Rev 0.0.0.1: June 19, 2014. Create by YU Zhi

# Design of SimPla

This document is created to describe and explain the basic requirement and design idea of project SimPla. The basic requirements are described in Sec.1, preliminary design is explained in Sec. 2, and some application examples are shown in Sec.3.

SimPla is a unified and hierarchical development framework for plasma simulation. **Its long term goal is to provide complete modeling of a fusion device.** “SimPla” is abbreviation of four words, **S**imulation, **I**ntegration, **M**ulti-physics and **Pla**sma.

## Requirements and analysis

### Background

In the tokamak, from edge to core, the physical processes of plasma have different temporal-spatial scales and are described by different physical models. To achieve the device scale simulation, these physical models should be integrated into one simulation system. A reasonable solution is to reuse and couple existing software, i.e. integrated modeling projects IMI, IMFIT and TRANSP. However, different codes have different data structures and different interfaces, which make it a big challenge to efficiently integrate them together. Therefore, we consider another more aggressive solution, implementing and coupling different physical models and numerical algorithms on a unified framework with sharable data structures and software architecture. This is maybe more challenging, but can solve the problem by the roots.

There are several important advantages to implement a unified software framework for the tokamak simulation system.

* Different physical models could be tightly and efficiently coupled together. Data are shared in memory, and inter-process communications are minimized.
* Decoupling and reusing physics independent functions, the implementation of new physical theory and model would be much easier.
* Decoupling and reusing physics independent functions, the performance could be optimized by non-physicists, without any affection on the physical validity. Physicist can easily take the benefit from the rapid growth of HPC.
* All physical models and numerical algorithms applied into the simulation system could be comprehensively reviewed.

To completely recover the physical process in the tokamak device, we need create a simulation system consisting of several different physical models. A unified development framework is necessary to achieve this goal.

### Requirements

The final objective of SimPla is to provide a complete and efficient framework for fusion plasma simulation. A list of strategic requirements has been established. They are explained in this section, and listed as:

1. **Requirements are expressed like this.**

### Physics model and numerical algorithm

In the tokamak, from edge to core, the physical processes of plasma have different temporal-spatial scales and are described by different **physical model**s.

The physical properties of simulation system are described by physical quantities. Physical quantity could be a spatial field, i.e. electric field, magnetic field, density, current, etc. or a phase space distribution function. The physical theory about the relation between different physical quantities and the temporal evolution of physical quantity are referred to as **physical model** or physical laws. Physical models are usually expressed by group partial differential equations (PDE), i.e. Maxwell equations, MHD equations or Vlasov Equations etc. The number of equations in the group should be same as the number of unknown physical quantities in the domain. One physical quantity may follow different physical models in different spatial domains. At the boundary of adjacent domains, physical models are coupled to each other through their common physical quantities.

The physical equations may be expressed on different coordinates systems, i.e. Cartesian coordinates, cylindrical coordinates, toroidal coordinates and magnetic flus coordinates etc.

To numerically solve physical equations, the physical quantities are approximately represented by values on discrete space points (mesh), and equations of continuous quantities are approximately converted into algebra equations of discrete values. The method to construct this discrete approximation is referred to as numerical algorithms, i.e. finite difference method (FDM), finite element method (FEM), finite volume method (FVM), Particle-in-cell method (PIC) etc. One physical equation may be solved by different numerical algorithm, and one numerical algorithm may be applied to different equation.

