НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

September 6

Beam dynamics calculation

S.B. Vorozhtsov, E.E. Perepelkin and V.L. Smirnov
Dubna, JINR

http://parallel-compute.com

Outline

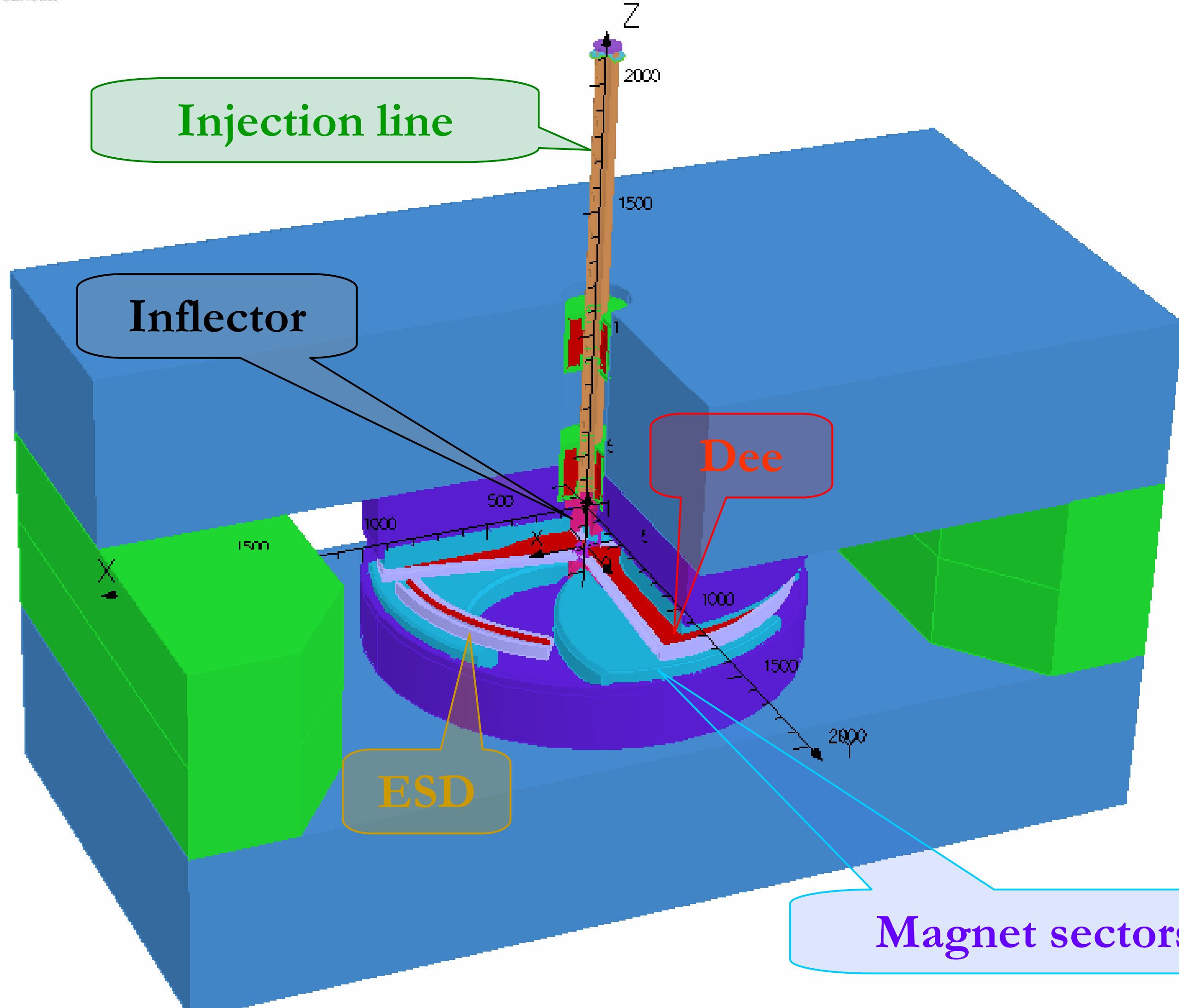
- Problem formulation
- Numerical methods
- OpenMP and CUDA realization
- Results

http://parallel-compute.com

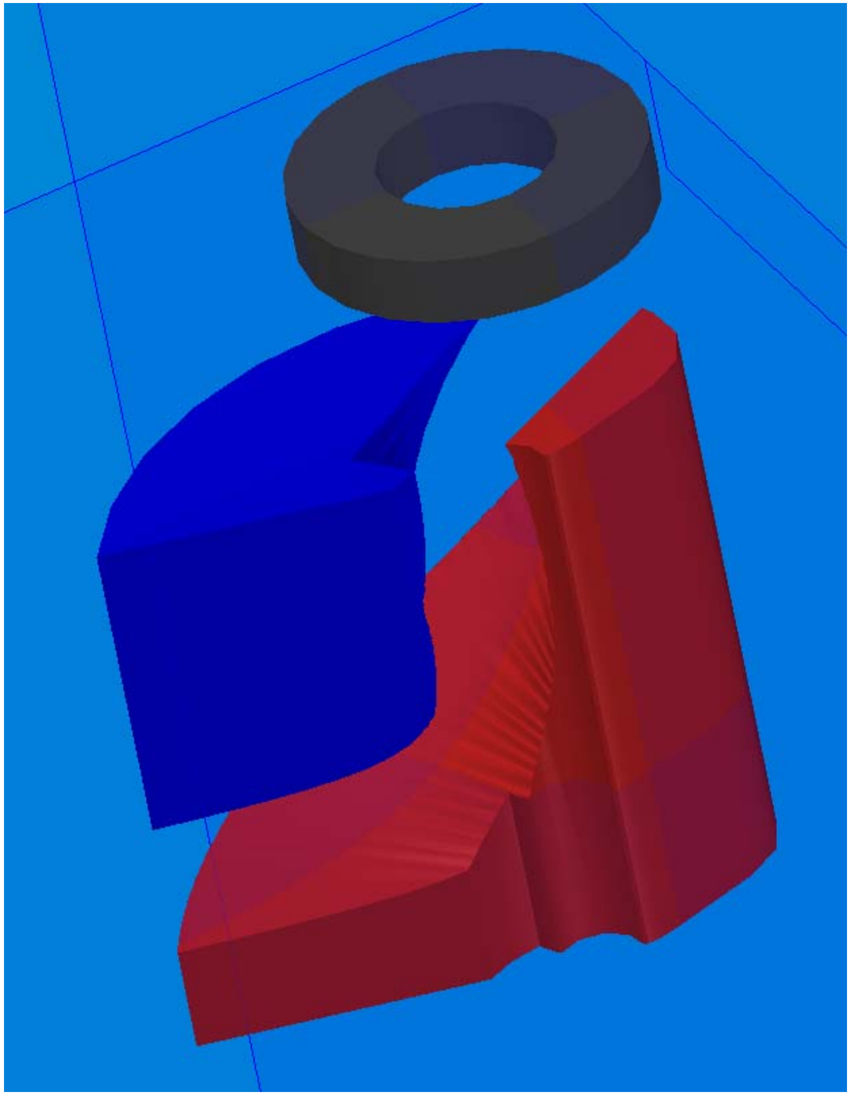
CBDA: Cyclotron Beam Dynamic Analysis code

НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

Computer model of the cyclotron

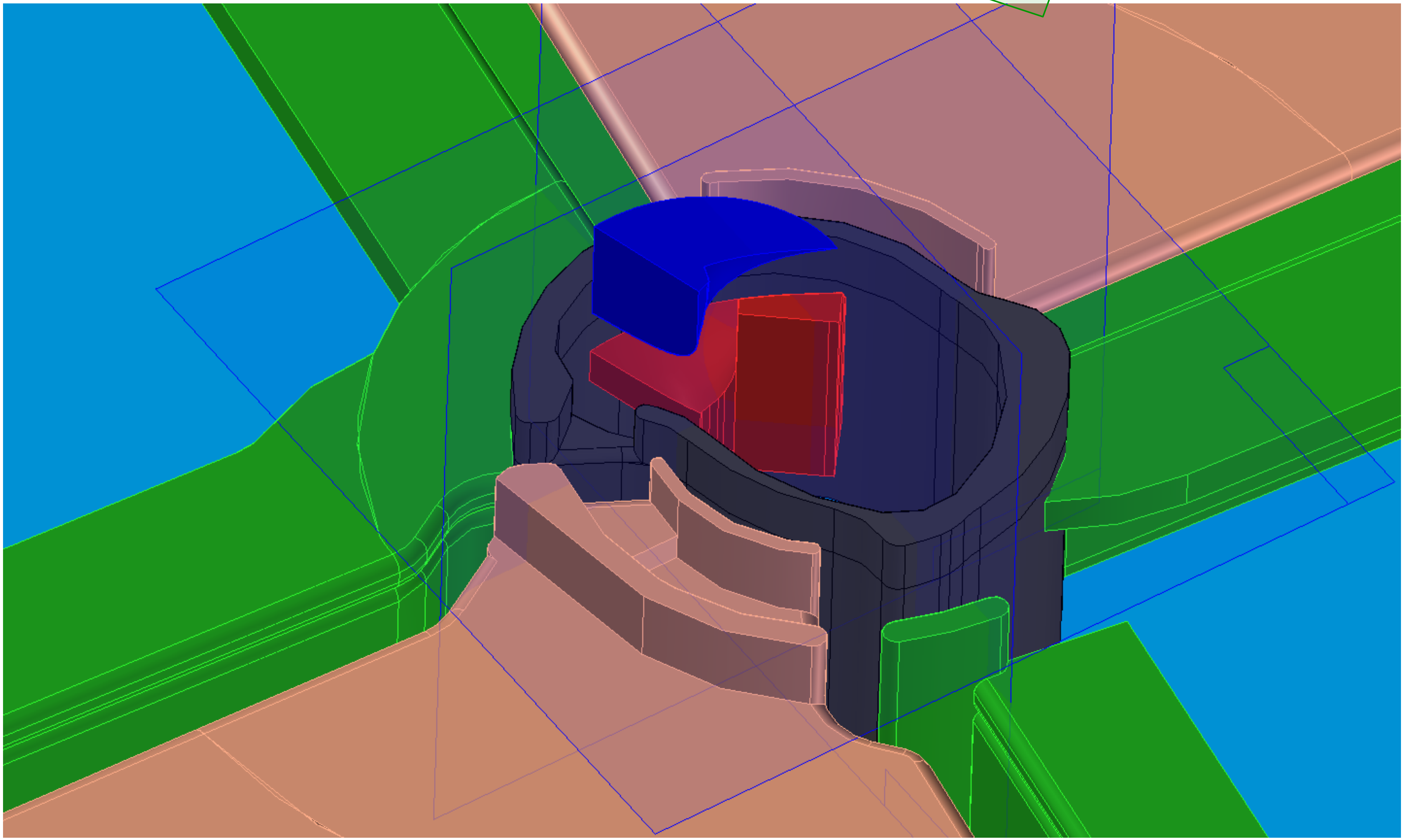


Inflector

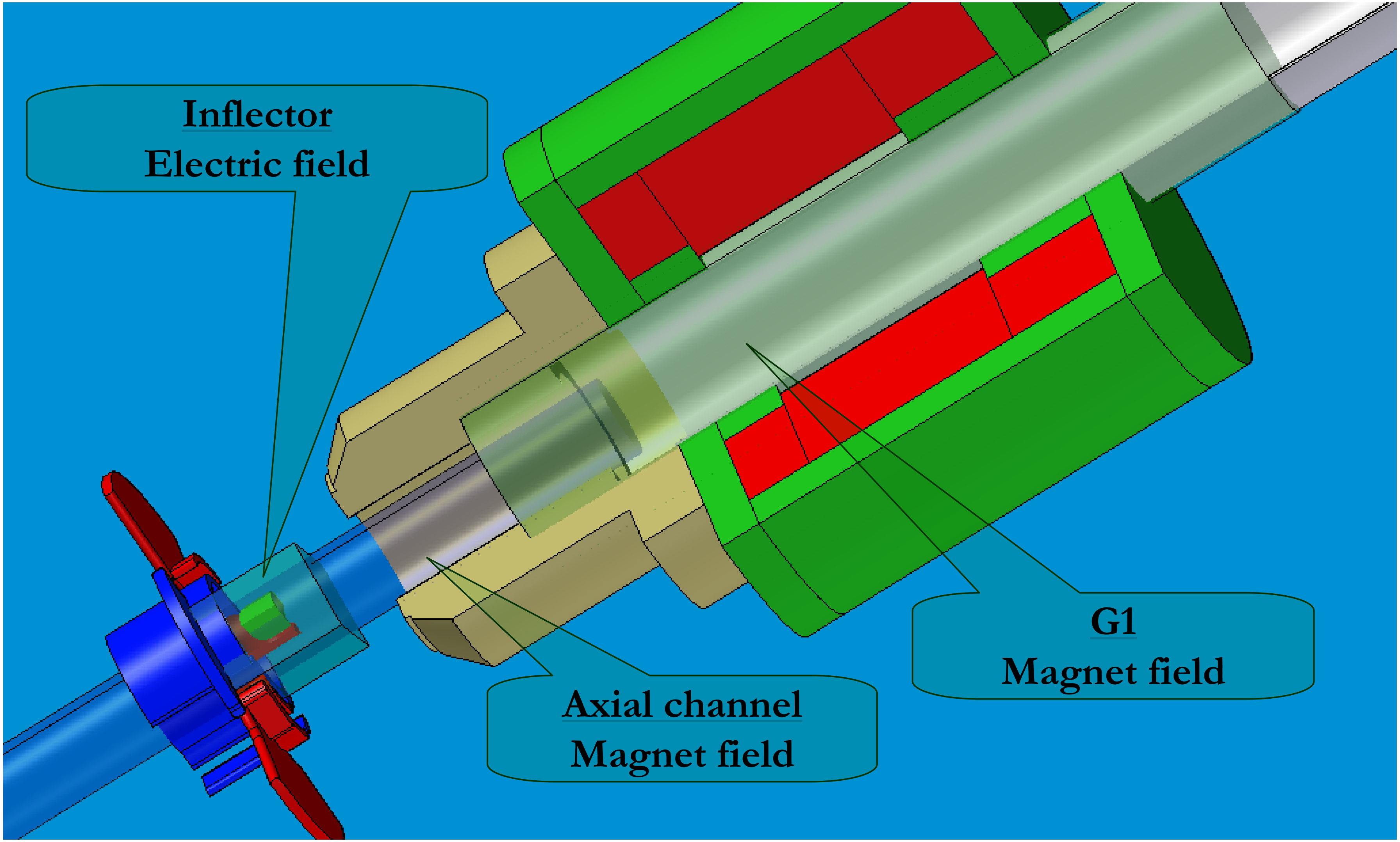


Central region

Courtesy A.S. Vorozhtsov



Regions of the field maps



Motion equation

$$\left(\vec{r}_i, \vec{p}_i, t\right),\ i = 1 \ldots N$$

$$\frac{d}{dt}\vec{p}_i = \vec{F}_i,\ i = 1 \ldots N$$

$$\begin{cases} m_i \dfrac{d}{dt}\left(\gamma_i \vec{v}_i\right) = q_i \left(\vec{E}_{ext}\left(\vec{r}_i, t\right) + \vec{E}_s\left(\vec{r}_i, t\right) + \left[\vec{v}_i, \vec{B}_{ext}\left(\vec{r}_i\right)\right]\right) \\ \gamma_i = 1 \Big/ \sqrt{1 - \beta_i^{\,2}},\ \beta_i = \dfrac{v_i}{c} \\ \vec{r}_i\Big|_{t=t_i} = \vec{r}_i^{\,(0)},\ \vec{v}_i\Big|_{t=t_i} = \vec{v}_i^{\,(0)},\ или\ \ \vec{r}_i\Big|_{t=0} = \vec{r}_i^{\,(0)},\ \vec{v}_i\Big|_{t=0} = \vec{v}_i^{\,(0)} \\ 1 \le i \le N, \vec{r}_i \in V \end{cases}$$

Space charge effect

$$div\vec{E}_s = \frac{\rho}{\varepsilon_0}, \qquad rot\vec{B}_s = \mu_0 \vec{J}_s + \frac{1}{c^2}\frac{\partial}{\partial t}\vec{E}_s$$

$$rot\vec{E}_s = -\frac{\partial}{\partial t}\vec{B}_s, \qquad div\vec{B}_s = 0, \quad \frac{1}{\mu_0\varepsilon_0} = c^2$$

PIC метоdays

$$\vec{E}_s = -\nabla\varphi$$

$$\begin{cases} \Delta\varphi(p) = -\dfrac{\rho(p)}{\varepsilon_0}, \quad p \in \Omega \\ \varphi\big|_{\Gamma_D} = \varphi_D, \quad \dfrac{\partial\varphi}{\partial n}\Big|_{\Gamma_N} = \psi_N, \quad \Gamma_D \cup \Gamma_N = \Gamma \end{cases}$$

PP метоdays

$$\vec{E}_s(\vec{r}_i) = \frac{1}{4\pi\varepsilon_0}\sum_{j\neq i}^{N}\frac{q_j}{\left|\vec{r}_i - \vec{r}_j\right|^3}\left(\vec{r}_i - \vec{r}_j\right), \quad i = 1\ldots N$$

$$\vec{E}_s(j \to i) = \frac{1}{4\pi\varepsilon_0}\frac{q_j}{R^3}\left(\vec{r}_i - \vec{r}_j\right), \quad \left|\vec{r}_i - \vec{r}_j\right| < R$$

Space charge effect

$$\begin{cases} \Delta\varphi(p) = -\dfrac{\rho(p)}{\varepsilon_0}, & p \in \Omega \\ \varphi|_\Gamma = 0 \end{cases}$$

FFT use for Furier set

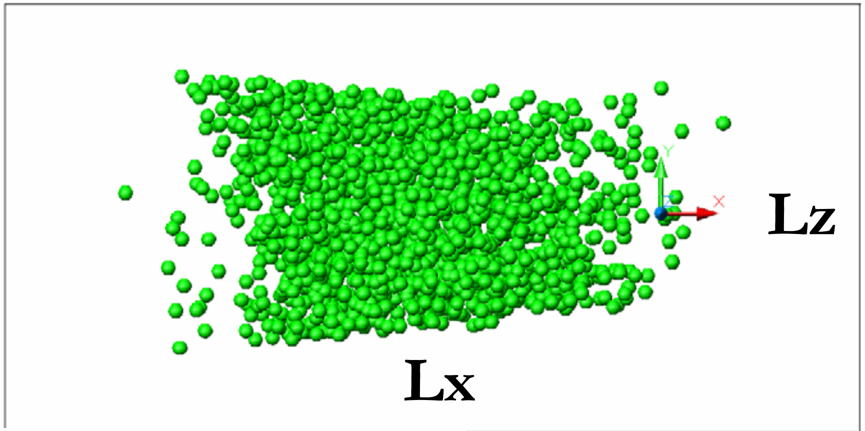
$\rho(x_i, y_j, z_s)$ – *obtain from the particle distribution*

