COSY INFINITY

Martin Berz, Kyoko Makino, Robert Hipple, Eremey Valetov

Department of Physics and Astronomy
Michigan State University

The Crux of Symplectic Tracking

- Symplecticity governs all Hamiltonian systems
- Symplecticity is rather hard to enforce; thus:
- Either try hard to track the **right** system, end up being non-symplectic
- Or track the **wrong** system with symplectic models

The Crux of Symplectic Tracking

- Symplecticity governs all Hamiltonian systems
- Symplecticity is rather hard to enforce; thus:
- Either try hard to track the **right** system, end up being non-symplectic
- Or track the **wrong** system with symplectic models

Right System, Non-Symplectic

- Best possible fields, potentials
- Exact Hamiltonian
- Good integrators
- Examples: numerical integrators,
  Map codes

The Crux of Symplectic Tracking

- Symplecticity governs all Hamiltonian systems
- Symplecticity is rather hard to enforce; thus:
- Either try hard to track the **right** system, end up being non-symplectic
- Or track the **wrong** system with symplectic models

Right System, Non-Symplectic

- Best possible fields, potentials
- Exact Hamiltonian
- Good integrators
- Examples: numerical integrators, Map codes

Wrong System, Symplectic

- Approximate Hamiltonian
- Approximate Fields
- Symplectic Integrators
- Examples: Kick codes

The Crux of Symplectic Tracking

- Symplecticity governs all Hamiltonian systems
- Symplecticity is rather hard to enforce; thus:
- Either try hard to track the **right** system, end up being non-symplectic
- Or track the **wrong** system with symplectic models

Right System, Non-Symplectic

- Best possible fields, potentials
- Exact Hamiltonian
- Good integrators
- Examples: numerical integrators, Map codes

Wrong System, Symplectic

- Approximate Hamiltonian
- Approximate Fields
- Symplectic Integrators
- Examples: Kick codes

Goal: Search **wrong** system nearest to **righ**t system

- Start with best possible **right** system
- High-order transfer map using “best” fields
- This makes it **wrong** - finite order, numerical error
- Symplectify using “nearest” via Hofer’s metric

Transfer Map Method and Differential Algebras

- The transfer map $\mathcal{M}$ is the flow of the system ODE.

$$\vec{z}_f = \mathcal{M}(\vec{z}_i, \vec{\delta}),$$

where $\vec{z}_i$ and $\vec{z}_f$ are the initial and the final condition, $\vec{\delta}$ is system parameters.

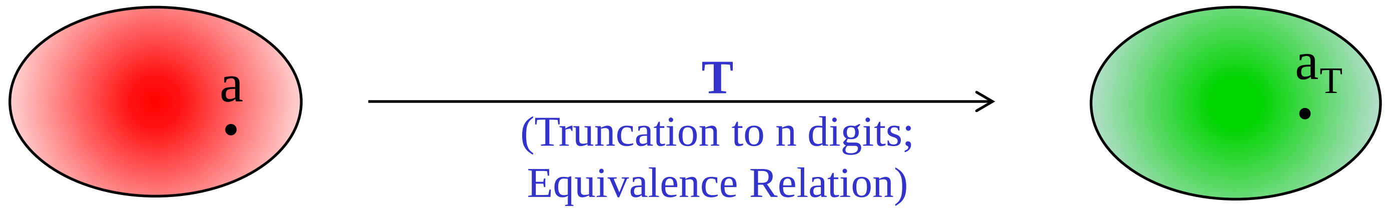
- For a repetitive system, only one cell transfer map has to be computed. Thus, it is much faster than ray tracing codes (i.e. tracing each individual particle through the system).
- The Differential Algebraic method allows a very efficient computation of high order Taylor transfer maps.
- The Normal Form method can be used for analysis of nonlinear behavior.

Differential Algebras (DA)

- it works to arbitrary order, and can keep system parameters in maps.
- very transparent algorithms; effort independent of computation order.

