BEAM DYNAMICS SIMULATION VIA PARALLEL AND DISTRIBUTED COMPUTING

S.N. Andrianov, SPbSU, Saint-Petersburg, Russia

Abstract

In this paper some approaches to beam dynamics simulation via parallel and distributed computing are discussed. This approach is based on two modeling levels. On the first level we find symbolic presentations both for a beam propagator (using the matrix formalism for Lie transformations)and for particle beam description (using a set of model distribution functions). On this level the LEGO-technology is used. This allows creating special date bases of LEGO-objects. On the second level we generate some distributed simulation flows synchronized with each other. For this purpose the natural parallel and distributed structures of the beam dynamics are used. The matrix formalism presentation for the beam propagator (Lie transformation) permits us to reduce numerical operations to routine matrix algebra operations. This allows us to realize the numerical simulation process on a computer cluster or on a set of computer clusters. A version of the prediction-correction method is used for space-charge-dominated beam dynamics.

1 INTRODUCTION

Although we have had undoubted advances of parallel and distributed computer systems, such systems are not widely adopted in accelerators physics. The main problem of such status is the emergency cost of supercomputers. However modern computer cluster systems are more cheaper and available compared with the supercomputers. Available evidence suggests that clusters can come to more applicable in computer modeling for beam physics problems. The major aim of this report is to discuss mathematical methods which are maximal adequate not only physical models, but the modern soft- and hardware. For this purpose a researcher should use a special representation for beam dynamics which is more congenial to cluster computations features. Usually there are several cluster systems connected into a local computational net. In the Saint-Petersburg State University (in the Petrodvoretz Campus) there are six clusters consolidating more 6 processors. The clusters each have $8 \div 12$ processors. The basic concept is that the whole problem is decomposed into several subproblems. Each of subproblems is computed on a cluster, while the common results are computed with the help of synchronization processes. These procedures of subproblems synchronization can be realized in a different way depending on concrete physical problem. It is obvious that used mathematical models have to admit similar decomposition naturally. All previous investigations [1]–[3] demonstrated that the matrix formalism for Lie algebraic methods give the best fit. We proceed from the knowledge that matrix algebra is very adaptable for parallel and distributed processing. In this reason it is a challenge to present beam dynamics problems in the matrix terms. In this approach all manipulated objects are two dimensional matrices. Usual matrix operations are supplemented with extended by Kronecker product and sum operations.

In this paper we consider the decomposition procedure based on the matrix representation for well known Lie algebraic methods [4].

2 THE MATHEMATICAL BACKGROUND

For an arbitrary dynamical system we can write

$$\mathbf{X} = \mathcal{M}(\mathbf{U}; s|s_0) \circ \mathbf{X}_0, \tag{1}$$

where $\mathbf{X}_0$, $\mathbf{X}$ are n-dimensional initial and current phase vectors, $\mathcal{M}(\mathbf{U}; s|s_0)$ a map (a propagator) generated by the dynamical system under study. Here $\mathbf{U}$ is a control vector describing control parameters and functions. According to the usual notations the motion equation can be written in the form of ordinary differential equation for the propagator

$$\frac{d\mathcal{M}(\mathbf{U}; s|s_0)}{dt} = \mathcal{L}(\mathbf{U}; s) \circ \mathcal{M}(\mathbf{U}; s|s_0), \tag{2}$$

where $\mathcal{L}(\mathbf{U}; s)$ is a Lie operator, $\mathbf{U}$ — the vector of control functions, describing, for example, the external (control) electromagnetic fields. In the arbitrary case the solution of the Eq. (1) can be written in the form of so called time-ordered presentation:

$$\mathcal{M}(s|s_0) = \mathrm{T}\exp\left(\int\limits_{s_0}^{s} \mathcal{L}_{\mathbf{F}(\tau)} d\tau\right),$$

