格子QCDシミュレーションの共通コード開発

[Development of a common code system for lattice QCD simulations]

野秋淳一 [Jun Noaki] (KEK)

for

新学術領域研究・HPCI戦略プログラム分野5 共通コードプロジェクト

JICFuS

Current Members (as of Nov. 2011)

S. Aoki, T. Aoyama, G. Cossu, T. Doi,
S. Hashimoto, N. Ishii, K-I. Ishikawa,
K. Kanaya, T. Kaneko, Y. Kuramashi,
H. Matsufuru, S. Motoki, Y. Namekawa,
H. Nemura, J. Noaki, K. Ogawa,
H. Saito, S. Sasaki, Y. Taniguchi,
S. Ueda, N. Ukita, N. Yamada, T. Yoshié

Programmers, reviewers and users are wanted.
Any interested people are welcome anytime!

Supported by

- Grant-in-Aid for Scientific Research on Innovative Areas

  "Research on the Emergence of Hierarchical Structure of Matter by Bridging Particle, Nuclear and Astrophysics in Computational Science"

  - The A04 squad

    "Interdisciplinary algorithms and computer simulations"

    http://bridge.kek.jp/A04/ see H. Matsufuru's talk

- HPCI Strategic Program Field 5

  "The origin of matter and the universe"

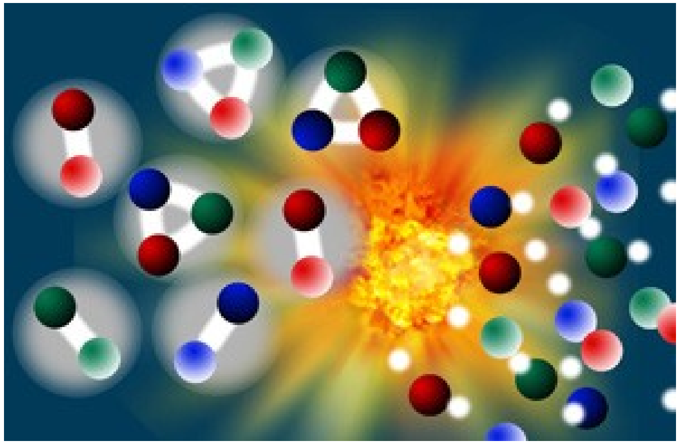
  http://www.jicfus.jp

  計算基礎科学連携拠点
  JICFuS Joint Institute for Computational Fundamental Science

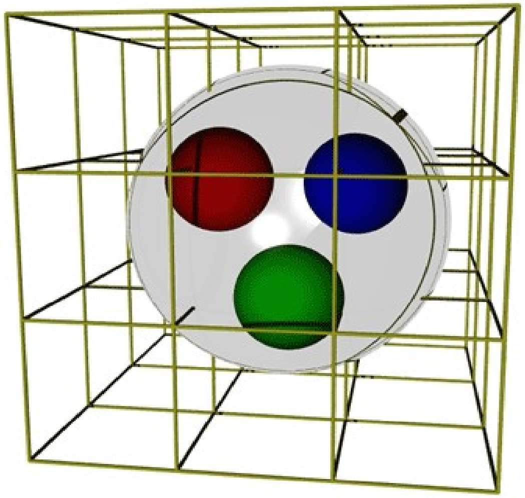
  - So far, 36 meetings held every 2-3 weeks
  - Help available from experts in computer science

Numerical simulation in lattice QCD

- QCD
  - Theory of quarks, gluons and their dynamics
  - Total analytic treatment is impossible
    - Theoretical calculation is challenging



- Lattice QCD
  - QCD on the lattice
    - ab initio simulation of QCD
  - Most promising way to study the strong dynamics
  - Tools: computer + code