The code **COSY Infinity** has many tools and algorithms necessary.

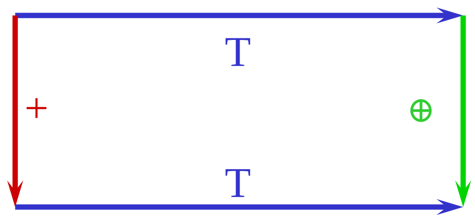
Number Fields and Floating Point Numbers



Real Numbers

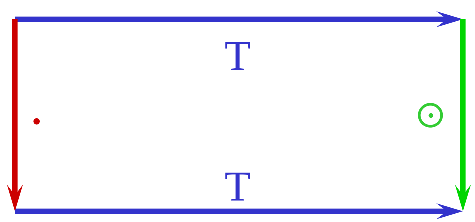
Floating Point Numbers

$c = a + b$



$c_T = a_T \oplus b_T$

$c = a \cdot b$



$c_T = a_T \odot b_T$

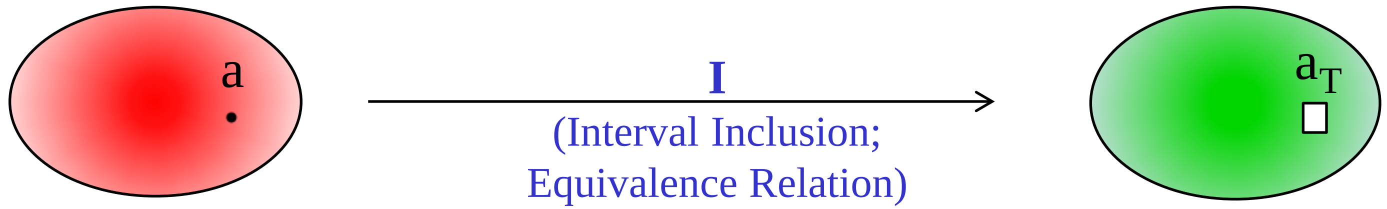
Field
(Also want "exp", "sin"
etc: Banach Field)

Diagrams commute
"approximately"

Field
("approximately")

T: Extracts information
considered relevant

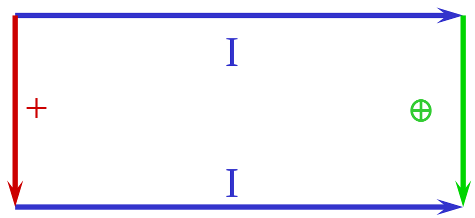
Number Field Inclusions (Intervals)



