Saint-Petersburg State University
Faculty of Applied Mathematics and Control Processes

Symplectic integrator of the spin-orbital motion in the matrix representation

Andrei Ivanov

St.Petersburg State University, Russia

November 27, 2012

Outline

- Our previous research.
- Matrix integration of ordinary differential equations (ODEs):
  - map building;
  - map symplectification.
- Code MODE demonstration (Matrix integration of Ordinary Differential Equations).
- Examining of fringe fields.

Our previous research

Beam dynamics is decribed by ODEs and numerical methods are usually used for it solving.

- Step-by-step integration approaches:
  - high precision and flexibility, but low performance (random and systematic errors can be easy taken into account, tracking for any field distribution, etc.)
- Map building algorithms:
  - high performance methods, but reduced flexibility (may be difficult to use in non-periodic systems, with length and time dependences, e.g. RFE/RFB).

The aim

To develop different approaches and techniques to ensure the correctness of simulation.

Our previous research

The conditions on the solutions from the point of view of physics are

- symplecticity;
- energy conservation (especially in electric fields).

The result of our previous research

- symplectic step-by-step integration algorithm based on a 4th order Runge-Kutta method is developed;
- condition of the energy conservation is satisfied by the mathematical model of particle motion;

The research shows good coinsidence between step-by-step integration code and COSY Infinity program in spin-orbital motion.

The key idea is

- to develop a new computational program that implemented well known idea of Taylor maps;
- to be sure in mathematical model and its realisation via computer tools (what exactly reference orbit means in our sumulation, which coordinate system we use, etc.)
- to apply the approach to EDM investigation:
  - adding new physical elements;
  - long-term evolution;
  - beam simulation.

Matrix Formalism

is an integration method based on map building in 2-dim matrix form

$$\frac{d}{dt}X = F(t, X); \qquad P^{1k}(t) = \frac{1}{(k)!}\frac{\partial^k F(t, X_0)}{\partial (X^{[k]})^T};$$

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

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

Matrix Formalism

is an integration method based on map building in 2-dim matrix form

$$\frac{d}{dt}X = F(t, X); \qquad P^{1k}(t) = \frac{1}{(k)!}\frac{\partial^k F(t, X_0)}{\partial (X^{[k]})^T};$$

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

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

Evaluation of elements of map

To build a map means to find the elements of it up to the necessary order. This task can be solved by several approaches, e.g.:

- DA technique:
  - COSY Infinity (FORTRAN);
  - Tech-X Corporation (C++ code)[1]
- matrix formalism:
  - analytical formulas for map elements derivation;
  - numerical map estimation.

Comparison of the approach

In each case (expect of analytical forms of map) map is founded by propagating of identity map through the system. Differences in implementation. In ideal case the resulting map must be the same but in different notations.

[1] J. Cary, S. Shasharina. Efficient differential algebra computations. Proceedings of the 1999 Particle Accelerator Conference, New York, 1999. P. 377–381.

Example

One dimensional sextupole:

$$\frac{dx}{dt} = kx^2.$$

Let's write solution as a Taylor series with unknown coefficients:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$

and derive ODEs for $a, b, c, \dots$ coefficients as function of t:

$$\frac{dx}{dt} = \frac{da}{dt} + \frac{db}{dt}x_0 + \frac{dc}{dt}x_0^2 + \frac{de}{dt}x_0^3 + \dots,$$
$$\frac{dx}{dt} = k(a + bx_0 + cx_0^2 + ex_0^3 + \dots)^2.$$

Combinig like terms we can obtain:

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \ldots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \ldots,$$

$\frac{da}{dt} = ka^2,$ $\quad \frac{da}{dt} = ka^2, a(0) = 0, \quad a(t) = 0;$

$\frac{db}{dt} = 2kab,$ $\quad \frac{db}{dt} = 0, b(0) = 1, \quad b(t) = 1;$

$\frac{dc}{dt} = k(2ac + b^2),$ $\quad \frac{dc}{dt} = k, c(0) = 1, \quad c(t) = kt;$

$\frac{de}{dt} = k(2ad + 2bc),$ $\quad \frac{de}{dt} = 2k^2t, c(0) = 1, \quad e(t) = k^2t^2;$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \ldots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$$\frac{da}{dt} = ka^2, \qquad \frac{da}{dt} = ka^2, a(0) = 0, \qquad a(t) = 0;$$

$$\frac{db}{dt} = 2kab, \qquad \frac{db}{dt} = 0, b(0) = 1, \qquad b(t) = 1;$$

$$\frac{dc}{dt} = k(2ac + b^2), \qquad \frac{dc}{dt} = k, c(0) = 1, \qquad c(t) = kt;$$

$$\frac{de}{dt} = k(2ad + 2bc), \qquad \frac{de}{dt} = 2k^2t, c(0) = 1, \qquad e(t) = k^2t^2;$$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$$\frac{da}{dt} = ka^2, \qquad \frac{da}{dt} = ka^2, a(0) = 0, \qquad a(t) = 0;$$

$$\frac{db}{dt} = 2kab, \qquad \frac{db}{dt} = 0, b(0) = 1, \qquad b(t) = 1;$$

$$\frac{dc}{dt} = k(2ac + b^2), \qquad \frac{dc}{dt} = k, c(0) = 1, \qquad c(t) = kt;$$

$$\frac{de}{dt} = k(2ad + 2bc), \qquad \frac{de}{dt} = 2k^2t, c(0) = 1, \qquad e(t) = k^2t^2;$$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$$\frac{da}{dt} = ka^2, \qquad \frac{da}{dt} = ka^2, a(0) = 0, \qquad a(t) = 0;$$

$$\frac{db}{dt} = 2kab, \qquad \frac{db}{dt} = 0, b(0) = 1, \qquad b(t) = 1;$$

$$\frac{dc}{dt} = k(2ac + b^2), \qquad \frac{dc}{dt} = k, c(0) = 1, \qquad c(t) = kt;$$

$$\frac{de}{dt} = k(2ad + 2bc), \qquad \frac{de}{dt} = 2k^2t, c(0) = 1, \qquad e(t) = k^2t^2;$$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$\frac{da}{dt} = ka^2,$ $\quad \frac{da}{dt} = ka^2, a(0) = 0,$ $\quad a(t) = 0;$

