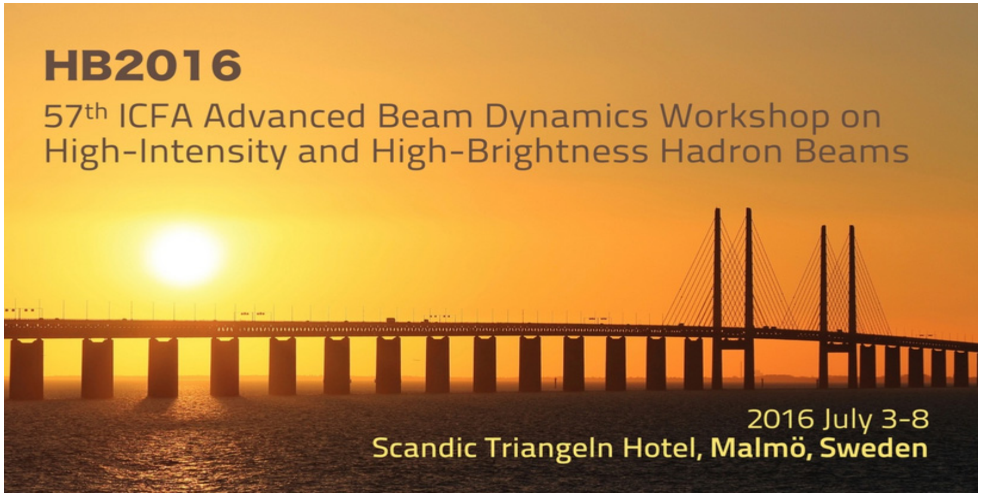
APPLICATION OF THE OPTIMIZATION ALGORITHM IN THE COLLIMATION SYSTEM FOR CSNS/RCS

Hongfei Ji, Sheng Wang, Mingyang Huang, Shouyan Xu
CSNS/IHEP, CAS, Dongguan, 523803, P.R. China

Yi Jiao, Na Wang. IHEP, CAS, Beijing 100049, P.R. China



jihf@ihep.ac.cn.

Outline

1. Introduction
2. Physics analysis and modeling
3. Optimization
4. Summary

Outline

1. Introduction
    1.1 Optimization and applications
    1.2 The Robust Conjugate Direction Search method
    1.3 Motivation of our work
2. Physics analysis and modeling
3. Optimization
4. Summary

Introduction-*optimization and applications*

- A particle accelerator is typically a complex system that consists of many components. There are many variables to be tuned to achieve good machine performance. Therefore, optimization usually needs to be performed in a multi-dimensional variable space.

- In the design phase, optimization and applications:
  - Programs:MAD, SAD, AT, ORBIT, …
  - Algorithms: Genetic algorithm (GA), Particle swarm optimization (PSO), …

Single-objective optimization methods	Multi-objective optimization methods	Parallel optimization techniques
GA (1975); PSO (1995)	MOGAs; MOPSO	PGA; PPSO

- Have been developed to model the system and to further optimize the expected machine performance.

jihf@ihep.ac.cn

Introduction-*optimization and applications*

- It is noted that, along with the unceasing development of computer technology and optimization algorithms, online optimization of accelerator performance has become both imperative and feasible.

- Several algorithms have been proposed for different purposes in many laboratories and have been explored for their application to online optimization problems;

  - Response matrix method APS, …, BEPCII 【*Y. Chung, G. Decker, K. Evans, 10.1109/PAC.1993.309289*】【*D. H. Ji, Q. Qin, et al., Proceedings of IPAC 2010, p.1644*】
  - The simplex method KEKB 【*Y. Funakoshi, et al., EPAC2008, Genoa, Italy, 2008, p.1893*】
  - Random Walk SLS 【*M. Aiba, et al., IPAC2012, New Orleans, Louisiana, USA, TUPPC033*】
  - Genetic Algorithm SLAC 【*K. Tian, J. Safranek, Y. Yan. Phys. Rev. ST Accel. Beams, 2014, 17: 020703*】
  - Robust Conjugate Direction Search method SLAC, BEPCII
    【*X. Huang, J. Corbett, J. Safranek, J. Wu, NIM A 726 (2013) 77-83*】
    【*X. B. Huang, J. Safranek, Phys. Rev. ST Accel. Beams, 18: 084001 (2015)*】
    【*H. F. Ji, Y. Jiao, S. Wang, et al., Chinese Physics C Vol. 39, No. 12 (2015) 127006*】
  - ……

Introduction-*the RCDS method*

- The Robust Conjugate Direction Search method，RCDS

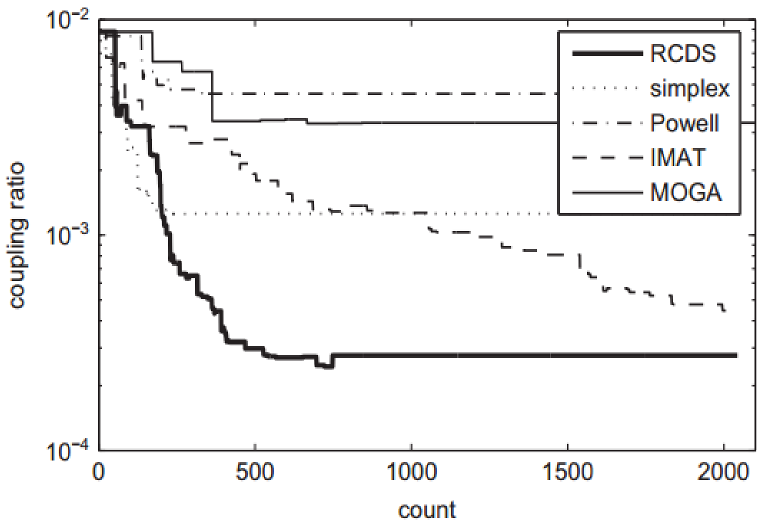
  - The RCDS method combines the conjugate direction set approach of Powell's method with a robust line optimizer that is robust against random noise and outliers. Simulation and experimental studies have been carried out to demonstrate the strength of the algorithm.