Here $\mathrm{T}\exp\{\cdot\}$ is the time-ordered exponential operator. Using the Magnus representation [5] for the map $\mathcal{M}$ one can write

$$\mathcal{M}(s|s_0) = \exp\left(\mathcal{L}_{\mathbf{G}(s|s_0)}\right),$$

where the new vector function $\mathbf{G}$ can be evaluated using the continuous analog of the Baker-Campbell-Hausdorff formula. The operator equations (1), (2) are generated by the system of ordinary differential equations (”motion equations”)

$$\frac{d\mathbf{X}}{ds} = \mathbf{F}(\mathbf{E}, \mathbf{B}, \mathbf{X}; s). \tag{3}$$

Following to previous papers [1]–[3] we present the force function $\mathbf{F}(\mathbf{E},\mathbf{B},s)$ in the following series

$$\mathbf{F}\left(\mathbf{E},\mathbf{B},s\right)=\sum_{k=0}^{\infty}\mathbb{P}^{1k}(\mathbf{E}_{\{k\}},\mathbf{B}_{\{k\}},s)X^{[k]},$$

where $\mathbf{X}^{[k]}$ is the Kronecker power of k-th order of phase vector $\mathbf{X}$, $\mathbf{E}_{\{k\}}=\mathbf{E}_{\{k\}}(s)$, $\mathbf{B}_{\{k\}}=\mathbf{B}_{\{k\}}(s)$, are external field functional characteristics (for example, field distribution along the reference orbit). The vector function $\mathbf{E}_{\{k\}},\mathbf{B}_{\{k\}}$ are included to the control function vector $\mathbf{U}(s)$. This expansion leads us to the following solution form for particle motion equations:

$$\mathbf{X}(s)=\sum_{k=0}^{\infty}\mathbb{M}^{1k}\left(\mathbf{E}_{\{k\}},\mathbf{B}_{\{k\}},s\right)\mathbf{X}_0^{[k]},$$

where $\mathbf{X}_0=\mathbf{X}(s_0)$ for some initial value of the phase vector $\mathbf{X}$ and $\mathbb{M}^{1k}$ are aberration matrices of k-th order. We must note that the matrix formalism allows to computer these matrices in advance in a symbolic form (using, for example, computer algebra code Maple). Using the Poincare–Witt $\left\{\mathbf{X}^{[k]}\right\}|_{k\geq 1}$ basis for the motion equation (1) one can write the following expansion

$$\frac{d\mathbf{X}}{dt}=\sum_{k=0}^{\infty}\mathbb{P}^{1k}(\mathbf{U};t)\mathbf{X}^{[k]}. \qquad (4)$$

For the corresponding propagator $\mathcal{M}_{\mathbf{F}}$ one can obtain the following matrix presentation

$$\mathcal{M}_{\mathbf{F}}(\mathbf{U};t|t_0)\circ\mathbf{X}=\sum_{k=0}^{\infty}\mathbb{M}^{1k}(\mathbf{F},\mathbf{U};t|t_0)\mathbf{X}^{[k]}. \qquad (5)$$

In the Eqs. (4) and (5) $\mathbb{P}^{1k}$, $\mathbb{M}^{1k}$ are two dimensional matrices (block-matrices): the matrices $\mathbb{P}^{1k}$ describe the source dynamical system (for example, a beam line) in k-th order and the matrices $\mathbb{M}^{1k}$ — the solution of the motion equation (5). Here it should be note that an investigator should truncate the series in the Eq. (5) for a some beam line, depending on his knowledge or needs on acting nonlinear effects. Let N note the approximation order for similar truncated series. It is known that after truncation procedure the effecting propagator losses its qualitative properties (for example, symplecticity). In the case of Hamiltonian formalism we apply the correction procedure [6]. In more general cases for this correction procedure we use the knowledge of approximation symmetries and invariants [7]. Similar restrictions have the form of linear algebraic equations for the block-matrices $\mathbb{M}^{1k}$, $k\leq N$, which can be solved in advanced using computer algebra codes (we use Maple codes). This approach allows to guarantee conservation necessary properties for the truncated map up to N-th order of approximation.