1. **Supports different physical model, i.e.** Maxwell equations, MHD equations or Vlasov Equations etc.
2. **Supports different coordinates system, i.e.** Cartesian coordinates, cylindrical coordinates, toroidal coordinates and magnetic flus coordinates etc.
3. **Supports different numerical algorithm,** i.e. FDM, FVM, FEM, DG-FEM, PIC, Delta-f etc.
4. **Automatize or simplify the conversion from physical equation to numerical algorithm;**

### Flexible and verifiability

1. **Physical model and numerical algorithm can be verified independently.**
2. **Be flexible to efficiently solve different scale problems, from one-dimensional slab model to three-dimensional global toroidal model.**

### Module and Integration

1. **Different physical model and numerical algorithms shall be coupled in memory;**
2. **Each module can be independent validated;**
3. **Support flexible task flow control;**
4. **Support global (device-scale) multi-module integration;**

### High performance computing

1. **Supports large scale high performance computing;**
2. **Has good portability on different hardware architectures, supports main-stream high performance hardware architectures, i.e. multi-core CPU, many-core GPU;**

### Input/output

1. **Provide interface for external pre-process software, i.e. modeling and mesh generate tools;**
2. **Provide interface for external post-process software, i.e. visualization tools**
3. **Flexible configuration;**
4. **Comprehensive and expressive logging information;**

### Preliminary design

***SimPla*** is a unified and hierarchical development framework for plasma simulation. **Its long term goal is to provide complete modeling of a fusion device.** “SimPla” is abbreviation of four words, **S**imulation, **I**ntegration, **M**ulti-physics and **Pla**sma.

The principle design ideas are explained in this section, and listed as:

1. **Ideas are expressed like this.**

### Design ideas

1. **SimPla is a *unified* framework.**

“***Unified***” means all different physical models are implemented by this framework and *tight coupled* to each other under this framework. “*Tight coupled*” means physical models share data and data structure as much as possible, which is to reduce the overhead of data/data structure conversion and communication between models. SimPla is designed to couple and reuse physical models and numerical algorithms, but not to couple legacy codes. All physical models and some numerical algorithm need be recoded using this framework, before they are integrated into the simulation system. This decision bases on two reasons,

* 1. Data conversion and communication between legacy codes are very complicate and will cause too much time lagged in the high performance simulation;
  2. The physical models and numerical algorithms used in the simulation system shall be comprehensive reviewed and verified;

This decision is reasonable, but the cost for recoding all physical models will be very expensive. We need reduce the development cost for the implement of physical model.

1. **SimPla is a *hierarchical* framework.**

“**Hierarchical**” means physical model, numerical algorithm and computing implement shall be decoupled.