$\frac{db}{dt} = 2kab,$ $\quad \frac{db}{dt} = 0, b(0) = 1,$ $\quad b(t) = 1;$

$\frac{dc}{dt} = k(2ac + b^2),$ $\quad \frac{dc}{dt} = k, c(0) = 1,$ $\quad c(t) = kt;$

$\frac{de}{dt} = k(2ad + 2bc),$ $\quad \frac{de}{dt} = 2k^2t, c(0) = 1,$ $\quad e(t) = k^2t^2;$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$\frac{da}{dt} = ka^2,$ $\frac{da}{dt} = ka^2, a(0) = 0,$ $a(t) = 0;$

$\frac{db}{dt} = 2kab,$ $\frac{db}{dt} = 0, b(0) = 1,$ $b(t) = 1;$

$\frac{dc}{dt} = k(2ac + b^2),$ $\frac{dc}{dt} = k, c(0) = 1,$ $c(t) = kt;$

$\frac{de}{dt} = k(2ad + 2bc),$ $\frac{de}{dt} = 2k^2t, c(0) = 1,$ $e(t) = k^2t^2;$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$\frac{da}{dt} = ka^2,$ $\frac{da}{dt} = ka^2, a(0) = 0,$ $a(t) = 0;$

$\frac{db}{dt} = 2kab,$ $\frac{db}{dt} = 0, b(0) = 1,$ $b(t) = 1;$

$\frac{dc}{dt} = k(2ac + b^2),$ $\frac{dc}{dt} = k, c(0) = 1,$ $c(t) = kt;$

$\frac{de}{dt} = k(2ad + 2bc),$ $\frac{de}{dt} = 2k^2 t, c(0) = 1,$ $e(t) = k^2 t^2;$

$$x(t) = x_0 + ktx_0^2 + k^2 t^2 x_0^3 + \dots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \ldots,$$

$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \ldots,$$

$$\frac{da}{dt} = ka^2, \qquad \frac{da}{dt} = ka^2, a(0) = 0, \qquad a(t) = 0;$$

$$\frac{db}{dt} = 2kab, \qquad \frac{db}{dt} = 0, b(0) = 1, \qquad b(t) = 1;$$

$$\frac{dc}{dt} = k(2ac + b^2), \qquad \frac{dc}{dt} = k, c(0) = 1, \qquad c(t) = kt;$$

$$\frac{de}{dt} = k(2ad + 2bc), \qquad \frac{de}{dt} = 2k^2 t, c(0) = 1, \qquad e(t) = k^2 t^2;$$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \ldots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \ldots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \ldots,$$

$$\frac{da}{dt} = ka^2, \qquad \frac{da}{dt} = ka^2, a(0) = 0, \qquad a(t) = 0;$$

$$\frac{db}{dt} = 2kab, \qquad \frac{db}{dt} = 0, b(0) = 1, \qquad b(t) = 1;$$

$$\frac{dc}{dt} = k(2ac + b^2), \qquad \frac{dc}{dt} = k, c(0) = 1, \qquad c(t) = kt;$$

$$\frac{de}{dt} = k(2ad + 2bc), \qquad \frac{de}{dt} = 2k^2t, c(0) = 1, \qquad e(t) = k^2t^2;$$

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \ldots$$

Example

Taking into account identity of the map in the initial time:

$$x = a + bx_0 + cx_0^2 + ex_0^3 + \dots,$$
$$x(0) = x_0 = 0 + 1x_0 + 0x_0^2 + 0x_0^3 + \dots,$$

$$\frac{da}{dt} = ka^2, \qquad \frac{da}{dt} = ka^2, a(0) = 0, \qquad a(t) = 0;$$

$$\frac{db}{dt} = 2kab, \qquad \frac{db}{dt} = 0, b(0) = 1, \qquad b(t) = 1;$$

$$\frac{dc}{dt} = k(2ac + b^2), \qquad \frac{dc}{dt} = k, c(0) = 1, \qquad c(t) = kt;$$

$$\frac{de}{dt} = k(2ad + 2bc), \qquad \frac{de}{dt} = 2k^2 t, c(0) = 1, \qquad e(t) = k^2 t^2;$$

$$x(t) = x_0 + ktx_0^2 + k^2 t^2 x_0^3 + \dots$$

Example

$$\frac{dx}{dt} = kx^2, x(0) = x_0.$$

The solution obtained by the approach is

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

The precise solution is

$$x(t) = \frac{x_0}{1 - ktx_0}.$$

Considering formula $\frac{1}{1-q} = 1 + q + q^2 + q^3 + \dots$ the precise solution can be pesented (within the region of convergence) as

$$x(t) = x_0 + ktx_0^2 + k^2t^2x_0^3 + \dots$$

Symplectic of the Jacobian matrix

If system of ODE and map is presented throw canonical variables $X = (P, Q)^T$:

$$\frac{d}{dt}\begin{pmatrix} P \\ Q \end{pmatrix} = -J\frac{\partial H}{\partial X^T}, \qquad \begin{pmatrix} P \\ Q \end{pmatrix} = R \circ \begin{pmatrix} P_0 \\ Q_0 \end{pmatrix}, J = \begin{pmatrix} 0 & E \\ -E & 0 \end{pmatrix},$$

then the symplectic condition for operator R is

$$M^T JM = J, \forall X_0.$$

In case of matrix map this condition provides set of links between elements of matrices. For instance:

Example

$$\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} \begin{pmatrix} p_0^2 \\ p_0 q_0 \\ q_0^2 \end{pmatrix}$$

The Jacobian matrix is equal to

$$M = \begin{pmatrix} a_{11} + 2b_{11}p_0 + b_{12}q_0 & a_{12} + 2b_{13}q_0 + b_{12}p_0 \\ a_{21} + 2b_{21}p_0 + b_{22}q_0 & a_{22} + 2b_{23}q_0 + b_{22}p_0 \end{pmatrix}.$$

The condition $M^T JM = J, \forall X_0$ provides system of equation:

$$\begin{aligned}
&a_{11}a_{22} - a_{12}a_{21} = 1, \\
&b_{11}b_{23} - b_{13}b_{21} = 0, \\
&b_{11}b_{22} - b_{12}b_{21} = 0, \\
&b_{12}b_{23} - b_{13}b_{22} = 0, \\
&2a_{11}b_{23} + a_{22}b_{12} - a_{12}b_{22} - 2a_{21}b_{13} = 0, \\
&2a_{22}b_{11} + a_{11}b_{22} - a_{21}b_{12} - 2a_{12}b_{21} = 0.
\end{aligned}$$

- Presenting of ODE system in matrix form with respect to the state powers up to the necessary order:

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

- Deriving the ODE system of map elements $\frac{dR^{1j}(t)}{dt} = \ldots$ end evaluate matrices R^{1j} by a numerical step-by-step integration (we use symplectic RK method of 4th order):

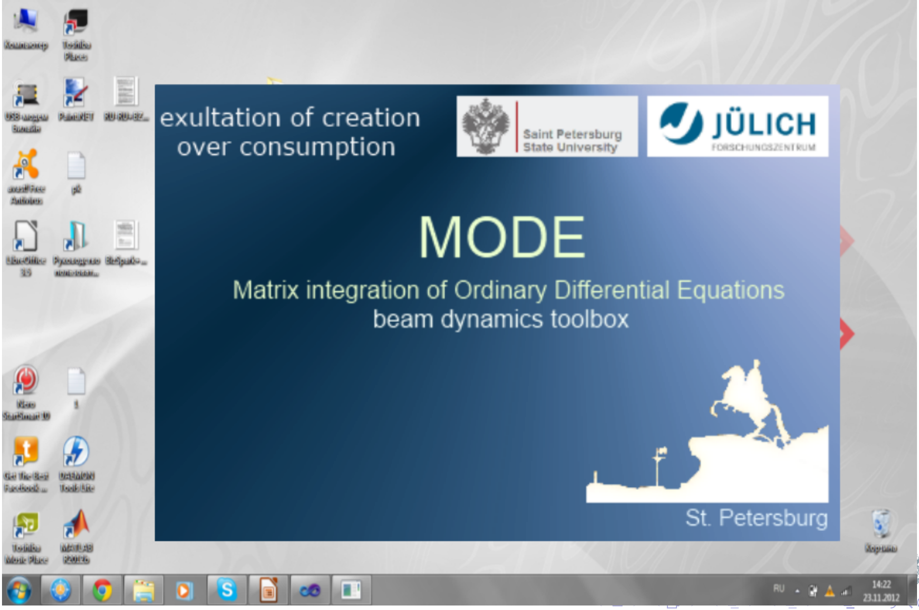
$$X = \sum_{i=0}^{k} R^{1i}(t)X_0^{[i]}.$$

Background, e.g:

S. Andrianov. The convergence and accuracy of the matrix formalism approximation. Proceedings of ICAP2012, Rostock, Germany. P. 93–95.
S. Andrianov. Symplectification of truncated maps for Hamiltonian systems. Mathematics and Computers in Simulation 57 (2001). P. 139–145.

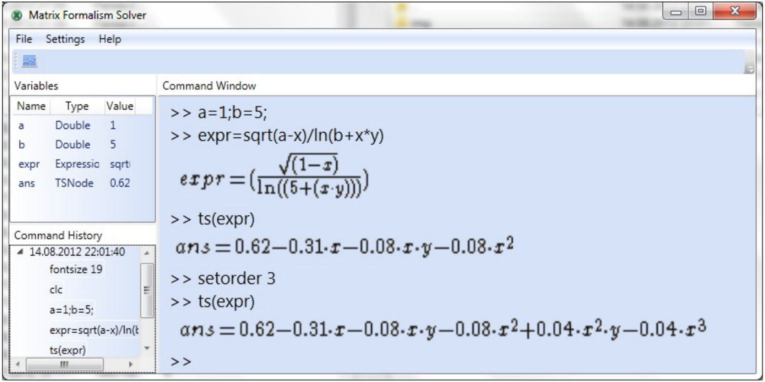
Tasks for software development

- Parsing of the analyical formulas to automatically expansion it the Teylor series:
  - subrouting for symbolic computation
- Building of ODE system for map and solving it by a numerical approach:
  - subrouting for Kronecker powers computation
- Automatically generation of symplectic condition with respect to the map elements
- For beam dynamics: preparing equation of spin-orbital motion in analytical form, that allows generate map by any field distribution (both electric and magnetic), that described in an analytical form.


exultation of creation
over consumption
Saint Petersburg
State University
JÜLICH
FORSCHUNGSZENTRUM
MODE
Matrix integration of Ordinary Differential Equations
beam dynamics toolbox
St. Petersburg

Taylor series expansion

The libraries for automatically expansion of a nonlinear function to corresponded Taylor series up to the necessary order have been implemented.