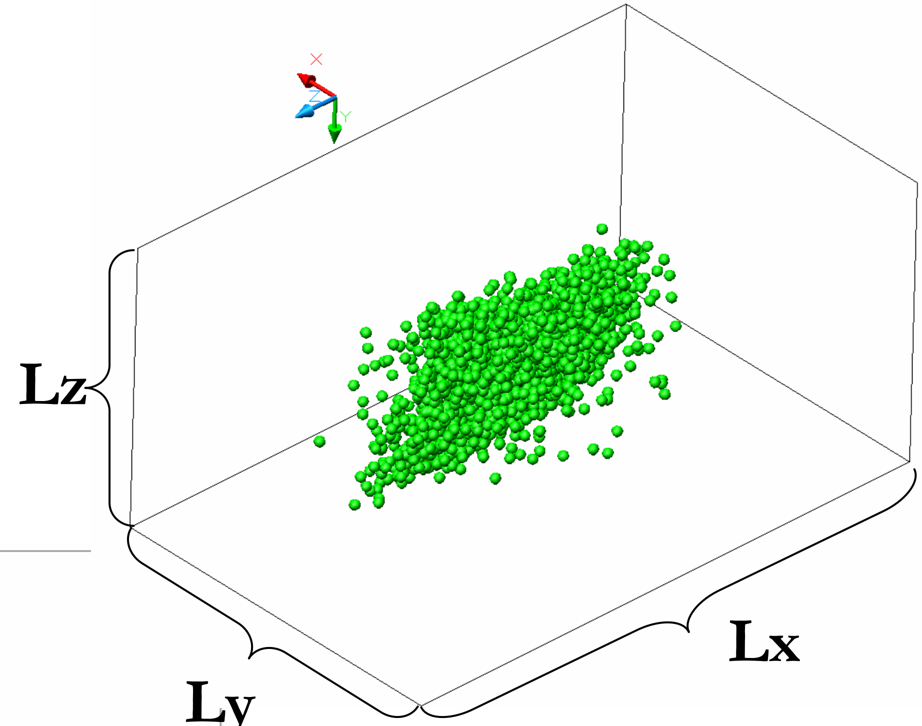
$$\bar{\rho}(n,m,k) = \frac{8}{N_x N_y N_z} \sum_{s=1}^{N_z-1} \sum_{j=1}^{N_y-1} \sum_{i=1}^{N_x-1} \rho(x_i, y_j, z_s) \sin\left(\frac{\pi n i}{N_x}\right) \sin\left(\frac{\pi m j}{N_y}\right) \sin\left(\frac{\pi k s}{N_z}\right)$$

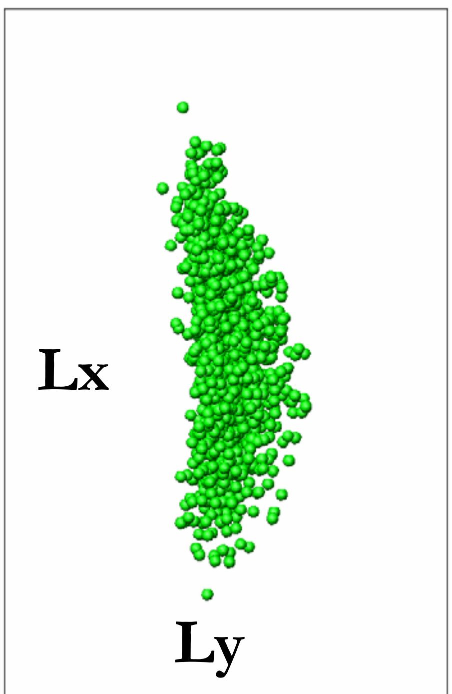
$$\bar{\varphi}(n,m,k) = -\bar{\rho}(n,m,k)\left[\left(\frac{\pi n}{L_x}\right)^2 + \left(\frac{\pi m}{L_y}\right)^2 + \left(\frac{\pi k}{L_z}\right)^2\right]^{-1}$$

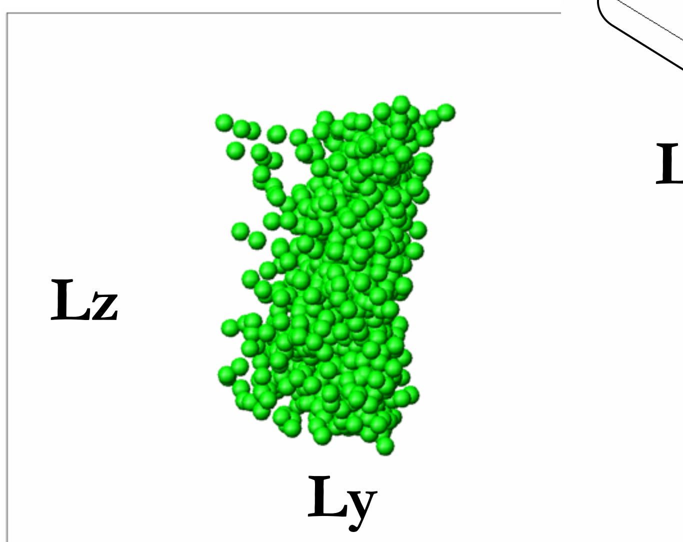
$$\varphi(x_i, y_j, z_s) = \sum_{k=1}^{N_z-1} \sum_{m=1}^{N_y-1} \sum_{n=1}^{N_x-1} \bar{\varphi}(n,m,k) \sin\left(\frac{\pi n i}{N_x}\right) \sin\left(\frac{\pi m j}{N_y}\right) \sin\left(\frac{\pi k s}{N_z}\right)$$

