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

Nataliia Kulabukhova

Virtual Accelerator Laboratory: the synbolic presentation for space charge fields

NEC 2015, September 27– October 3, Budva, Montenegro

Overview

- Beam (Accelerator) Physics;
- Computer Modeling, Concept of VA;
- Space Charge Fields;
- The Possibility of Parallelisation;
- Conclusion.

Motivation for Virtual Accelerator Laboratory (VAL)

- Computer modeling environment for beam physics: model, predict and assist construction of real accelerators;

- Multiple software modeling packages exist;

- Cross-check modeling results of different packages;

- Combination of software packages into a workflow.

Problems for computer scientist

- Provide access for scientists: easy-to-use GUI (PSE – problem solving environment; SG – science gateway; VL – virtual laboratory etc);
- Easy integration and control of different application components (e.g. software packages);
- Transparent and coordinated use of distributed computing resources.

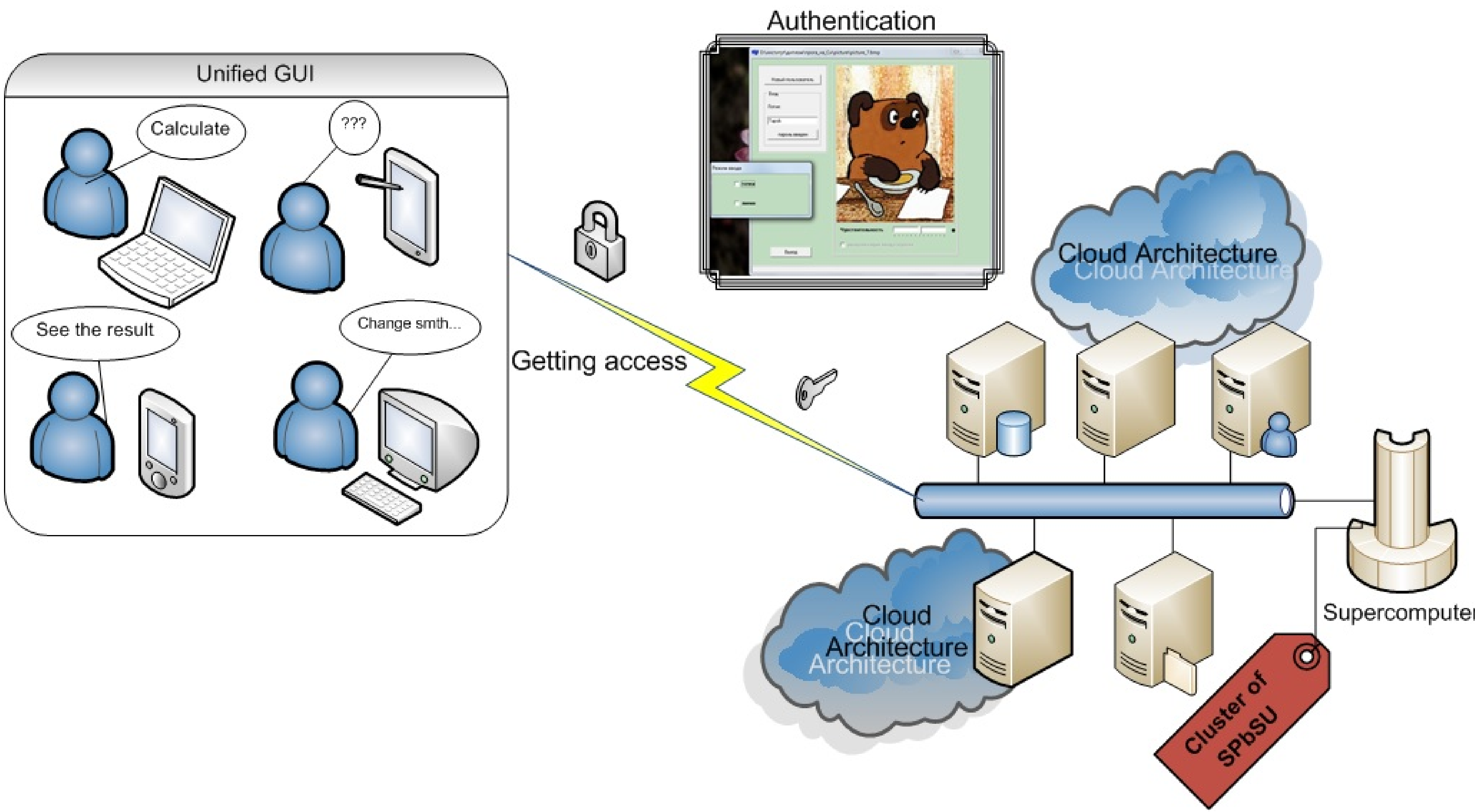
Overview of Virtual Accelerators (VA)

