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

A. Ivanov, S. Andrianov

Matrix Formalism for long-term evolution of charged particle and spin dynamics in electrostatic fields

speaker:
Andrei Ivanov

2012

The aims are

- to derive the equations for numerical implementation of matrix formalism for ODE solving;
- to develop tool for Taylor series expansion and building solution of ODE in matrix form;
- to apply the approach to beam dynamics modeling;

Matrix Formalism

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

$$\frac{dX(t)}{dt} = F(t, X)$$

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

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

A solution can be found as a series with respect to the powers of initial point

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

An example

Orbital motion of a particle in cylindrical deflector can be described by *

$$x'' + \frac{2}{R_{eq}^2}x - \frac{1}{R_{eq}^3}x^2 = 0,$$
$$y'' = 0,$$

$$\frac{d}{ds}\begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} x' \\ -2/R_{eq}^2\, x + 1/R_{eq}^3\, x^2 \end{pmatrix}.$$

In matrix form we have

$$\frac{d}{ds}\begin{pmatrix} x \\ x' \end{pmatrix} = \sum_{\mathbf{k}=0}^{\infty} \mathbb{P}^{1k}(s)\begin{pmatrix} x \\ x' \end{pmatrix}^{[k]},$$

$$\mathbb{P}^{11} = \begin{pmatrix} 0 & 1 \\ -2/R_{eq}^2 & 0 \end{pmatrix},\ \mathbb{P}^{12} = \begin{pmatrix} 0 & 0 & 0 \\ 1/R_{eq}^3 & 0 & 0 \end{pmatrix},\ \mathbb{P}^{1k} = \mathbb{O},\ k = \overline{3,\infty}.$$

*** Senichev Yu., Moeller S.P. Beam Dynamics in Electrostatic Rings.**

An example

Following the algorithm for building matrices R *

$$\mathbf{X}(s) = \mathbb{R}^{11}(s)\,(x_0, x_0')^T + \mathbb{R}^{12}(s)\,\left(x_0^2, x_0 x_0', {x_0'}^2\right)^T$$

where

$$\mathbb{R}^{11}(s|0) = \begin{pmatrix} \cos(bs) & \frac{1}{b}\sin(bs) \\ -b\sin(bs) & \cos(bs) \end{pmatrix},$$

$$\mathbb{R}^{12}(s|0) = \begin{pmatrix} R^{12}_{11} & R^{12}_{12} & R^{12}_{13} \\ R^{12}_{21} & R^{12}_{22} & R^{12}_{23} \end{pmatrix},$$

$$R^{12}_{11} = \frac{b^2}{2^{3/2}}\left(-\frac{(\cos(2bs)-3)}{6b} - \frac{\cos(bs)}{3b}\right), \quad R^{12}_{21} = \frac{b^3}{2^{3/2}}\left(\frac{\sin(2bs)}{3b} + \frac{\sin(bs)}{3b}\right),$$

$$R^{12}_{12} = \frac{b}{2^{1/2}}\left(\frac{\sin(bs)}{3b} - \frac{\sin(2bs)}{6b}\right), \quad R^{12}_{22} = \frac{b^2}{2^{1/2}}\left(\frac{\cos(bs)}{3b} - \frac{\cos(2bs)}{3b}\right),$$

