

Proposal for FY2021 Laboratory Directed Research and Development Funds

Title: Feasibility Study on Implementing Quantum Computing to Multiple Particle Beam Dynamic Simulations

Topic: Quantum Information Science

|  |  |
| --- | --- |
| **Lead Scientist or Engineer:** | He Zhang |
| **Phone:** | 757-269-6657 |
| **Email:** | hezhang@jlab.org |
| **Date:** | 05/31/2020 |
| **Department/Division:** | CASA/Accelerator |
| **Other Personnel:** |  |
| **Mentor (if needed)** |  |
| **Proposal Term:** | **From:** 10/2020**Through:** 09/2021**If continuation, indicate year (2nd/3rd)**:  |

|  |  |
| --- | --- |
| **Division Budget Analyst** | Annie Ungaro |
| **Phone:** | 757-269-5752 |
| **Email:** | atran@jlab.org |

This document and the material and data contained herein were developed under the sponsorship of the United States Government. Neither the United States nor the Department of Energy, nor the Thomas Jefferson National Accelerator Facility, nor their employees, makes any warranty, express or implied, or assumes any liability or responsibility for accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use will not infringe privately owned rights. Mention of any product, its manufacturer, or suppliers shall not, nor it is intended to imply approval, disapproval, or fitness for any particular use. A royalty-free, non-exclusive right to use and disseminate same for any purpose whatsoever, is expressly reserved to the United States and the Thomas Jefferson National Accelerator Facility.

# Abstract

This project addresses the focus area of quantum information science, especially the field of quantum computing. We propose to develop a quantum computing code that simulates the motion of multiple charged particles under their self-field and test run the code on a virtual quantum computer or a real quantum computer if available. Through the process of developing and running the code, we will investigate the challenges and opportunities to implement quantum computing in beam dynamics simulations for particle accelerators and provide a report.

# Summary of Proposal

## 1.1 Description of Project

Collective effects play an important role into high intensity beam dynamics. A start-to-end simulation of a particle accelerator with real particles based on first principles requires extensive computational resources and is currently not achievable even with the most powerful cluster machines. A quantum computer uses quantum mechanics to do computations and potentially can provide exponential improvement on the efficiency. As the commercial companies competing to demonstrate the “quantum supremacy” [1,2], in the near future, quantum computing may be available to the public as a service, such as IBM Q experience [3]. Taking advantage of the quantum computing and implementing the technique will significantly enhance our beam dynamics simulation capability.

To our best knowledge, there has been no implementation of quantum computing in beam dynamics simulations by now. We proposed to develop the first program that uses the quantum algorithm to simulate the motion of charged particles under their self-field. However, rather than a ready-to-use program for users, this program will serve as the vehicle for us to explore the field of quantum computing, to identify the challenges and opportunities in the design and implementation of quantum algorithms, and to test our approaches. We will record our experiences, summarizes the lessons learned in this process and provide a report as part of the deliverables.

This project studies the implementation of quantum computing to numerical simulations of accelerator physics and addresses the LDRD topic: Quantum information science, including quantum computing and quantum sensing devices, which is not only a topic with particular interest for JLab LDRD program but also one for DOE Advanced Scientific Computing Research program [4]. To implement quantum computing in a specific field needs the effort from the field experts. **This project will be the first step of us, the accelerator physicists, into the quantum computing field. With a modest investment, the result of this project will provide guidance or at least valuable opinions on how we can proceed in the field.**

## 1.2 Expected Results

1. An open source program using quantum algorithms to simulate the motion of multiple charged particles under their self-field.
2. A report on challenges and opportunities of implementing quantum computing to beam dynamics simulations, including the lessons learned in developing and running the code.

