# A NEW SCALABLE SOFTWARE PACKAGE FOR LARGE SCALE BEAM DYNAMIC SIMULATIONS

R. Zhao, J. Xu,
Institute of Software, Chinese Academy of Sciences, Zhong Guan Cun, Beijing, China
Ch. Li, X. Qi, L. Yang, Y. He,
Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, China

*Abstract*

Large scale Beam Dynamics Simulations (BDS) are important in accelerator design and optimization. With the fast development of supercomputers, new software packages need to be developed in order to fully make use of hardware and software progresses. In this paper, we will introduce new BDS software, LOCUS3D, which is developed for efficient use of these new techniques. It is based on Particle-In-Cell (PIC) method, and includes space charge effect by solving the Poisson's equation. Parallel Poisson solver has been developed with MPI. Standard accelerator devices can be simulated and new devices can be added. Benchmark results have been obtained on BG/P at ANL. Large-scale simulation with $10^9$ particles can be done now. LOCUS3D will be used for more realistic accelerator simulations in the near future.

## INTRODUCTION

Although there are many beam dynamics codes [1- 4] been developed so far, few of them are designed particularly to efficiently use newly emerged software and hardware technologies for large supercomputers. The beam dynamics code LOCUS3D is now being developed to efficiently use these new technologies for designing and commissioning of various medium-energy high-intensity accelerators worldwide. It solves the Poisson's equation for counting the space charge effect, and the time integration algorithm uses sub-steps for effects of electric and magnetic fields separately [5,6].

In this paper, we present the preliminary efforts on the developments of full version LOCUS3D, including particle initialization module, time integration module, Poisson solver module, parallel model and parallel I/O. The parallel code LOCUS3D has been tested to simulate beam dynamics in the basic accelerator devices, such as multi-pole lenses, solenoid, and Drift Tube Linac (DTL). The parallel Poisson solver has been successfully developed and benchmarked separately. Large number of particles has been successfully simulated on modern supercomputer with thousands of processors. This will greatly increase the capability and accuracy of beam dynamics simulations by using large number particles, which has numbers close to the actual particles per bunch. Thus, beam halo formation in both transverse and longitudinal phase spaces can be clearly observed. The paper will introduce the time integration and parallel schemes first, and then give benchmark and simulation results.

## NUMERICAL METHOD

The numerical methods for LOCUS3D can be divided into two parts: particle tracking and electromagnetic (EM) field computation, including external fields and space charge (SC) fields. Particle tracking is designed to have different time integration schemes for choices. And a parallel Poisson solver using fast Fourier transform (FFT) has been developed for SC fields computation. Currently LOCUS3D uses sub-steps to separates the effects of electric and magnetic fields, which is similar to the algorithm of BEAMPATH code [6]. But it has many differences from BEAMPATH, such as units of quantities, Poisson solver, parallel model, etc.

## PARALLEL FRAMEWORK

LOCUS3D is designed from beginning to be parallel software for complete beam dynamics simulations. It is based on MPI library, and object-oriented C++ language. And it targets on modern supercomputers, such as Tianhe, BG/P, etc.

### *Overall Parallel Layout*

LOCUS3D comprises two major parts: particle pushing and EM field calculation. For particle tracking, the particles are distributed evenly to all processors. They are pushed every $\Delta t$ time simultaneously on all processors. For EM field calculation, SC fields' calculation is usually the major time consumer. As far as we know, the two parts, in the traditional serial code, run in alternation manner [6]. As shown in Fig. 1, in step 1, one can calculate the EM field $\boldsymbol{f}_i$ at time $t_i$ by particle-in-cell method, Poisson equation solver and external field interpolation, since every particle's coordinate information $\boldsymbol{r}_i$ at time $t_i$ is known. In step 2, using the got EM field, one can push every particle over $\Delta t$ time to get the particle velocity $\boldsymbol{v}_{i+1}$ at time $t_i+\Delta t$ by $\boldsymbol{v}_i$ and $\boldsymbol{f}_i$ and to get the particle coordinate $\boldsymbol{r}_{i+1}$ at time $t_i+\Delta t$ by $\boldsymbol{r}_i$ and $\boldsymbol{v}_{i+1}$. Since step 2 depends on $\boldsymbol{f}_i$, which is got in step 1, and particle-in-cell method needs all particles' coordinate information, step 1 and step 2 can't be parallelized.



ISBN 978-3-95450-122-9

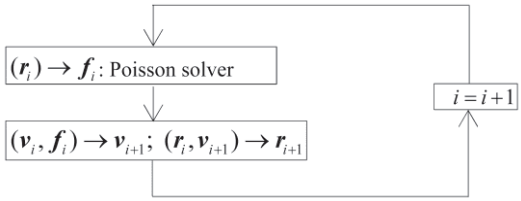


Figure 1: Traditional calculating layout in serial code.