✓ Powell ＋ robust line optimizer

✓ Search on a conjugate direction set

✓ have high tolerance to noise

✓ Find a local minimum



【X. Huang, J. Corbett, J. Safranek, J. Wu, NIM A 726 (2013) 77-83】

✓ Be effective in optimizing a single-objective function of several variables

Introduction-*Motivation of our work*

- The RCDS method is much more robust against noise than traditional algorithms. Based on the success of the RCDS method with a light source and a collider, it is interesting to find possible applications of this method in optimizing the performance of the collimation system for CSNS/RCS.
- To improve the efficiency of the optimization in the application, the parallel technique of the RCDS method is explored.
- Try to implement the RCDS method to optimize the collimation system of CSNS/RCS, and reduce the uncontrolled beam loss.
- To present a way to find an optimal parameter combination of the secondary collimators for a machine model in preparation for CSNS/RCS commissioning.

Outline

1. Introduction

2. Physics analysis and modeling

    2.1 Physics variables

    2.2 Objective function

    2.3 Implementation of the simulation code

    2.4 Parallel computing of ORBIT instances

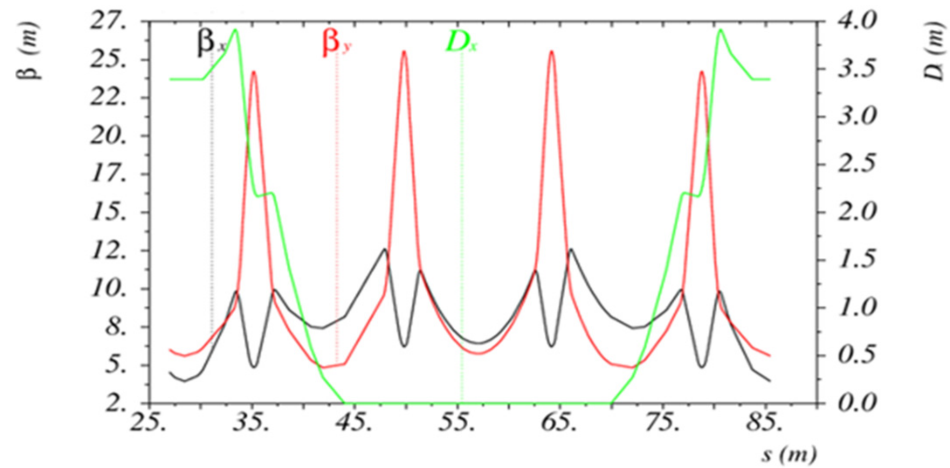
3. Optimization

4. Summary

The transverse collimation system for CSNS/RCS

- To meet the requirements for hands-on and safe maintenance, a two-stage collimation system was designed to localize the beam loss in the collimation section in the RCS.
- The transverse collimation system consists of one primary collimator and four secondary collimators.

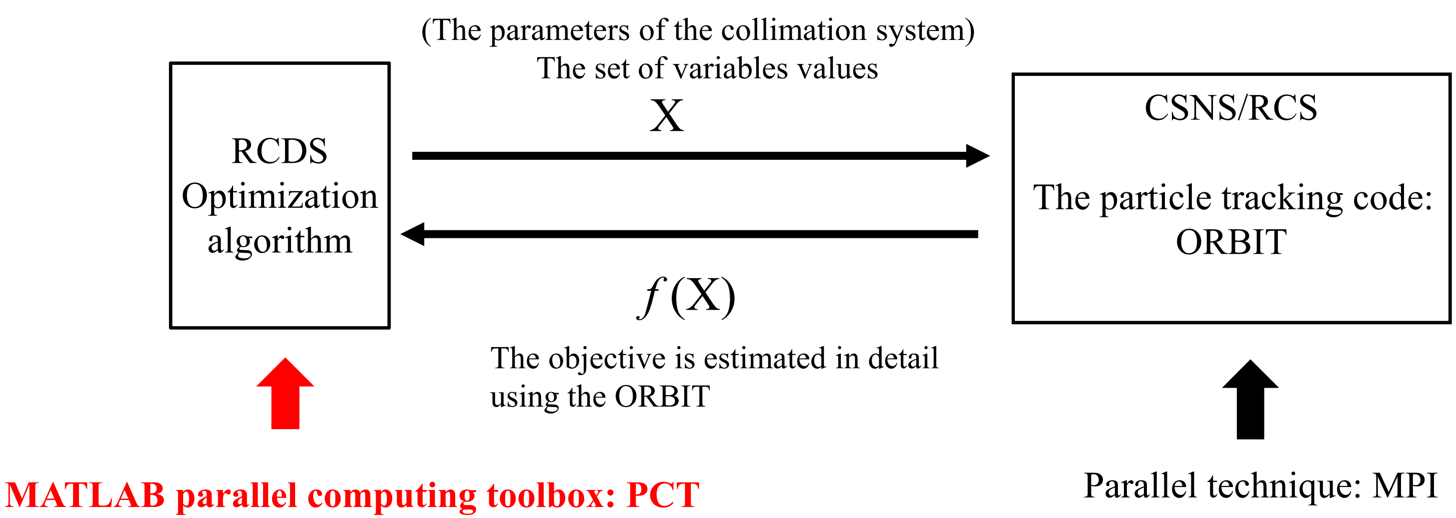
Parameters	Symbol, unit	Value
Circumference	C, m	227.92
Injection energy	E_{inj}, MeV	80
Extraction energy	E_{ext}, GeV	1.6
Betatron tune (H/V)	υ_x/υ_y	4.86/4.78
Accumulated particles per bunch	N_p, $\times 10^{12}$	7.8
Harmonic number	h	2
Repetition frequency	f_0, Hz	25
Accumulated and accelerated time per cycle	t, ms	~20
Transverse acceptance	ε, πmm·mrad	540



