A SELF CONSISTENT MULTIPROCESSOR SPACE CHARGE ALGORITHM THAT IS ALMOST EMBARRASSINGLY PARALLEL[*]

E. Nissen, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606,
B. Erdelyi, S. Manikonda, Argonne National Laboratory, Argonne, IL, 60439,

Abstract

We present a space charge code that is self consistent, massively parallelizeable, and requires very little communication between computer nodes; making the calculation almost embarrassingly parallel. This method is implemented in the code COSY Infinity where the differential algebras used in this code are important to the algorithm's proper functioning. The method works by calculating the self consistent space charge distribution using the statistical moments of the test particles, and converting them into polynomial series coefficients. These coefficients are combined with differential algebraic integrals to form the potential, and electric fields. The result is a map which contains the effects of space charge. This method allows for massive parallelization since its statistics based solver doesn't require any binning of particles, and only requires a vector containing the partial sums of the statistical moments for the different nodes to be passed. All other calculations are done independently. The resulting maps can be used to analyze the system using normal form analysis, as well as advance particles in numbers and at speeds that were previously impossible.

INTRODUCTION

The study of intense charged particle beams often requires an understanding of the effects of space charge, the effect of the electrostatic fields of the many particles comprising the beam on each other. A large number of computational methods have been created to analyze the manner in which these effects manifest themselves in the beam [1]. Since space charge is a collective effect, in order to accurately model the motion of one particle in the beam, it must have information about the motion of every other particle in the beam. This will require information about the position of every particle to be communicated to every other particle.

The use of distributed, parallel computations has greatly sped up some aspects of modeling charged particle beams. Moving non-interacting particles through a magnetic element is an example of what is known as an embarrassingly parallel system. Since the particles don't communicate with each other they can be placed on separate computer nodes and allowed to move completely separately; two computers could literally finish in half the time of one.

[*] Notice: Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

Much progress has been made to adapt space charge solvers for use in parallel systems. Most solvers sort particles by their position into groups, a process known as binning, which provides a logical way to spread the particles out to the different nodes. However, both the communication of the positions of the particles to other nodes, and the possibility that particles can move between bins, requires extensive communication between the nodes, keeping it from being embarrassingly parallel. Furthermore, in some methods such as the FFT the number of useable nodes are constrained.

PARALLEL MOMENT METHOD

We have developed a type of space charge algorithm that is both self consistent [2], and can be made parallel to as many nodes as desired [3]. With one communication between the nodes, it can be as close to embarrassingly parallel as desired. The serial version of this code is outlined in Fig. 1. This version models the transverse space charge effect, though conceptually longitudinal is no different.

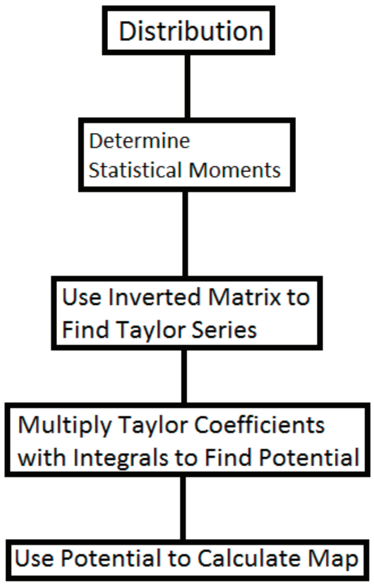
Distribution
Determine Statistical Moments
Use Inverted Matrix to Find Taylor Series
Multiply Taylor Coefficients with Integrals to Find Potential
Use Potential to Calculate Map

Figure 1: An outline of the serial moment method.

The parallelized version of the code is shown in Fig. 2. The changes are made in the original moment method at the point where the moments are calculated.

As is seen in Fig. 1 and Fig. 2 the method begins with a distribution of test particles. In the parallel version they