The function may be a composition of elementary functions ($sin(x), tan(x), exp(x), sqrt(x), ln(x) \ldots$) and operators $+, -, *, /$.

9 dimensional state vector for a particle

$$\begin{pmatrix} x \\ y \\ x' \\ y' \\ S_x \\ S_y \\ S_s \\ \delta k \\ \Delta t \end{pmatrix} = R^{10} + R^{12} \begin{pmatrix} x_0 \\ y_0 \\ x'_0 \\ y'_0 \\ S_{x0} \\ S_{y0} \\ S_{s0} \\ \delta k_0 \\ \Delta t_0 \end{pmatrix} + R^{12} \begin{pmatrix} x_0^2 \\ x_0 y_0 \\ x_0 x'_0 \\ x_0 y'_0 \\ x_0 S_{x0} \\ . \\ . \\ . \\ \Delta t_0^2 \end{pmatrix} + \dots$$

Coordinate system

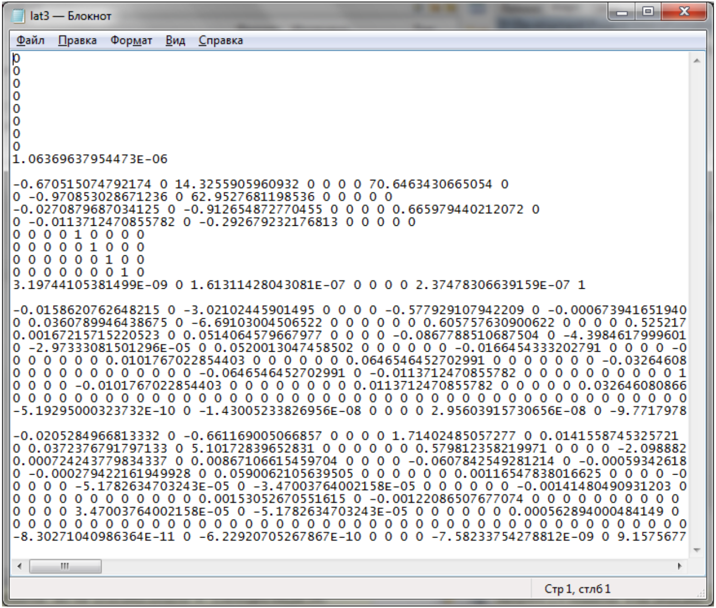
Particles moove along **designed orbit**, not along reference orbit. Simulation runs in physical elements without lattice realigned and "jumps" in a space. Reference orbit can be found after matrix map is builded.