层次化是指：把物理、数值算法、计算实现相分离。

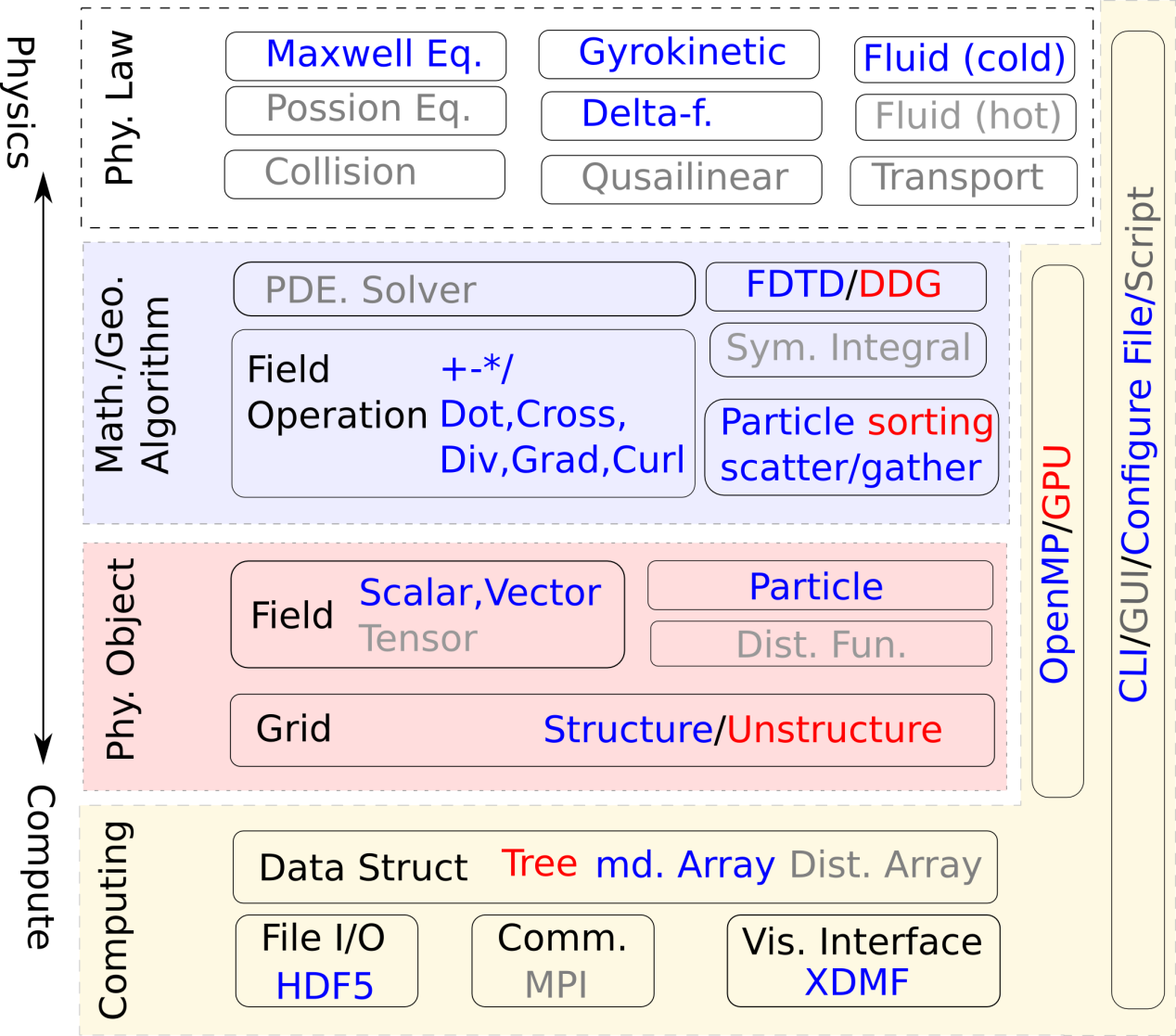
其中，物理是指上述不同的物理模型，具体表现为一些物理对象（电场、磁场、等体电流、粒子分布函数等）上不同几何位形（平板、柱位形、环位形、磁面坐标等）下的不同物理方程（MHD、Vlasov等）。数值算法包括物理对象的离散化表示和数值求解物理方程的算法。求解同一个物理方程可以有不同的数值算法（FD、FVM、FEM、DG、PIC、Delta-F等），同一个数值算法可以用来求解不同的物理方程。将物理问题和数值算法问题相分离，不仅可以实现数值算法、离散化表示的复用，以减少数据量、代码量，降低数据转换和通信开销，还便于在解决同一物理问题时在不同数值算法之间进行替换，以比较不同数值算法的结果。此外，在增加新的物理方程时，可以通过数值算法、离散化表示的复用来降低开发难度、减少工程量。

计算实现包含三个方面：（1）为实现某一具体的离散化表示、数值算法而进行的数据结构设计、程序流程设计、并行化设计（MPI）等；（2）通用的、平台无关的多线程计算模型（CPU、GPU、MIC等）；（3）通用支撑功能设计，例如数据输出/输出、日志、配置文件等等。将数值算法和计算实现相分离，不仅有利于实现代码和数据结构复用、提高平台的可扩展性，特别地，还可以在不影响到物理模型/数值算法/计算结果的情况下对性能进行优化，例如在不同的硬件架构（CPU、GPU、MIC等）上进行移植。

1. **Discretization of partial differential equations using a human-readable syntax;**

**Usually, MHD theory is expressed by a group PDEs. Using C++ expression template technology, we can directly write these PDEs into the simulation code, and automatize the discretization process when the code is compiled. That means the code could be written in a physicist-friend style, and** the development cost will be sharply decreased. The C++ expression template technology is realizable, which has a lot successful applications, i.e. OpenFOAM.