- ➢ The primary collimator has minimum acceptance in the ring, and thus is used to scatter the protons with large deviations from the beam center.
- ➢ The scattered protons are then absorbed by four secondary collimators located downstream of the primary collimator.

The layout of the application

- In the RCS, the aperture of each secondary collimator can be varied by adjusting the positions of four movable blocks. To achieve satisfactory machine performance, it is necessary to optimize the parameters of collimators, so as to absorb most of the undesirable protons by the collimation system and minimize the beam loss in other regions of the ring.
  - In the study, we introduce an algorithm, the RCDS method, in the optimization.
  - The collimation process in the presence of space charge is simulated with the Objective Ring Beam Injection and Tracking (ORBIT) code.



Physics variables

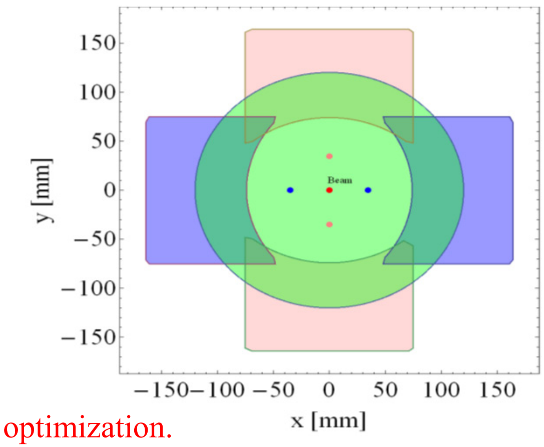
- The beam power is 100 kW; the acceptance of the primary collimator is fixed to 350 πmm·mrad all the time, and the secondary collimators are tuned to optimize the performance of the collimation system.
- Each of the secondary collimators is composed of four movable copper blocks. Each block has a circular surface based on the equation:

$$\frac{\left(x\cos(\frac{\pi\theta}{180^\circ})+y\sin(\frac{\pi\theta}{180^\circ})\pm c\right)^2}{a^2}+\frac{\left(-x\sin(\frac{\pi\theta}{180^\circ})+y\cos(\frac{\pi\theta}{180^\circ})\right)^2}{a^2}=1,$$

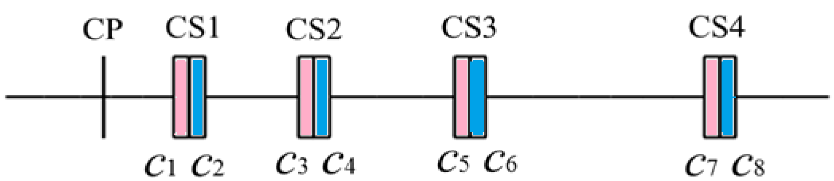
$$\theta=\begin{cases}0^\circ, & \text{The horizontal blocks}\\ 90^\circ, & \text{The verticle blocks}\end{cases}$$

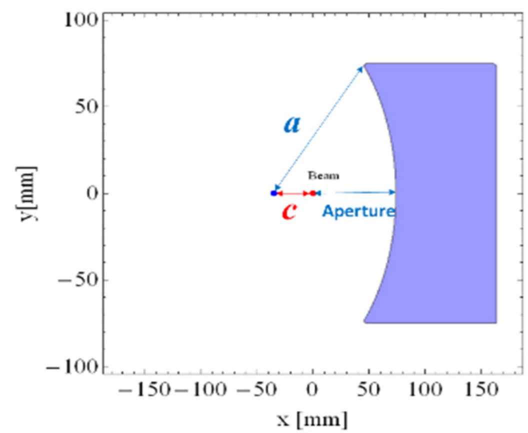
a=108.8 mm
Aperture -> (40 mm, 74 mm)

c -> (34.8 mm, 68.8 mm)
The parameter c is selected to be the variable for the optimization.



- Considering the symmetry of beam distribution in simulations, the parameters of the blocks on the same direction of each secondary collimator are the same. Then there are eight variables to be tuned for four secondary collimators





Objective function

- A single objective function is constructed to measure the performance of the collimation system:

$$f = -\eta_{system} \cdot R_{\varepsilon_{xm}} \cdot R_{\varepsilon_{ym}} \cdot R_{\varepsilon_{add}} \cdot R_{\varepsilon_{flag}}$$

- A minus sign is added to form a minimization problem.
- The cleaning efficiency of the system:

$$\eta_{\text{system}} = \frac{N_{\text{incol}} + N_{\text{indrift}}}{N_{\text{loss}}},$$

- The weight factors:

$$R_{\varepsilon_{xm}} = \begin{cases} 1, & if\ \varepsilon_{xm} \le Ref\_collimator; \\ Ref\_collimator / \varepsilon_{xm}, & if\ \varepsilon_{xm} > Ref\_collimator. \end{cases}$$