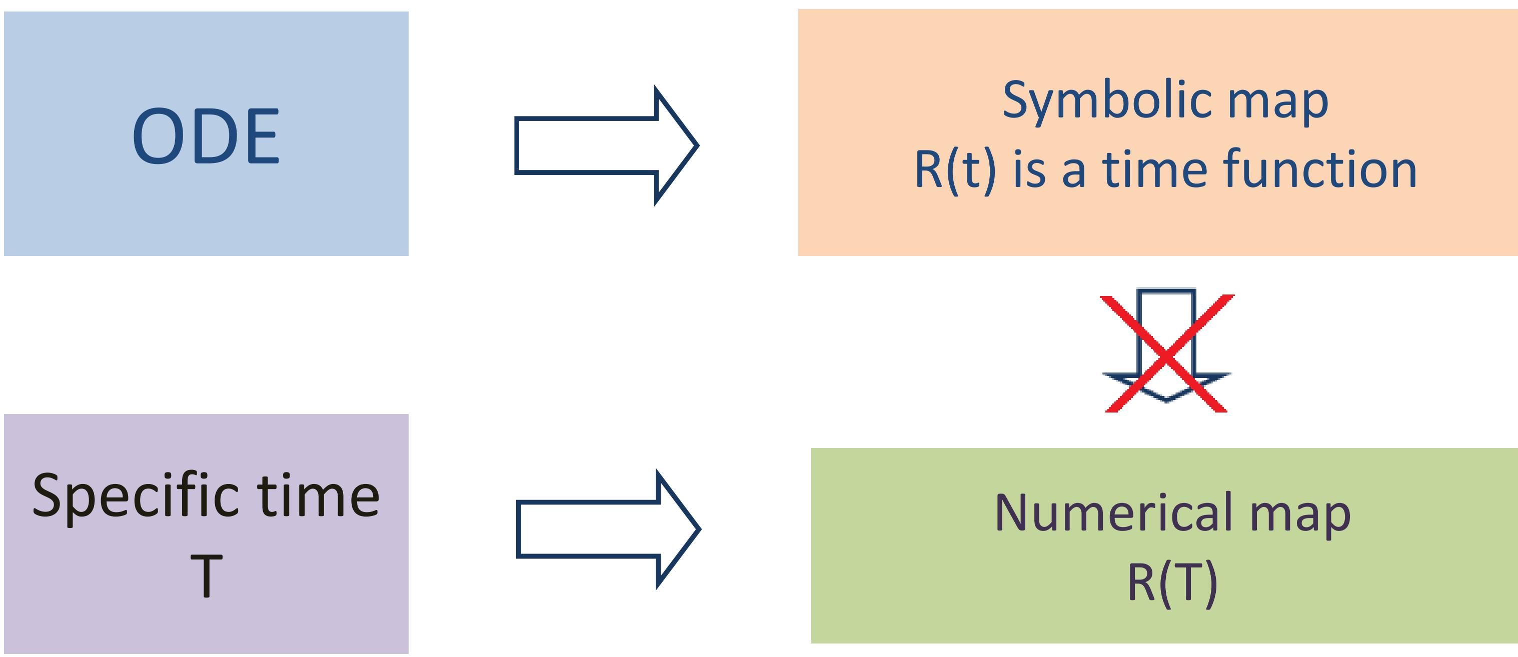
$$R^{12}_{13} = \frac{b^2}{2^{3/2}}\left(\frac{(\cos(2bs)+3)}{6b} - \frac{2\cos(bs)}{3b}\right), \quad R^{12}_{23} = \frac{b^2}{2^{3/2}}\left(\frac{2\sin(bs)}{3b} - \frac{\sin(2bs)}{3b}\right).$$

* Andrianov S. Dynamic modeling of beam control systems.

Numerical implementation

There are 2 ways for map building:

- symbolic mode;
- numerical mode.



Numerical implementation

According to the initial equation for system of ODE and its solution

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

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

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

By partial deriving of this equation we can obtain:

$$\frac{dR^{1k}(t)}{dt}=P^{1k}(t)\frac{\partial X^{[k]}}{\partial \left(X_0^{[k]}\right)^T} \quad (*)$$

Using a step-by-step integration method for solving equation (*) we can evaluate matrices R

Map building

- desciption of the equation of the motion and spin dynamics in analytical form (Newton-Lorentz and BMT equation);

- Taylor series expansion of these equation;

- buildin ODE with corresponds to the matrices R following to the algorithm that described above;

- evaluating elements of matrices R by a numerical integration;

Map building

$$S_x' = S_s/R + \frac{Q}{m_0c^2}\left(G + \frac{1}{1+\gamma}\right)((h_sE_x - x'E_s)S_s -$$
$$-(x'E_y - y'E_x)S_y),$$