It is noted that, $\boldsymbol{r}_{i+1}$ is calculated by [6]:

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \Delta t * \boldsymbol{v}_{i+1} \quad (1)$$

In our opinion, $\boldsymbol{r}_{i+1}$ can also be calculated by:

$$\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \Delta t * \boldsymbol{v}_i \quad (2)$$

and we expect that Eq. (2) will not significantly affect the precision of symplectic integration. Our parallel layout is shown in Fig. 2.

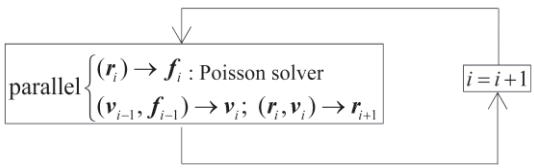


Figure 2: Parallel calculating layout.

The symbols are just like those in Fig. 1. Since step 2 doesn't depend on step 1 anymore, the two step can be parallelized.

### *Parallel Poisson Solver*

LOCUS3D's Poisson solver uses 3 dimensions FFT (3D-FFT) and is based on the fftw3 package [7]. Parallel FFT has been included in fftw3-mpi library for 1D data distribution. For example, if the charge density grid has the size of $n_0 \times n_1 \times n_2$, the grid data can be distributed to at most $n_0$ processes. In step 1, $n_0$ independent 2D-FFTs, with the size of $n_1 \times n_2$, are implemented in parallel. In step 2, grid transpose is made. In step 3, $n_1 \times n_2$ independent 1D-FFTs, with the size of $n_0$, are implemented in parallel. This distribution scheme works well in general situation. However, this is not a good choice if we have a big amount of processes for a 3D-FFT, since the communication volume caused by global transpose in step 2 is too large.

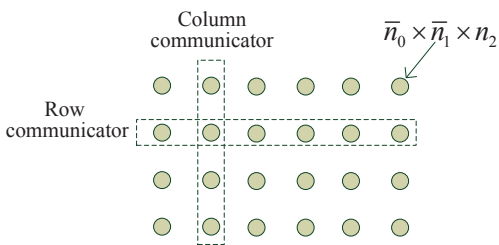


Figure 3: Schematic diagram of 2D data distribution.

In LOCUS3D, a 2D data distribution for big grid size is implemented, as shown in Fig. 3. A data section with size of $\overline{n}_0 \times \overline{n}_1 \times n_2$ is assigned on every process, where $\overline{n}_0$ and $\overline{n}_1$ indicate part of $n_0$ and $n_1$ respectively. The processes in a column (row) communicator have the same $\overline{n}_0$ ($\overline{n}_1$). In step 1, $\overline{n}_0$ 2D-FFTs, with the size of $n_1 \times n_2$, is implemented in every column communicator. In step 2, $\overline{n}_1 \times n_2$ 1D-FFTs, with the size of $n_0$, is implemented in every row communicator. In these two steps, all transpose are in local communicator, so the communication volume is reduced greatly. In theory, $n_0 \times n_1$ processes at most can take part in the 3D-FFT. But the best combination should be obtained by testing.

To validate parallel Poisson solvers, the results have been compared with known analytical solutions and the deviation is within machine round-off error.

## BENCHMARK AND SIMULATION

### *Poisson Solver Benchmark*

The parallel efficiency of Poisson solver is tested individually. Figure 4 shows that the solver's strong scaling result on IBM BlueGene/P supercomputer, with the grid size of $1024^3$ and $256^3$. This means that it can achieve good scalability using large meshes. In addition, the larger of the mesh been used, the better is the parallel efficiency. With 512 processes, the elapsed time is only 0.09 s for the grid size of $256^3$.

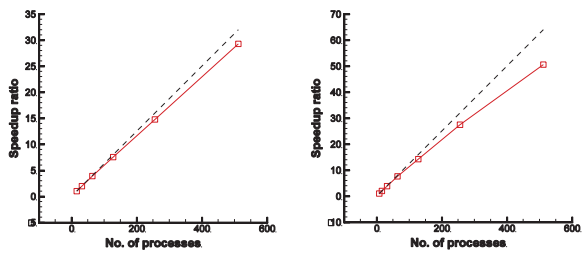


Figure 4: The speed-up factor of Poisson solver for the SC grid of $1024^3$ (left) and $256^3$ (right).

Figure 5 shows the results of weak scaling on BG/P. The size of part of grid on each processor is kept constant, which is $64^3$. As shown, the scalability of the solver is good enough.

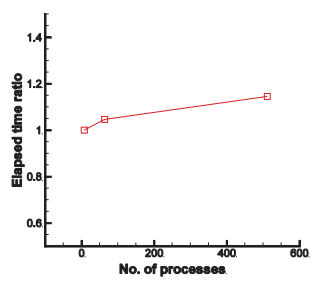


Figure 5: Weak scaling on IBM BG/P.




