MATRIX FORMALISM FOR LONG-TERM EVOLUTION OF CHARGED PARTICLE AND SPIN DYNAMICS IN ELECTROSTATIC FIELDS

A. Ivanov*, S. Andrianov, Saint-Petersburg State University, Russia

Abstract

The matrix formalism as a numerical approach for solving of ODE equations is considered. It is a map method and has several advantages over classical step-by-step integration methods. This approach allows to present the solution as set of numerical matrices. A complete derivation of the equations this method is based on will be shown. Problems of symplectification and computing performance are discussed. We have developed an application that provides a tool for differential equations solving. The developed program allows to generate the final programming codes on C++, Fortran, MATLAB, C#, Java languages. The given approach is applied to long-term evolution of charged particle and spin dynamics in electrostatic fields.

MATRIX FORM OF ODE

Let's introduce a nonlinear system of ordinary differential equations

$$\frac{d}{dt}X = F(t, X). \tag{1}$$

Under the assumptions of $F(0, X_0) = 0$ the system (1) can be presented in the following form

$$\frac{d}{dt}X = \sum_{k=0}^{\infty} P^{1k}(t)X^{[k]}, \tag{2}$$

where $X^{[k]}$ is kronecker power of vector X, matrices P^{1k} can be calculate as

$$P^{1k}(t) = \frac{1}{(k)!}\frac{\partial^k F(t, X_0)}{\partial (X^{[k]})^T}, \quad k = 1, 2, \dots$$

Note that vector X is equal to $(x_1^{k_1}, \dots, x_n^{k_n})$, where x_i means ith component of state, $(k)! = k_1! \dots k_n!$

NUMERICAL IMPLEMENTATION

Solution of system (2) can be written in form

$$X = \sum_{k=0}^{\infty} R^{1k}(t)X_0^{[k]}. \tag{3}$$

Elements of matrices R^{1k} are depended on t and can be calculated in symbolic mode [1]. But such algorithms are quite complex. In this paper we propose a numerical implementaton of it. In this case matrices R^{1k} are evaluate in the specific time.

* 05x.andrey@gmail.com

After differentiating the equation (3) and taking into account (2) we get

$$\frac{dX}{dt} = \sum_{k=0}^{\infty} \frac{dR^{1k}(t)}{dt}X_0^{[k]},$$
$$\sum_{k=1}^{\infty} \frac{dR^{1k}(t)}{dt}X_0^{[k]} = \sum_{k=1}^{\infty} P^{1k}(t)X^{[k]}.$$

The partial derivatives of this equations with respect to $X_0^{[j]}$ are equal to

$$\begin{aligned} \frac{dR^{10}(t)}{dt} &= \sum_{k=1}^{\infty} P^{1k}(t)(R^{1k})^{[k]}, \\ \frac{dR^{1j}(t)}{dt} &= \sum_{k=1}^{\infty} P^{1k}(t)\frac{\partial X^{[k]}}{\partial (X_0^{[j]})^T}, \quad k = 1, 2, \dots \end{aligned} \tag{4}$$

and define the system of ordinary differential equations. Solution of this system is deterined matrices R^{1k}.

For integration of equations (4) numerical approach can be used. Note that step-by-step integration use only for map building. After that the solution that corresponds to the initil point X_0 can be calculate with the same map (3). In the research we use symplectic 2 stage Runge-Kutta scheme of 4 order (see [2]).

SYMPLECTICATION

The relation (3) can be presented as map transformation

$$X = R \circ X_0. \tag{5}$$

This map R is symplectic if

$$M^* J M = J, \forall X_0, \tag{6}$$

where $M = \partial X / \partial X_0$ and M^* is the transponse of M, E is identity matrix,

$$J = \begin{pmatrix} 0 & E \\ -E & 0 \end{pmatrix}. \tag{7}$$

Relation (6) in case of numerical matrices R^{1k} leads to a system of equations

$$a_0 + A_1 \mathbf{X_0}^{[1]} + \cdot + A_k \mathbf{X_0}^{[k]} = 0,$$

where A_i is a numerical vector. Note that this equation must be satisfied for any X_0. It means that the coefficients of each polynom are equal to zero and in this way appropriate corrections of the elements of the matrices R^{1k} can be found.