Lattice elements

$$\Phi(x,y) = -V + \frac{2V}{\ln\frac{R_2}{R_1}}\ln\frac{r}{R_1}$$

Matrix Formalism Tools

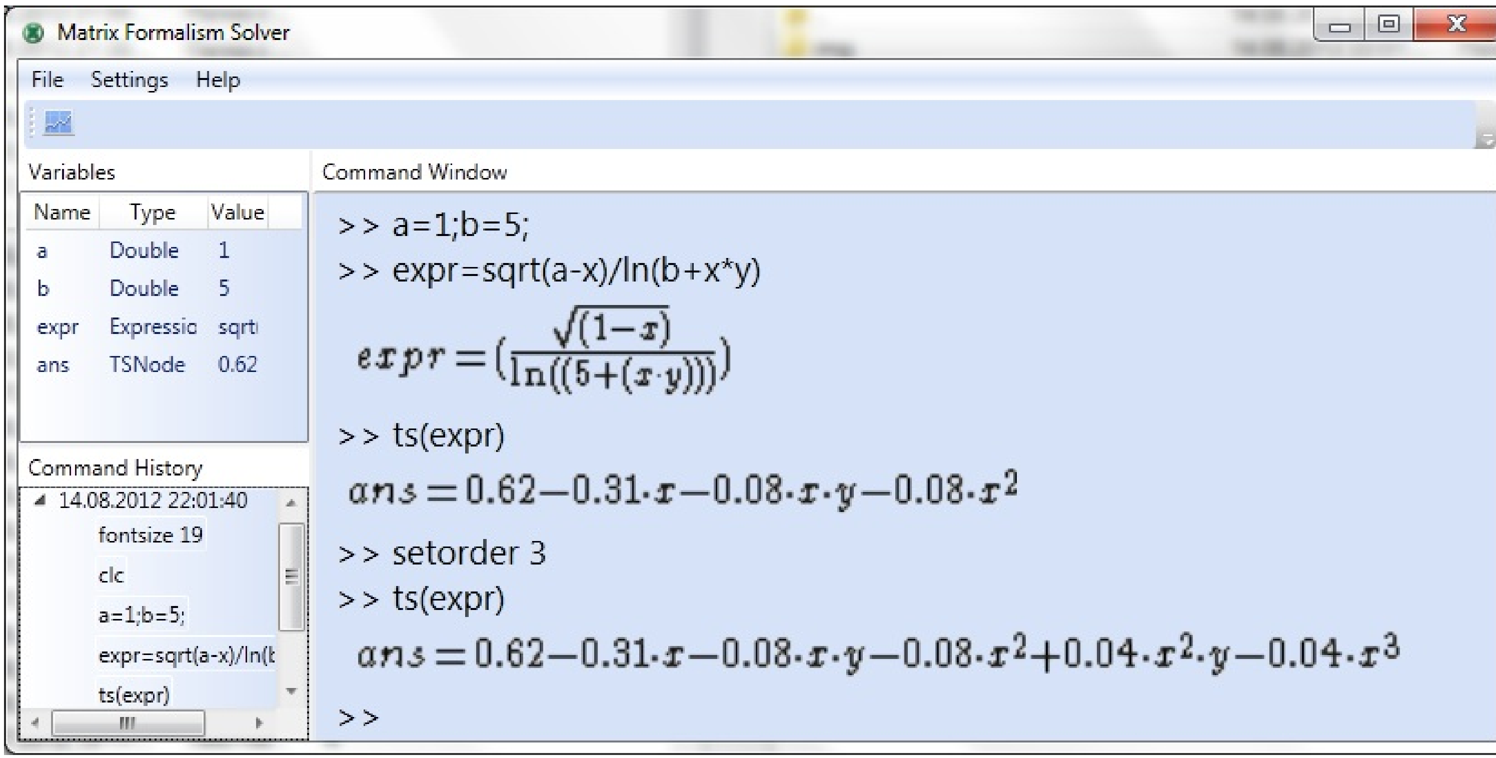
- Taylor series expansion;
- Building ODE's for matrices form;
- Numerical integration;