ε_{xm} is the maximum 99% horizontal emittance of the beam
ε_{ym} is the maximum 99% vertical emittance of the beam

$$R_{\varepsilon_{ym}} = \begin{cases} 1, & if\ \varepsilon_{ym} \le Ref\_collimator; \\ Ref\_collimator / \varepsilon_{ym}, & if\ \varepsilon_{ym} > Ref\_collimator. \end{cases}$$

$Ref\_collimator$=350 πmm·mrad

$$R_{\varepsilon_{add}} = \begin{cases} 1, & if\ \varepsilon_{add} \le Ref\_add; \\ Ref\_add / \varepsilon_{add}, & if\ \varepsilon_{add} > Ref\_add. \end{cases}$$

Ref_add =700 πmm·mrad

$$\varepsilon_{add} = \varepsilon_{xm} + \varepsilon_{ym}.$$

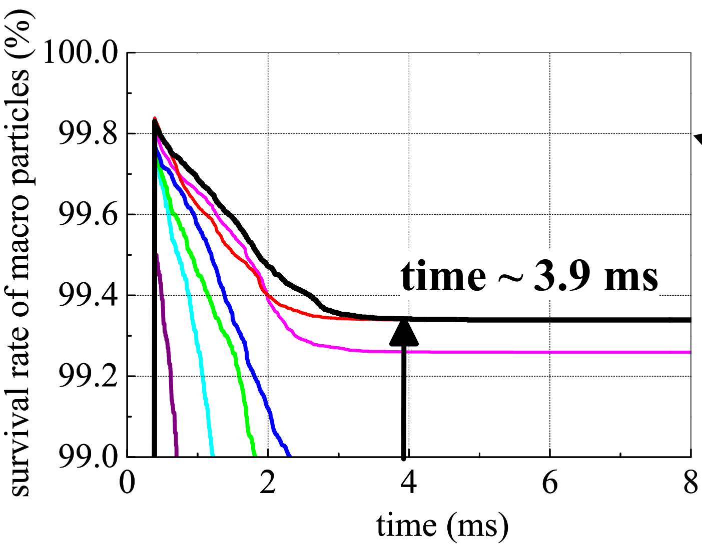
$$R_{\varepsilon_{flag}} = \begin{cases} 1, & if\ \varepsilon_{flag} \le Ref\_flag; \\ Ref\_flag / \varepsilon_{flag}, & if\ \varepsilon_{flag} > Ref\_flag. \end{cases}$$

$$\varepsilon_{flag} = \max\left(\frac{\varepsilon_{xm}}{\varepsilon_{ym}}, \frac{\varepsilon_{ym}}{\varepsilon_{xm}}\right) - 1.$$

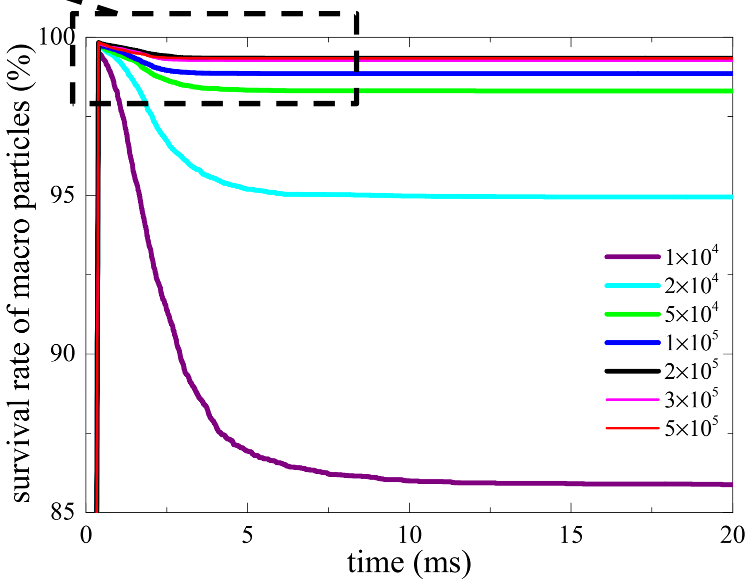
$Ref\_flag = 0.5$

An ORBIT instance

- To achieve high computing precision and fast execution speed of the program, model parameters of an ORBIT instance are studied in detail, including the number of macro particles and the number of turns for the beam being turned around the ring.



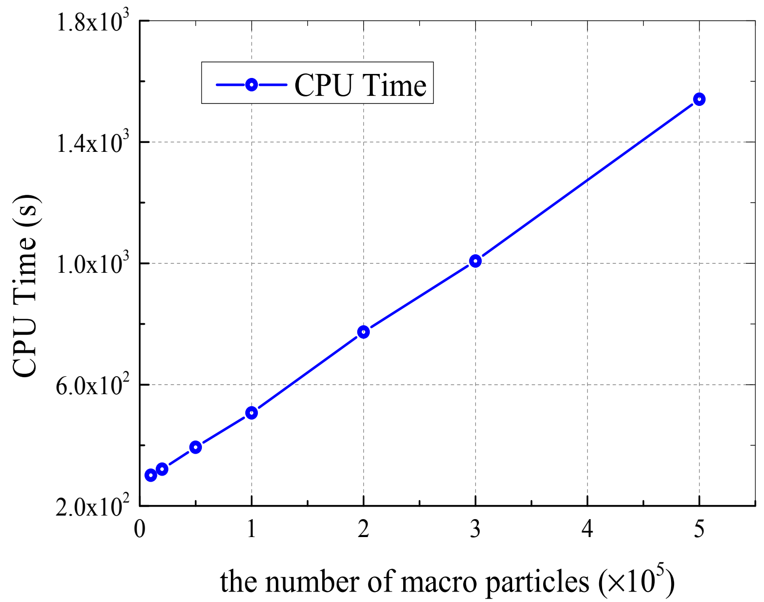
- In the RCS, the proton beam is accumulated through an anti-correlated painting scheme within 200 turns, and accelerated to 1.6 GeV in about 20 000 turns. It takes about 20 ms per cycle.