# 2. Proposal Narrative

## 2.1 Purpose/Goals

Collective effects, such as space charge effect, halo effect, intrabeam scattering, etc., are important for modern particle accelerators with high intensity beams. Strong collective effects often result in phase space distortion to the particles. In such a case, analytical model, which often assumes a regular phase space distribution, can only give a first order estimation on the dynamic property of the beam, so the numerical study is inevitable to fully understand the physics in accelerator design or performance improvement. The start-to-end simulation of an accelerator using a real beam is the dream for computational accelerator physicists. The particle-based numerical simulation of the collective effects is usually composed of two parts: (1) calculation of the interaction between the charged particles, and (2) solving the dynamic ODEs to move the particles. The computational cost of the simulation is determined by the number of simulation steps and the cost of each step. The cost in one step mostly comes from Part (1), which is often the bottleneck when the particle number is large because the computation cost scales with the square of the particle number. Approximate algorithms [5,6] are often employed to reduce the computational cost. Complex boundary condition may require an iterative field solver, which is even more expensive in computation than a direct solver. The number of steps is affected by the length of the machine under study and the property of the field. Smoother field allows larger step size. But if the field changes drastically or if the collision of the particles plays an important role, such as in the intrabeam scattering effect or the electron cooling process, the step size has to be small enough to catch the physics. Since the particle number in a real beam easily goes above 109, the start-to-end simulation using real beams for an accelerator is still a dream even with current cluster machines.

The development of the quantum computer provides new opportunities to enhance our ability in simulations. Commercial companies as Google are pushing forward the techniques of building quantum computers and quantum computing with tens of qubits may be available as a service in the near future. [7] In academia, studies on quantum algorithm design is also booming in the past a few years. The work most related to our topic is to solve linear system [8-12] and to solve ODEs [13-17] with quantum algorithms. Some papers [18,19] discuss specifically how to solve Poisson’s equation, which is often used to model the electrostatic interaction in the beam frame when relative speed between the particles are low. However, few papers provide numerical examples besides the algorithm study and we have not seen a real scientific problem in physics that has been solved by quantum computing. Clearly there is a gap between the development of the algorithm and the implementation of them. We propose to build a bridge between the quantum computing and the accelerator physics. In this project we want to achieve the following two goals: (1) develop a code that simulates the interaction and the motion of a group of charged particles using quantum computing, which could serve as the foundation for future implementation of quantum computing in real beam dynamics simulations when the resources are available; and (2) identify the challenges and opportunities in the implementation of quantum computing in accelerator physics study, address and present this issues in a report.

## 2.2 Approach/Methods

### 2.2.1 Quantum computer [20-22]

Quantum computer is the device that utilizes special properties of quantum mechanics to perform computation. The three quantum properties that makes the quantum computing different with the classic one are superposition, interference, and entanglement. As any linear combination of two quantum states forms a valid quantum states, quantum superposition allows a quantum system to exist in multiple states simultaneously. Similar to the probabilistic computation, each state in quantum computing is assigned a complex weight. The combination of states with different weight can result in the enhancement, reduction or even annihilation of a state. Quantum interference provides the way to manipulate a quantum system to a preferred state (the correct answer). Entanglement is another special phenomenon in quantum system: a quantum entity in an entangled state cannot be described independently from the system and any change to the quantum entity leads to changes to other entities in the system. This property is also used in some quantum algorithms. [23]

Qubit, the quantum bit, is the fundamental component of a quantum computer. The concept of qubit had been developed even before a physical qubit was realized and is defined as a quantum system that has two base states, which represents 0 and 1. According to the superposition phenomena, the state of a qubit can be represented as

where and are the complex weights and a measurement will make the state collapse to either or , with respective probability ofor. Due to the superposition, the number of states of a quantum system with n qubits goes up to *2n* from to . The information stored in a quantum system scales exponentially with the number of qubits while it scales linearly with the number of bits in a classical system. This is one important reason that quantum computing can provides higher efficiency than classical computers on some problems.

The state of a qubit can be manipulated by unitary operators. These operators are called qugates, i.e. quantum gates, as the analogue of the logic gates in classical computers. Qugates are the basic units that constitute a quantum algorithm. Some qugates have counterparts in classical logic gates, such as the NOT gate, which changes into or into , while others have not, such as the Hadamard gate, which maps a basic gate into a superposition of the basic gates, and the phase operator, which adjust the phase of a qubit. As in classical computing, any computation can be carried out by the combination of a group of fundamental gates, one can build quantum algorithms using the qugates.

Although the quantum computer can easily solve some problems that are extremely difficult to handle in a classical computer, not all the problems can benefit from it. It is not a superfast computer, which will replace the classical computer. Actually, at least for now, from the user’s point of view, a quantum computer is more like a co-processor, such as a GPU (Graphics Processing Unit), which works together with the CPU. A code may primarily run on a CPU and transfer the suitable computation to the co-processors and receives the results from them.

### 2.2.2 Related quantum algorithms

The research in quantum algorithms has borne fruitful results after around two decades of great work. A summary on quantum algorithms is given in [24] with 60 topics and 420 references. In the following, we briefly introduce a few of them, which are related to our purpose.