Real Numbers

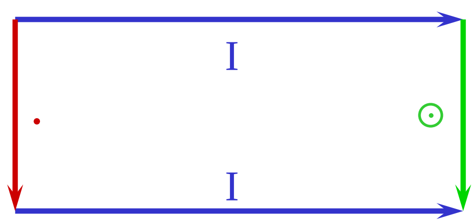
Floating Point Intervals

$c = a + b$



$c_I = a_I \oplus b_I$

$c = a \cdot b$



$c_I = a_I \odot b_I$

Field

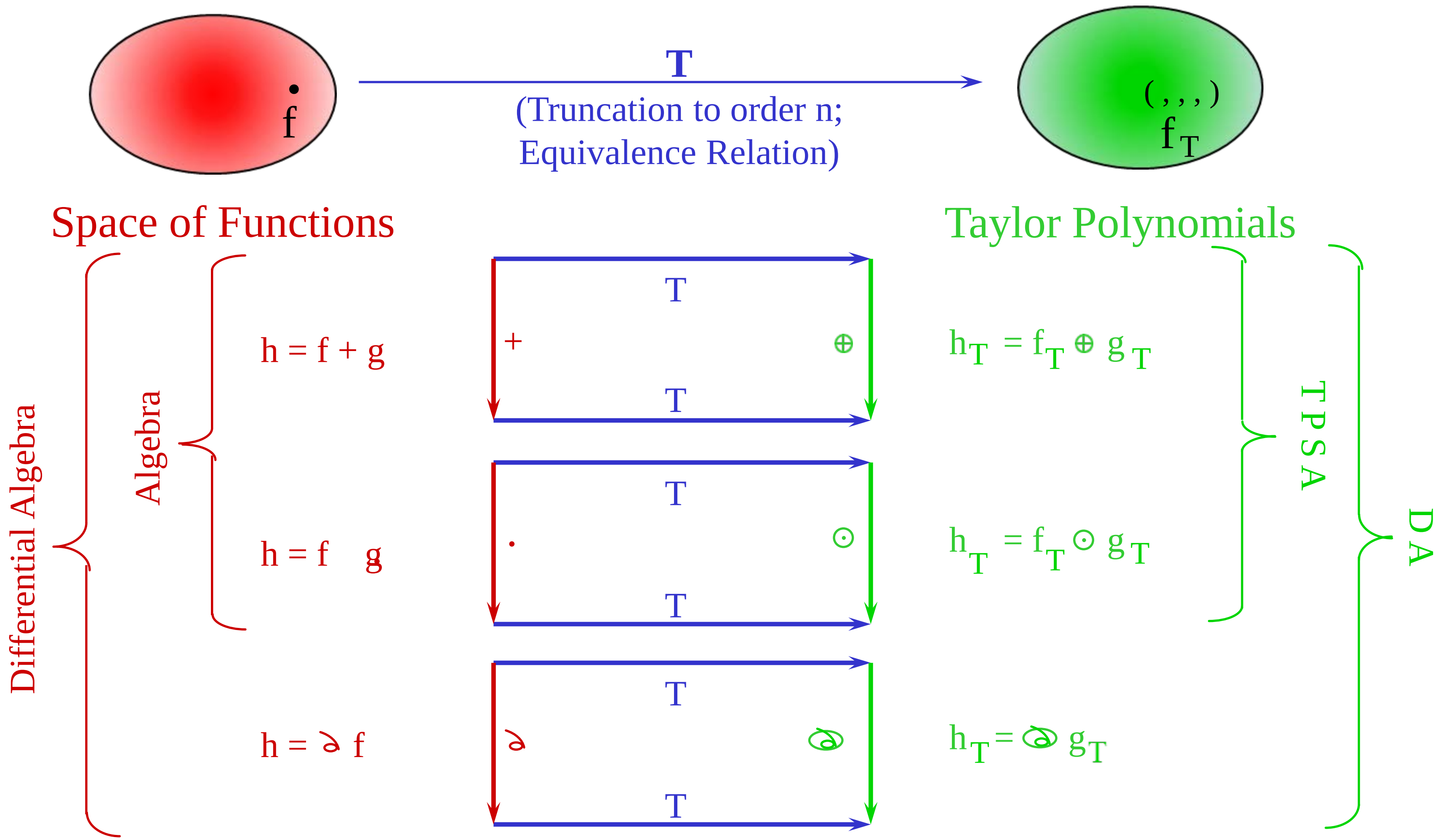
(Also want "exp", "sin" etc: Banach Field)

Diagrams commute exactly!

Little Algebraic Structure

I: Extracts information considered relevant

Function Algebras

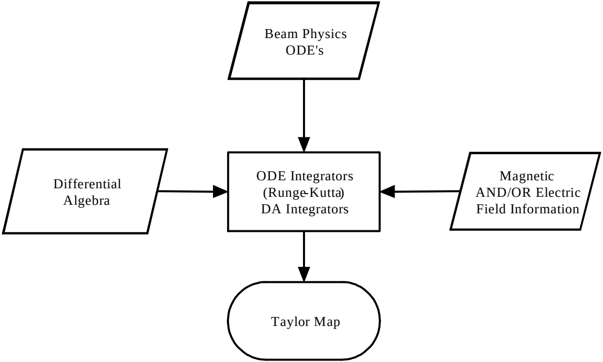


Differential Algebra
(also want "exp", "sin" etc: Banach DA)

