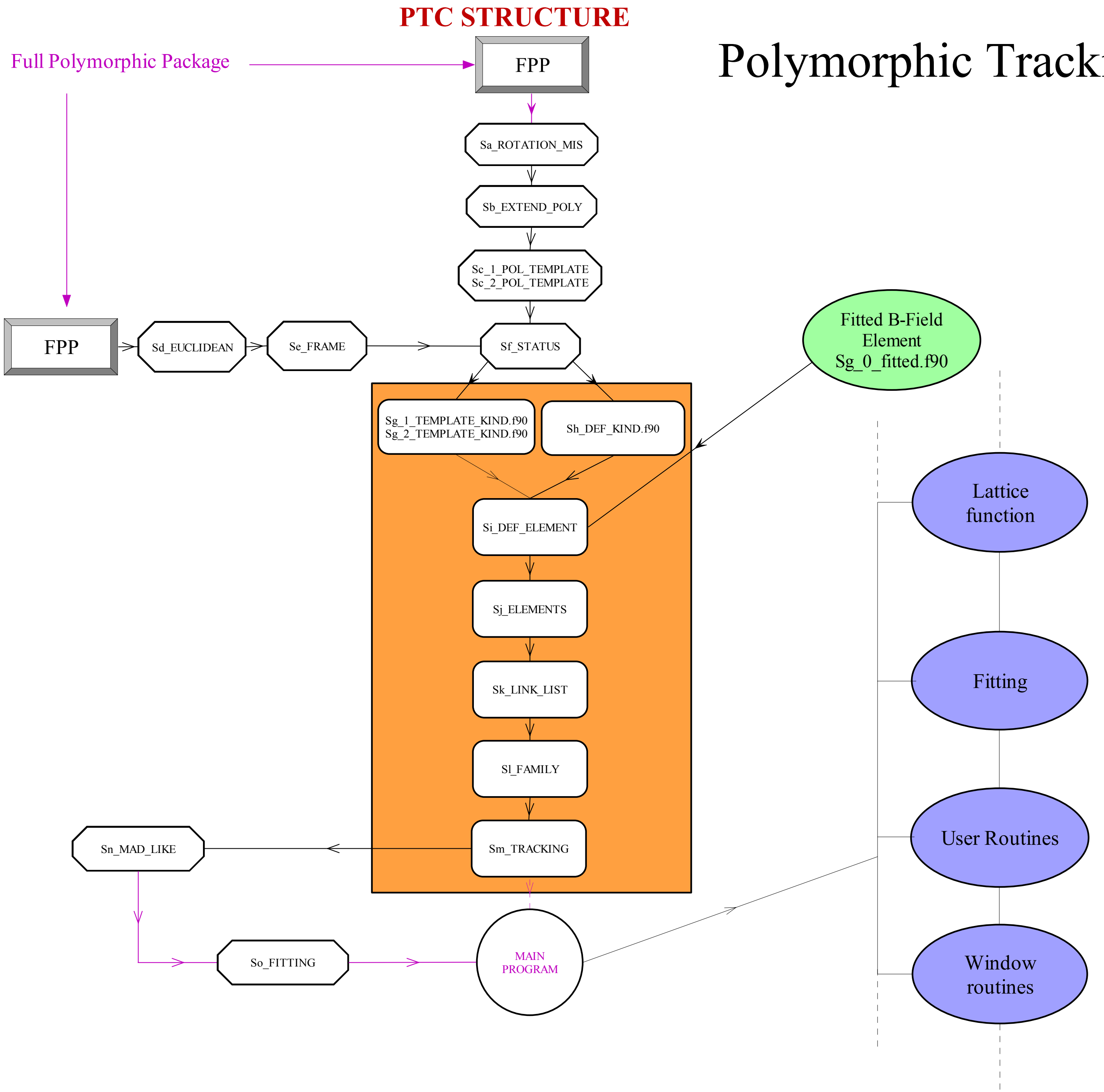


Frames

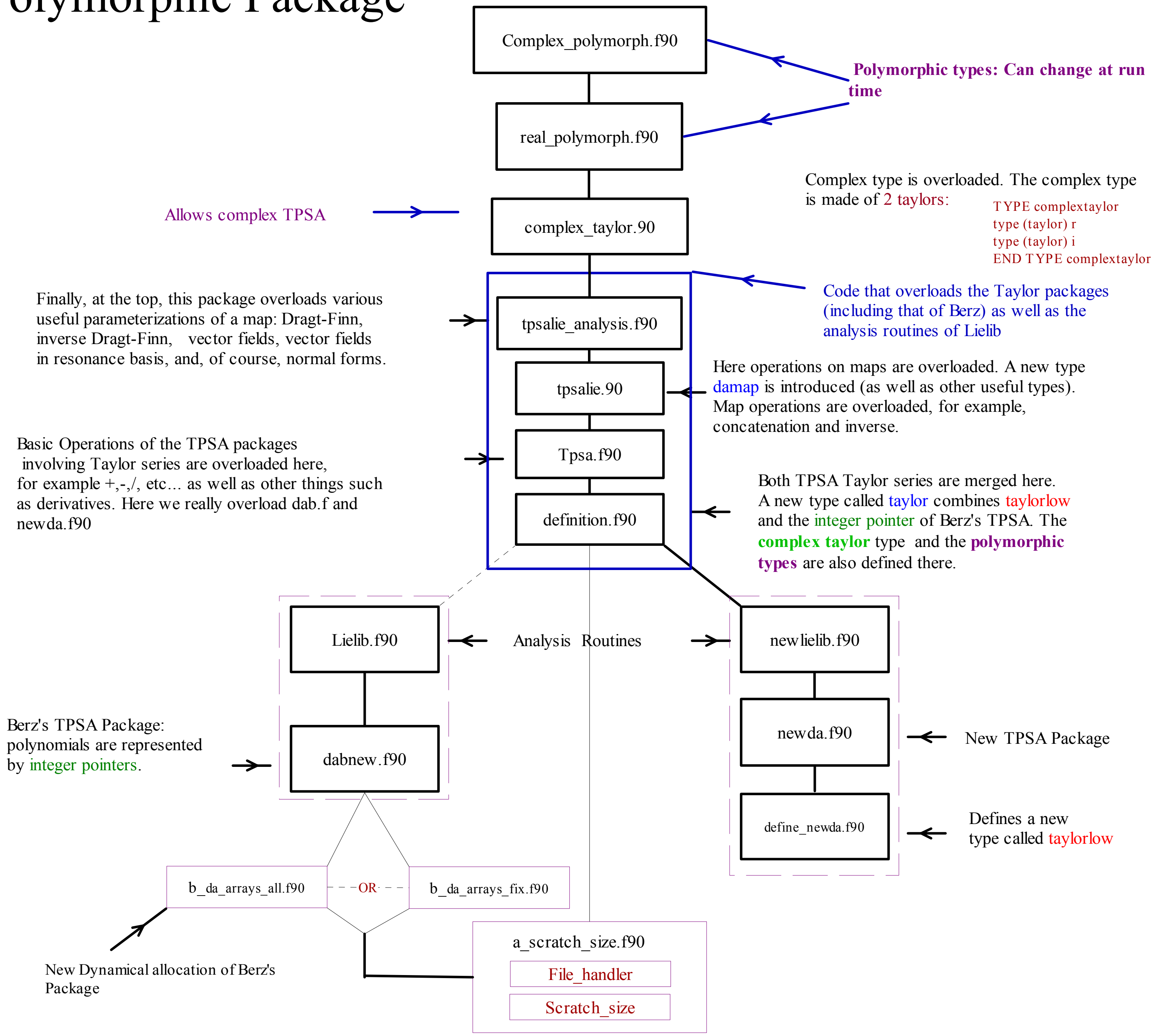


Misaligned Element
D_IN
MID(3,3)
ENT(3,3)
D_OUT
EXI(1,3)
ENT(1,3)
L
MID(1,3)
EXI(3,3)
A(3)
O(3)
ALPHA
B(3)

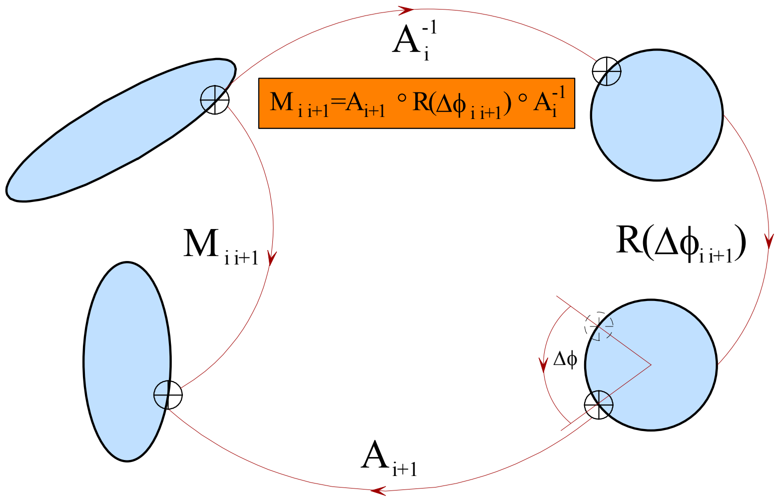
Polymorphic Tracking Code

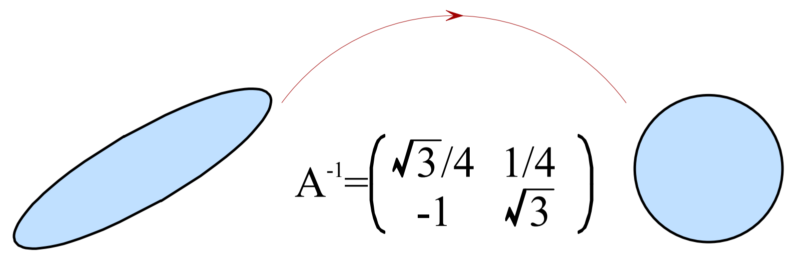


Fully Polymorphic Package



Algorithms????





$$\underbrace{M_{i\;i+1} \circ A_i}_{\text{Tracked transformation}} \circ R(\Delta\phi^{-1}_{i\;i+1}) = \underbrace{A_{i+1}}_{\text{Courant}-\text{Snyder}}$$

```
y=x+norm%a_t

i=1
p=>psr%start
   write(16,*) i,p%mag%name, (y(1).sub.'10')**2+(y(1).sub.'01')**2
   write(6,*) i,p%mag%name,  (y(1).sub.'10')**2+(y(1).sub.'01')**2

do i=1,psr%n
 call TRACK(PSR,y,i,i+1,DEFAULT)
        p=>psr%start
   write(16,*) i+1,p%mag%name,(y(1).sub.'10')**2+(y(1).sub.'01')**2
enddo
```

$$\underbrace{M_{i\ i+1} \circ A_i}_{\text{Tracked transformation}} \circ R(\Delta\phi^{-1}_{i\ i+1}) = \underbrace{A_{i+1}}_{\text{Courant - Snyder}}$$

Algorithms are nowhere to be found in PTC!