**Quantum Fourier transform (QFT)** [20,25] is the key ingredient in many important quantum algorithms. Similar with the classical discrete Fourier transform, the QFT is defined as an linear operator that does the following on a base state, which is a base state within .

When applying the QFT on an arbitrary state, we have

where the amplitude are the discrete Fourier transform of the amplitude *.* Now if we considerwith an integer , the base stateof a quantum system withqubits can be written in binary format as *.* Then the QFT can be represented as

With the guidance of the above formula, the QFT can be achieved by applying the Hadamard gate and a series of controlled gate *,* defined as

to each bit inconsequently, and then be applying the swap operation to reverse the order of the qubits. The first step including the Hadamard gate and is shown in fig. 1.

**

Figure 1. Circuit for the quantum Fourier transform

If a unitary operator U has an eigenvectorwith eigenvalue, using the QFT we can find an approximate value ofby the process called **quantum** **phase estimation (QPE)** [25]. This operation needs two quantum registers. The first register has t qubits. The number of qubits determines the accuracy (the number of digits) of the result and the probability for the phase estimation to be successful. Initially all the qubits are in the state . The second register stores the state with as many as qubits that are needed and remains in this state all through the procedure. The schematic of the procedure is shown in fig. 2. In the first step of the procedure, we apply the Hadamard gate to each qubit in the first register, followed by the controlled-U operations on the second register. For the *ith* qubit, the U is applied 2i-1 times. Then the first register will be in the state as

In the second step, we apply the inverse QFT on the first register, which leads to the state *.* A measurement will give us an estimate ofwith *t* digits.

**

Figure 2. Phase estimation procedure

The **HHL algorithm** [8,26,27] is a widely adapted quantum algorithm to solve a linear system,with a matrixand a vector. To find the solution in quantum computing, we assumeandcan be represented as respective quantum stateandand *A* is a Hermitian operator. If the *jth* eigenvector of *A* iswith eigenvalue *,*  can be expressed as a linear combination of the base state as

The purpose of the algorithm is to find the state, which satisfies

The procedure of the HHL algorithm is demonstrated in fig. 3, which needs three registers: ancilla, clock, and input. Initially the ancilla register and the clock register are in state and the input register loads the state *.* The algorithm then proceeds in three steps. First, apply a QPE to estimate the eigenvalue of *A.* Then apply a conditioned rotation to get the eigenvalue inversion. Finally, apply the inverse QPE to resets the clock register back to , which makes the remaining states to be

**

Figure 3. HHL algorithms

Now measure the output of the ancilla register. If the output is *1,* we have obtainin the input register. Otherwise repeat the procedure until the ancilla register gives the output 1. The amplitude amplification technique can be applied to boost the success probability.

The HHL algorithm can be utilized to solve the **Poisson’s equation** and the strategy is to discretize the Poisson’s equation on a regular grid and construct a linear system. [18] For example, we can discretize the 1D Poisson’s equation

on a uniformed grid with *M+1* grid points. Using finite difference method and ignoring the truncated error, the second order derivative on the left side of the equation can be written as

whereand *.* Then the differential equation on the whole grid can be represented as a linear system as

which can be solved using the HHL algorithm. This approach can be extended to multiple dimensional Poisson’s equation using Kronecker products.

It is more complicated to solve **linear differential equations** with respect to time, which describes how a system evolves as time goes. Several approaches based on HHL algorithm [14], spectral method [16], and Taylor expansion [28] have been proposed.

### 2.2.3 Particle-based beam dynamic simulation

Using particles to represent the beam and simulating the motion of the particles inside a device is a widely used method in beam dynamic study. The motion of each particle can be described by the following linear differential equations in the lab frame: [29]

where is the position of the particle,is the momentum of the particle, with *c* the speed of light and the time, is the Lorentz factor, is the electric field and ***B*** is the magnetized field. and in the dynamic equation include both the external field and the self-field between the particles. If the relative velocity between the particles are low, the self-field can be modeled as the electrostatic field in the beam frame, which can be described by the Poisson’s equation as

After solving the Poisson’s equation in the beam frame and convert the result into the lab frame by Lorentz transform, we can solve the dynamic equations. This procedure needs to be repeated at each time step in the simulation.