$R^{10} =$

$R^{11} =$

$R^{12} =$

$R^{13} =$

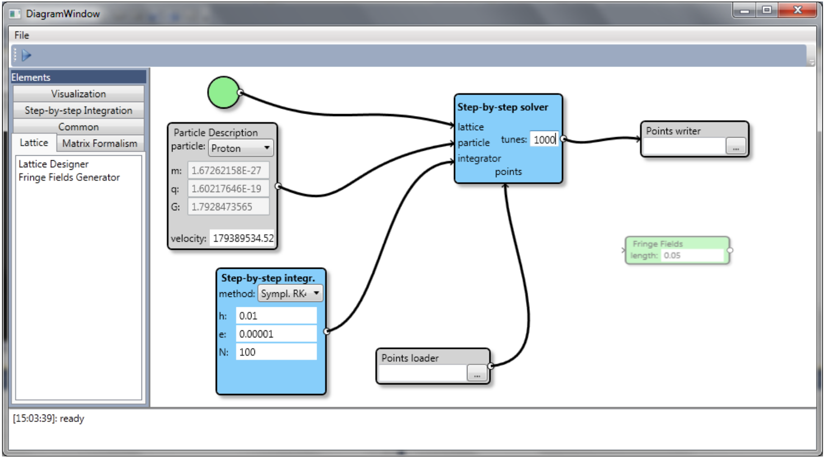


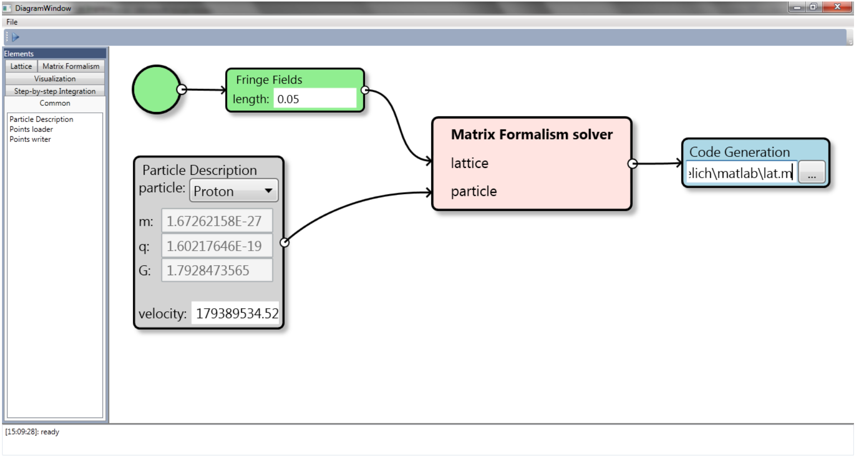
lat3 — Блокнот

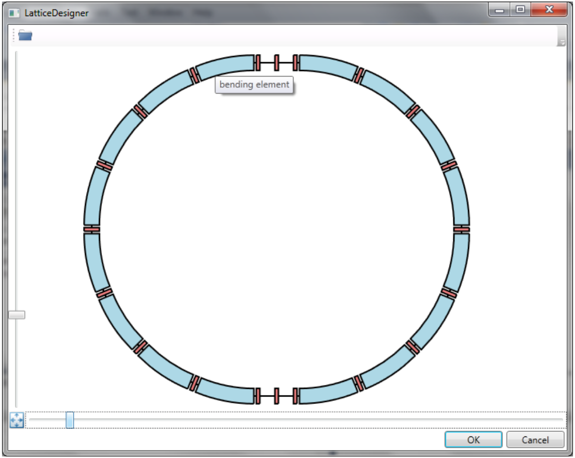
Файл Правка Формат Вид Справка

```
p
0
0
0
0
0
0
0
0
1.06369637954473E-06

-0.670515074792174 0 14.3255905960932 0 0 0 0 70.6463430665054 0
0 -0.970853028671236 0 62.9527681198536 0 0 0 0 0
-0.0270879687034125 0 -0.912654872770455 0 0 0 0 0.665979440212072 0
0 -0.0113712470855782 0 -0.292679232176813 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
3.19744105381499E-09 0 1.61311428043081E-07 0 0 0 0 2.37478306639159E-07 1

-0.0158620762648215 0 -3.02102445901495 0 0 0 0 -0.577929107942209 0 -0.000673941651940
0 0.0360789946438675 0 -6.69103004506522 0 0 0 0 0 0 0.605757630900622 0 0 0 0 0.525217
0.00167215715220523 0 0.0514064579667977 0 0 0 0 -0.0867788510687504 0 -4.3984617999601
0 -2.97333081501296E-05 0 0.0520013047458502 0 0 0 0 0 0 -0.0166454333202791 0 0 0 0 -0
0 0 0 0 0 0 0.0101767022854403 0 0 0 0 0 0 0.0646546452702991 0 0 0 0 0 0 0 -0.03264608
0 0 0 0 0 0 0 0 0 0 0 0 -0.0646546452702991 0 -0.0113712470855782 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 -0.0101767022854403 0 0 0 0 0 0 0 0 0.0113712470855782 0 0 0 0 0 0.032646080866
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-5.19295000323732E-10 0 -1.43005233826956E-08 0 0 0 0 2.95603915730656E-08 0 -9.7717978

-0.0205284966813332 0 -0.661169005066857 0 0 0 0 1.71402485057277 0 0.0141558745325721
0 0.0372376791797133 0 5.10172839652831 0 0 0 0 0 0 0.579812358219971 0 0 0 0 -2.098882
0.000724243779834337 0 0.00867106615459704 0 0 0 0 -0.0607842549281214 0 -0.00059342618
0 -0.000279422161949928 0 0.0590062105639505 0 0 0 0 0 0 0.00116547838016625 0 0 0 0 -0
0 0 0 0 -5.1782634703243E-05 0 -3.47003764002158E-05 0 0 0 0 0 0 -0.00141480490931203 0
0 0 0 0 0 0 0 0 0 0 0 0 0.00153052670551615 0 -0.00122086507677074 0 0 0 0 0 0 0 0 0 0
0 0 0 0 3.47003764002158E-05 0 -5.1782634703243E-05 0 0 0 0 0 0 0.000562894000484149 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-8.30271040986364E-11 0 -6.22920705267867E-10 0 0 0 0 -7.58233754278812E-09 0 9.1575677
```

Стр 1, стлб 1


DiagramWindow
File
Elements
Visualization
Step-by-step Integration
Common
Lattice
Matrix Formalism
Lattice Designer
Fringe Fields Generator
Step-by-step solver
lattice
particle
tunes: 1000
integrator
points
Points writer
Particle Description
particle: Proton
m: 1.67262158E-27
q: 1.60217646E-19
G: 1.7928473565
velocity: 179389534.52
Fringe Fields
length: 0.05
Step-by-step integr.
method: Sympl. RK4
h: 0.01
e: 0.00001
N: 100
Points loader
[15:03:39]: ready


DiagramWindow
File
Elements
Lattice
Matrix Formalism
Visualization
Step-by-step Integration
Common
Particle Description
Points loader
Points writer
Fringe Fields
length: 0.05
Matrix Formalism solver
lattice
particle
Code Generation
lich\matlab\lat.m
Particle Description
particle: Proton
m: 1.67262158E-27
q: 1.60217646E-19
G: 1.7928473565
velocity: 179389534.52
[15:09:28]: ready