Bunch area





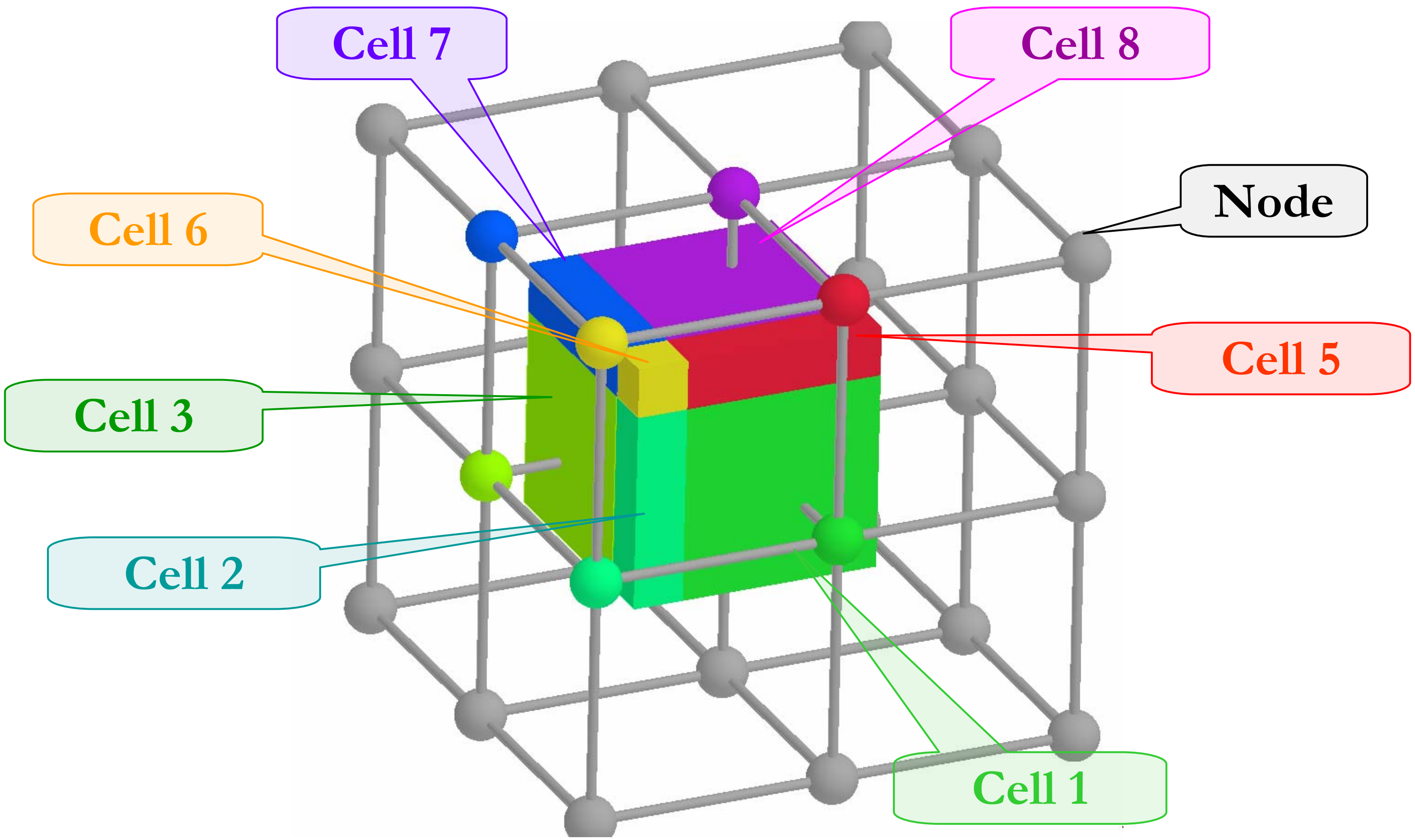




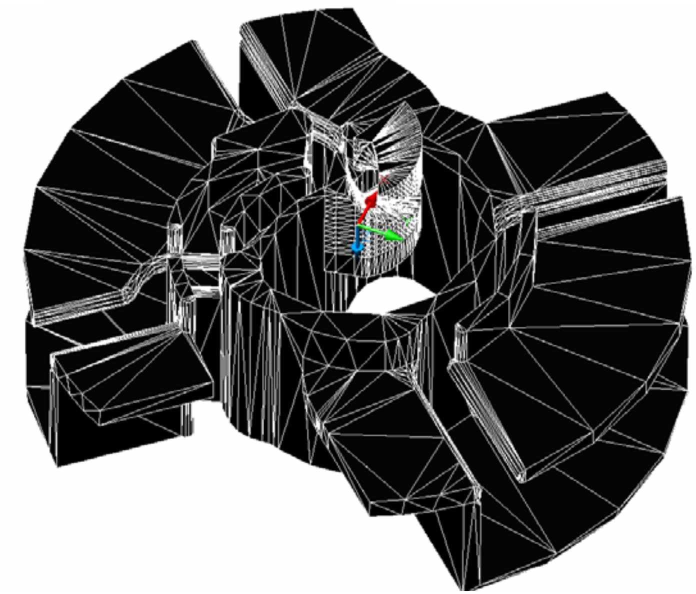
$$Mesh\ \ Nx \cdot Ny \cdot Nz$$

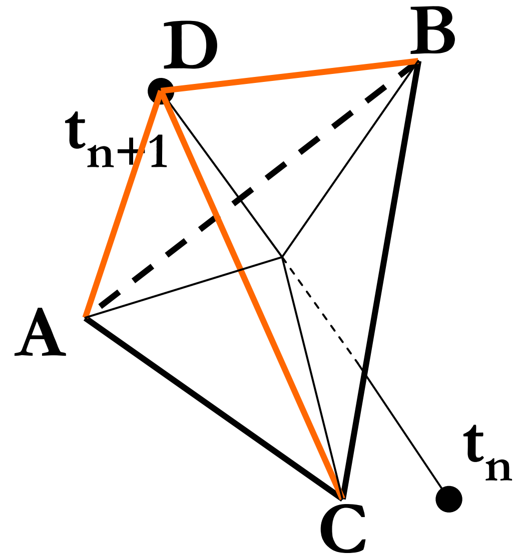
$$Mesh\ steps\ \ hx = \frac{Lx}{Nx}, hy = \frac{Ly}{Ny}, hz = \frac{Lz}{Nz}$$

Charge density calculation



Particle losses





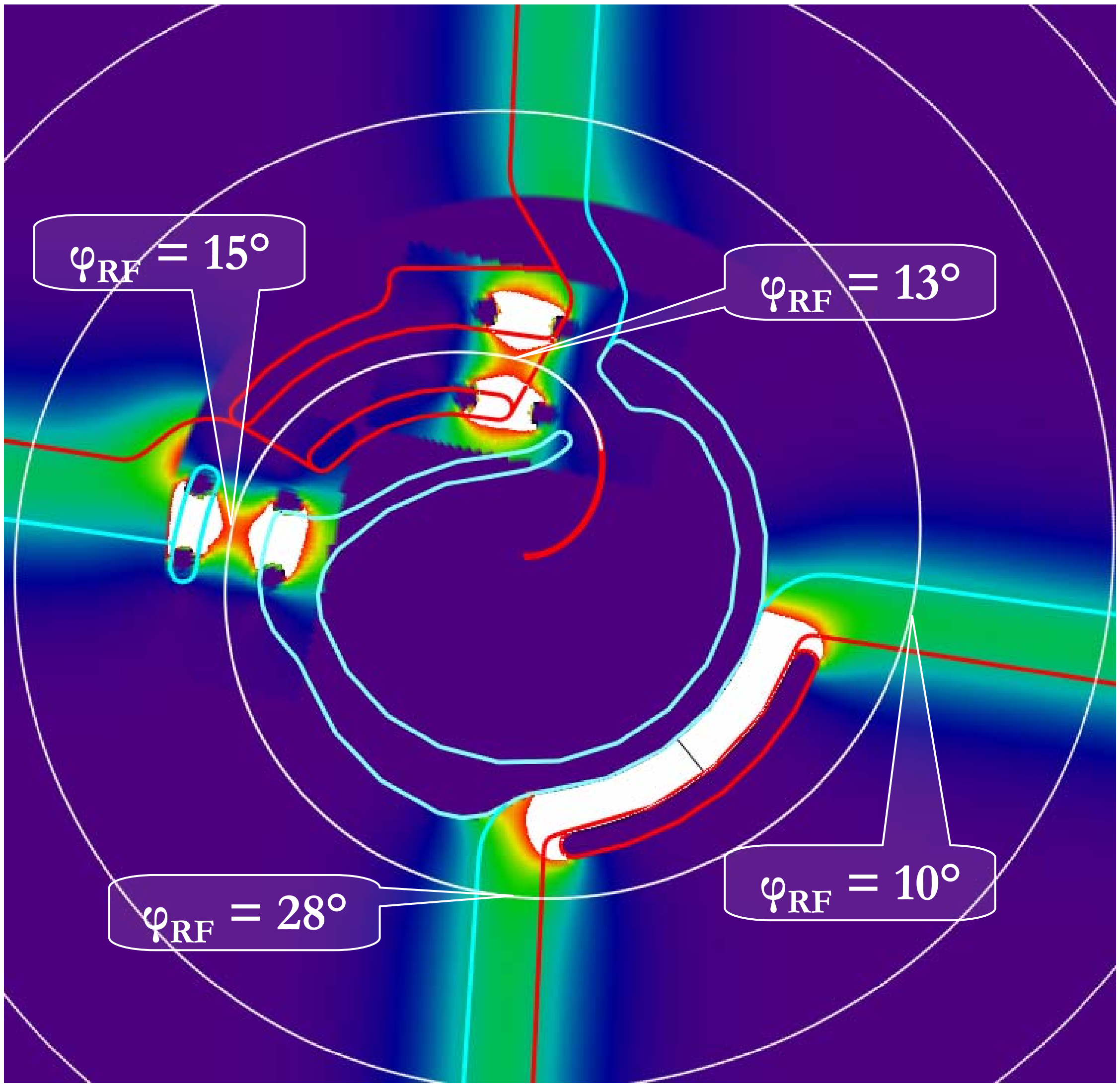
If point D belongs the ABC triangle then

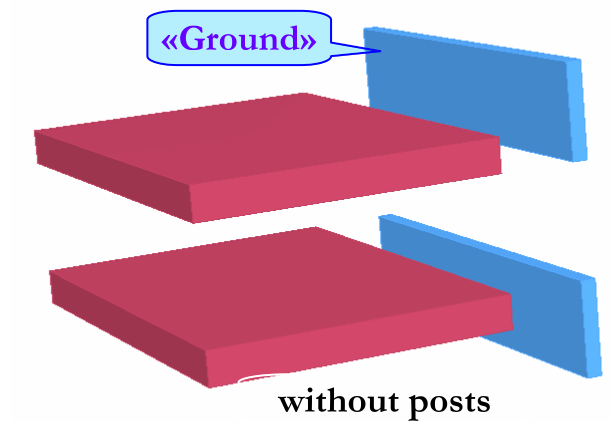
$$S_{\Delta ADC} + S_{\Delta ADB} + S_{\Delta CDB} = S_{\Delta ABC}$$

Cross condition $$S_{\Delta ADC} + S_{\Delta ADB} + S_{\Delta CDB} < S_{\Delta ABC} + \varepsilon_{\Delta}$$

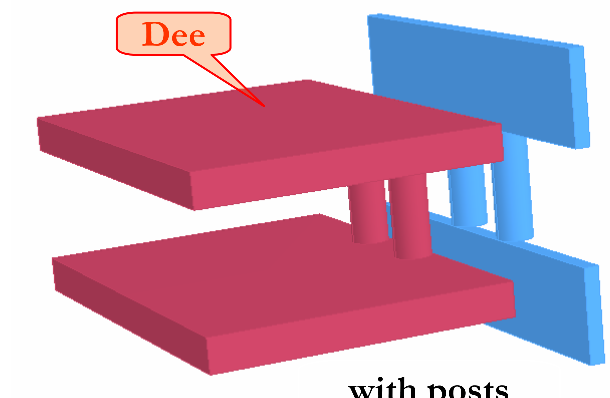
where ε_{Δ} – maximal deviation from surface