### Design of Framework architecture



### 物理层

物理层是

### 数值算法层

数值算法层？？？

### 计算层

计算层包括三个部分：（1）为实现某一具体的离散化表示、数值算法而进行的数据结构设计、程序流程设计、并行化设计（MPI）等；（2）通用的、平台无关的多线程计算模型（CPU、GPU、MIC等）；（3）通用支撑功能设计，例如数据输出/输出、日志、配置文件等等。

### Design of Input

### Design of Output

### 关键数据结构设计

### 运行设计

### Program Language

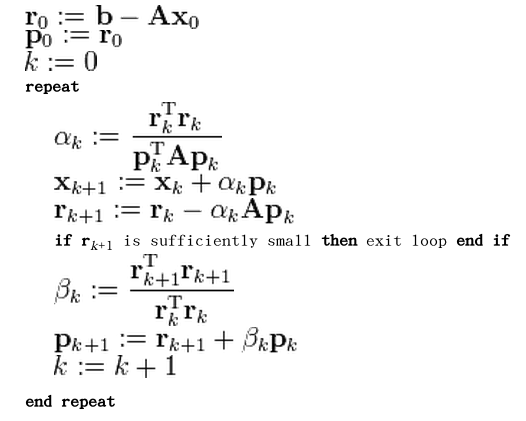
• Decomposing the computing domain into a series of “Context”, the physics in a single “Context” is simple and uniform.

– All data and functions are contained and managed by the “Context”.

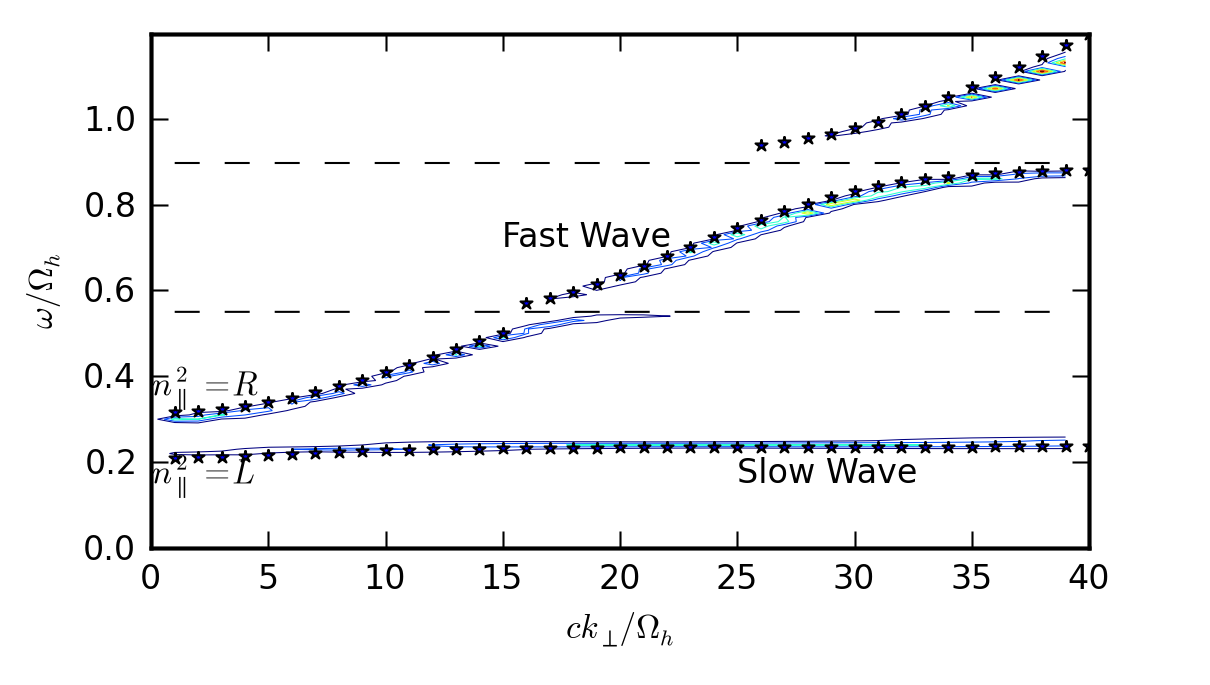
• Separate the physical equations from numerical discretization and computational implementation;