ISBN 978-3-95450-115-1

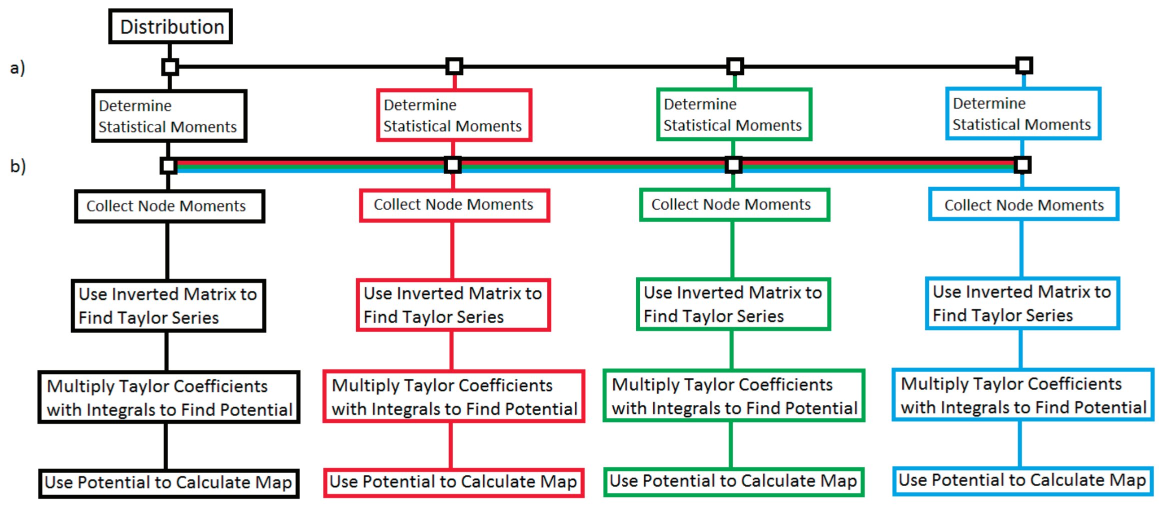
Distribution
a)
b)
Determine Statistical Moments
Collect Node Moments
Use Inverted Matrix to Find Taylor Series
Multiply Taylor Coefficients with Integrals to Find Potential
Use Potential to Calculate Map

Figure 2: Example of the parallelized moment method. The Distribution is sent to the nodes only once, since they do not need to be sorted in any way they will stay on the node they are placed for the duration of the calculation, This is the communication shown at point a). The only other communication is shown after the moments have been calculated, at point b). This information can then be used to advance the particles under the influence of space charge.

must be distributed to the different nodes. One of the advantages of this method is that no sorting is required, we want the different nodes to have distributions that are all roughly representative of the distribution as a whole. This means that there is no need to have particles transfer between different bins, and thus no need to move them between nodes.

The moment method works by transforming the test particles into a Taylor series representation of the potential and electrostatic fields which are combined with the map of the machine element that the beam is passing through using Strang splitting. The moments are defined as,

$$M_{nm} = \sum_{i=1}^{N_{Particles}} x_i^n y_i^m. \tag{1}$$

We assume that the distribution can be modeled using w^{th} order polynomial series, of the form,

$$\rho(x, y) = \sum_{i=0}^{w} \sum_{j=0}^{w-i} C_{ij} x^i y^j \tag{2}$$

We can then link the moments to the distribution,

$$M_{nm} = \sum_{i=0}^{w} \sum_{j=0}^{w-i} \int \int C_{ij} x^{n+i} y^{m+j} dx dy, \tag{3}$$

The integrals can be trivially solved, leading to a matrix equation, which is solved using truncated Singular Value Decomposition methods. This gives the Taylor coefficients of the distribution which are combined with pre-stored integrals to find the potential and electrostatic fields of the system.

The parallelization of the system is introduced in Eq. 1. This is the point where all of the information about the distribution of the particles is determined; everything else can be done separately from that. Therefore, if we have split the distribution into several nodes then we can re-imagine Eq. 1. as,

$$M(j)_{nm} = \sum_{i=1}^{N(j)_{Particles}} x_i^n y_i^m, \tag{4}$$

where j indicates which node this set of moments represents. These $M(j)_{nm}$ terms form an array that is passed to the other nodes using MPI. Once this passing has occurred, each node will perform the operation,

$$M_{nm} = \sum_{j=1}^{N_{Nodes}} M(j)_{nm}. \tag{5}$$

Each node now has all of the information it needs to determine the field of its own particles combined with those of the other nodes, after only one communication.

THE METHOD IN OPERATION

This method has been implemented using the MPI version of COSY Infinity 9.0 [4]. This package uses the message passing interface for parallel computing. This package was installed on the Northern Illinois University Beowulf cluster, which had 24 dual core computer nodes available for this test, for a series of trial computations.

The first trial used lower numbers of particles to determine the parallel speedup factor of the method. The speedup factor is defined as,

$$S_P = \frac{T_{Serial}(N)}{T_{Parallel}(N,P)} \qquad (6)$$

where S_P is the speedup factor, $T_{Serial}(N)$ Is the execution time of the serial algorithm, while $T_{Parallel}(N,P)$ is the execution time of the parallel algorithm with N particles using P nodes. An example of the speedup factors for 50,000 and 100,000 particles over a single kick using a uniform distribution is shown in Fig. 3.