Central region optimization

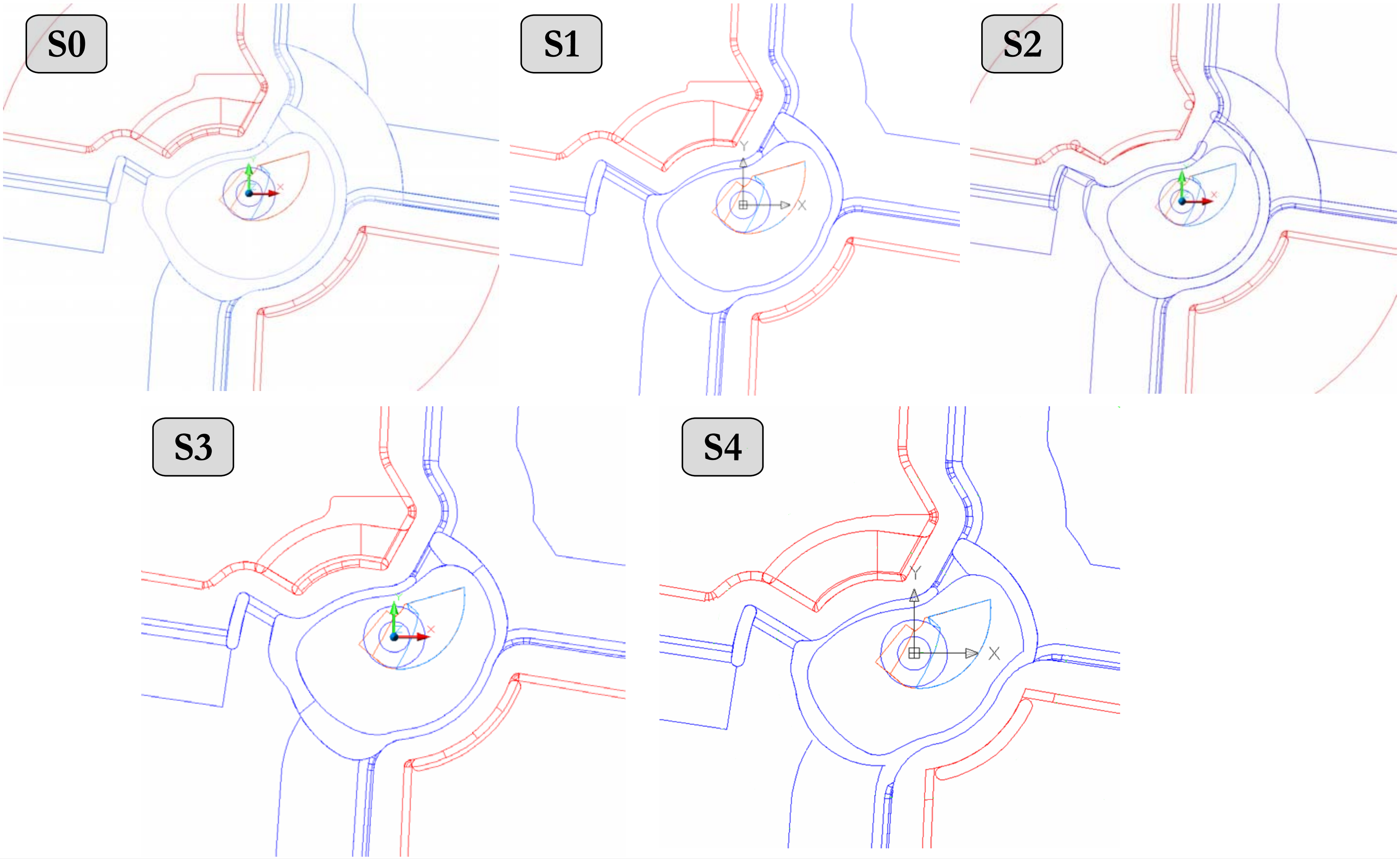




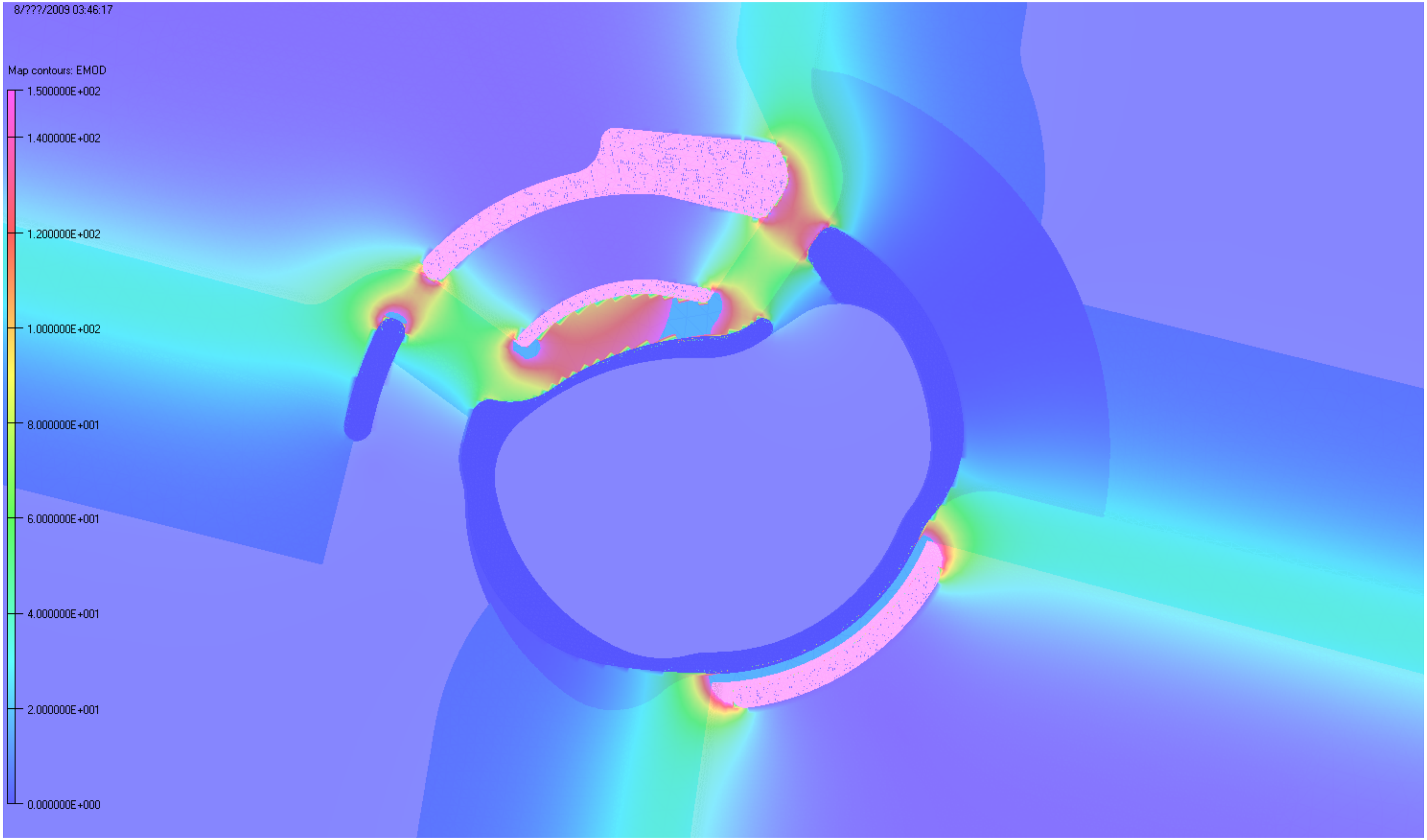
$$F = ZU_{RF} - W_{GAP}$$



Choice optimal configuration



Accelerating field distribution



Very time consuming problem

- About 5 different variants - minimum
- Many ions species - accelerated
- Very complicated structure
- Multi macro particle simulation for SC dominated beams

One run requires ~
several days of computer time

НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

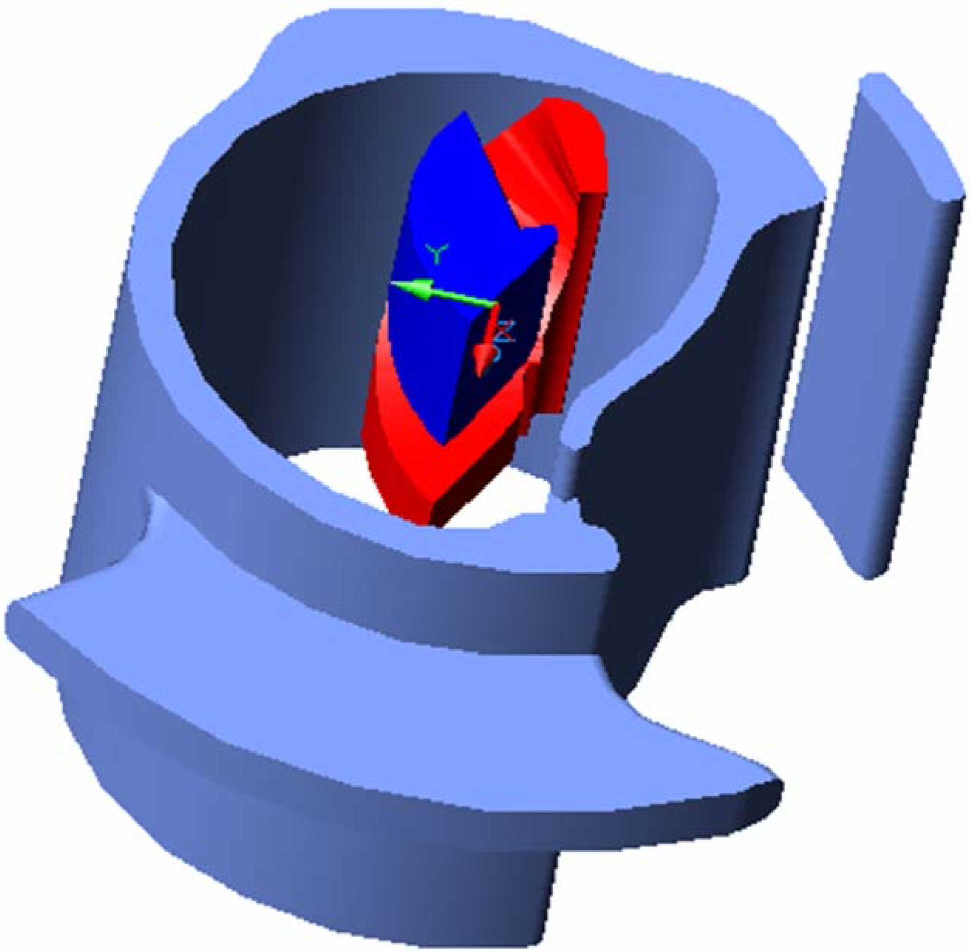
Numerical methods

- Runge-Kutta 4th order
  - (motion equations)
- FFT for Poisson equation
  - (space charge effect)
- Beam losses
  - («ray tracing» algorithm)