ISBN 978-3-95450-116-8

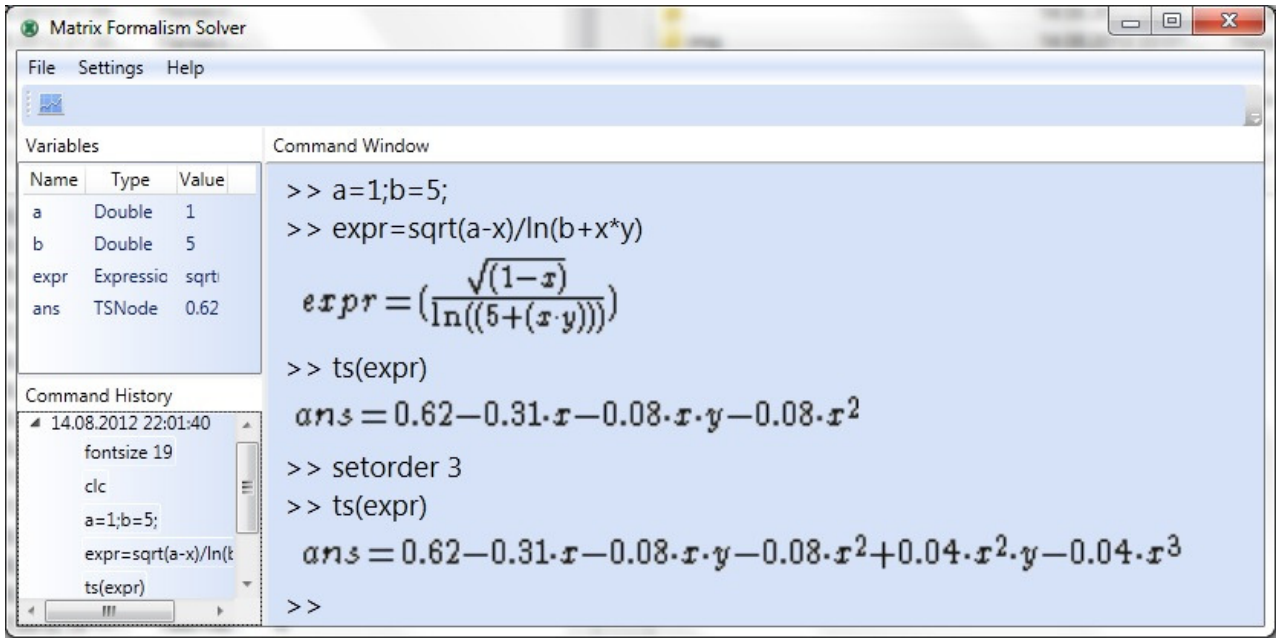

Figure 1: Command window for Taylor series expansion.

ODE SOLVING

For solving an ordinary differential eqation it is neccessary to implement following steps (see Fig. 2):

- to expanse function F in (1) in a Taylor series up to the order k;
- to build matrices $P^{1i}, i = 1 \dots k$;
- to solve the ordinary differential equations (4) with respect to the elements of matrices $R^{1i}, i = 1 \dots k$;
- calculate solution (3).

Taylor Series Expansion

We have implement libraries for automatically expansion of a nonlinear function to corresponded Taylor series up to the any order. The function may be a composition of such elementary functions as $\sin x, \cos x, \tan x, e^x, \sqrt{x}, \ln x$, and operators $+, -, *, /$.

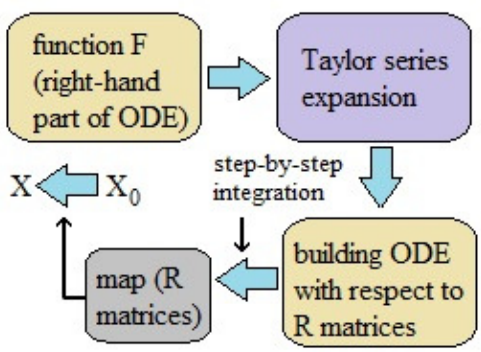

Figure 2: Map building of ODE.

In Fig. 1 developed command window is shown. For Taylor series representation symbolic polinoms are used.

SIMULATION OF ELECTROSTATIC STORAGE RING

Electrostatic storage ring consist of elements with different electric field distribution. In this research quadrupole lenses, cylindrical deflectors and drifts are used. The orbital motion and spin dynamics of the particle are described in [2]. Using such equations and library for Taylor series expansion that described above it is possible to build matrix form for each lattice element (see Fig. 3).

This software allows to build computational model and consist of pre build components. Each of this components can be added to the design area by drag and drop events.

Building of the Resulting Map

After maps are built for each elements of lattice it is possible to evaluate map for one tune of whole lattice. Such map is constructed using serial concatenation procedure. Imaging we have two numerical serial maps

$$X_1 = \sum_{k=0}^{k_1} R_1^{1k}(t) X_0^{[k]},$$

$$X_2 = \sum_{k=0}^{k_2} R_2^{1k}(t) X_1^{[k]}.$$

Substituting X_1 to the equation for X_2 we obtain

$$X_2 = \sum_{k=0}^{k_1 \cdot k_2} \tilde{R}_2^{1k}(t) X_0^{[k]}.$$

As you can see the resulting map has order $k_1 \cdot k_2$. But we can use terms of order not higher than $max(k_1, k_2)$, because other members are errors in the form of our initial in Taylor series expansions.