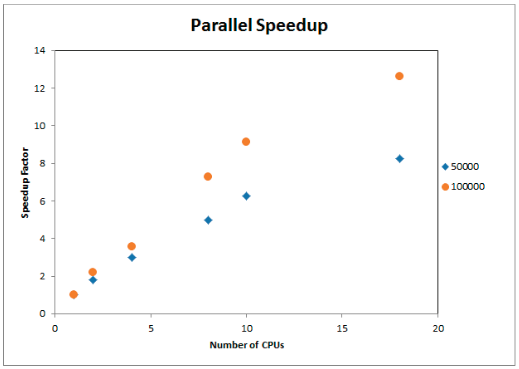

Figure 3: A plot of the speedup factor of the parallel method for 50,000 and 100,000 particles.

The speedup factor in an embarrassingly parallel system would increase perfectly linearly with a slope of P. It is reduced by the need to pass information about the distribution between the nodes. In the case of the parallel moment method however, the size of the data being exchanged between the nodes depends only on the order of the Taylor expansion, so as the number of particles increases the speedup will increase as well.

The true abilities of the moment method are shown when they are applied to very large numbers of particles, as can be seen in Fig. 4 the execution time of the method increases linearly with the number of particles, up through the tens of millions. If we are using two variables to eighth order we would be passing 45 moments per CPU, independant of partcle number. Assuming 8 bytes per moment on 24 CPUS, the all to all type communication is passing 8640 bytes.

Since determining the execution time of 1×10^7 particles was computationally prohibitive, the speedup factor can be estimated using different numbers of CPUs with a fitline applied to it. The results of these experiments are shown in Fig. 5. The fitline gives an expected speedup of $S_P = P^{0.987}$ for 1×10^7 particles.

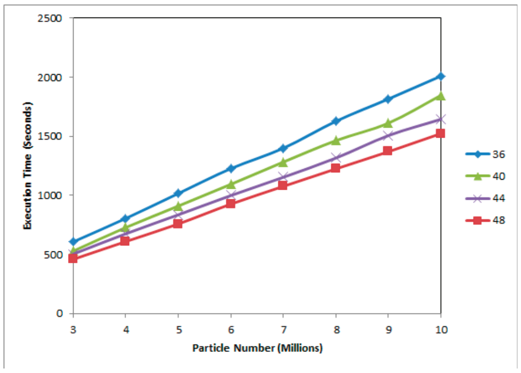

Figure 4: The execution time as it scales with the number of particles for four different sets of numbers of CPUs.

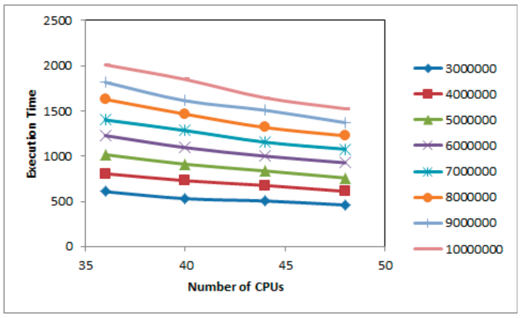

Figure 5: The execution time for different numbers of particles as they scale with the number of CPUs.

CONCLUSIONS

Using the moment method for parallelized space charge calculations allows a space charge method that can be parallelized across as many nodes as are available, and gets very close to being embarrassingly parallel. Since only one communication is required for each space charge evaluation, and the size of the array passed between each node depends on the order of the evaluation, the method is exceptionally suited to very large numbers of particles. This method combined with larger distributed computer networks will allow the possibility of one to one simulation of particles in charged particle beams.

REFERENCES

[1] A. Adelmann, “3d Simulations of Space Charge Effects in Particle Beams,” Ph.D. Thesis, Paul Scherrer Institut (2002).

[2] E. Nissen and B. Erdelyi, “A New Paradigm for Modeling, Simulation, and Analysis of Intense Beams,” Proceedings of the High Brightness High Intensity Beams Workshop, Morschach Switzerland (2010).

[3] E. Nissen, “Differential Algebraic Methods for Space Charge Modeling and Applications to the University of Maryland Electron Ring,” Ph.D. Thesis, Northern Illinois University, (2011).

[4] Y. Kim and M. Berz, “Parallel Constructs in COSY Infinity,” Tech. Rep. MSUHEP-060805, Michigan State University (unpublished) (2006).