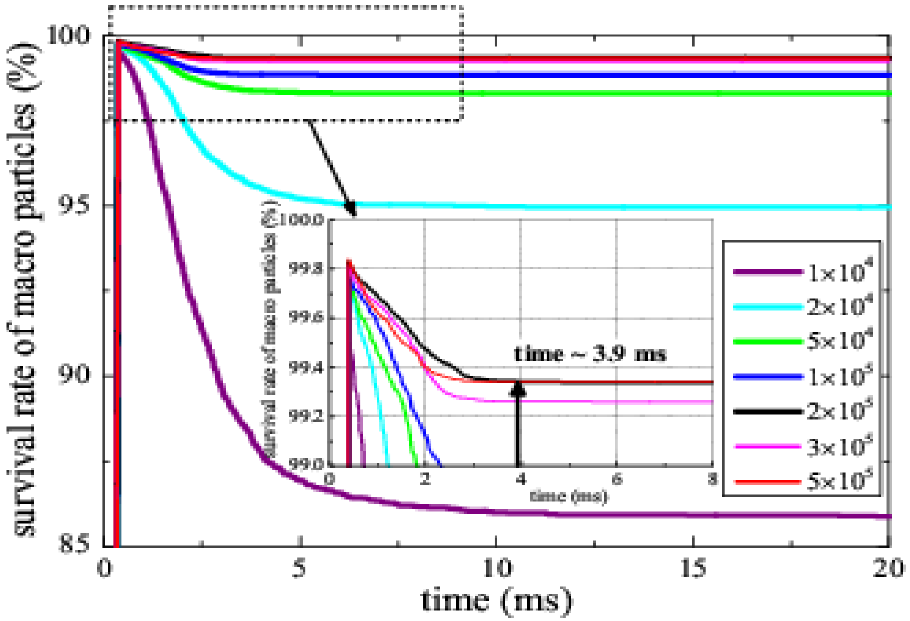


- The stability of simulation results increases with the number of macro particles.
- The simulation time is related to the number of turns for the acceleration.
- With 2×10^5 macro particles, the beam loss mainly occurs within the first 3.9 ms, which corresponds to 2200 turns for beam tracking.

An ORBIT instance

- To achieve high computing precision and fast execution speed of the program, model parameters of an ORBIT instance are studied in detail, including the number of macro particles and the number of turns for the beam being turned around the ring.
- The stability of simulation results increases with the number of macro particles.
- The number of macro particles directly affects the simulation time.



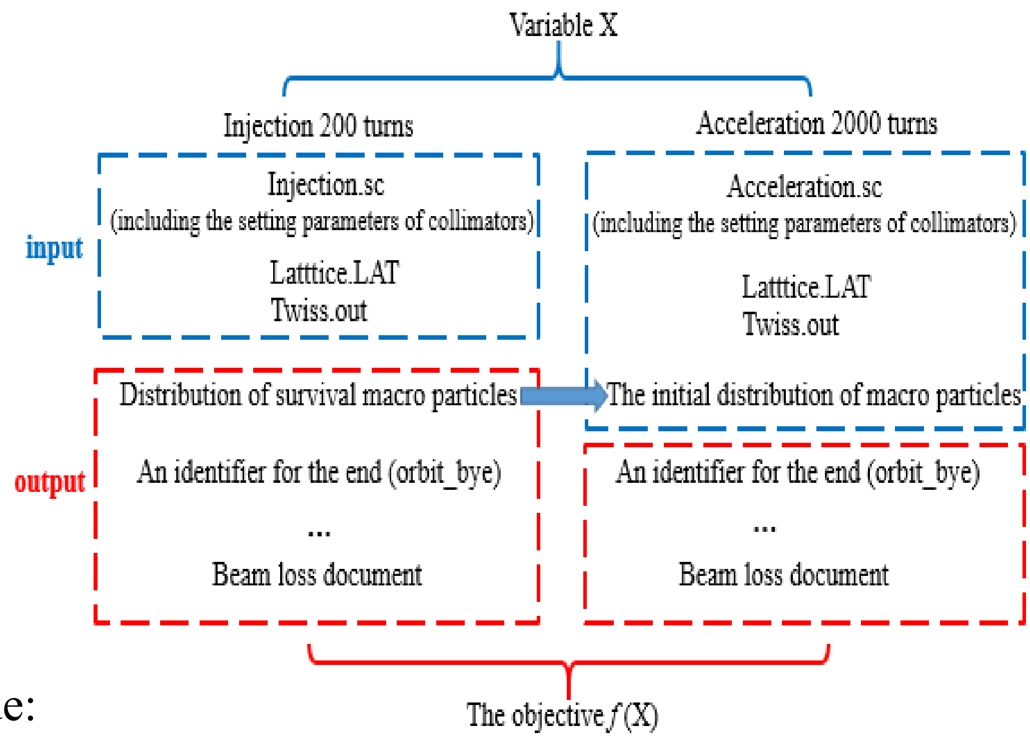


➢ Apparently, the CPU time increases approximately linearly with the number of macro particles with the same computing resources at IHEP.