ISBN 978-3-95450-122-9

### *Validation and Simulations*

To validate LOCUS3D, the results have been compared with known code BEAMPATH. If the parallel layout in Fig. 1 is adapted, the differences between the two codes can be ignored. However, if the parallel layout in Fig. 2 is adapted, there are some obvious differences, since BEAMPATH use the layout in Fig. 1. Some differences are shown in Fig. 6, when a particle bunch with the Kapchinsky-Vladimirsky (KV) distribution passes a magnetic quadrupole FODO channel. In Fig. 6, the top figures are Twiss Parameters $\alpha_x$ and $\alpha_y$; the bottom figures are bunch envelope; The horizontal axis is the position of the bunch center in the transport direction. It is noted that the differences of Twiss parameter $\alpha$ and bunch envelop of x direction is greater than that of y direction. On the differences, we think the reason can be found and a proper choice or modification can be made.

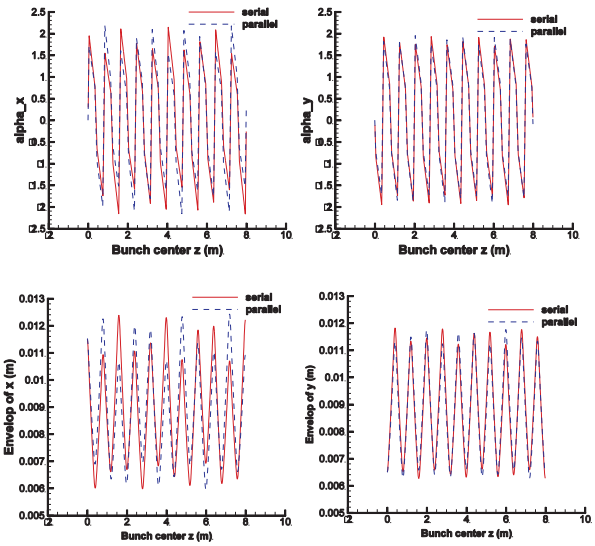


Figure 6: Comparisons between LOCUS3D (dash line) and BEAMPATH (solid line).

Figure 7 shows the initial phase space plots for 50000 particles. The initial parameters are $\alpha_x = 1.5$, $\alpha_y = 1.2$, $X_1 = 2.0$, $X_2 = 0.4$, $Y_1 = 3.0$, $Y_2 = 0.6$, where $\alpha_x$ and $\alpha_y$ are the tilts of the beam ellipses and $X_1$, $X_2$, $Y_1$, $Y_2$ are the semi-axes of the beam ellipses, which is descripted in Ref. 6.

Figure 7: Initial phase space plots for $(x, x')$ (left) and $(y, y')$ (right) with 50000 particles.

Up to $10^9$ particles are simulated on BG/P, with 4096 processes and 250k particles on every process. The time for every pushing step are shown in table 1 with different processes number. From this table, it can be seen that the weak scaling of LOCUS3D is good.

Table 1: Elapsed Time for Every Pushing Step

| processes no. | particles no. (M) | time per step (s) |
|---|---|---|
| 64 | 16 | 20.24 |
| 256 | 64 | 20.43 |
| 1024 | 256 | 20.59 |
| 4096 | 1024 | 20.93 |

## SUMMARY

This paper presents our preliminary efforts on developing parallel beam dynamics software, LOCUS3D. It targets on efficiently use the large supercomputers available now. LOCUS3D is based on the particle-in-cell method and solving the Poisson's equation for the space charge effect. The particles have been distributed evenly to all processors, and parallel Poisson solver has been successfully developed. Parallel particle initialization module and I/O have been developed to fully make use of tens of thousands processors on large supercomputers. Up to $10^9$ particles can be simulated with LOCUS3D now. Several accelerating devices have already been developed, such as multi-pole lenses, solenoid, and DTL. Benchmarks and simulation results have been shown.

## REFERENCES

[1] V.N. Aseev, P.N. Ostroumov, E.S. Lessner and B. Mustapha, “TRACK: The New Beam Dynamics Code”, in Proceedings of PAC-05 Conference, Knoxville, Tennessee, May 16-20, 2005.
[2] J. Xu, B. Mustapha, V.N. Aseev and P.N. Ostroumov, Phys. Rev. ST. Accel. Beams 10, 014201 (2007).
[3] J. Qiang, R. D. Ryne, S. Habib, and V. Decyk, J. Comput. Phys. 163 (2000) 434.
[4] S.Kowalsky and H.A. Enge, RAYTRACE, MIT Report, Cambridge, Massachusets, July 1, 1987.
[5] R. Hockney and J. Eastwood, Computer Simulations Using Particles, Institute of Physics Publishing, London (1988).
[6] Y. K. Batygin, Nucl. Instr. and Meth. in Phys. Res. A 539 (2005) 455.
[7] M. Frigo, S. G. Johnson, "The Design and Implementation of FFTW3," Proceedings of the IEEE 93 (2), 216–231 (2005).