Data Base

quadrupole lenses, cylindrical deflectors, drifts, RF, etc

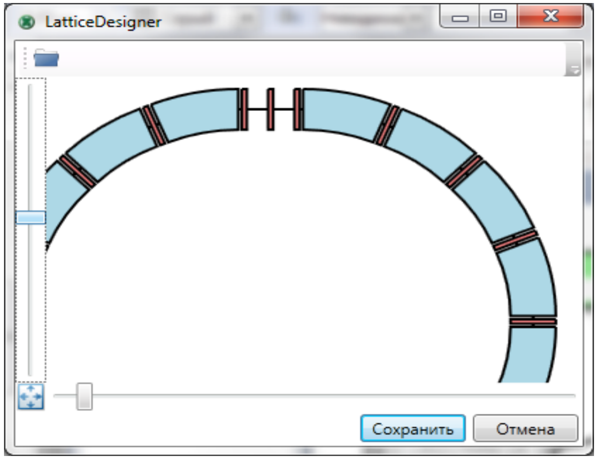
Taylor series expansion

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

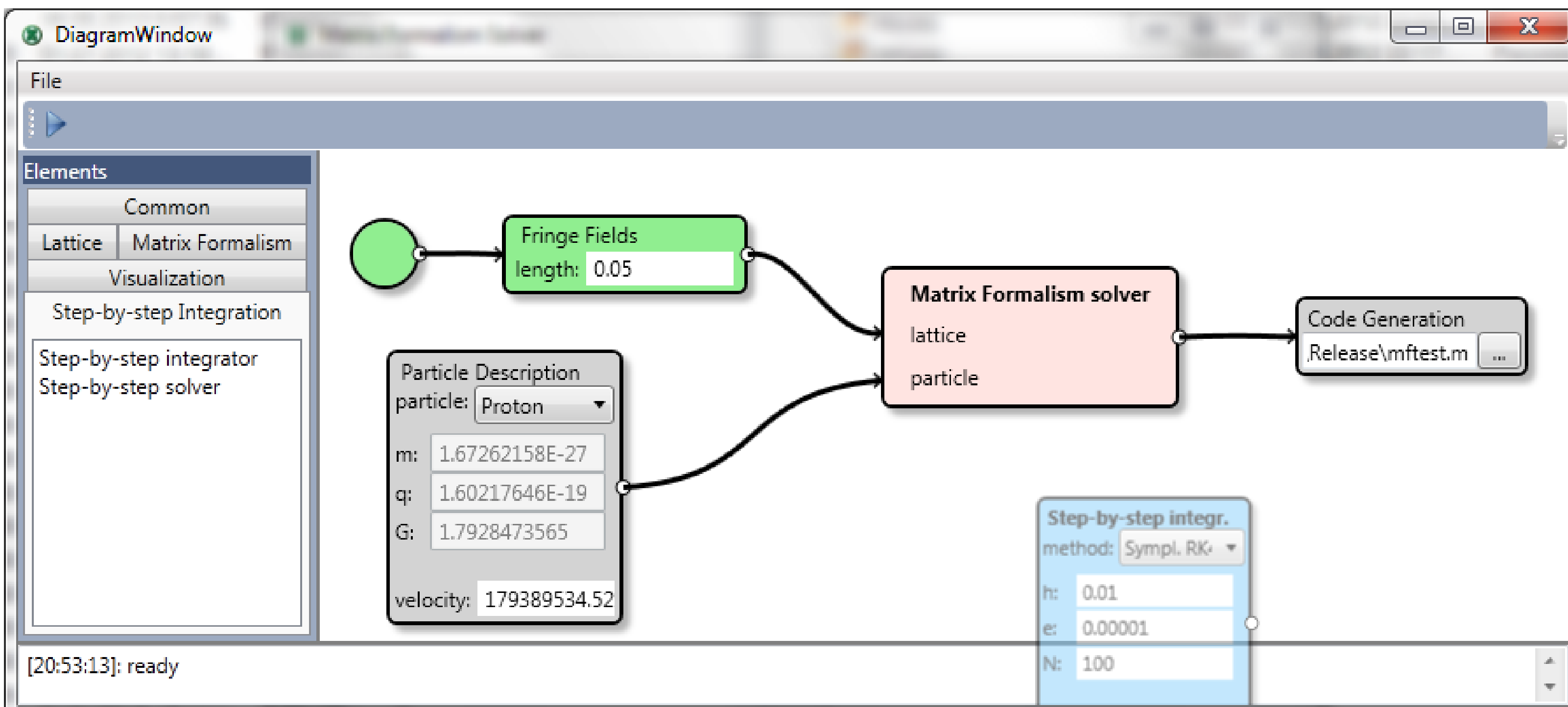


The function may be a composition of elementary functions (sin(x), cos(x), tan(x), exp(x), sqrt(x), ln(x)) and operators +, -, *, /.