– Using operator overload and expression template encapsulate the finite difference scheme

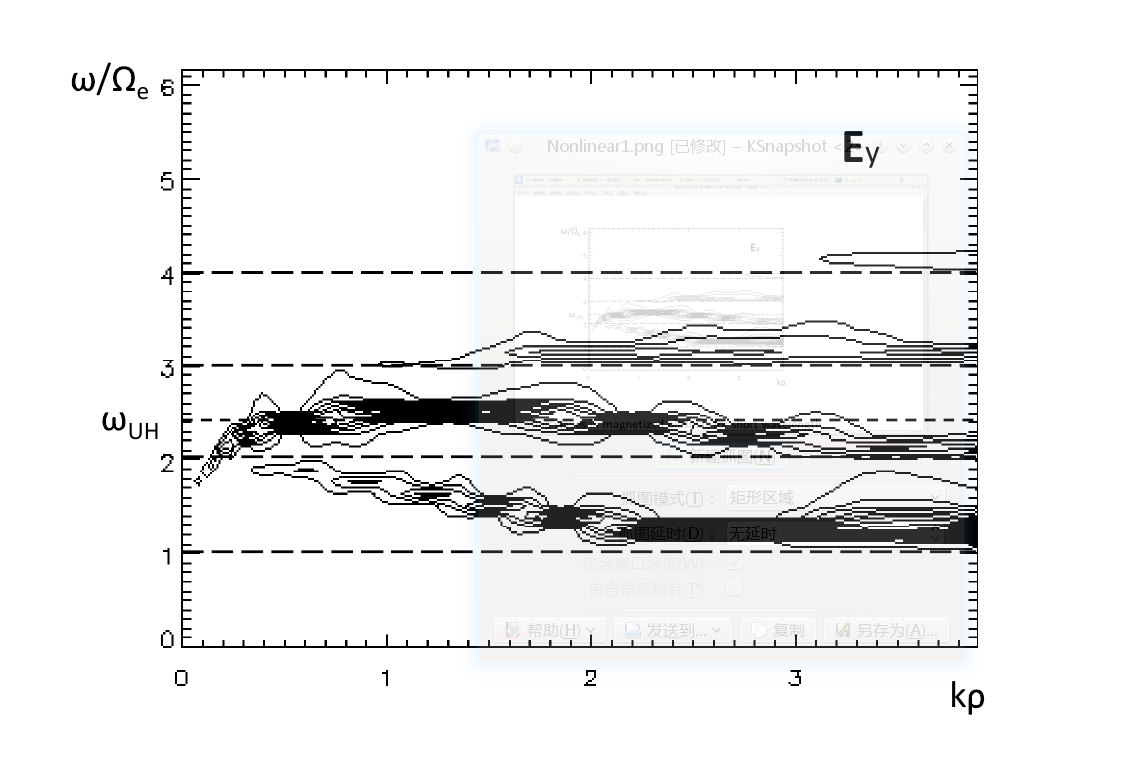
## Application



**ICRF，离子采用 G-Gauge模型，电子冷流体模型**



* 离子成分　H 25% ,D 25% and He3 25%
* 参数 ,
* **EBW ,Full-F GyroGauge模型**



* Jian Liu, Zhi Yu and Hong Qin ，Commun. Comput. Phys., 15 (2014), pp. 1167-1183. Published Online: January 21, 2014.