In order to control large-scale accelerators efficiently, a control system with a virtual accelerator model was constructed by many facilities. In many papers by the words Virtual Accelerator an on-line beam simulator provided with a beam monitor scheme is meant. It works parallel with real machine. The machine operator can access the parameters of the real accelerator through the client and then feed them to the virtual accelerator, and vice versa.
Until now there is no virtual accelerator working without real machine.

Goal

Construct such Virtual Accelerator Laboratory application used independently from any machine.

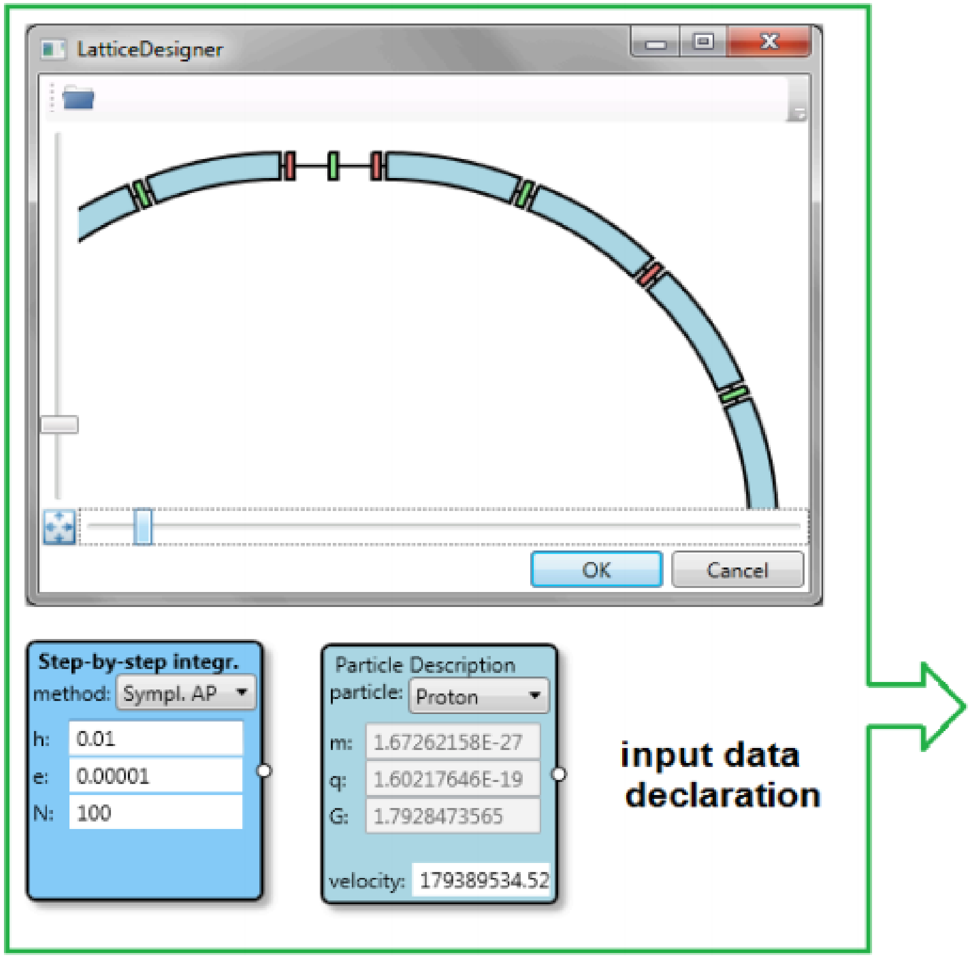
Concept of Virtual Accelerator Laboratory



Concept of Virtual Accelerator Laboratory

User sets initial conditions and parameters:

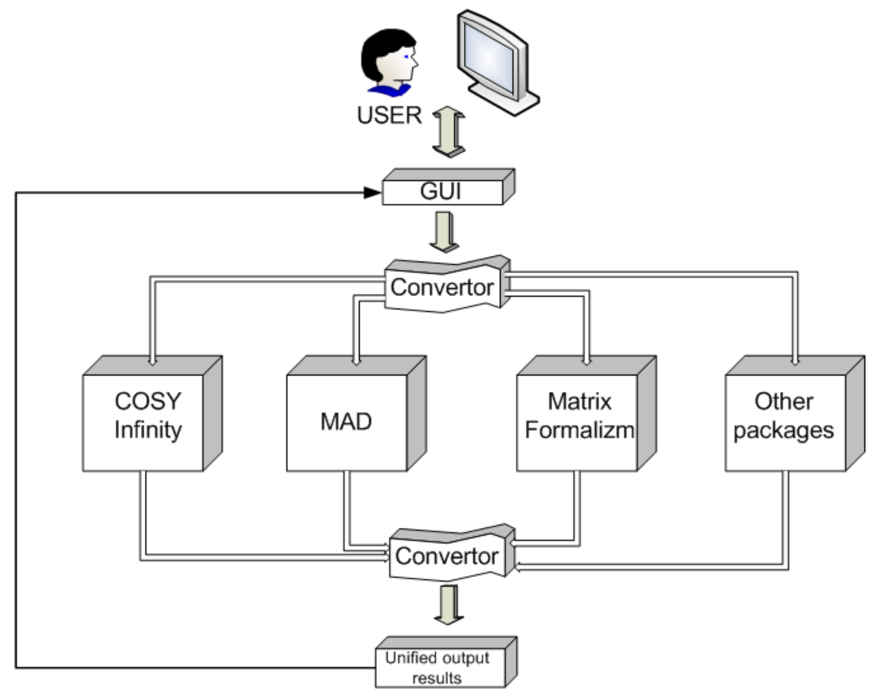
- quadrupole;
- sextupole;
- octupole;
- solenoid.



Concept of Virtual Accelerator Laboratory

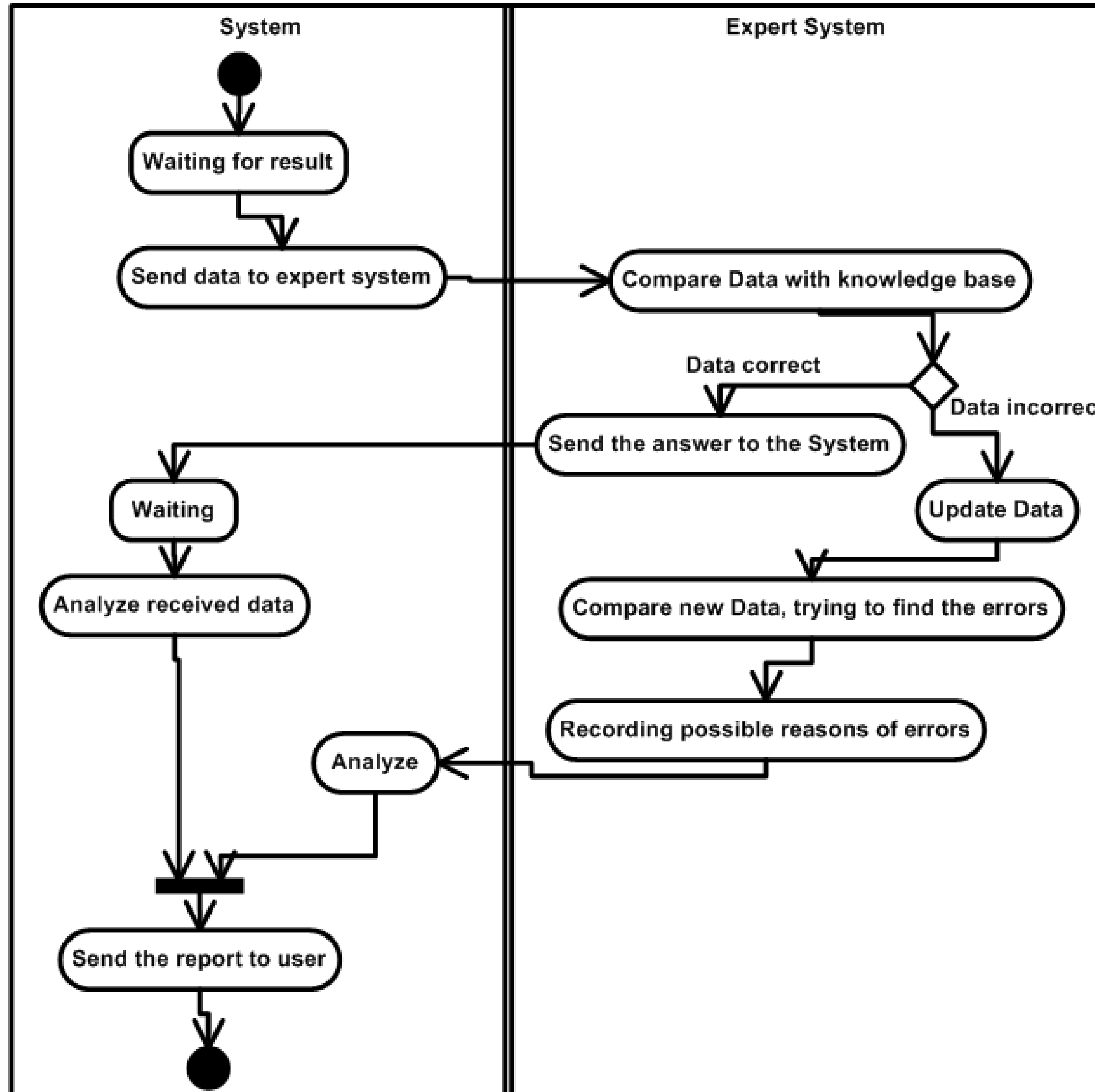
- directly in the language of one of the packages, which will be used for calculations;