We will study the feasibility to perform the simulation using quantum computing. The Poisson’s equation may be solved using the aforementioned quantum algorithm, if we choose to discretize the equation on a regular grid. Another choice is to use integral method, which will generate a different linear system with an intense matrix. The aforementioned algorithm designed for sparse matrix may not have good efficiency in this case, but other quantum algorithms [30] may be employed. There are also a few candidates [14-17,28] to solve the dynamic equations. We will explore all the possibilities and choose a proper one to work on. Then we need to make the two algorithms work together, which is not as straightforward as it looks like. Some difficulties can be listed immediately, such as (1) We need to calculate the derivative of the potential obtained from solving the Poisson’s equation to obtain the field; (2) If the field is calculated on the grid, we need to calculate the field on particles by interpolation for the dynamic equations; (3) How to feed the result from one equations to the other ones? These operations are all very straightforward in classical computing, but have to be carefully concerned in quantum computing. One important reason is the results from a quantum algorithm are stored in quantum states. Although a measurement will read out the result in numbers, the measurement procedure is expensive in time and should be avoided as much as possible. It is one of the purpose of this project to identify and address these issues in implementing the quantum algorithms in beam dynamic studies.

### 2.2.4 Development and test platform

Although there are cloud services to quantum computers provided by commercial companies such as IBM [3], D-Wave [31], Amazon [32] and Microsoft [33], we still need a quantum computer simulator for the daily developing and debugging work. The simulators are available for almost all mainstream programming languages, e.g. C/C++, Python, Java, Matlab, etc. There also exist some languages specially designed for quantum computing. One good example is the Q# and the quantum develop kit [34] for it from Microsoft, which can be integrated into Microsoft Visual studio and Azure quantum cloud service. We will investigate these platforms and choose one that is free, suitable for Linux environment and cluster machine and convenient for developing and debugging. It would be better if the simulator provides an intrinsic access to a cloud service of quantum computing.

### 2.2.5 Work plan

We plan to complete the project in the following steps: 1. Investigate the quantum algorithms on solving linear system and solving differential equations. 2. Investigate the available quantum algorithm developing platforms and quantum machine emulators. Select one platform and the respective programming language for code development. 3. Set up the numerical model of the problem and make sure it is compatible with the quantum algorithm. 4. Design the algorithm and develop the code. This is the most challenging part of this project. We need to prepare the quantum state to begin the simulation, develop the field solver and the ODE solver, let them collaborate with each other, and finally read out the result from the quantum states. 5. Test run the code in a virtual quantum machine or a real one if available. 6. Document the process as the project goes on and generate a final report when the project ends. The quarterly goals are listed as follows.

**Goals for FY2021**

* Quarter 1
	+ Choose the developing platform and quantum machine emulator and set up the developing and testing environment.
	+ Decide the numerical model and the quantum algorithms to implement.
	+ Code and test the fundamental pieces for the quantum algorithms, including QFT, HHL, and others if identified.
	+ Document the process.
* Quarter 2
	+ Code and test run the Poisson solver.
	+ Document the process.
* Quarter 3
	+ Code and test run the ODE solver.
	+ Document the process.
* Quarter 4
	+ Design, code and test run the quantum algorithm that combined the Poisson solver and the ODE solver.
	+ Run simulations with a few particles on a virtual quantum machine and a real quantum machine if have access.
	+ Finish the report.

## 2.3 Required Resources

This project will be carried out at Jefferson Lab. We will set up the quantum computer simulator and test run the code in the cluster machines managed by the scientific computing group.

## 2.4 Anticipated Outcomes/Results

1. An open source program using quantum algorithms to simulate the motion of multiple charged particles under their self-field.

2. A report on challenges and opportunities of implementing quantum computing to beam dynamic simulations, including the lessons learned in developing and running the code.

## 2.5 Qualification of the Team

This work will be carried out by the PI, Dr. He Zhang. As a computational accelerator physicist at CASA, he has extensive experience and expertise in algorithm design, scientific computing using high performance parallel platform and beam dynamic simulations. His previous work in the these fields includes: create the differential algebra-based fast multipole method [6] and develop the code that runs in both a personal computer and a cluster machine; perform particle-based simulations on the photoemission process for femtosecond electron bunch generation, [35] which is space charge effect dominated; implement the traceless property to enhance the efficiency of the Cartesian tensor-based fast multipole method for Coulomb interaction, [36] expand the Cartesian tensor-based fast multipole method for *r-v* kernel to general non-oscillating kernels; [37] and develop the intrabeam scattering and electron cooling simulation code JSPEC. [38,39] He will be responsible to finish the proposed work.