Implementation of the simulation code

- An ORBIT instance:

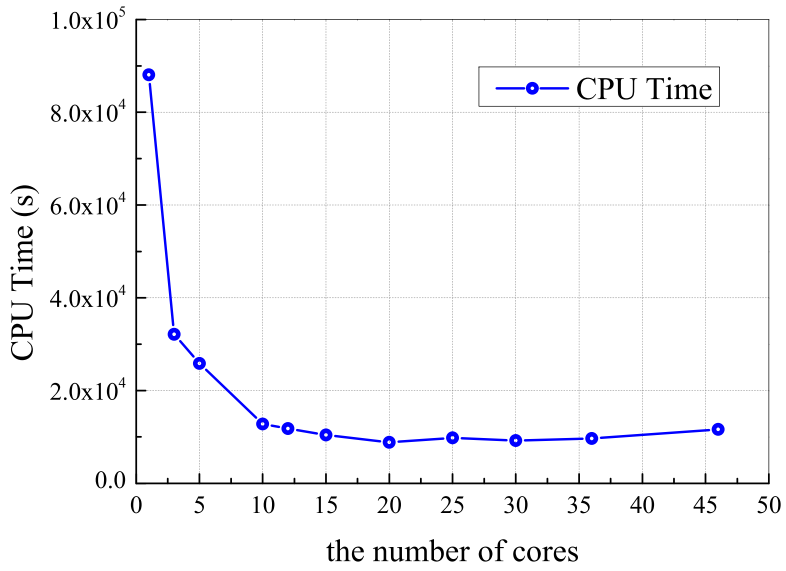
Model parameters / unit	Value
number of real protons / $\times 10^{12}$	7.8
number of macro particles / $\times 10^{5}$	2
number of injection turns	200
number of acceleration turns	2000
beam tracking, turns / time	2200 turns / 3.9 ms



- Implementation of the simulation code:
  - A complete ORBIT instance for CSNS/RCS consists of two parts, the injection and acceleration.
  - Macro particles are generated and put into the injection simulation. The output of the injection simulation is then used as input for the acceleration simulation.
  - To accommodate the searching process of the algorithm in an automatic way, a code is first written in MATLAB to create and submit an ORBIT instance automatically.

The computing resources and the limitation

- A cluster in a Gigabit network environment at IHEP
  - Each node of the cluster has 2 CPUs;
  - Each CPU has 6 cores.

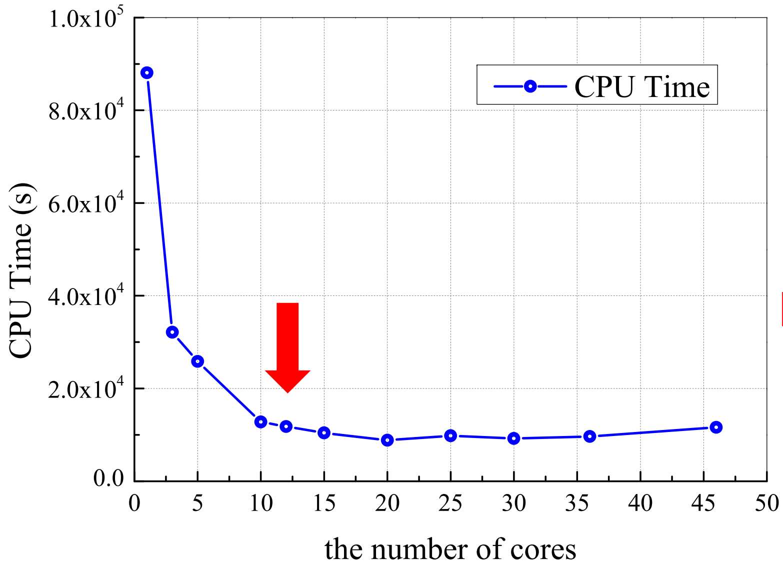


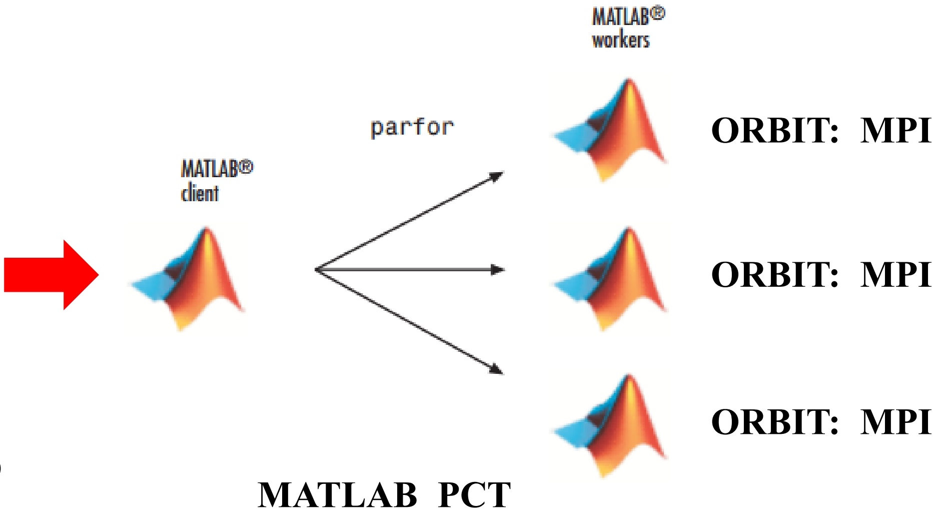
- Compared the CPU time of running an ORBIT instance with different numbers of cores, and the distributions of the cores over different nodes are listed in the Table:
  - The CPU time decreases obviously with the increase of the number of cores on the same node.
  - When the cores belong to different nodes, an increase of the number of cores would not lead to a proportionate decrease in the CPU time of running an ORBIT instance.
  - The communication loss between nodes is not ignorable.

cores	1	3	5	10	12	15	20	25	30	36	46
nodes	1	1	1	1	1	2	2	3	3	4	5

Parallel computing of ORBIT instances

- A cluster in a Gigabit network environment at IHEP
  - Each node of the cluster has 2 CPUs;
  - Each CPU has 6 cores.