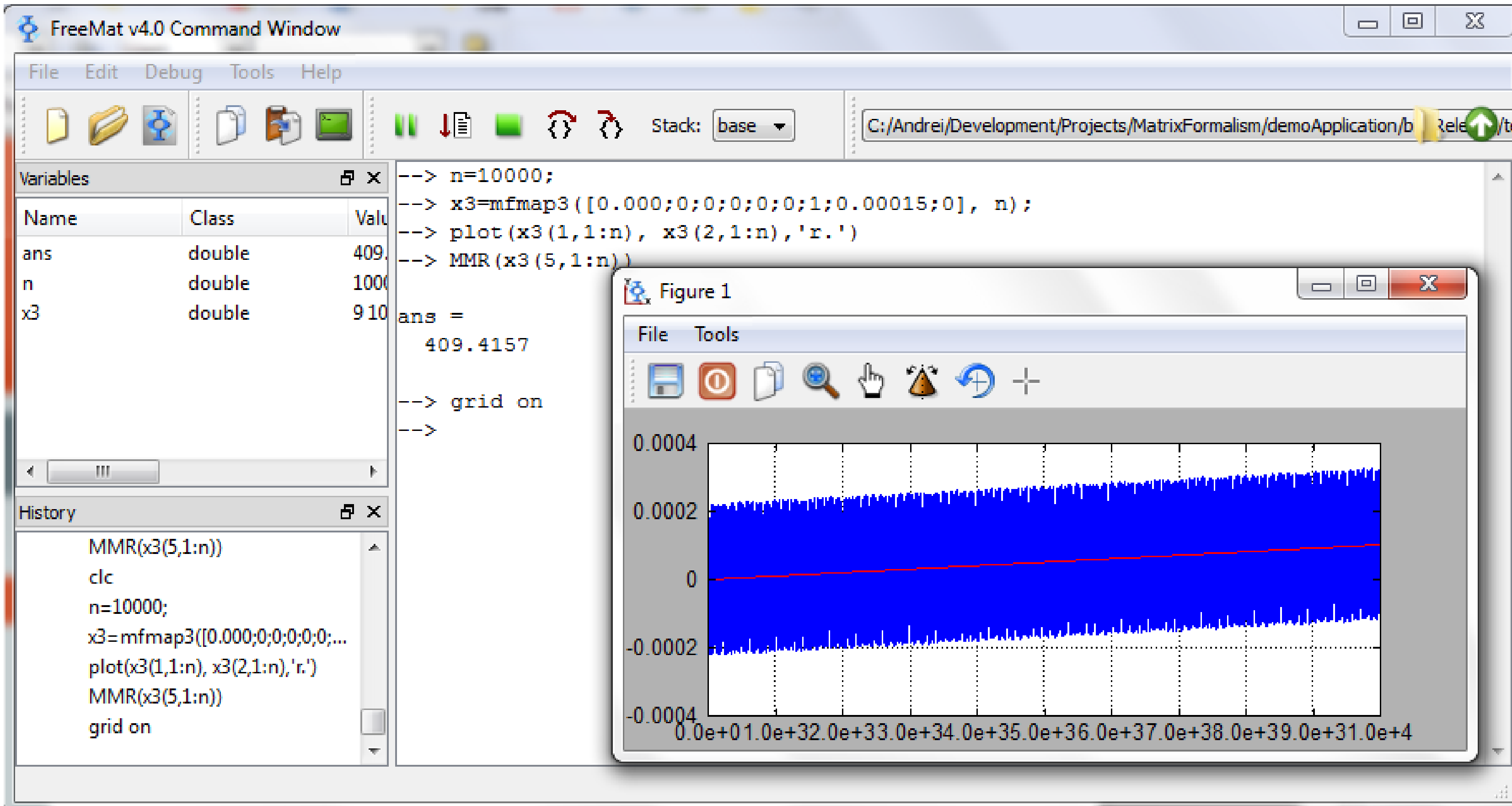
Map building for a lattice



You can specify any field distribution (both electric and magnetic) in an analytical form for automatically map bilding. Although computational code generation is available.



Particle simulation: demo



X = (x, x', y, y, Sx, Sy, Sz, dv, t)

Code generation

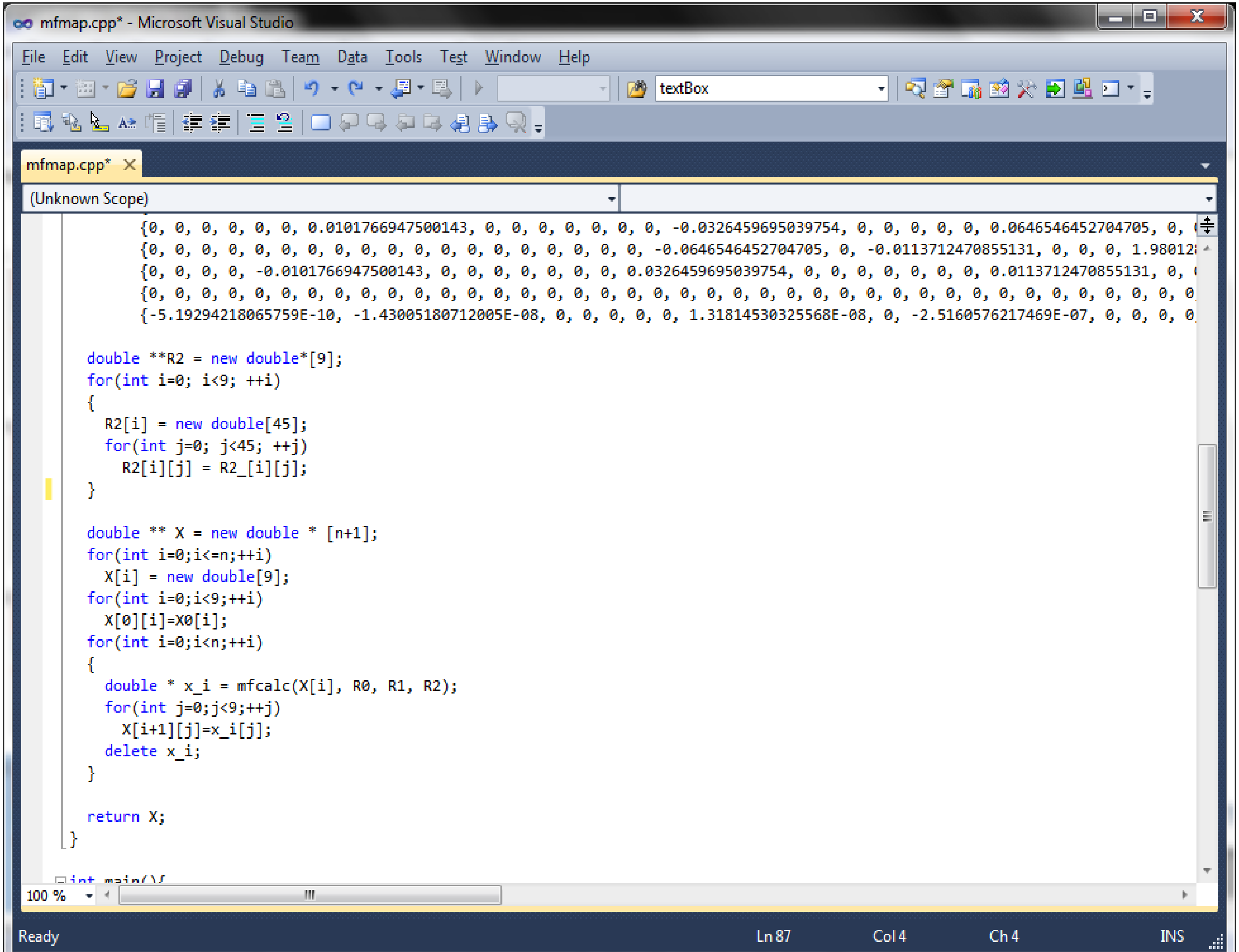
mfmap3.m* (C:/Andrei/Development/Projects/MatrixFormalism/demoApplication/bin/Release/test/) - FreeMat v4.0 Editor