3 BEAM PHASE PORTRAIT EVOLUTION

Practically all necessary information about beam evolution can be evaluated from phase portrait characteristics. Here there are three approaches. The first is traditional and based on point presentation of the phase manifold, occupied by beam particles or phase portrait of the dynamical system (for example, for the well known Henon–Heiles potential see the Fig. 1). The second uses the distribution functions and the third — the beam boundary equation in the form $\mathbf{G}(\mathbf{X},s)=0$. Last two approaches are consistent with the matrix formalism and described in [3]. The last approach is not consistent for space-charge problems, but the two first can include the space-charge forces in the corresponding computational schemes. It should be noted that in this case one has to use the analog of predictor-corrector method [3]. For all description pictures we use the matrix presentation of the propagator in the form (6).

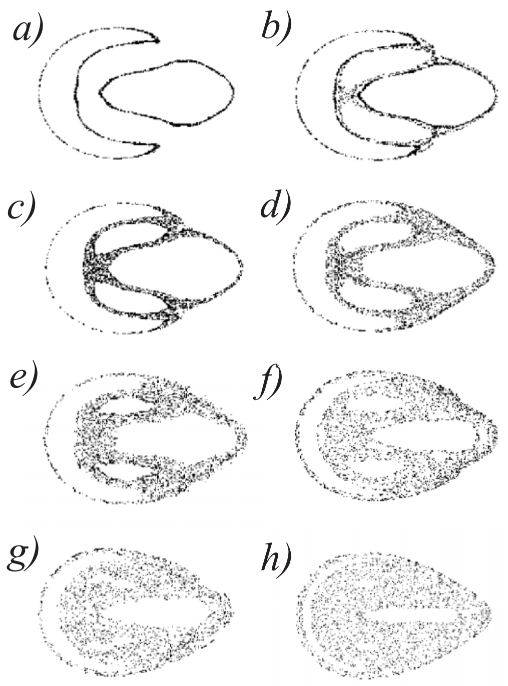


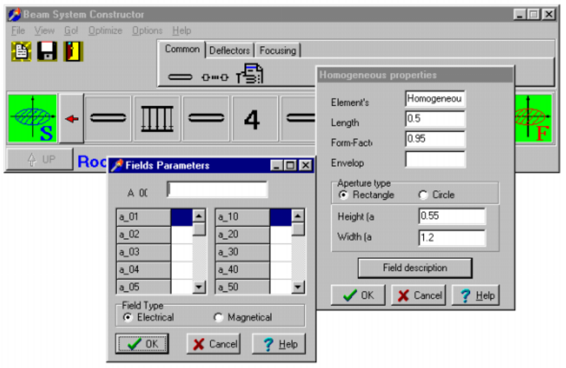
Figure 1: The example of interface window for long time evolution simulation.

4 THE COMPUTER PARALLEL FLOWS

It is known that the most used numerical methods are parallelized with significant effort (see, for example, [9]–[10]), while matrix algebra methods are parallelized naturally. It is necessary note that all the operations for the matrix formalism have parallel structure elementally. For matrix operations one can talk about homogeneous operations, which can be realized on computers separate entering into a computational cluster. Basically here one can use general parallel packages for linear algebra. Here one has to pay attention to computer load as so all processors do roughly the same amount of work the load balance issue and that interprocessor communication does not seriously a affect performance. As a result a researcher should pay attention to creation of optimal algorithms for parallel computers. Standard libraries of MPI (Message Passing Interface) are quite mature for these problems. But software challenges remain particularly for computer systems such as clusters.