# Budget Explanation

The budget includes 0.6 FTE of Dr. He Zhang, who will carry out all the research work, including investigating quantum algorithms and developing platforms, developing and testing the code, performing simulations and documenting the project, and a $2500 travel fund for domestic trip to attend workshop of training on quantum computing. The total budget is $104,419.

References

[1] Preskill, John. "Quantum computing and the entanglement frontier." arXiv preprint arXiv:1203.5813 (2012).

[2] Harrow, Aram W., and Ashley Montanaro. "Quantum computational supremacy." Nature 549.7671 (2017): 203-209.

[3] https://www.ibm.com/quantum-computing/technology/experience/

[4] Cho, Adrian. "DOE pushes for useful quantum computing." Science 359.6372 (2018): 141-142.

[5] Vay, Jean-Luc, et al. "Novel methods in the particle-in-cell accelerator code-framework warp." Computational Science & Discovery 5.1 (2012): 014019.

[6] Zhang, He, and Martin Berz. "The fast multipole method in the differential algebra framework." Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 645.1 (2011): 338-344.

[7] Gomes, Lee. "Quantum computing: Both here and not here." IEEE Spectrum 55.4 (2018): 42-47.

[8] Harrow, Aram W., Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations." Physical review letters 103.15 (2009): 150502.

[9] Clader, B. David, Bryan C. Jacobs, and Chad R. Sprouse. "Preconditioned quantum linear system algorithm." Physical review letters 110.25 (2013): 250504.

[10] Childs, Andrew M., Robin Kothari, and Rolando D. Somma. "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision." SIAM Journal on Computing 46.6 (2017): 1920-1950.

[11] Bravo-Prieto, Carlos, et al. "Variational quantum linear solver: A hybrid algorithm for linear systems." arXiv preprint arXiv:1909.05820 (2019).

[12] Lee, Yonghae, Jaewoo Joo, and Soojoon Lee. "Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience." Scientific reports 9.1 (2019): 4778.

 [13] Leyton, Sarah K., and Tobias J. Osborne. "A quantum algorithm to solve nonlinear differential equations." arXiv preprint arXiv:0812.4423 (2008).

[14] Berry, Dominic W. "High-order quantum algorithm for solving linear differential equations." Journal of Physics A: Mathematical and Theoretical 47.10 (2014): 105301.

[15] Arrazola, Juan Miguel, et al. "Quantum algorithm for nonhomogeneous linear partial differential equations." Physical Review A 100.3 (2019): 032306.

[16] Childs, Andrew M., Jin-Peng Liu, and Aaron Ostrander. "High-precision quantum algorithms for partial differential equations." arXiv preprint arXiv:2002.07868 (2020).

[17] Xin, Tao, et al. "Quantum algorithm for solving linear differential equations: Theory and experiment." Physical Review A 101.3 (2020): 032307.

 [18] Cao, Yudong, et al. "Quantum algorithm and circuit design solving the Poisson equation." New Journal of Physics 15.1 (2013): 013021.

[19] Wang, Shengbin, et al. "Quantum Fast Poisson Solver: the algorithm and modular circuit design." arXiv preprint arXiv:1910.09756 (2019).

[20] Gamble, Sara. "Quantum Computing: What It Is, Why We Want It, and How We’re Trying to Get It." Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2018 Symposium. National Academies Press, 2019.

[21] Kimmel, Shelby. "Quantum Algorithms: Promise and Perspective." Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2018 Symposium. National Academies Press, 2019.

[20] Johnston, Eric, et al. "Programming Quantum Computers: Essential Algorithms and Code Samples." O’Reilly Media Inc., (2019)

[22] Pittenger, Arthur O. An introduction to quantum computing algorithms. Vol. 19. Springer Science & Business Media, 2012.

[23] Zidan, Mohammed, et al. "A novel algorithm based on entanglement measurement for improving speed of quantum algorithms." Appl. Math. Inf. Sci 12.1 (2018): 265-269.

[24] <https://quantumalgorithmzoo.org/>

[25] Nielsen, Michael A., and Isaac Chuang. "Quantum computation and quantum information." Cambridge University Press, (2002): 558-559.