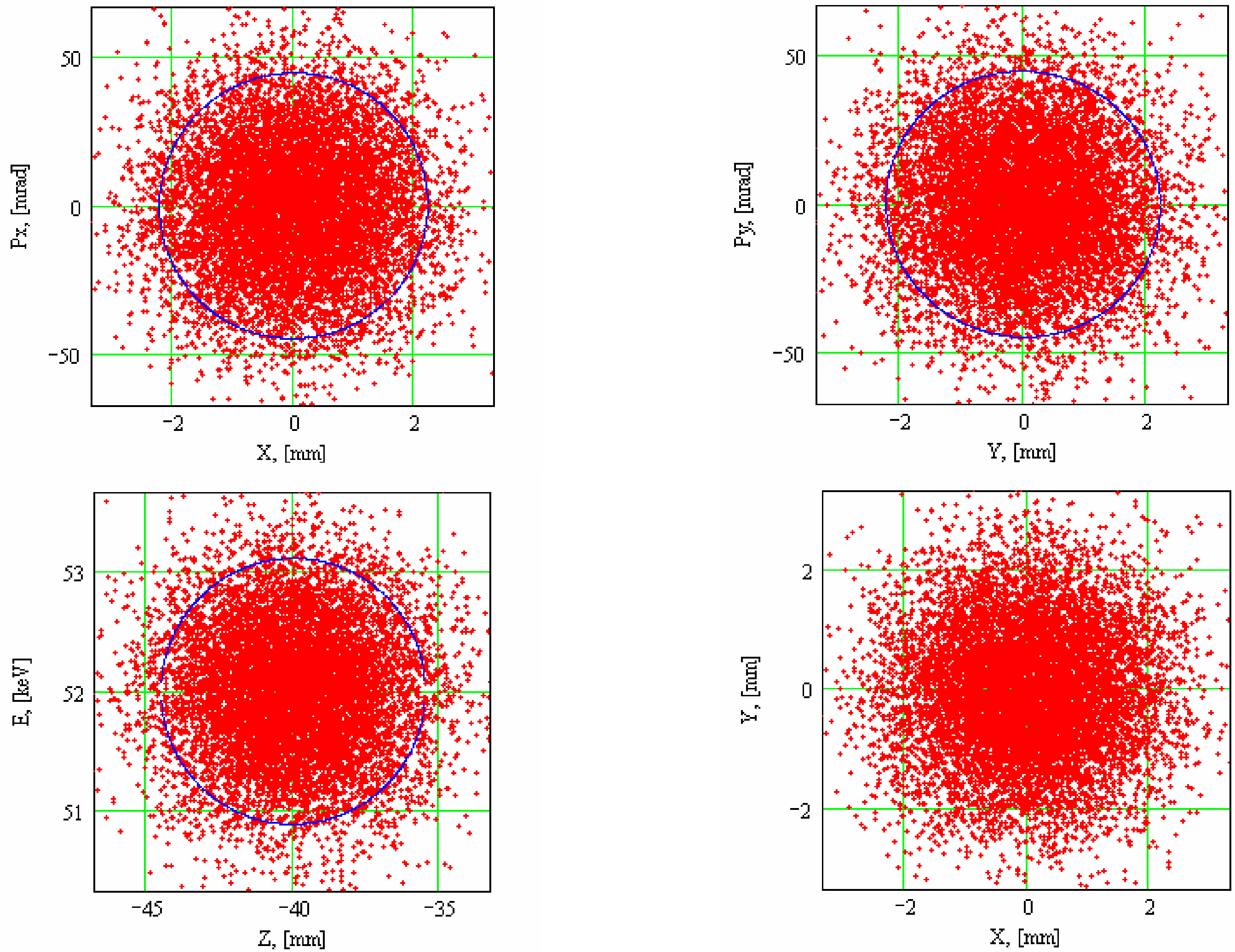
Open Multi-Processing

OpenMP

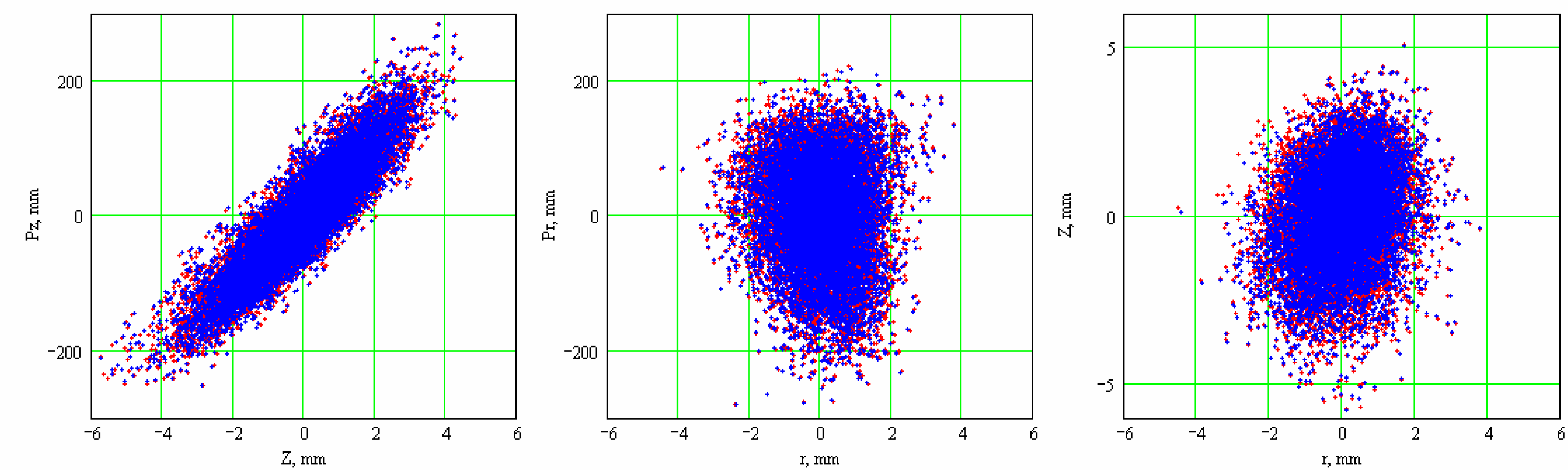
Spiral inflector



Beam parameters at the inflector entrance



Beam parameters at the inflector exit



Blue points – PIC by FFT (mesh: 2^5 x 2^5 x 2^5)

Red points – PP

НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

Calculation time

Method	Without OpenMP	With OpenMP	Computer platform
PP	4 h. 53 min.	2 h. 34 min.	AMD Turion 64×2, 1.60 GHz
	4 h. 38min	1 h. 25 min.	Intel Core2 Quad 2.4 GHz
PIC $2^5 \times 2^5 \times 2^5$	~11 min.	~6 min.	AMD Turion 64×2, 1.60 GHz
	7 min.	~2 min.	Intel Core2 Quad 2.4 GHz

10,000 particles
No geometry losses

CUDA

Compute Unified Device Architecture

CUDA kernel-functions

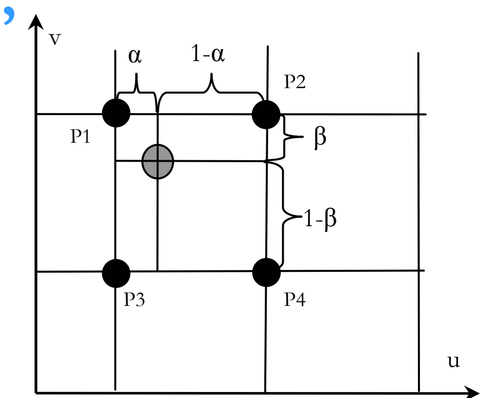
- **Track** (field maps, particle coordinate and velocity)
- **Losses** (setup geometry, particle coordinate)
- **Rho** (particle coordinate)
- **FFT** (charge density function)
- **PoissonSolver** (Furies coefficients)
- **E_SC** (electric field potentials)

__global__ void Track ( )

- Function with many parameters. Use variable type __constant__ :
  - __device__ __constant__ float d_float[200];
  - __device__ __constant__ int d_int[80];
- Particle number corresponds:
  - int n = threadIdx.x+blockIdx.x*blockDim.x;
- Number of "if, goto, for" should be dicreased
- Linear interpolation field maps
  - texture < float, 2, cudaReadModeElementType > Ex_TexRef;
  - Ex_TexRef.filterMode = cudaFilterModeLinear;

$$P = \left[(1-\alpha)P1 + \alpha P2\right](1-\beta) + \left[(1-\alpha)P3 + \alpha P4\right]\beta$$

