

PARALLEL COMPUTING IN BEAM PHYSICS PROBLEMS

S.N. Andrianov, SPbSU, St.Petersburg, Russia

Abstract

In recent years, sufficient interest has been displayed great activity in creating of modern computer codes allowing to release high performance computing for long beam evolution including space charge effects. Here we can mention the dynamical aperture problem and the halo formation problem. The most proposed methods and algorithms are based on on special parallel codes for enormous number of ordinary differential equations, described charged particles motion. In this paper we discuss an approach having two levels of modeling process presentation. The first level uses matrix representation of Lie algebraic tools for nonlinear beam dynamics maps (including space charge forces). Usage of computer algebra methods and codes and object-oriented modeling allows to decompose our simulation process on some independent processes. This decomposition procedure automatically leads to a natural parallel structure of evaluation process. The second level based on usually numerical parallel codes, for example, parallel programming of matrix operations using MPI or PVM.

1 INTRODUCTION

Probably the main part of any computer codes is the mathematical core which should be adequate to using computer architecture. Modern computers with paralleling processing are used first of all for numerical computing for standard mathematical models. As an example it can be mentioned such computing method as parallel Particle-In-Cell (PIC) simulations in the frame of 2D- and 3D-dimensional particle-core model (see for example, [1]). At the same time the deep hierarchy of physical problems demonstrates necessity to use corresponding mathematical models. Indeed in the last time there appear different approaches (see, for example, [2], [3]), which use the object-oriented modeling paradigm. But many of them are not made with modern parallel computer architecture and as a result the corresponding parallel procedure does not lead to saving computer time. So the state of modern computer art pushes us to create new mathematical methods which can use all advantages of parallel processing.

In this paper we discuss an approach based on following fundamental ideas:

- *any beam line system is a dynamical system with control [4];*
- *all control elements (accelerator lattices) can be presented as a structured set of virtual elements [5];*

- *time displacement operators for the dynamical systems can be described using Lie transformations [6];*
- *in appropriate basis any Lie transformation has a matrix representation as a set of two dimensional matrices [7];*
- *any beam state can be describe in the terms of phase space distribution functions admitted matrix representation [7];*
- *matrix formalism for all mathematical objects is a base of parallel procedure of simulation process.*

2 TWO SCHEMES OF SIMULATION PROCESS FOR BEAM DYNAMICS

At present there two schemes used in the practice of beam dynamics simulation: the first of them can be named *the tracking method* (or the ray tracing method) and the second one – *the mapping method*. Let discuss the basic features of these approaches.

2.1 The Tracking Method

This method is based on solution of enormous number of ordinary differential equations, described charged particles motion. For each equation one should formulate an initial problem (Cauchy's problem). So these initial problems correspond to an initial state of the beam, described with the help of a phase set $\mathfrak{M}_0 \subset \mathbf{R}^{2n}$. If the space charge forces can be neglected these initial problems can be realized on parallel computers using usual parallel algorithms for ODE's on the computers with the SIMD (Single Instruction – Multiple Data) technology. Here program commands play a role of instructions for the motion equations solver, and the set of initial data plays a role of multiple data. This process allows to create a required result: the current beam images and so called phase trajectories. This approach is widely used for study of nonlinear beam dynamics, in particular chaotic behaviour. We should note that the necessary information on accelerator elements (guiding and focusing elements) is included in the computational procedure step-by-step as functional coefficients of the motion equations. If the space charge forces stand appreciable than one should include a process of a self-consistent field of the beam calculation (see [8]). For this process there are parallel algorithms too (see, for example, [2]). We should note that the described approach has only

numerical realization and can not be used for a deep analysis without additional procedures. On the Fig.1 one can see two levels of the simulation process using the tracking method. The inner level admits parallelization procedure naturally. The outer level has sequential character and usually can not be realized on parallel computers (see the Fig.1).

2.2 The Mapping Method

This method is based on the priority of map creation. Using a database of accelerator elements (drifts, dipoles, multipole lenses and so on) we computer a map generated by the machine. In this case an accelerator or its part are described as a dynamical system with control. In this work we use the matrix formalism for Lie algebraic methods developed in previous works [7]–[9].

The Lie Algebraic Tools. Let be

$$\frac{dX}{ds} = F(X, U, s) \quad (1)$$

a motion equation for beam particles in external and space-charge fields. Here the vector $U(s)$ describes control functions corresponding to guiding and focusing fields. Any solution of this equation can be written in the form

$$X(s) = \mathcal{M}(s|s_0) \cdot X_0, \quad X_0 \in \mathfrak{M}_0,$$

where $\mathcal{M}(s|s_0)$ is so called Lie map (transformation). This map satisfies to the following operator equation

$$\mathcal{M}(s|s_0)(s) = \mathcal{L}_F \cdot \mathcal{M}(s|s_0), \quad \mathcal{M}(s_0|s_0) = Id,$$