ISBN 978-3-95450-116-8

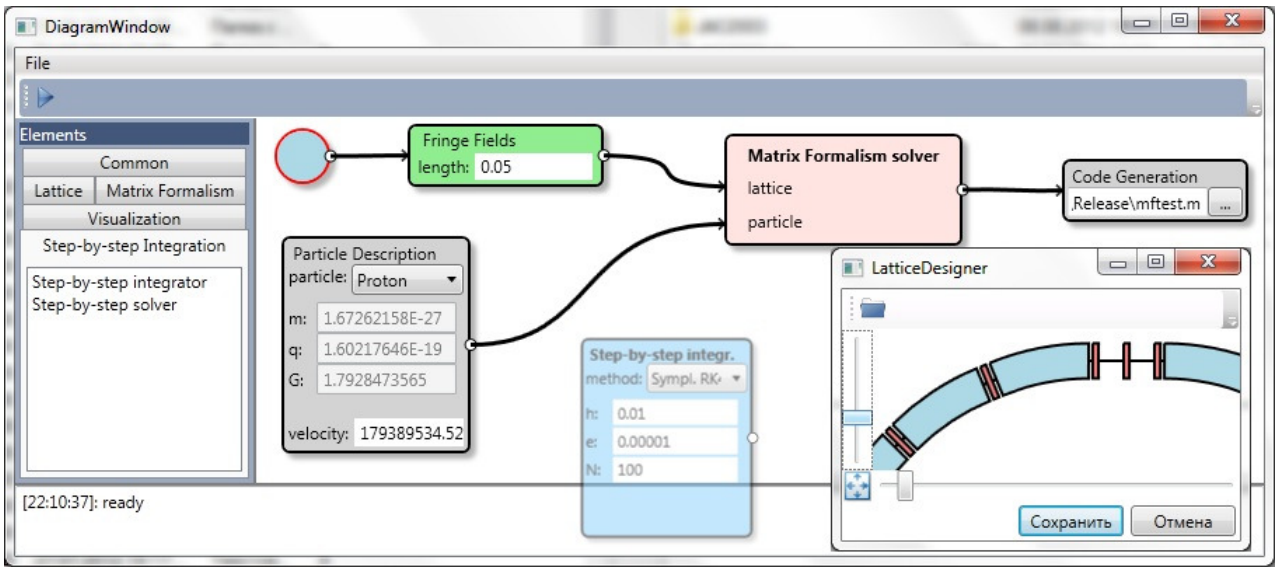

Figure 3: GUI environment for map building and code generation.

Evaluation of the Reference Orbit

Reference orbit of the map can be found from the equation (3). It is a nonlinear equation that can be solved by a numerical method. But in linear case we have:

$$X_{ref} = (E - R^{11})^{-1} R^{10}. \tag{8}$$

Symplectification Sample

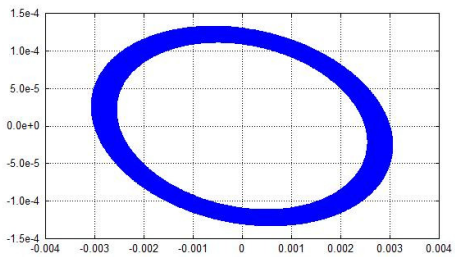

Figure 4: Map without symplectification.

Using a symplectic integration algorithm to build a map does not guarantee the symplectic map at all. For example, after 100000 turns becomes significant violation of the symplectic condition (see Fig. 4). To solve this problem we can increase the accuracy of step-by-step integration. But more accurate and efficient solution is map symplectification that described in this paper. In Fig. 5 a phase plane after additional symplectification is shown.

CONCLUSION

Matrix formalism is a high performance approach for ordinary differential equations solving. It allows present a solution as numerical matrix multiplication. This implies the possibility of implementation of the method in parallel codes [3, 4]. The result of the research is software for solving ODE in matrix form up to the neccessary order of nonlinearity. Direction of further development is the realization of matrix formalism in a symbolic form.

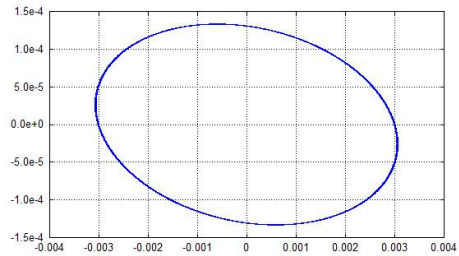

Figure 5: Map symplectification.

ACKNOWLEDGMENTS

The research is a part of JEDI[1] collaboration. The author would like to thank Yu. Senichev for advices and remarks and D. Zyuzin for comparative calculations on COSY Infinity.

REFERENCES

[1] S. Andrianov, "The Convergence and Accuracy of the Matrix Formalism Approximation," TUSCC2, these proceedings.

[2] A. Ivanov, "Particle tracking in electrostatic fields with energy conservation," WEP06, these proceedings.

[3] S. Andrianov, "Role of parallel and distributed computing in beam physics," Nuclear Instruments and Methods, 2004. Vol. 519. P. 37–41.

[4] N. Kulabukhova, "GPGPU Implementation of Matrix Formalism for Beam Dynamics Simulation," MOSDC2, these proceedings.

[1] Juelich Electric Dipole Moment Invenstigations, Spokes-persons: A. Lehrach, J. Pretz, and F. Rathmann

ISBN 978-3-95450-116-8