$$0 \le \alpha, \beta \le 1$$

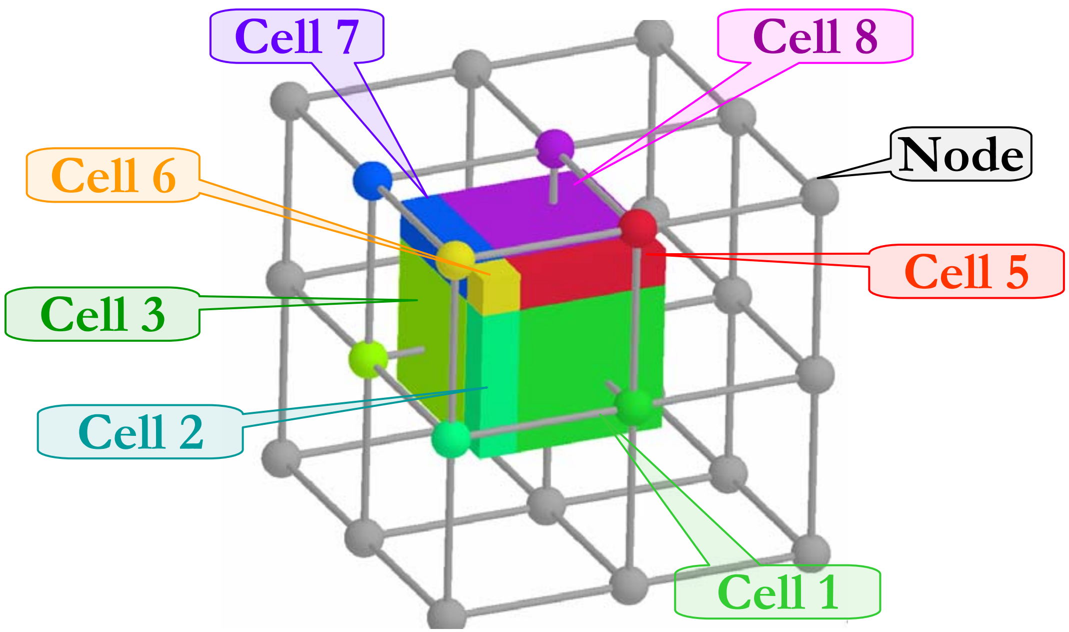


__global__ void Losses ( )

- Geometry structure consist of triangles.
- Block threads copies the triangle nodes from global to shared memory.
- Threads synchronize after copy operation __syncthreads()
- Particle number corresponds to thread number:
  - int n = threadIdx.x+blockIdx.x*blockDim.x;
- Check particles and triangles match

__global__ void Rho

- Calculate charge impact in nodes of the mesh from particle with number n = threadIdx.x + blockIdx.x*blockDim.x
- Particle gives impact to 8 nodes. Therefore, each thread store 16 values: 8 – numbers of node and 8 – values of charge density.
- For each node summarize all impacts of charge density.



__global__ FFT ( )

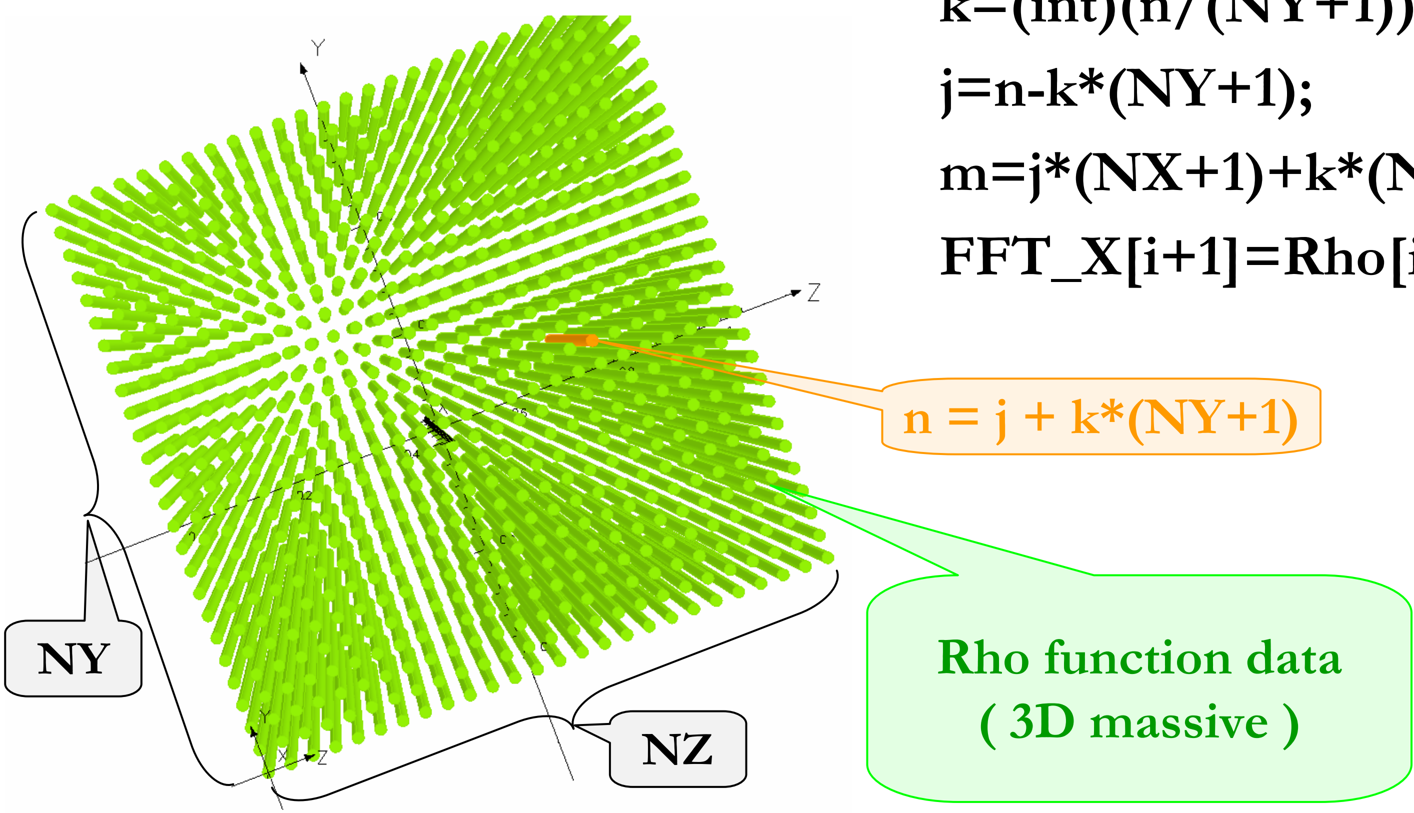
- Used real FFT for sin(πn/N) basis function
- 3D transform consist of three 1D FFT for each axies: X, Y, Z
- int n = threadIdx.x+blockIdx.x*blockDim.x;

k=(int)(n/(NY+1));

j=n-k*(NY+1);

m=j*(NX+1)+k*(NX+1)*(NY+1);

FFT_X[i+1]=Rho[i+m];



__global__ PoissonSolver ( )

- Thread number

  int n = threadIdx.x+blockIdx.x*blockDim.x;

- Each thread calculate Furier coefficient PhiF of potential function Phi

  $PhiF_{ind(i,j,k)} = -RhoF_{ind(i,j,k)} / (kx_i^2 + ky_j^2 + kz_k^2)$

  in the node:

  ind(i,j,k)=i+j*(NX+1)+k*(NX+1)*(NY+1),

  where

  k=(int)(n/(NX+1)*(NY+1));

  j=(int)(n-k*(NX+1)*(NY+1))/(NX+1);

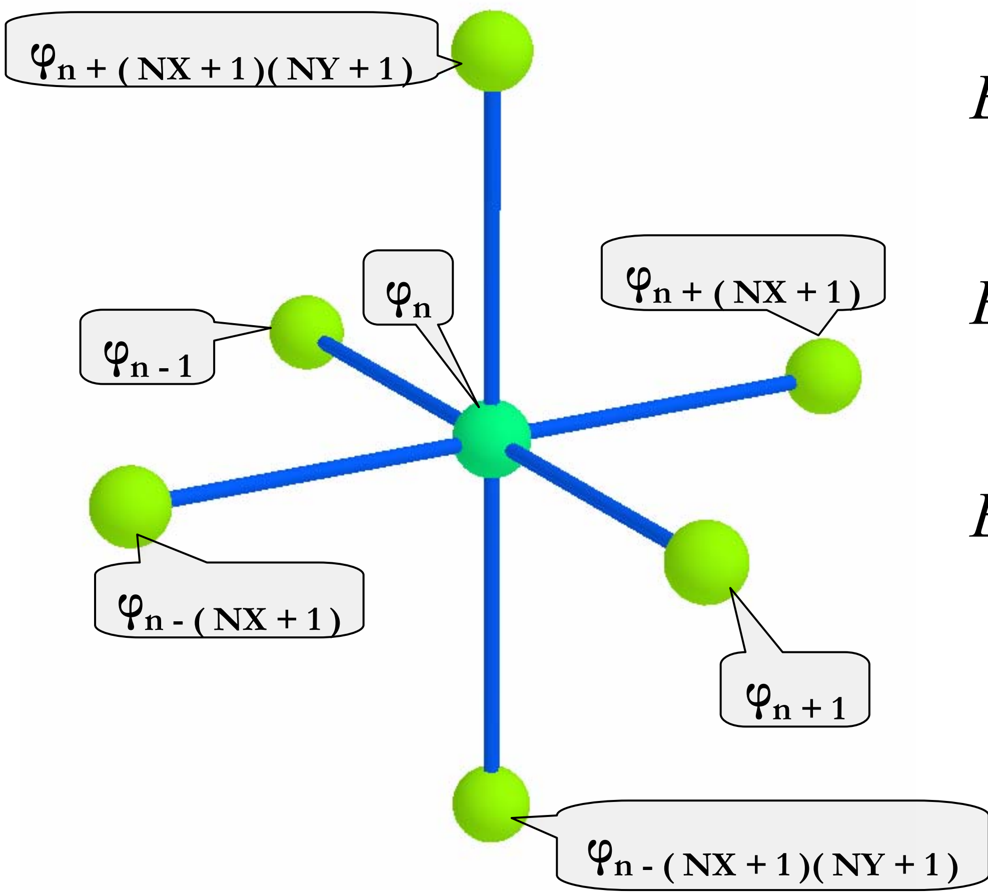
  i=n-j*(NX+1)-k*(NX+1)*(NY+1);

- RhoF –Furier coefficients of charge density function Rho.