where \mathcal{L}_F is a Lie operator associated with the function F from the Eq.(1). For non-autonomous systems the so called Magnus's representation [7] is used. This approach allows to pass from the time-ordered exponent operator to a routine exponential operator. The expansion of the function $F(X, s)$:

$$F(X, s) = \sum_{k=0}^{\infty} \mathbf{P}^{1k}(s) X^{[k]}$$

generates an expansion of the function $G(X; t|t_0) = \sum_{k=0}^{\infty} \mathbf{G}_k(t|t_0) X^{[k]}$ which appears in the Magnus's representation and one can write

$$\mathcal{M}(t|t_0) = \exp \left\{ \sum_{k=0}^{\infty} \mathcal{L}_{G_k}(X, U; t|t_0) \right\},$$

$$G_k(X, U; t|t_0) = \mathbf{G}_k(U; t|t_0) X^{[k]}.$$

The similar to the Dragt-Finn factorization for the Lie transformations allows to rewrite the exponential operator as an infinite product of exponential operators of Lie operators

$$\begin{aligned} \mathcal{M} &= \dots \exp\{\mathcal{L}_{H_2}\} \cdot \exp\{\mathcal{L}_{H_1}\} = \\ &= \exp\{\mathcal{L}_{V_1}\} \cdot \exp\{\mathcal{L}_{V_2}\} \cdot \dots, \end{aligned}$$

where $H_k = \mathbf{H}_k X^{[k]}$, $V_k = \mathbf{V}_k X^{[k]}$ are homogeneous polynomials of k -th order. The matrices \mathbf{H}_k or \mathbf{V}_k can be calculated with the help of the continuous analogue of the CBH-and Zassenhauss formulae and by using the Kronecker product and Kronecker sum technique for matrices [7]. Moreover, using the matrix representation for the Lie operators one can write a matrix representation for the Lie map generated by these Lie operators

$$\begin{aligned} \mathcal{M} \cdot X &= \mathbf{M} X^\infty = \\ &= (\mathbf{M}^{10} \mathbf{M}^{11} \mathbf{M}^{12} \dots \mathbf{M}^{1k} \dots) X^\infty = \\ &= \sum_{k=0}^{\infty} \mathbf{M}^{1k} X^{[k]}, \end{aligned} \quad (2)$$

$$X^\infty = (1 X X^{[2]} \dots X^{[k]} \dots)^*,$$

where the matrices \mathbf{M}^{1k} (**solution matrices**) can be calculated according to the recurrent sequence of formulae of the following types:

$$\begin{aligned} \mathcal{M}_k \cdot X^{[l]} &= \exp\{\mathcal{L}_{G_k}\} \cdot X^{[l]} = \\ &= X^{[l]} + \sum_{m=1}^{\infty} \frac{1}{m!} \prod_{j=1}^m \mathbf{G}_m^{\oplus((j-1)(k-1)+l)} X^{[m(k-1)+l]}, \end{aligned}$$

where $\mathbf{G}^{\oplus l} = \mathbf{G}^{\oplus(l-1)} \otimes \mathbf{E} + \mathbf{E}^{[l-1]} \otimes \mathbf{G}$ denotes the Kronecker sum of l -th order. For the inverse map $\mathcal{M}^{-1}: X \rightarrow X_0 = \mathcal{M}^{-1} \cdot X$ one can compute the corresponding block-matrices using the generalized Gauss's algorithm.

Computer Algebra and Knowledge Bases. The desired solution is created in the form of power series. It is clear that this way can be realized only with truncated procedures for some chosen order of expansions – the approximation order N . In the referred works the corresponding matrices \mathbf{P}^{1k} , \mathbf{G}_k , \mathbf{H}_k , \mathbf{V}_k and \mathbf{M}^{1k} are calculated up to seventh order in symbolic forms using the computer algebra codes (MAPLE V). These tools have algebraic character and can be easily realized on parallel computers.

The Map Creation. Knowledge of the Lie map $\mathcal{M}(s|s_0)$ in the matrix form (see the Eq.(2)) up to some approximation order N allows to create necessary criteria for beam line working and current images $\mathfrak{M}(s) = \mathcal{M}(s|s_0) \cdot \mathfrak{M}_0$. We should note that the approximation order N has to make consistent with the order of approximation for the external and space charge fields. In this approach one follows current phase beam portraits. These portraits allow to calculate additional criteria of beam evolution. From computing point of view this approach has two levels of realization too. But in contrast to the tracking method the mapping method has another structure of computational process (see the Fig.2). The map computing (the inner level) is based on a set of ready matrices $\mathbf{M}^{1k}(s|s_0)$ for different accelerator elements calculated in symbolic forms (using computer algebra codes REDUCE

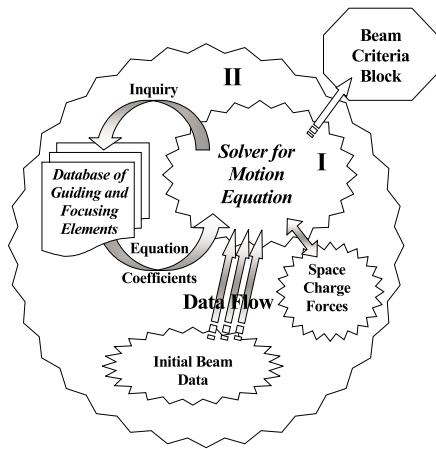


Figure 1: The calculation scheme according to the tracking method. There two levels of calculations: the inner level **I** is based on the motion equations solver. The initial data flow realizes the computing beam phase portraits on the outer level **II**.