- using "generic" description language, that can be converted into a specific language used in the individual package;



Concept of Virtual Accelerator Laboratory

Starting the simulation user may wish to see the intermediate results;

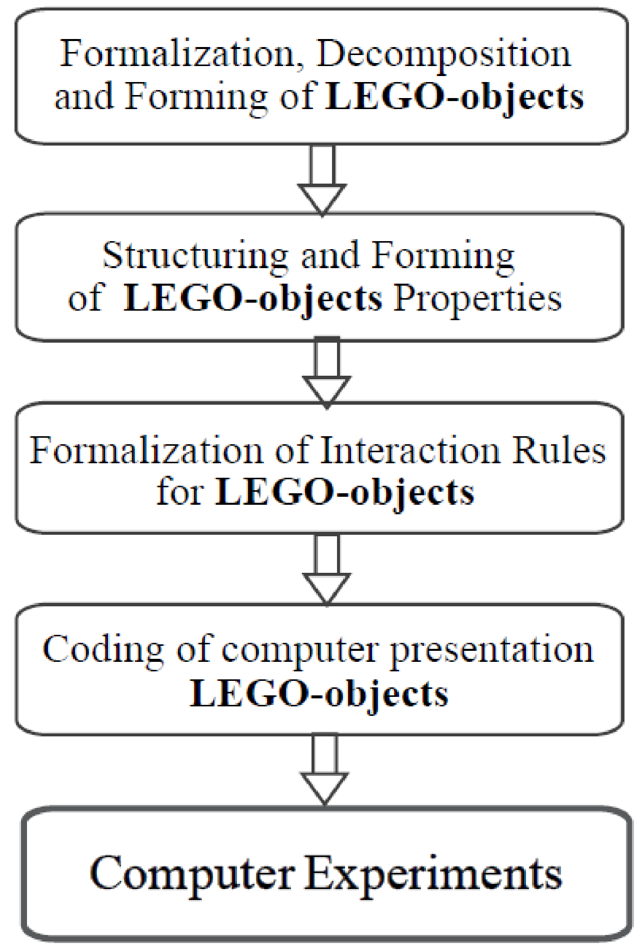


Concept of Virtual Accelerator Laboratory

VA offers the possibility of organizing a workflow – sequentially running packages, where the next step uses the data obtained on the previous step(s).