LatticeDesigner
bending element
OK
Cancel

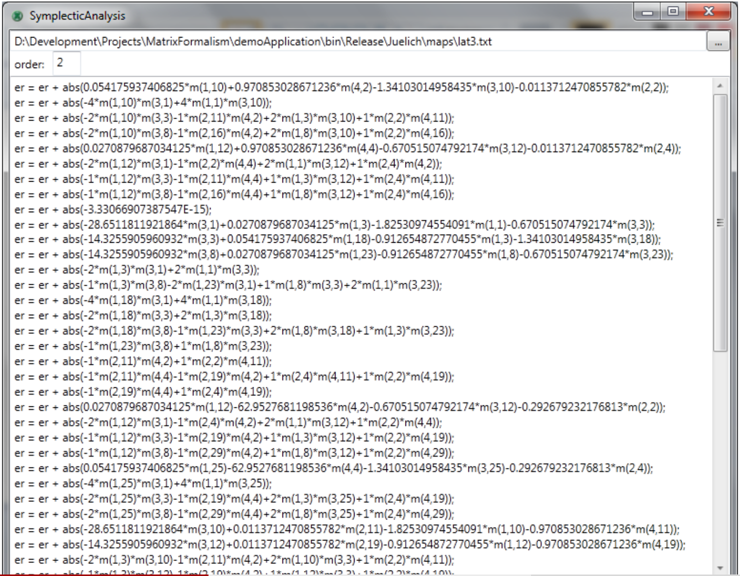
Symplectic conditions

lat3 — Блокнот

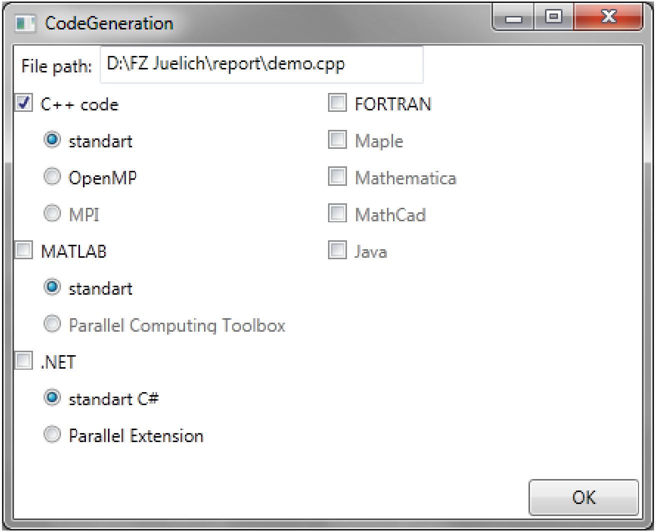
Файл Правка Формат Вид Справка

```
0
0
0
0
0
0
0
0
1.06369637954473E-06

-0.670515074792174 0 14.3255905960932 0 0 0 0 70.6463430665054 0
0 -0.970853028671236 0 62.9527681198536 0 0 0 0 0
-0.0270879687034125 0 -0.912654872770455 0 0 0 0 0.665979440212072 0
0 -0.0113712470855782 0 -0.292679232176813 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
3.19744105381499E-09 0 1.61311428043081E-07 0 0 0 0 2.37478306639159E-07 1

-0.0158620762648215 0 -3.02102445901495 0 0 0 0 -0.577929107942209 0 -0.000673941651940
0 0.0360789946438675 0 -6.69103004506522 0 0 0 0 0 0 0.605757630900622 0 0 0 0 0.525217
0.00167215715220523 0 0.0514064579667977 0 0 0 0 -0.0867788510687504 0 -4.3984617999601
0 -2.97333081501296E-05 0 0.0520013047458502 0 0 0 0 0 0 -0.0166454333202791 0 0 0 0 -0
0 0 0 0 0 0 0.0101767022854403 0 0 0 0 0 0 0.0646546452702991 0 0 0 0 0 0 0 -0.03264608
0 0 0 0 0 0 0 0 0 0 0 -0.0646546452702991 0 -0.0113712470855782 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 -0.0101767022854403 0 0 0 0 0 0 0 0 0.0113712470855782 0 0 0 0 0 0.032646080866
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-5.19295000323732E-10 0 -1.43005233826956E-08 0 0 0 0 2.95603915730656E-08 0 -9.7717978

-0.0205284966813332 0 -0.661169005066857 0 0 0 0 1.71402485057277 0 0.0141558745325721
0 0.0372376791797133 0 5.10172839652831 0 0 0 0 0 0 0.579812358219971 0 0 0 0 -2.098882
0.000724243779834337 0 0.00867106615459704 0 0 0 0 -0.0607842549281214 0 -0.00059342618
0 -0.000279422161949928 0 0.0590062105639505 0 0 0 0 0 0 0.00116547838016625 0 0 0 0 -0
0 0 0 0 -5.1782634703243E-05 0 -3.47003764002158E-05 0 0 0 0 0 0 -0.00141480490931203 0
0 0 0 0 0 0 0 0 0 0 0 0.00153052670551615 0 -0.00122086507677074 0 0 0 0 0 0 0 0 0 0
0 0 0 0 3.47003764002158E-05 0 -5.1782634703243E-05 0 0 0 0 0 0 0.000562894000484149 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-8.30271040986364E-11 0 -6.22920705267867E-10 0 0 0 0 -7.58233754278812E-09 0 9.1575677
```