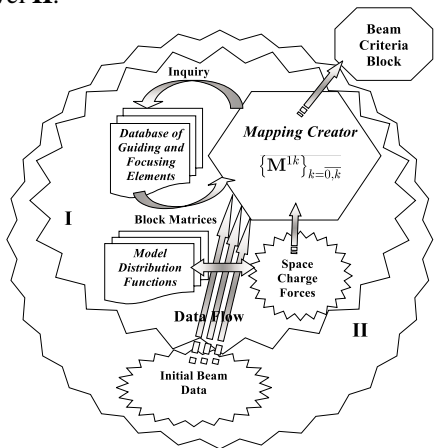


Figure 2: The calculation scheme according to the mapping method. There two levels of calculations: the inner level **I** is based on the map calculation in the matrix forms. This process has a parallel character as it is based on the matrix algebra. The initial data flow enters on the outer level **II**.

and MAPLE V). These matrices are the fulfilment of the corresponding database. It is known that the matrix algebra admits the parallelization in a natural way. Then one can calculate necessary criteria using only matrix elements of current matrices M^{1k} , $k \leq N$ (see, for example, [10]). On the outer level we use the beam information in the different forms: 1) the model initial distribution function $f_0(X)$, $X \in \mathcal{M}_0$ [10]; 2) the equipotential surfaces or lines for initial phase portrait \mathcal{M}_0 presentation: $G_0(X) = C$, $C = \{g_0^1, \dots, g_0^m\}$, where m is the number of levels; 3) the pointwise presentation of the \mathcal{M}_0 . This information treat in to the calculation process for those sections for which one wants to study the beam behaviour. So we deal with the so called MIMD (**M**ultiple **I**nstruction, **M**ultiple **D**ata) technology.

The Advantage of the Mapping Method. From computing point of view the second approach based on the mapping method is more preferable. Indeed at first we have two parallel processes: for maps computation and for phase portraits computation. In the second we can use the prepared in advance matrices M^{1k} (extracting them from the corresponding database). From analytical point of view the mapping method is also preferable as one can analyze corresponding criteria without beam images calculation.

3 CONCLUSION

The discussed approaches is verified on some test problems (for example, for halo formation process). For some models of beam lines we approved these approaches on the base of the High Performance Center of St.-Petersburg State University. These computer experiments confirmed our estimates of effectiveness of these approaches.

REFERENCES

- [1] J. A. Holmes, V. V. Danilov, J. D. Galambos, D. Jeon, and D. K. Olsen, Phys.Rev.Special Topics - Accel.and Beams, **2**, 114202, 1999.
- [2] J. Qiangy ,R. D. Ryne, S. Habib, Proc. of the 1999 Particle Accelerator Conference, N.Y., (1999), pp.366–368.
- [3] Tai-Sen F. Wang and Thomas P. Wangler, Proc. of the 1999 Particle Accelerator Conference, N.Y., (1999),pp.1848–1850.
- [4] S.N. Andrianov, A.I. Dvoeglazov, Proc. of the 1997 Particle Accelerator Conference, Vancouver, BC, Canada, 1997, pp.2419–2421.
- [5] S.N. Andrianov, A.I. Dvoeglazov, Proc. of the Sixth European Particle Accelerator Conference, Stockholm, Institute of Physics Publishing, Bristol, UK, pp.1150–1152.
- [6] A.J. Dragt, Physics of High Energy Particle Accelerators. Eds. Carrigan, Huson F.R., Month M. AIP Conf. Proc., **87**, NY, 1982, pp.147–313.
- [7] S.N. Andrianov, Proc. of the 1996 Computational Accelerator Physics Conference – CAP’96, Sept. 24–27, Williamsburg, Virginia, USA, eds. J.J.Bisognano, A.A.Mondelli, AIP Conf. Proc. **391**, NY, 1997, pp.355–360.
- [8] S.N. Andrianov, Proc. of the Second Int.Workshop Beam Dynamics & Optimization, 1995, St.Petersburg, Russia, SPbSU, St.Petersburg, 1996, pp.25–32.
- [9] S.N.Andrianov, Proc. of the Sixth European Particle Accelerator Conference, Stockholm,1998, Institute of Physics Publishing, Bristol, UK, pp.1091–1093.
- [10] S.N.Andrianov, V.M.Taryanik, in this Proc.