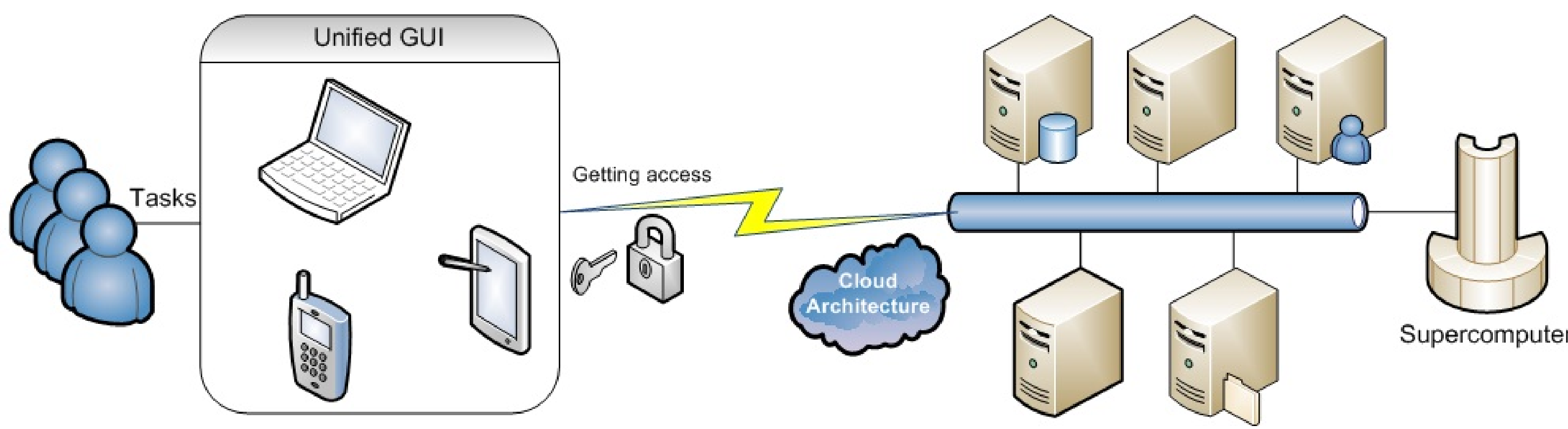
Lego blocks have property of minimality - the further decomposition is impossible. The blocks have properties of universality, abstractness and can be implemented in the form of VAL.

The LEGO blocks are constructed for all levels of the computer model.



Approach

- **Graphical representation (GUI);**
- **Workflow-based PSE;**
- **Unified description processed by different packages;**
- **Personal, Cluster, Grid and Cloud. computing**



Space Charge Forces

The problem of dispersion in the presense of significant space charge effects and outline two possible approaches:

- **Particles simulation with the help of distribution functions;**
- **Envelope simulation.**

Predictor-corrector method

Currently the distribution function can be presented as:

– Uniform distibution :

$$\varphi(\varkappa^2) = \frac{2\sqrt{detA}}{\pi^2}\Theta(1-\varkappa^2), \Theta(x) = \begin{cases} 1, x \geq 0, \\ 0, x < 0. \end{cases}$$

– VK-distribution:

$$\varphi(\varkappa^2) = \frac{\sqrt{detA}}{\pi^2}\delta(1-\varkappa^2).$$

– Normal distribution:

$$\varphi(\varkappa^2) = \frac{\sqrt{detA}}{4\pi^2} exp\left(-\frac{\varkappa^2}{2}\right)$$

Matrix Formalism

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

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

Non-linear system of ordinary differential equations

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

Matrix form of ODE

Matrix Formalism

General view of equation:

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

The solution of X can be found this way:

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

Beam dynamics simulation

Envelope mapping

$$X_0^T A X_0 < 1$$

initial particle distribution

$$X = R X_0$$

where R is a linear map

$$X_0 = R_0^{-1} X$$

$$X^T \underbrace{R^{-1} A R}_{\text{new envelope}} X < 1$$

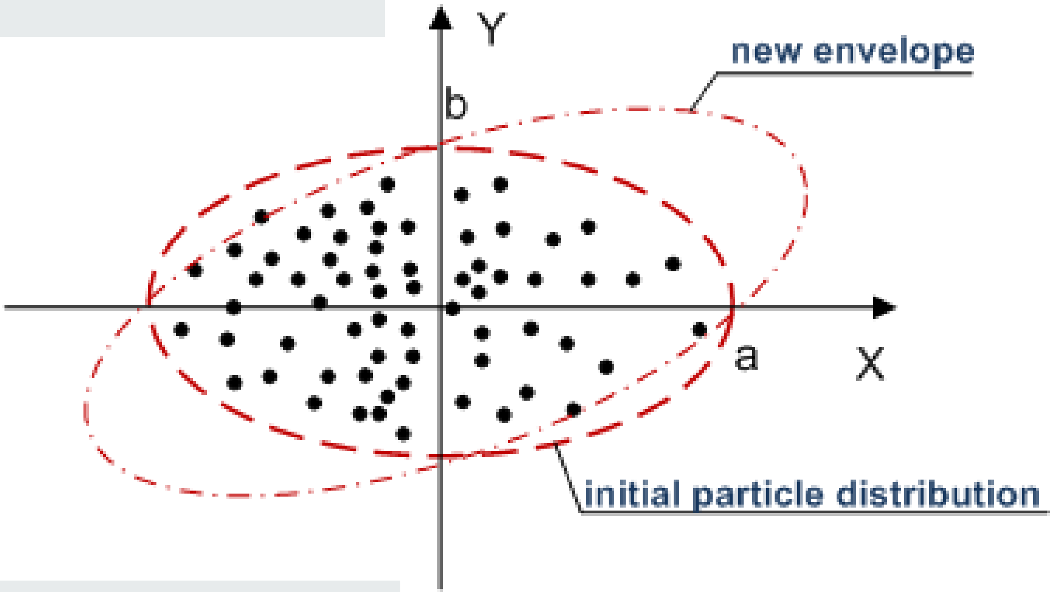
in case of non-linear map the transformations are similar

Beam dynamics simulation

Envelope mapping

$$X_0^T A X_0 < 1$$

initial particle distribution



$$X^T \underbrace{R^{-1} A R}_{\text{new envelope}} X < 1$$

new envelope

in case of non-linear map the transformations are similar

Extension space: example of 2 order

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \end{pmatrix} \begin{pmatrix} x_0^2 \\ x_0 y_0 \\ y_0^2 \end{pmatrix}$$

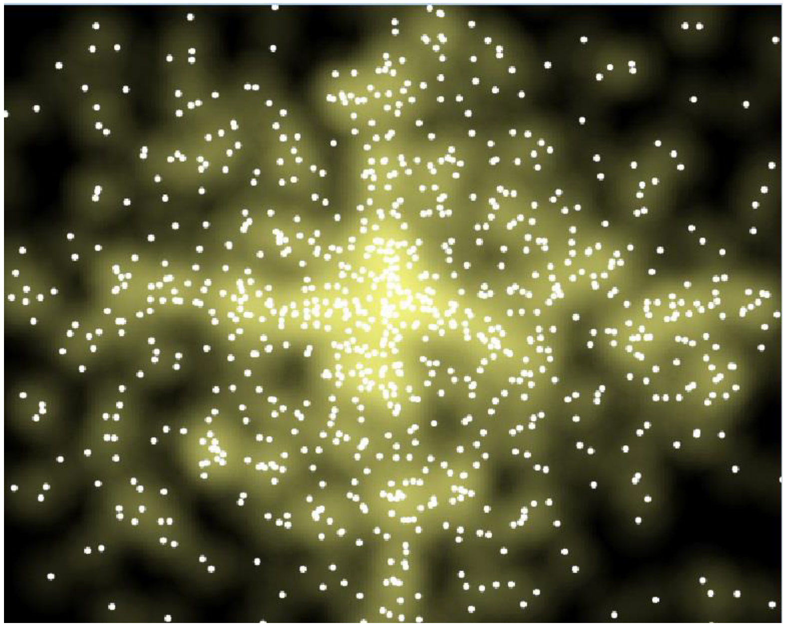
A non-linear map can be represented as linear in extension space:

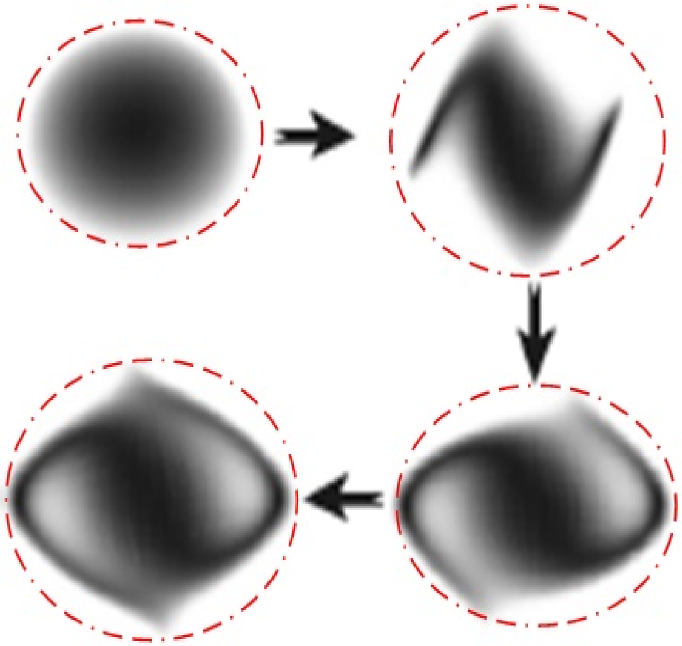
$$\begin{pmatrix} x \\ y \\ x^2 \\ xy \\ y^2 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & b_1 & b_2 & b_3 \\ a_3 & a_4 & b_4 & b_5 & b_6 \\ 0 & 0 & a_1^2 & 2a_1a_2 & \dots \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ x_0^2 \\ y_0 \\ y_0^2 \end{pmatrix}$$

Extension space : example of 2 order

$$X = R^{11} X_0 + R^{12} X^{[2]}$$

$$\begin{pmatrix} X \\ X^{[2]} \end{pmatrix} = \begin{pmatrix} R^{11} & R^{12} \\ 0 & R^{22} \end{pmatrix} \begin{pmatrix} X_0 \\ X_0^{[2]} \end{pmatrix}$$

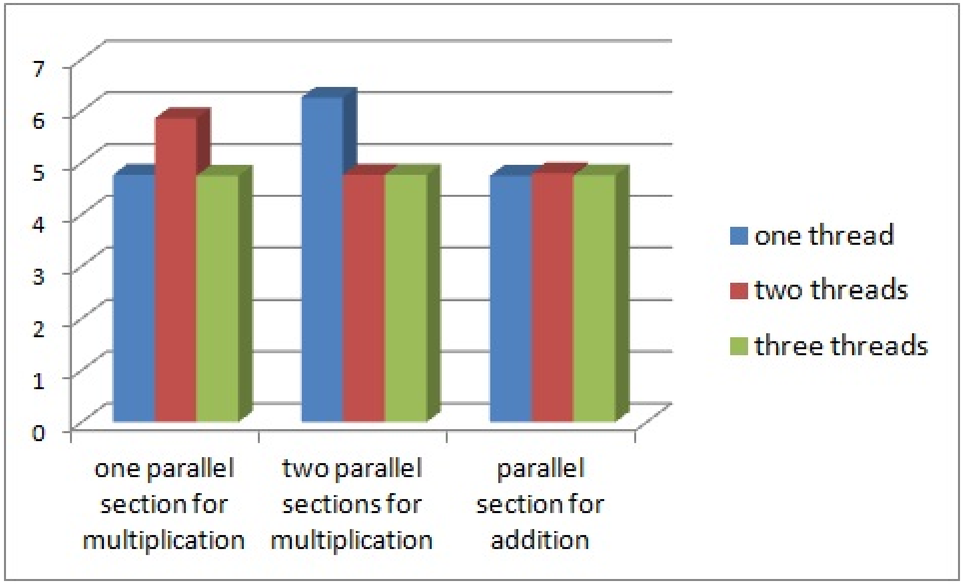




Results on CPU

For 1 000 000 turns we have:

	one parallel section for multiplication	two parallel sections for multiplication	parallel section for addition
one thread	4,77 sec	6,26 sec	4,75 sec
two threads	5,86 sec	4,77sec	4,8 sec
three threads	4,75 sec	4,77 sec	4,76 sec



More than three threads are not profitable

Conclusion

The model of Virtual Accelerator Laboratory should consist of a simulation engine that reads in all relevant lattice settings from the accelerator control system and computes data corresponding to the real diagnostic data output from the real accelerator.

Matrix formalism is a high-performances approach for beam dynamic modeling. The method can be implemented in parallel codes on GPU. It allows simulate both long-term evolution of a set of particles, and evaluation based on envelope description.

Aknowligments

- Special thanks for professor Alexander Bogdanov and professor Alexander Degtyarev;
- The author would like to thank Vladimir Korkhov for advices and remarks;
- Research was carried out using computational resources provided by Resource Center "Computer Center of SPbU" (http://cc.spbu.ru).
- The work is supported by the grants 0.37.155.2014 and 9.38.674.2013

Thank you