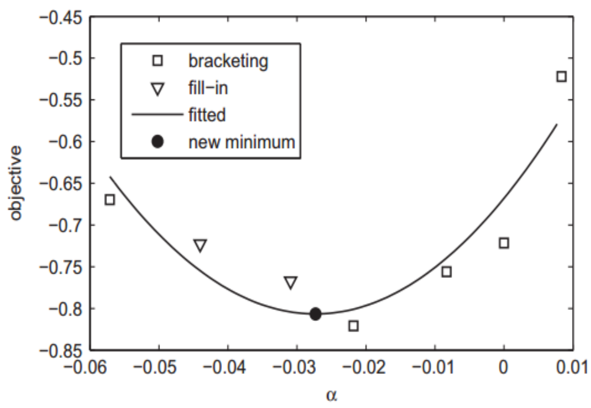


- To make better use of nodes in the cluster and speed up computation of the RCDS method:
  - 12 cores inside the same node are used to run an instance with the MPI parallel computing.
  - A function is programmed within the RCDS algorithm to run several instances **on different nodes simultaneously** with the MATLAB parallel computing toolbox.

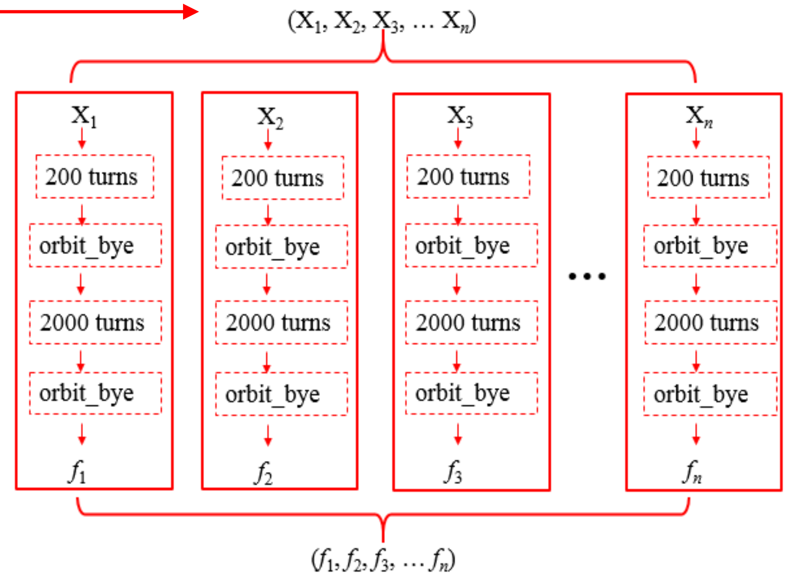
Parallel computing of ORBIT instances

- x0, step, range

Parallelorbit.m



【X. Huang, J. Corbett, J. Safranek, J. Wu, NIM A 726 (2013) 77-83】



- For a certain direction, the starting point, the step size and the range for the variables can be determined before the line optimizer is executed. So the candidate values of variables are calculated to give out a variables vector set (X_1, X_2, …, X_n), and the corresponding ORBIT instances are created.
- The function we defined in MATLAB is called to distribute these ORBIT instances over n nodes individually with PCT, and then run each ORBIT instance inside a separate node with MPI.
- Finally, the corresponding objective values across all the separate nodes are obtained to form a library.
- Set up a candidate function library, {(X_1, f_1), (X_2, f_2), …, (X_n, f_n)}, for the present direction
- In this method, a group of candidate evaluations of the objective are obtained in a period of time during which the original method can get only one evaluation.

Outline

1. Introduction
2. Physics analysis and modeling
3. Optimization
    3.1 The beam distribution
    3.2 The initial values of variables
    3.3 The noise level
    3.4 Optimization result
4. Summary

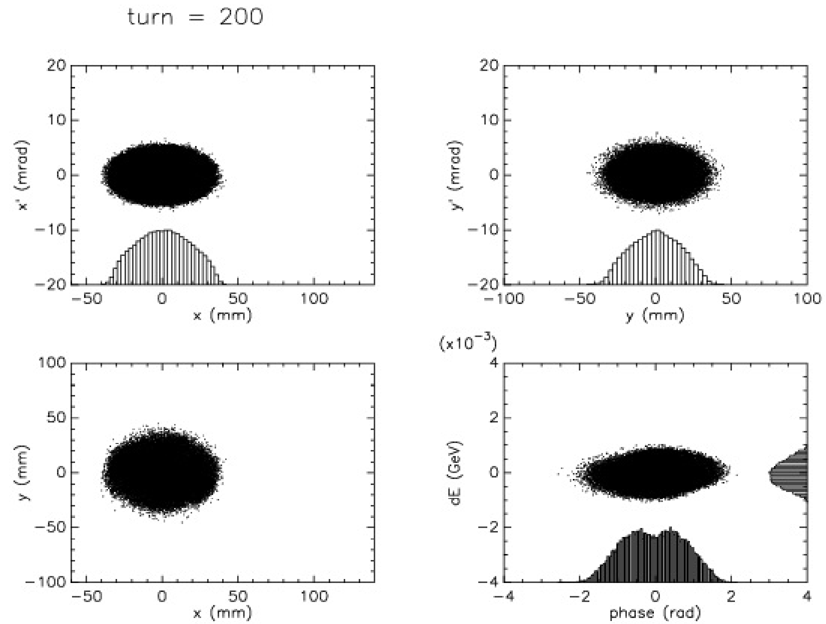
The beam distribution

- In order to confirm the optimization of the performance of the collimation system, we fix the particle distribution for the input of the acceleration process.