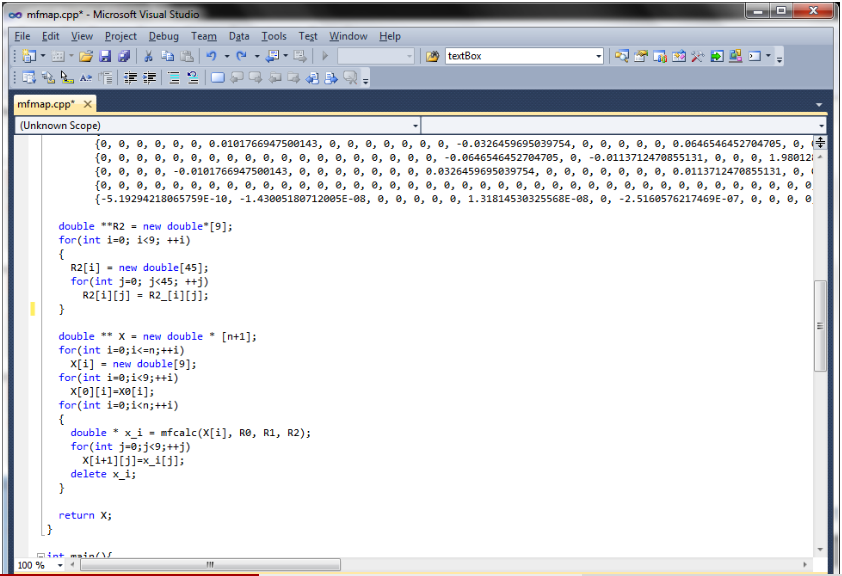
Symplectic conditions



Code generation



Code generation



Code generation

mfmap3.m* (C:/Andrei/Development/Projects/MatrixFormalism/demoApplication/bin/Release/test/) - FreeMat v4.0 Editor

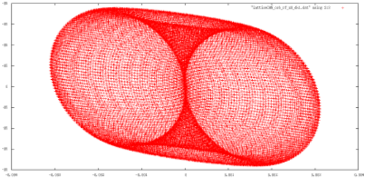
File Edit Tools Debug Help

mfmap3.m*

```
%code generation
%mapping function
function X = mfmap(X0, n)
%map generation:
    R0=[0;0;0;0;0;0;0;0;1.06369637954473E-06;];
    R1=[-0.670513915242833 14.3255631639029 0 0 0 0 0 35.3231449743497 0;
        -0.0270879171669592 -0.912653250594104 0 0 0 0 0 0.332988869053756 0;
        0 0 -0.970853028677094 62.9527681188037 0 0 0 0 0;
        0 0 -0.0113712470855131 -0.292679232191561 0 0 0 0 0;
        0 0 0 0 1 0 0 0 0;
        0 0 0 0 0 1 0 0 0;
        0 0 0 0 0 0 1 0 0;
        0 0 0 0 0 0 0 1 0;
        3.19743598523805E-09 1.61311298409876E-07 0 0 0 0 0 -2.80213794351364E-07 1;];
    R2=...      R3=...
    X = zeros(length(X0), n+1);
    X(:,1) = X0;
    for i=1:n
        X(:, i+1) = mfcalc(X(:,i), R0, R1, R2, R3);
    end
end
%one-tune solution
function X = mfcalc(X0, R0, R1, R2, R3)
%initial state:
    x0=X0(1);x1=X0(2);x2=X0(3);x3=X0(4);x4=X0(5);x5=X0(6);x6=X0(7);x7=X0(8);x8=X0(9);
    %calculation of kronecker pows:
    X1=[x0;x1;x2;x3;x4;x5;x6;x7;x8];
    X2=[x0*x0;x0*x1;x0*x2;x0*x3;x0*x4;x0*x5;x0*x6;x0*x7;x0*x8;x1*x1;x1*x2;x1*x3;x1*x4;x1*x5;x
    X3=[x0*x0*x0;x0*x0*x1;x0*x0*x2;x0*x0*x3;x0*x0*x4;x0*x0*x5;x0*x0*x6;x0*x0*x7;x0*x0*x8;x0*x
%solution:
    X=R0+R1*X1+R2*X2+R3*X3;
end
```

Spin-orbital motion in transverse plane (RF ON)

- COSY Infinity



- MODE (matrix map)

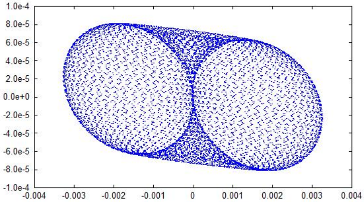
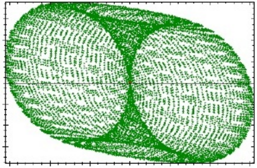


Table : SCT, sec

Case	MODE	RK 4	COSY Inf.
		RF = ON	
$\Delta x = 0mm$, $\Delta k/k = 10^{-4}$	7043	5260	7316
$\Delta x = 0mm$, $\Delta k/k = 310^{-4}$	862	639	774

- MODE (RK 4)



Computational code performnce: no parallel, Intel Core i5

- C++ compiled code:
  - 1 particle, 10^6 turns – 3.2 sec
- Matlab:
  - 100 particle, 10^3 turns – 3,7 sec
  - 1000 particle, 10^3 turns – 36 sec
  - 10000 particle, 10^3 turns – 38 sec
  - 10000 particle, 10^4 turns – 397 sec

Comparison with COSY Infinity

In the current stage of development MODE can generate matrix map up to 6 order (there is no restriction of algorithm for any order). Map generation is slower than in COSY Infinity, but particle tracking is faster by executing of the compiled code that generated by the MODE. It is not necessary to have a special version of MODE software for parallel systems, just generate code in you favorite language and run it in any platform.