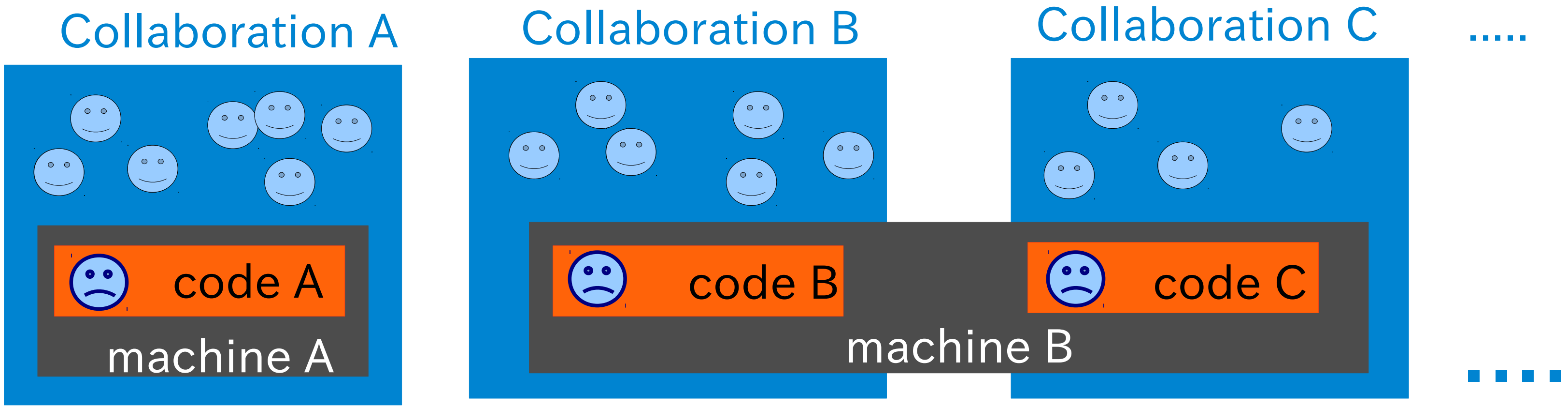


Why common code?

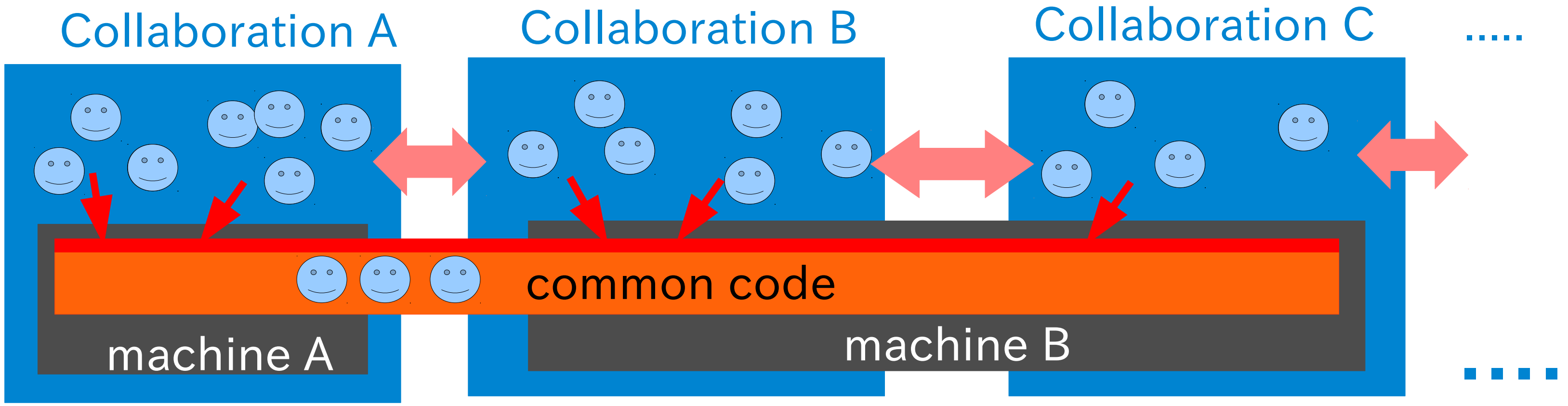
- Fluid research environment
  - Collaboration members always come and go.
    - who maintains the code?
    - communication problem might occur.
  - Machine architecture is changeable.
    - have to rewrite the code for updated machines?
- Demands for “standard”
  - Users can concentrate on their physics project.
  - Generated data can be shared by different groups.
  - Provides a common language on the calculations.
- Why not using existing codes?
  - Ex. Chroma and CPS++ are widely used in the community.
  - We want “Complete control from the basic to the advanced”.
    - Make it from the beginning.
    - Accumulating experiences of the development is important.

Why common code? (contd.)