File Edit Tools Debug Help

mfmap3.m*

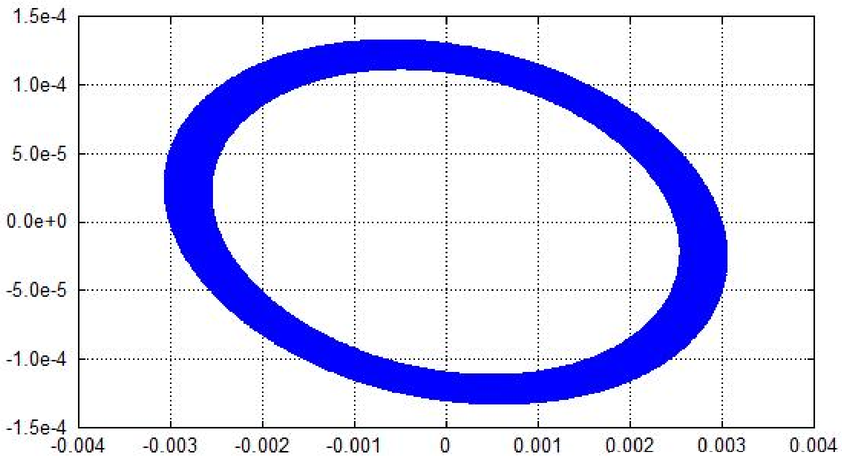
```
%code generation
%mapping function
function X = mfmap(X0, n)
%map generation:
    R0=[0;0;0;0;0;0;0;0;1.06369637954473E-06;];
    R1=[-0.670513915242833 14.3255631639029 0 0 0 0 0 35.3231449743497 0;
        -0.0270879171669592 -0.912653250594104 0 0 0 0 0 0.332988869053756 0;
        0 0 -0.970853028677094 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
```

Code generation



Matrix Formalism

for long-term evalution



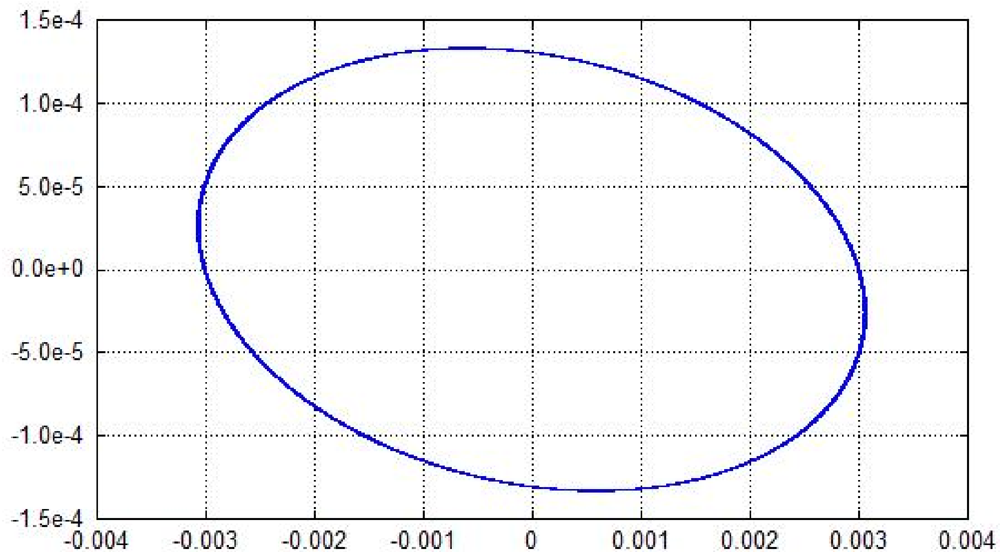
Using a symplectic integration algorithm to build a map does not guarantee the symplectic map (left: phase plane for 100000 turns, symplectic condition violation)

Symplectification is only matrix elements corrections:

$$M^* J M = J, \forall X_0,$$

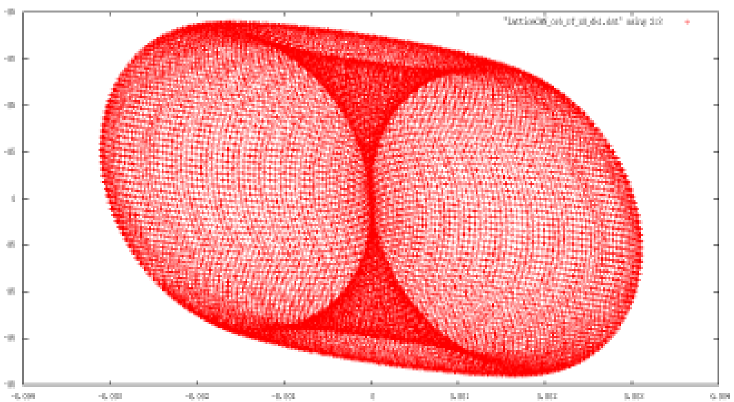
$$J = \begin{pmatrix} 0 & E \\ -E & 0 \end{pmatrix}.$$