5 THE COMPUTER DISTRIBUTED FLOWS

In order to ease and speed up the beam dynamics simulation we suggest to organize the computing in several computational flows. Each flow correspond to particular problems among the it is necessary to mention the following: beam propagator evaluation, beam evolution (using the phase portrait concept), space-charge forces computing and visualization procedure. Certainly the are another computational flows, but they play auxiliary role. This separation into flows is based on two modeling levels. On the first level we find symbolic presentations both for a beam propagator (using the matrix formalism for Lie transformations) and for particle beam description (using a set of model distribution functions and/or boundary functions, see, for example, [3]). On this level the LEGO-technology is used (see [11]). This allows creating special date bases of LEGO-objects. On the second level we generate some distributed simulation flows synchronized with each other. For this purpose the natural parallel and distributed structures of the beam dynamics are used. The matrix formalism presentation for the beam propagator (the Lie transformation) permits us to reduce numerical operations to routine matrix algebra operations. This allows us to realize the numerical simulation process on a computer cluster or on a set of computer clusters. The main difficulties are generated by of synchronization problem for two flows: beam propagator flow and space-charge forces flow. Other types of computational flows interchange by information more rarely in comparison with the first two flows. For space-charge beam dynamics we use a version of the special prediction-correction method [3].



Homogeneous properties

Element's: Homogeneou
Length: 0.5
Form-Fact: 0.95
Envelop:
Aperture type: [✓ Rectangle] Circle
Height (a: 0.55
Width (a: 1.2

A 0(:
a_01, a_02, a_03, a_04, a_05
a_10, a_20, a_30, a_40, a_50
Field Type: [✓ Electrical] Magnetical

Figure 2: The example of interface window for beam line design.

6 COMPUTER EXPERIMENTS

For beam numerical simulation described approach we use special object-oriented codes which generated using Fortran 90/High Performance Fortran and MPI codes (for Linux platform). All symbolic computations were evaluated using Maple codes. Usage of splitting on physical processes MPI codes allow to organize calculation independent (within the limits of the given temporary step) flows simultaneously. Obtained parallel computer program complex has property of a scalability, that leads to carry them on computer systems with any arbitrary number of processors. The necessary number of processors is determined for each particular problem. Besides MPI codes give the code developers flexibility of combination of different program units, organization of the necessary architecture of the program complex. The program complex is tested on the cluster system the Saint-Petersburg State University (in the Petrodvoretz Campus). Primarily it was realized on the cluster (eight Processors, Celeron 600 MGz, 256 Mb RAM, Fast Ethernet 100 Mbit/sec) of the Applied Mathematics & Control Processes Faculty. In the future this approach will be realized on the cluster system consists on 60 computers. In perspective we hope build a distributed computer system total $100 \div 200$ computers. Now a special interface for distant admittance is created.

The special interfaces were created for solving different problems (see examples such interface windows on the Fig. 2). These interfaces give the designer the comfortable possibility to manipulate by computer objects.

7 REFERENCES

[1] S.N. Andrianov, in: AIP Conf. Proc. No 391, NY, 1997. P. 355–360.

[2] S.N. Andrianov, in: Proc. of the 1999 Particle Accelerator Conference, 1999, NY (1999). P. 2701–2703.

[3] S.N. Andrianov, in: Proc. of the Sixth European Particle Accelerator Conference, Stockholm, Sweden, 1998. Bristol (UK) (1998). P. 1091–1093.

[4] A.J. Dragt, in: Physics of High Energy Particle Accelerators. AIP Conf. Proc., N 87. N.-Y. (1982). P. 147–313.

[5] W. Magnuss, in: Comm. Pure Appl. Math. Vol. 7. No 4 (1954). P. 649–679.

[6] S.N. Andrianov, in: Mathematics and Computers in Simulation. Vol. 57. Issue 3–5, (2001), P. 139–145.

[7] S.N. Andrianov, in: Mathematics and Computers in Simulation. Vol. 57. Issue 3–5, (2001), P. 146–154.

[8] S.N. Andrianov, in: Proc. of the First Int. Workshop Beam Dynamics & Optimization, SPb. 1996. P. 19–29.

[9] J. Galambos, J. Cobb, J.A. Holmes, D.K. Olsen, in: Proc. of the 2000 European Particle Accelerators Conference, Vienna, Ausria, 2000. P.1372–1374.

[10] M. Giovannozzi, in: Proc. of the 1998 European Particle Accelerators Conference, Stockholm, Sweden, 1998. P. 1189–1191.

[11] S.N. Andrianov, in: this Proceedings.