- Currently…

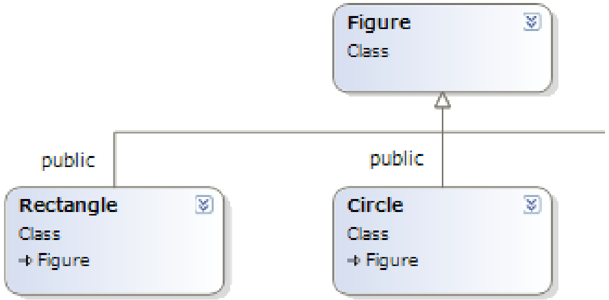


- We aim at …



Profile

- Our aim: well-organized portable code with a good performance
  - allows beginners to carry out “professional simulations”
- C++ language
  - Design by the object oriented programming
  - Stick to the standard libraries for portability
- Documentation
  - Doxygen is helpful
  - Detailed manual is to be published in English/Japanese
- Covering all basic calculations in LQCD
  - Gauge generation+ measurements
  - Compatible with various kinds of lattice fermions
  - ILDG data format is employed
  - Maximum flexibility of the simulation parameters



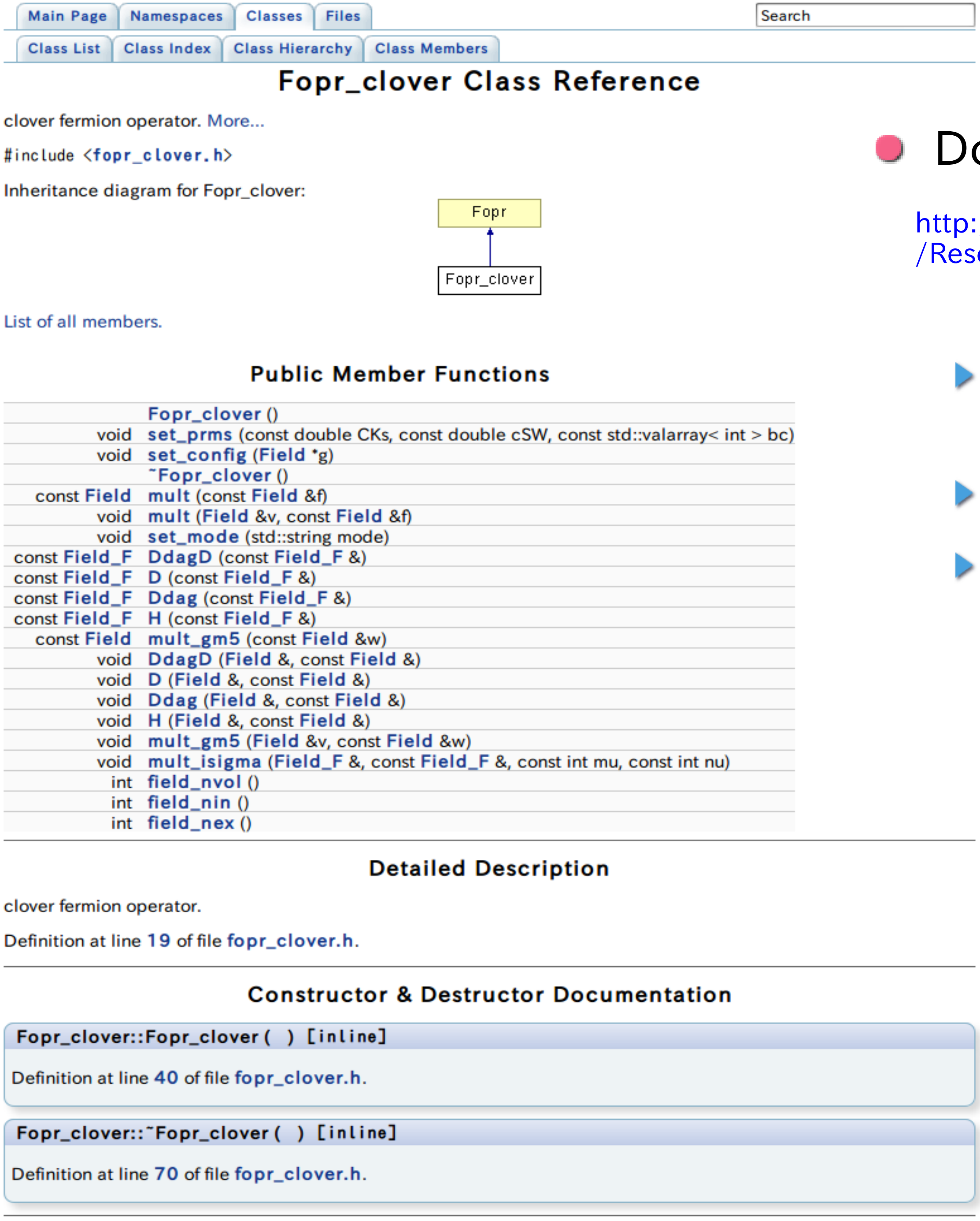
Two major applications

- Grand challenge of the Japanese LQCD community
  - to use K-Computer from next year
  - clover fermions + smeared links
    - to accomplish extremely large & fine lattice with physical quark masses
  - all core calculations are ready to run