1 000 000 turns:



Comparison:

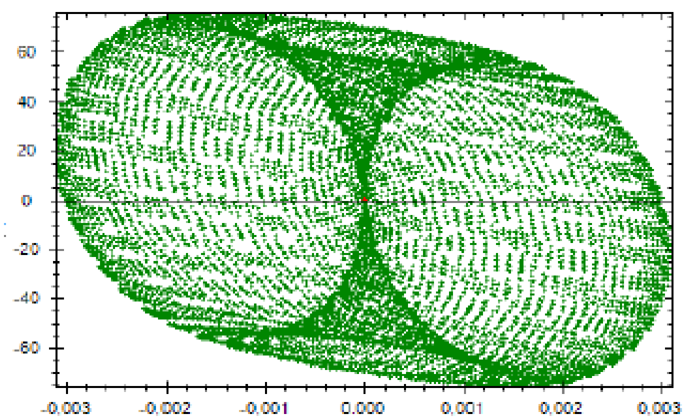
orbital motion in transverse plane (RF ON)

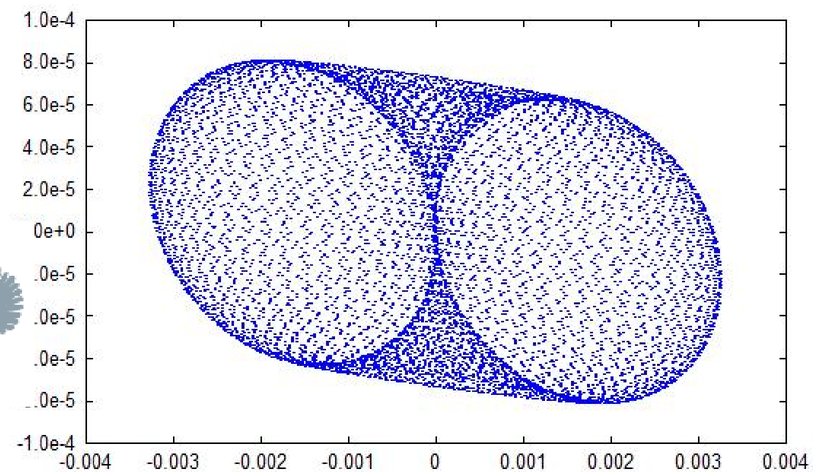


COSY INFINITY

MICHIGAN STATE UNIVERSITY

Thanks to D. Zyuzin for help with COSY INFINITY calculation



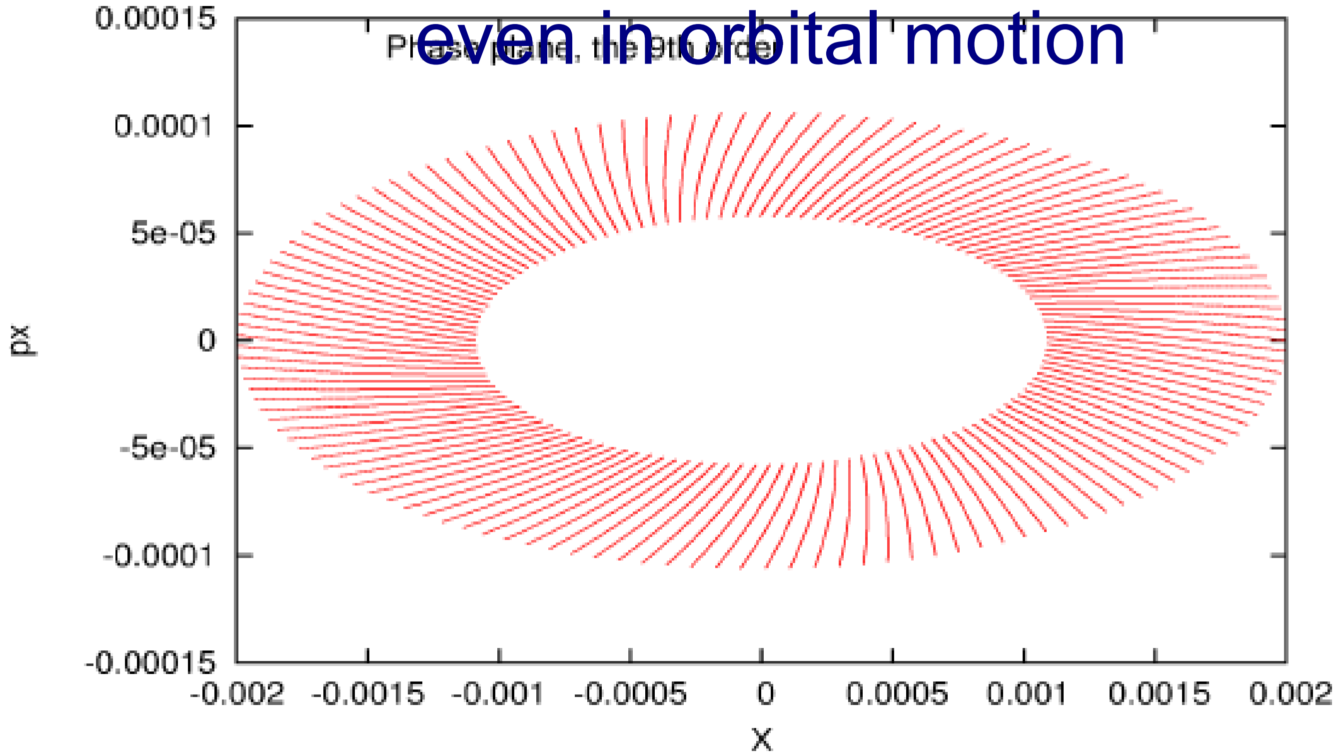


- Step-by-step integration
- Matrix Formalism

Comparison:
fringe fields influence

COSY INFINITY
MICHIGAN STATE UNIVERSITY

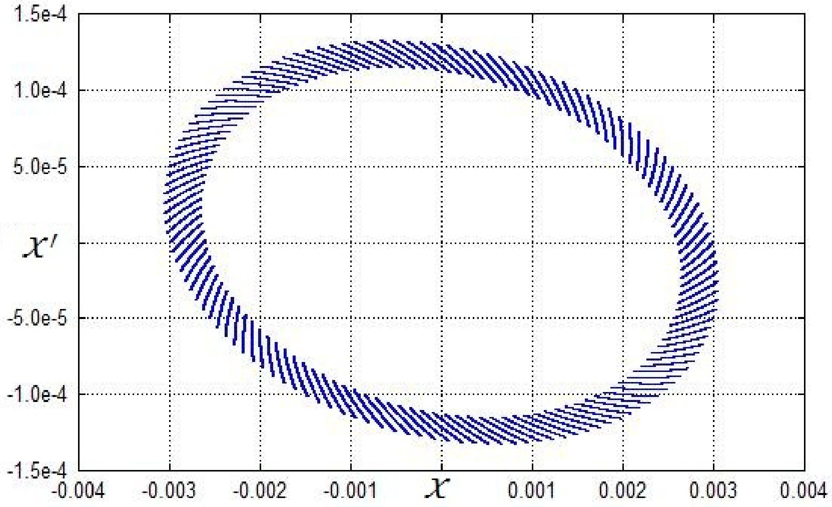
Strange behavior not only in spin dynamics
but
even in orbital motion



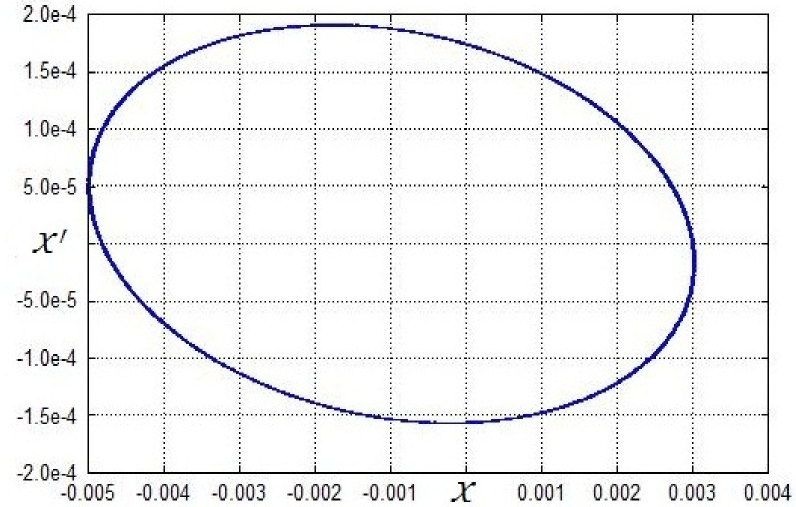
It seems that all particles shrink in time to the reference particle

Matrix Formalism:

fringe fields influence



a)



b)

a) transverse plane with fringe fields without energy conservation

b) energy conservation corrections, when reference orbit can be found by following way

$$X_{ref} = \sum_{i=0}^{k} R^{1i}(t) X_{ref}^{[i]}. \qquad X_{ref} = (E - R^{11})^{-1} R^{10}.$$

Conclusion

Results

- ✓ matrix formalism as a high-performance mapping approach for solving of ODE is implemented in numerical mode

Laptop Intel Core i5

3 order, 9-dim state
1 000 000 turns for
1 point per **3.2 sec**

!

- ✓ sofware for beam dynamics (spin-orbital motion) simulation is developed

Further development

- implementation of matrix formalism in symbolic mode;
- research on EDM project and spin dynamics investigation.

Acknowledgments

The research is a part of **JEDI** collaboration.

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

Thanks for

- Yu. Senichev
- D. Zyuzin

Thank you for your attention