[26] Hidary, Jack D. “Quantum Computing: An Applied Approach.” Springer International Publishing, 2019.

[27] Dervovic, Danial, et al. "Quantum linear systems algorithms: a primer." arXiv preprint arXiv:1802.08227 (2018).

[28] Berry, Dominic W., et al. "Quantum algorithm for linear differential equations with exponentially improved dependence on precision." Communications in Mathematical Physics 356.3 (2017): 1057-1081.

[29] Zhang, He. “The fast multipole method in the differential algebra framework for the calculation of 3D space charge fields.” PhD thesis, Michigan State University, (2013)

[30] Wossnig, Leonard, Zhikuan Zhao, and Anupam Prakash. "Quantum linear system algorithm for dense matrices." Physical review letters 120.5 (2018): 050502.

[31] <https://www.dwavesys.com/>

[32] <https://aws.amazon.com/quantum-solutions-lab/>

[33] <https://azure.microsoft.com/en-us/services/quantum/>

[34] <https://www.microsoft.com/en-us/quantum/development-kit>

[35] Portman, J., et al. "Computational and experimental characterization of high-brightness beams for femtosecond electron imaging and spectroscopy." Applied Physics Letters 103.25 (2013): 253115.

[36] Huang, He, et al. "Improve the efficiency of the Cartesian tensor based fast multipole method for Coulomb interaction using the traces." Journal of Computational Physics 371 (2018): 122-136.

[37] Zhang, He, et al. “Fast multipole method for non-oscillating kernel based on Cartesian tensor and differential algebra.” CSE19, Spokane, Washington, USA (2019).

[38] <https://github.com/zhanghe9704/jspec2>

[39] Zhang, He, et al. "Development of the Electron Cooling Simulation Program for JLEIC." Proceedings of IPAC 2016, Busan, Korea (2016).

Attachments

# VITA (Lead Scientist)

**He ZHANG**

Center for Advanced Studies of Accelerators Tel: 757-269-6657

Thomas Jefferson National Accelerator Facility Fax: 757-269-5024

Newport News, VA 23606 Email: hezhang@jlab.org

**Position Title:** Computational Accelerator Physicist, Jefferson Lab, Newport News, VA

**Education**

* Ph.D. Physics, 2012, Michigan State University, East Lansing, Michigan
* Ph.D. Engineering, 2006, University of Science and Technology of China, Hefei, Anhui, China
* B.S. Physics, 2001, University of Science and Technology of China, Hefei, Anhui, China

**Professional Experience**

* Computational Accelerator Physicist, Thomas Jefferson National Accelerator Facility, 2014- present
* Postdoctoral Research Fellow, Thomas Jefferson National Accelerator Facility, 2012- 2014
* Research Assistant and Teaching Assistant, Michigan State University, 2006-2012
* Research Assistant, University of Science and Technology of China, 2001-2006

**Selected Publications**

1. He Huang, Li-Shi Luo, Rui Li, Jie Chen, He Zhang, Improve the efficiency of the Cartesian tensor based fast multipole method for Coulomb interaction using the traces, Journal of computational Physics 371, 122-136, 2018
2. He Zhang, Ya Derbenev, Yuhong Zhang, Dispersive Electron Cooling for JLEIC, Proc. of IPAC’18, Vancouver, BC, Canada, 2018
3. He Zhang, He Huang, Rui Li, *et al.*, Fast multipole method using Cartesian tensor in beam dynamic simulation, AIP Conference Proceedings 1812(1) 050001, 2017
4. He Zhang, Jie Chen, Rui Li, *et al.*, Development of the electron cooling simulation program for JLEIC, Proc. Of IPAC’16, Busan, Korea, 2016
5. Jenny Portman, He Zhang, *et al.*, Computational and experimental characterizations of high-brightness beams for femtosecond electron imaging and spectroscopy ,*Applied Physics Letter*, 103, 253115 (2013)
6. He Zhang, Martin Berz, Efficient treatment of space charge effects using DA-based FMM Methods, *Vestnik Mathematica*, in print, 2014
7. He Zhang, Martin Berz, The fast multipole method in the differential algebra framework, *Nuclear Instruments and Methods A 645* pp.338-344, 2011
8. He Zhang, Martin Berz, Effects of the wiggler on the Hefei light source storage ring, *International journal of modern physics A*, Vol. 24, No. 5, pp.1057-1067, 2009