- JLQCD (+TWQCD) Collaboration(s)
  - KEK supercomputer system A+B
  - chiral fermions + smeared links
    - for the deeper study of chiral symmetry
  - under construction with test running

KEK system A (Hitachi SR16000M)

At a glance



Fopr_clover Class Reference

clover fermion operator. More...

`#include <fopr_clover.h>`

Inheritance diagram for Fopr_clover:

List of all members.

Public Member Functions

	Fopr_clover ()
void	set_prms (const double CKs, const double cSW, const std::valarray< int > bc)
void	set_config (Field *g)
	~Fopr_clover ()
const Field	mult (const Field &f)
void	mult (Field &v, const Field &f)
void	set_mode (std::string mode)
const Field_F	DdagD (const Field_F &)
const Field_F	D (const Field_F &)
const Field_F	Ddag (const Field_F &)
const Field_F	H (const Field_F &)
const Field	mult_gm5 (const Field &w)
void	DdagD (Field &, const Field &)
void	D (Field &, const Field &)
void	Ddag (Field &, const Field &)
void	H (Field &, const Field &)
void	mult_gm5 (Field &v, const Field &w)
void	mult_isigma (Field_F &, const Field_F &, const int mu, const int nu)
int	field_nvol ()
int	field_nin ()
int	field_nex ()

Detailed Description

clover fermion operator.

Definition at line 19 of file fopr_clover.h.

Constructor & Destructor Documentation

Fopr_clover::Fopr_clover ( ) [inline]

Definition at line 40 of file fopr_clover.h.

Fopr_clover::~Fopr_clover ( ) [inline]

Definition at line 70 of file fopr_clover.h.