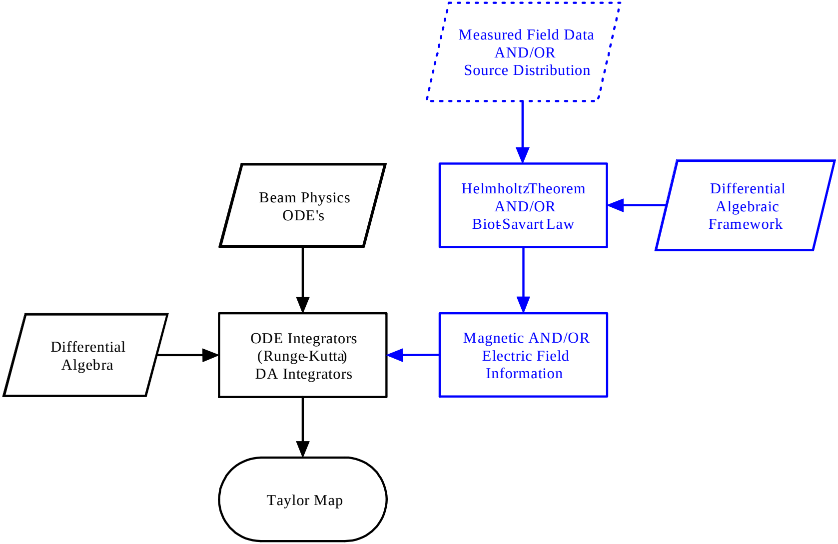
Diagrams commute exactly

T: Extracts information considered relevant

Differential Algebra
(even Banach DA)



Analytic formula or local expansion of the field should be specified



Quadrupole example: $\vec{B}(x, y, s) = (k_q y, k_q x, 0)$

COSY INFINITY

- Arbitrary order (in practice orders 7 to 11 are reasonable)
- Maps depending on parameters
- No approximations in motion or field description
- Large library of elements, magnetic or electric
- Arbitrary Elements (you specify fields)
- Very flexible input language
- Powerful interactive graphics
- Errors: position, tilt, rotation
- Symplectic tracking through maps
- Normal form methods
- Spin dynamics
- Fast fringe field models using SYSCA approach
- Reference manual (80 pages) and Programming manual (90 pages)

Elements in COSY

- Magnetic and electric multipoles
- Superimposed multipoles
- Combined function bending magnets with curved edges
- Electrostatic deflectors
- Wien filters
- Wigglers
- Solenoids, various field configurations
- 3 tube electrostatic round lens, various configurations
- Exact fringe fields to all of the above
- Fast fringe fields (SYSCA)
- General electromagnetic element (measured data)
- Glass lenses, mirrors, prisms with arbitrary surfaces
- Misalignments: position, angle, rotation

All can be computed to arbitrary order, and the dependence on any of their parameters can be computed.

The Operator ∂^{-1}on Taylor Models

Let (P_n, I_n) be an n-th order Taylor model of f. From this we can obtain a Taylor model for the indefinite integral $\partial_i^{-1} f = \int f \; dx_i'$ with respect to variable x_i.

Taylor polynomial part: $\int_0^{x_i} P_{n-1} dx_i'$,

Remainder Bound: $(B(P_n - P_{n-1}) + I_n) \cdot B(x_i)$, where $B(P)$ is a polynomial bound.