- A realistic distribution of macro particles is obtained with:
  - ➢ The acceptances of secondary collimators being set to 500 πmm·mrad.
  - ➢ An anti-correlated painting scheme.

- The horizontal 99% emittance of the beam distribution is 193 πmm·mrad and the vertical is 219 πmm·mrad.

parameters	values
$\varepsilon_{x,\,rms}$ (πmm·mrad)	0.639
$\varepsilon_{y,\,rms}$ (πmm·mrad)	0.619
α_x	0.003
α_y	-0.102
β_x (m)	1.631
β_y (m)	1.637
E_{mean} (MeV)	80.4
σ_E (MeV)	0.15



The configuration parameters

The initial values of variables

- The initial acceptances of secondary collimators are set to 420 πmm·mrad.
- Their ranges are tuned from 370 πmm·mrad to 500 πmm·mrad due to the acceptance of the primary collimator and the transverse acceptance of the ring.

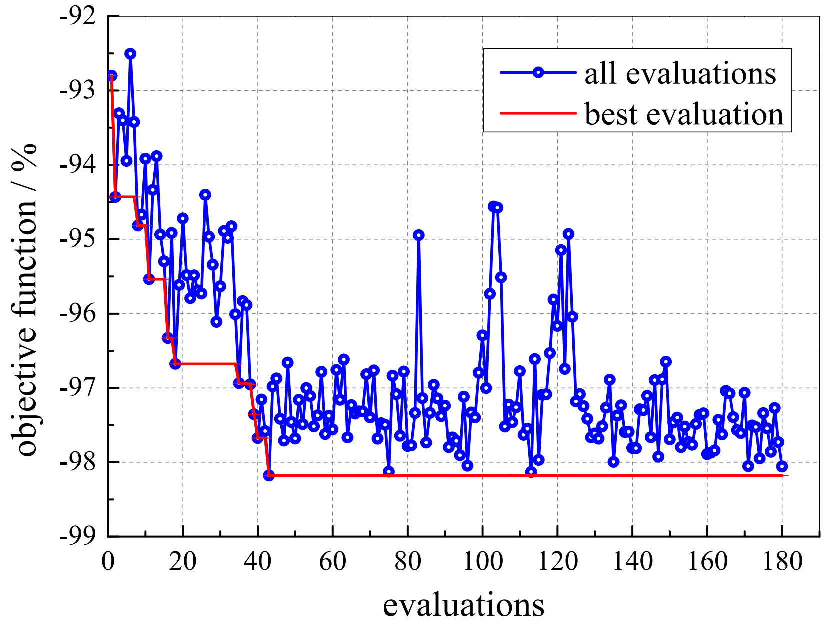
The noise level

- By running an instance with the beam being accelerated for 2000 turns repeatedly, the noise level of the cleaning efficiency is calculated to be 0.7%, and this value is used as the noise of the objective during the optimization.

$$\sigma_{\text{noise}} = 0.7\ \%$$

Optimization result

- Having configured the parameters of RCDS, the simulation is performed:



Element	variables	value / mm	Acceptance / πmm·mrad	λ_{co} / %
CS1	c_1	56.1	371	29.58
	c_2	48.3	470	
CS2	c_3	60.9	372	58.83
	c_4	55.8	419	
CS3	c_5	59.5	420	5.08
	c_6	56.6	420	
CS4	c_7	50.6	420	2.65
	c_8	48.2	420	

- The objective： -92.8% ➞ -98.2%
- The objective is optimized automatically within 180 evaluations. The simulation time is lower than the time of running one ORBIT instance 180 times.

jihf@ihep.ac.cn

Optimization result

- A comparison of the parameters reflecting the performance of the collimation system between the initial state and the optimal result:

parameters	η_{system} / %	λ_{un} / 10^{-4}	η_{co} / %	λ_{total} / %	ε_x /πmm·mrad	ε_y /πmm·mrad
initial state	92.8	4.9	91.9	0.7	193	219
optimal result	98.2	1.7	96.3	0.9	193	215

- The cleaning efficiency, η_{system}, is optimized to 98.2%.
- The uncontrolled beam loss of total beam outside the collimation section during the acceleration, $\lambda_{un,}$ is lower.
- The collimation efficiency of collimators, η_{co}, is higher.
- The total beam loss along the ring, λ_{total}, is acceptable for shielding, although it is a little higher than that of the initial state.
- The 99% horizontal (vertical) emittance of the beam shows a good quality beam.
- Considering even larger beam loss might be caused by various kinds of errors in the actual conditions, it is more important to reduce the uncontrolled beam loss.

Summary

- We have implemented the RCDS method to optimize the collimation system of CSNS/RCS.
- The uncontrolled beam loss of the total beam during the acceleration can be reduced to 1.7×10^{-4}, which is lower than that obtained by previous optimization.
- As a result, an approach is established to efficiently give an optimal parameter combination of the secondary collimators for the present machine model.

Thanks for your attention!

jihf@ihep.ac.cn