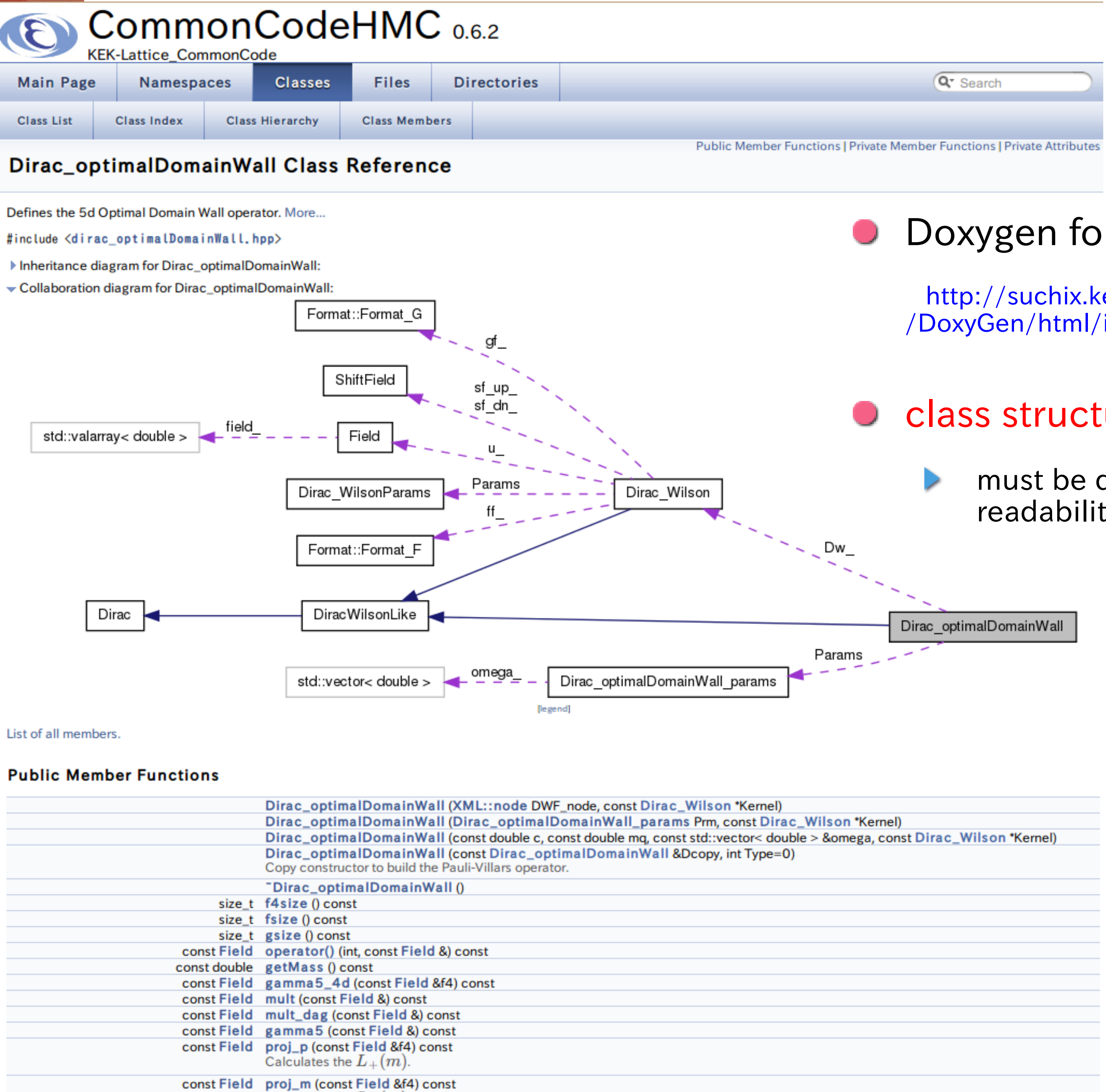
- Doxygen page for the core

  http://suchix.kek.jp/hideo_matsufuru//Research/Projects/Lattice_code/

  H.Masufuru, (Access is restricted currently.)

  - Major helper software which generates html files based on the source code.
  - provides graphical information of the code.
  - One of the advantages of the OO language.

At a glance (contd.)



CommonCodeHMC 0.6.2

KEK-Lattice_CommonCode

Dirac_optimalDomainWall Class Reference

Defines the 5d Optimal Domain Wall operator. More...

```
#include <dirac_optimalDomainWall.hpp>
```

Inheritance diagram for Dirac_optimalDomainWall:

Collaboration diagram for Dirac_optimalDomainWall:

List of all members.

Public Member Functions

	Dirac_optimalDomainWall (XML::node DWF_node, const Dirac_Wilson *Kernel)
	Dirac_optimalDomainWall (Dirac_optimalDomainWall_params Prm, const Dirac_Wilson *Kernel)
	Dirac_optimalDomainWall (const double c, const double mq, const std::vector< double > &omega, const Dirac_Wilson *Kernel)
	Dirac_optimalDomainWall (const Dirac_optimalDomainWall &Dcopy, int Type=0) Copy constructor to build the Pauli-Villars operator.
	~Dirac_optimalDomainWall ()
size_t	f4size () const
size_t	fsize () const
size_t	gsize () const
const Field	operator() (int, const Field &) const
const double	getMass () const
const Field	gamma5_4d (const Field &f4) const
const Field	mult (const Field &) const
const Field	mult_dag (const Field &) const
const Field	gamma5 (const Field &) const
const Field	proj_p (const Field &f4) const Calculates the $L_+(m)$.
const Field	proj_m (const Field &f4) const

- Doxygen for the JLQCD porsion

  http://suchix.kek.jp/guido_cossu/documents/DoxyGen/html/index.html

  G. Cossu

- class structure = expression of the theory
  - must be designed carefully to guarantee readability and expandability in the future

Road map

- End of Dec. 2011: Beta version
  - core functions for clover fermion + gauge smearing
  - reviewing and refactoring start
    - collect as many comments & requests from users as possible
- Jan. 2012 : RC1
  - merge of the JLQCD implementations with the beta version.
- Feb. 2012: RC2
  - user interfaces, accessories and documents developed
- End of Mar. 2012: Ver.1.0
  - release to public (GPL)
  - invite new users （a summer school is planned.)

Summary

This talk is …

- Not on physics, but on new research environments
  - to skip unnecessary efforts of coding
  - to remove barriers of communication between researches
  - to share experiences and data
- Even on a fusion of different fields
  - physics: numerical simulations
  - computer science: software design
  - one of the goal of this program

The project is still in the first stage.

We strongly expect feedbacks from the community.

Thank you for your attention.