So define the operator ∂_i^{-1}on space of Taylor models as

$$\partial_i^{-1}(P_n, I_n)$$
$$= \left(\int_0^{x_i} P_{n-1} dx_i' \;\; , \quad (B(P_n - P_{n-1}) + I_n) \cdot B(x_i) \right)$$

Taylor Models for the Flow

Goal: Determine a Taylor model, consisting of a Taylor Polynomial and an interval bound for the remainder, for the flow of the differential equation

$$\frac{d}{dt}\vec{r}(t) = \vec{F}(\vec{r}(t), t)$$

where $\vec{F}$ is sufficiently differentiable. The Remainder Bound should be fully rigorous for all initial conditions $\vec{r}_0$ and times t that satisfy

$$\begin{aligned} \vec{r}_0 &\in [\vec{r}_{01}, \vec{r}_{02}] = \vec{B} \\ t &\in [t_0, t_1]. \end{aligned}$$

In particular, $\vec{r}_0$ itself may be a Taylor model, as long as its range is known to lie in $\vec{B}$.

The Use of Schauder's Theorem

Re-write differential equation as integral equation

$$\vec{r}(t) = \vec{r}_0 + \int_{t_0}^{t} \vec{F}(\vec{r}(t'), t')\, dt'.$$

Now introduce the operator

$$A : \vec{C}^0[t_0, t_1] \to \vec{C}^0[t_0, t_1]$$

on space of continuous functions via

$$A\left(\vec{f}\right)(t) = \vec{r}_0 + \int_{t_0}^{t} \vec{F}\left(\vec{f}(t'), t'\right)\, dt'.$$

Then the solution of ODE is transformed to a fixed-point problem on space of continuous functions

$$\vec{r} = A(\vec{r}).$$

Theorem (Schauder): *Let A be a continuous operator on the Banach Space X. Let $M \subset X$ be compact and convex, and let $A(M) \subset M$. Then A has a fixed point in M, i.e. there is an $\vec{r} \in M$ such that $A(\vec{r}) = \vec{r}$.*

The Polynomial of the Self-Including Set

Attempt sets M^* of the form

$$M^* = M_{\vec{P}^* + \vec{I}^*} \text{ where}$$
$$\vec{P}^* = \mathcal{M}_n(\vec{r}_0, t),$$

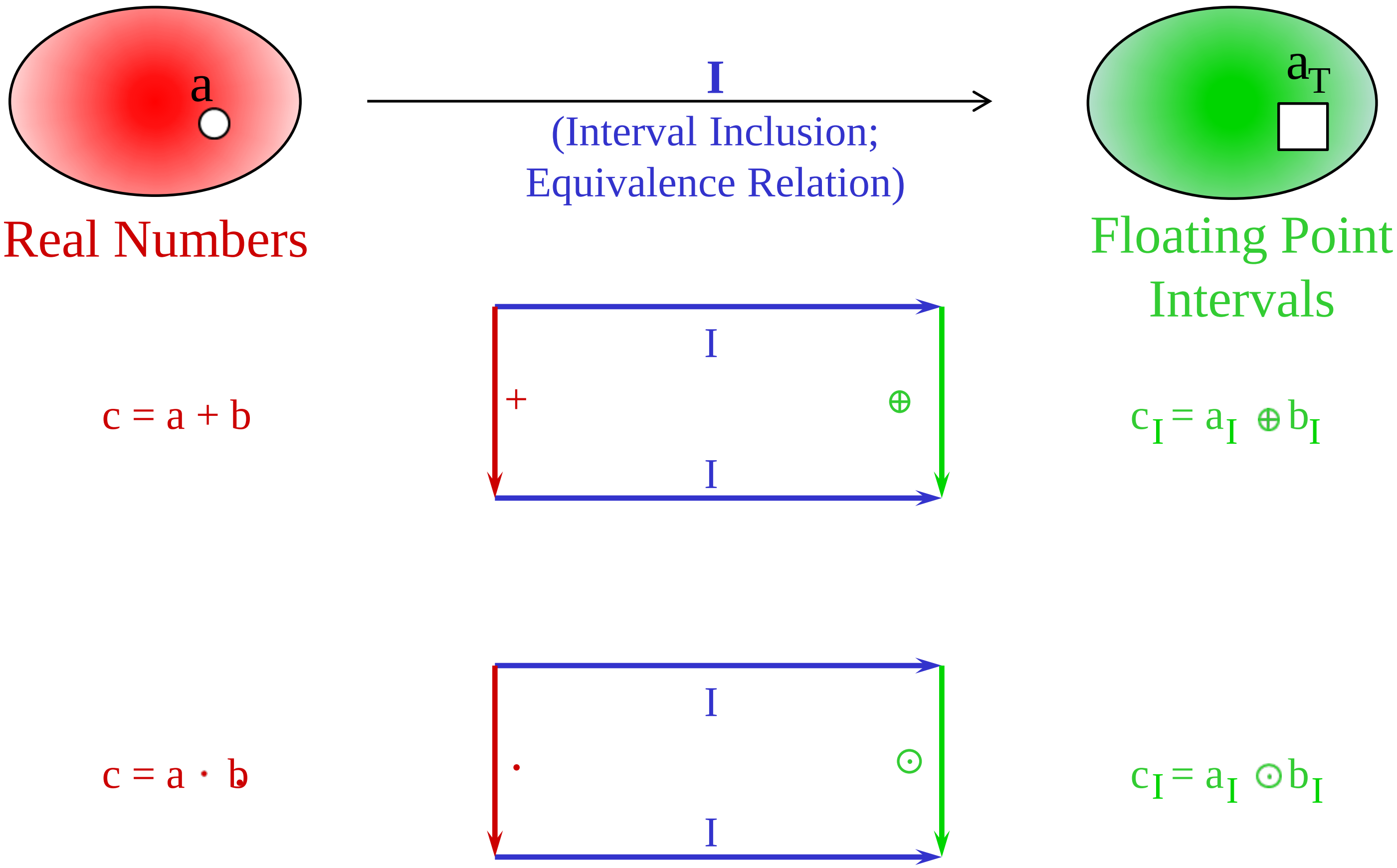
the n-th order Taylor expansion of the flow of the ODE. It is to be expected that $\vec{I}^*$ can be chosen smaller and smaller as order n of $\vec{P}^*$ increases.