Fringe field model

Potential distribution can be described in form (M.Szilgyi, Electron and ion optics, Plenum Press, New York, 1988)

$$u(x,y,z) = \sum_{m=0}^{\infty}\sum_{k=0}^{\infty}\frac{(-1)^k m!(x^2+y^2)^k}{4^k k!(m+k)!}\times$$
$$\times\left(U_m^{2k}(z)\sum_{i=0}^{m}\frac{(-1)^i m!}{(2i)!(m-2i)!x^{m-2i}y^{2i}}+\right.$$
$$\left.+\,W_m^{2k}(z)\sum_{i=0}^{m}\frac{(-1)^i m!}{(2i+1)!(m-2i-1)!x^{m-2i-1}y^{2i+1}}\right).$$

In case of planar field

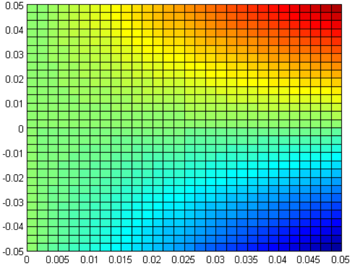
$$u(x,z) = \sum_{k=0}^{\infty}(-1)^k\left(\frac{U_0^{2k}(z)}{(2k)!}x^{2k} + \frac{U_1^{2k+1}(z)}{(2k)!}x^{2k+1}\right).$$

Fringe field model

Considering symmetry of field we can conclude:

$$u(x,z) = U_1(z)x - \frac{U_1''(z)}{6}x^3 - \ldots, U_1(z) = \frac{\partial u(x,z)}{\partial x}\,|_{x=0}\,.$$

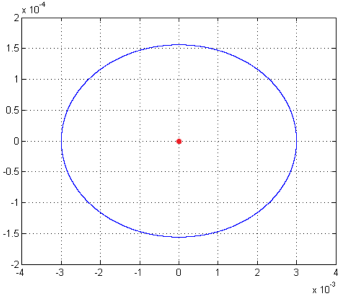
For example, for $U_1(z) = -E_0 k(z), k(z) = \frac{z}{L}$:

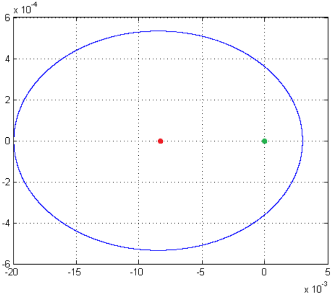


$$E_x = E_0 k(z) - \frac{E_0 x^2}{2} k''(z), \qquad E_z = E_0 k'(z) x - \frac{E_0 x^3}{6} k'''(z).$$

For $k(z) = z/L$ transverse planes in $x - x'$ space are

- Without fringe fields ($E_0 = 0$):
- With fringe field ($E_0 \neq 0$):





SCT, sec

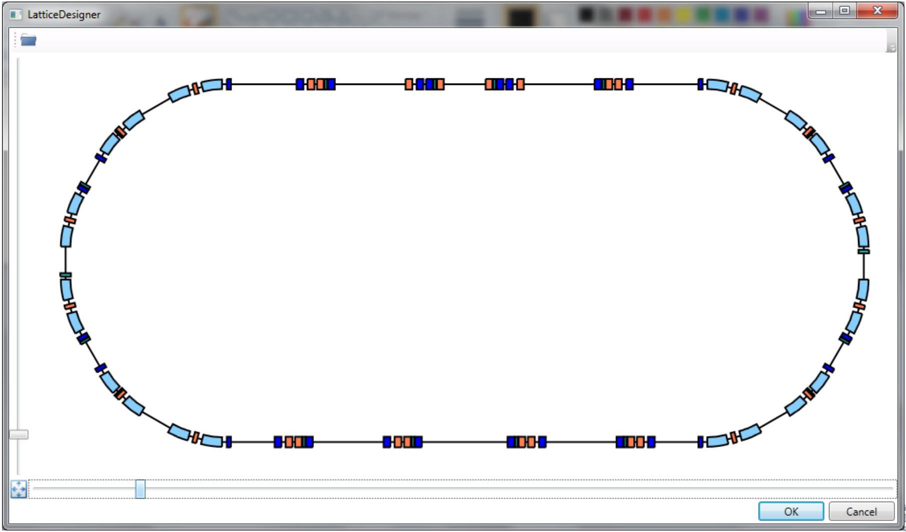
Fringe fields are considered as additional elements near cylindrical deflectors:

	Without fringe fields	With fringe fields
$\Delta x = 3mm$ $\delta k = 0$	654	668
$\Delta x = 3mm$ $\delta k = 3 \cdot 10^{-4}$	174	193
$\Delta x = 0mm$ $\delta k = 3 \cdot 10^{-4}$	862	1473
$\Delta x = 0mm$ $\delta k = 1 \cdot 10^{-4}$	7043	8469

Issue

SCT has the same order. Differences may be caused by different reference orbits and corresponded RF elements.

Magnet optics implementation



- EDM influence investigation:
  - including EDM to the BMT equation, precision problem (currently MODE supported either 15-16 (double) or 28-29 significant digits);
- beam dynamics simulation:
  - not particle tracking, but envelope analysis;
- parallelization on accelerators in St.Petersburg NVIDIA Tesla GPUs:
  - calculation based on matrix map can be easy parallelized.
- Introducing random and systematic errors in MODE simulation model (additional parameters in fields and physical elements).

Thank you for your attention