This requires knowledge of nth order flow $\mathcal{M}_n(\vec{r}_0, t)$, including time dependence. It can be obtained by iterating in polynomial arithmetic, or Taylor models without treatment of a remainder. To this end, one chooses an initial function $\mathcal{M}_n^{(0)}(\vec{r}, t) = \mathcal{I}$, where $\mathcal{I}$ is the identity function, and then iteratively determines

$$\mathcal{M}_n^{(k+1)} =_n A(\mathcal{M}_n^{(k)}).$$

This process converges to the exact result $\mathcal{M}_n$ in exactly n steps.

Set Inclusions (Intervals)



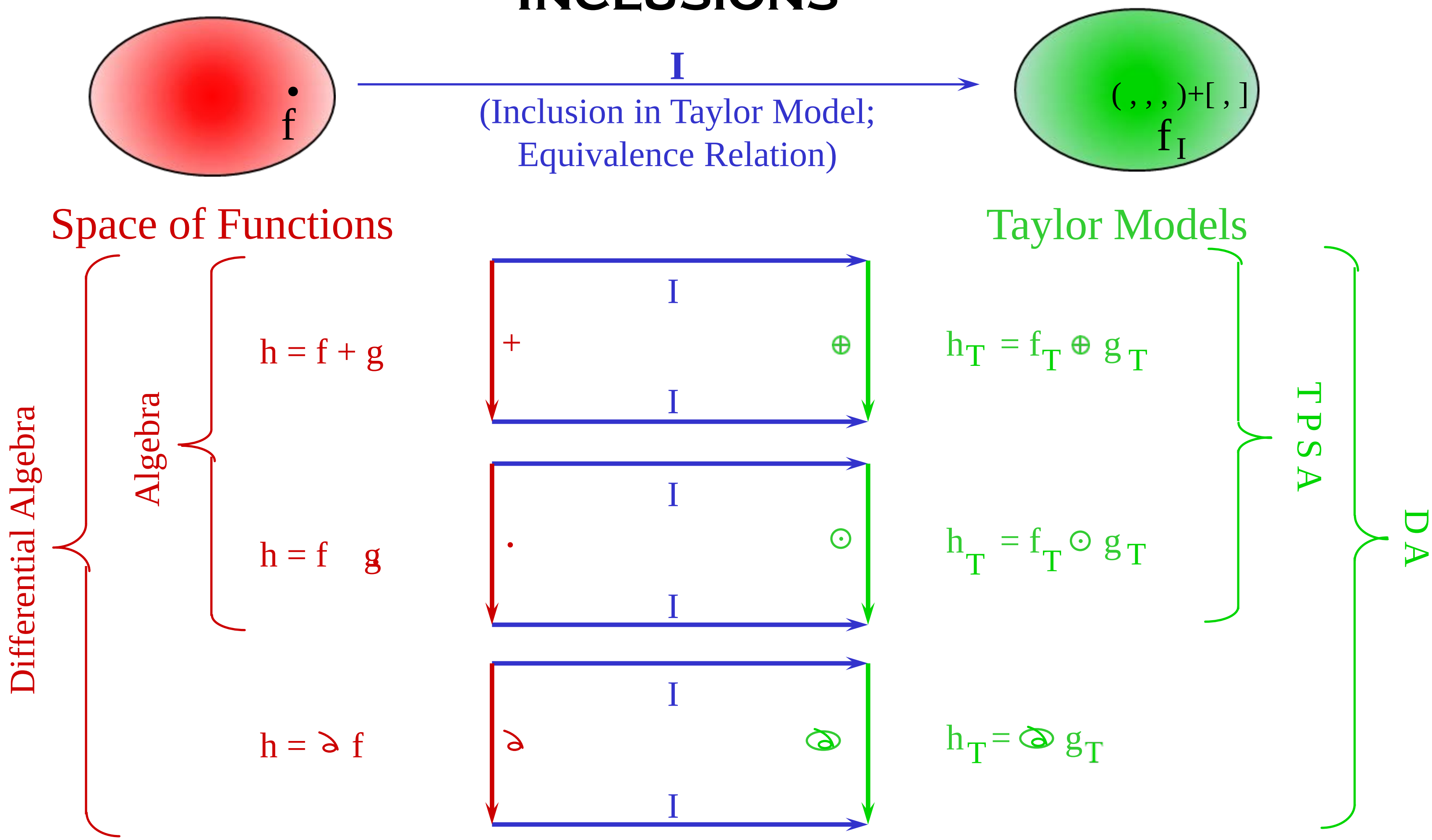
Field
(Also want “exp”, “sin”
etc: Banach Field)

Diagrams commute
exactly!

**Little Algebraic
Structure**

I: Extracts information
considered relevant

Function Algebra Inclusions



Differential Algebra
(also want "exp", "sin" etc: Banach DA)

Diagrams commute exactly

T: Extracts information considered relevant

Differential Algebra
(even Banach DA)

The Remainder of the Self-Including Set

Now try to find $\vec{I}^*$ such that

$$A(\mathcal{M}_n + \vec{I}^*) \subset \mathcal{M}_n + \vec{I}^*,$$

the Schauder inclusion requirement. Suitable choice for $\vec{I}^*$ requires experimenting, but is greatly simplified by the observation

$$\vec{I}^* \supset \vec{I}^{(0)} = A(\mathcal{M}_n(\vec{r},t) + [\vec{0},\vec{0}]) - \mathcal{M}_n(\vec{r},t).$$

Evaluating the right hand side in RDA yields a lower bound for $\vec{I}^*$, and a benchmark for the size to be expected. Now iteratively try

$$\vec{I}^{(k)} = 2^k \cdot \vec{I}^{(0)},$$

until computational inclusion is found, i.e.

$$A(\mathcal{M}_n(\vec{r},t) + \vec{I}^{(k)}) \subset \mathcal{M}_n(\vec{r},t) + \vec{I}^{(k)}.$$