__global__ E_SC ( )

- Electric field calculation at mesh node

int n = threadIdx.x+blockIdx.x*blockDim.x+st_ind

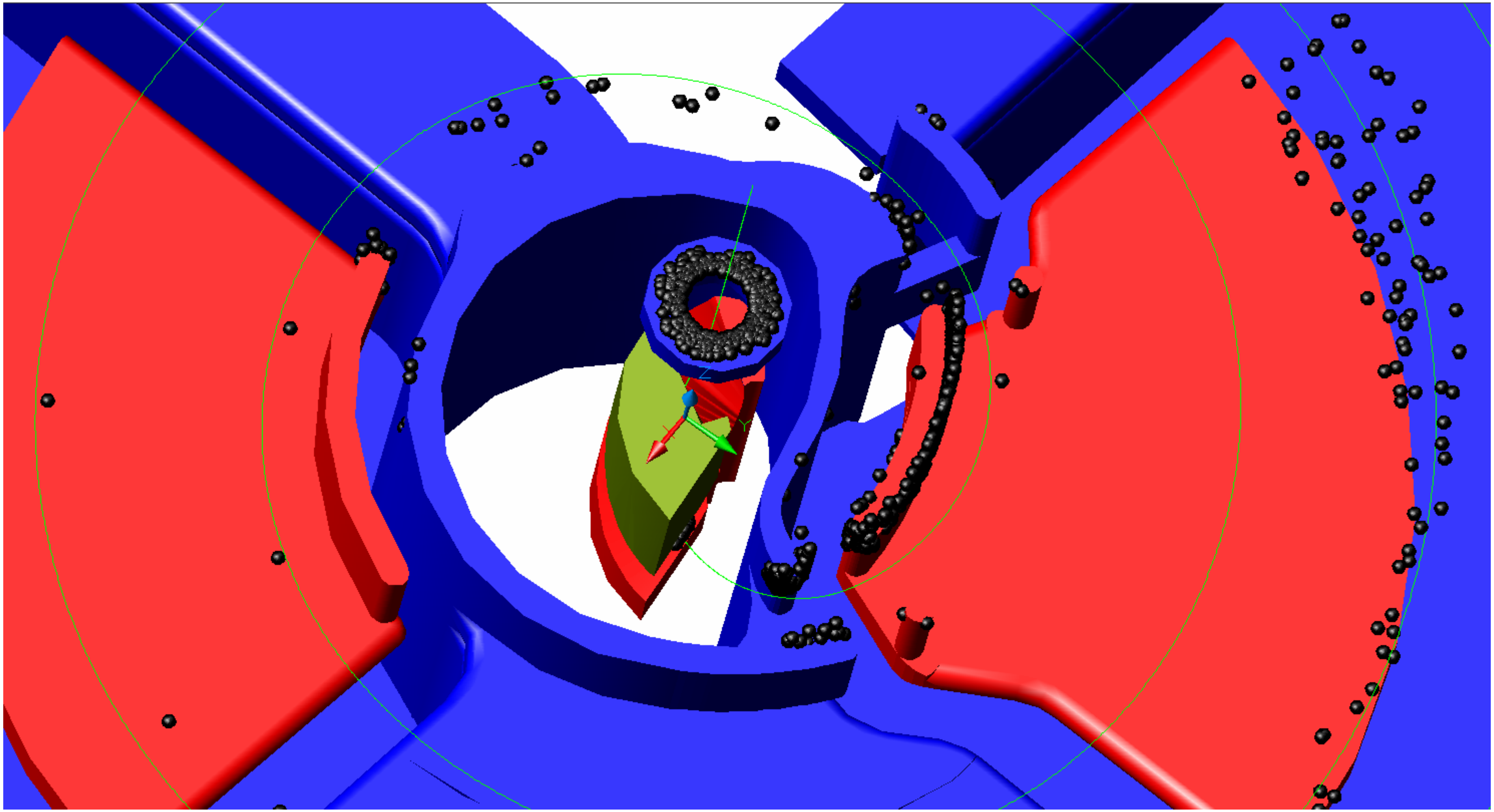


$$E_x = -\frac{\varphi_{n+1} - \varphi_{n-1}}{2h_x}$$

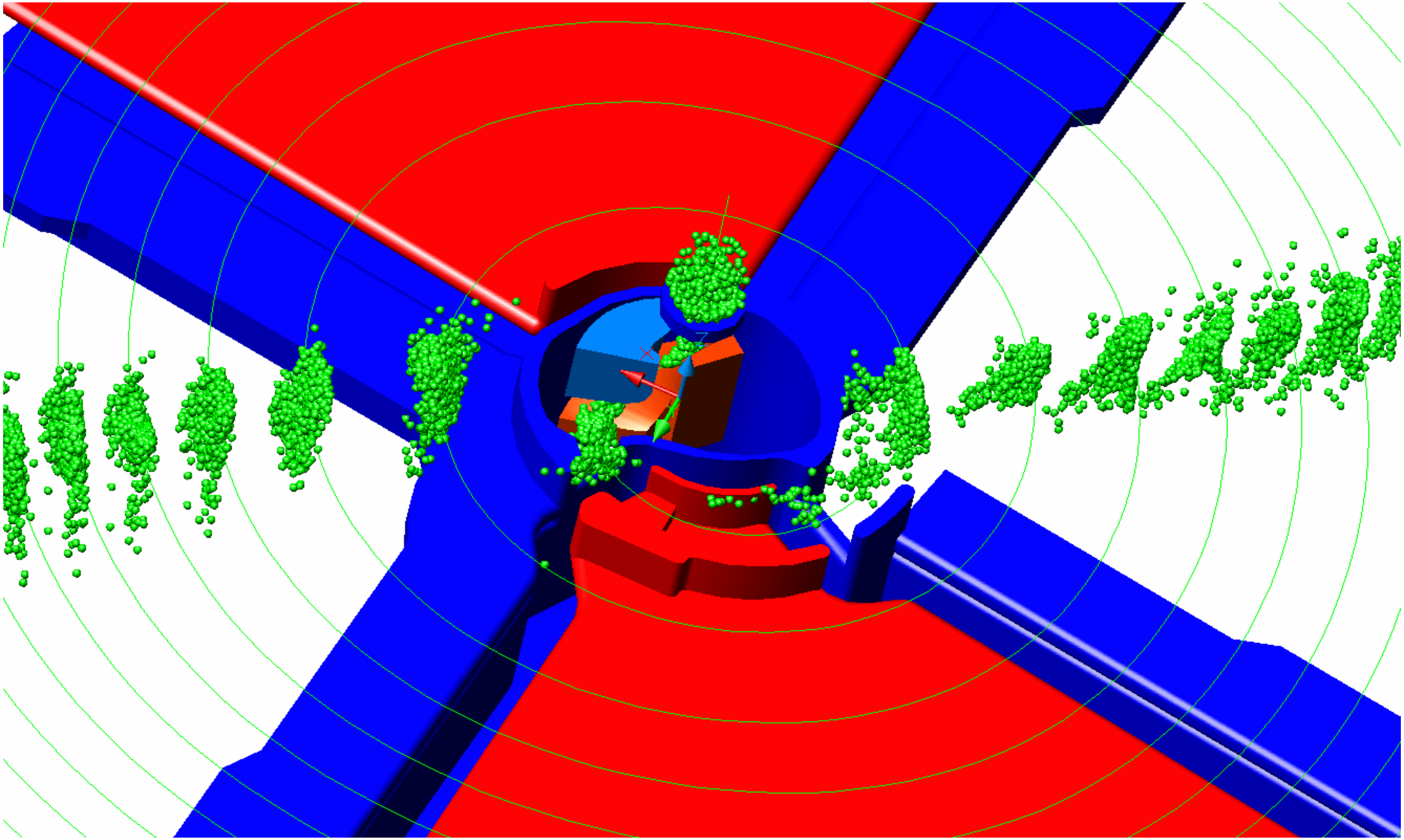
$$E_y = -\frac{\varphi_{n+(Nx+1)} - \varphi_{n-(Nx+1)}}{2h_y}$$

$$E_z = -\frac{\varphi_{n+(Nx+1)(Ny+1)} - \varphi_{n-(Nx+1)(Ny+1)}}{2h_z}$$

Particle losses



Bunch acceleration



GeForce 8800GTX

CUDA kernel-function*	Time, [ms]		Rate, [x]
	CPU**	GPU	
Track	486	30	16
Losses	6997	75	93
Rho	79	6	14
Poisson/FFT	35	3	13
E_SC	1.2	0.8	1.4
Total	7598	114	67

*Mesh size: 2^5 x 2^5 x 2^5. Number of particles: 100,000 triangles: 2054

**CPU 2.4 GHz

GeForce 8800GTX

Number of particles	Time calculation		Rate, [x]
	CPU*	GPU	
1,000	3 min. 19 sec.	12 sec.	17
10,000	34 min. 14 sec.	42 sec.	49
100,000	5 h. 41 min.	6 min.	56
1,000,000	2 days 8 h. 53 min.	1 h.	60

***CPU 2.4 GHz**

Tesla C1060

Number of particles	Time calculation		Rate, [x]
	CPU 2.5GHz	GPU C 1060	
1,000	3 min. 12 sec.	11 sec.	18
10,000	32 min. 24 sec.	27 sec.	72
100,000	5 h. 14 min. 31 sec.	3 min. 34 sec.	88
1,000,000	2 days 4 h. 25 min.	34 min. 29 sec.	91

without space charge effect

НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

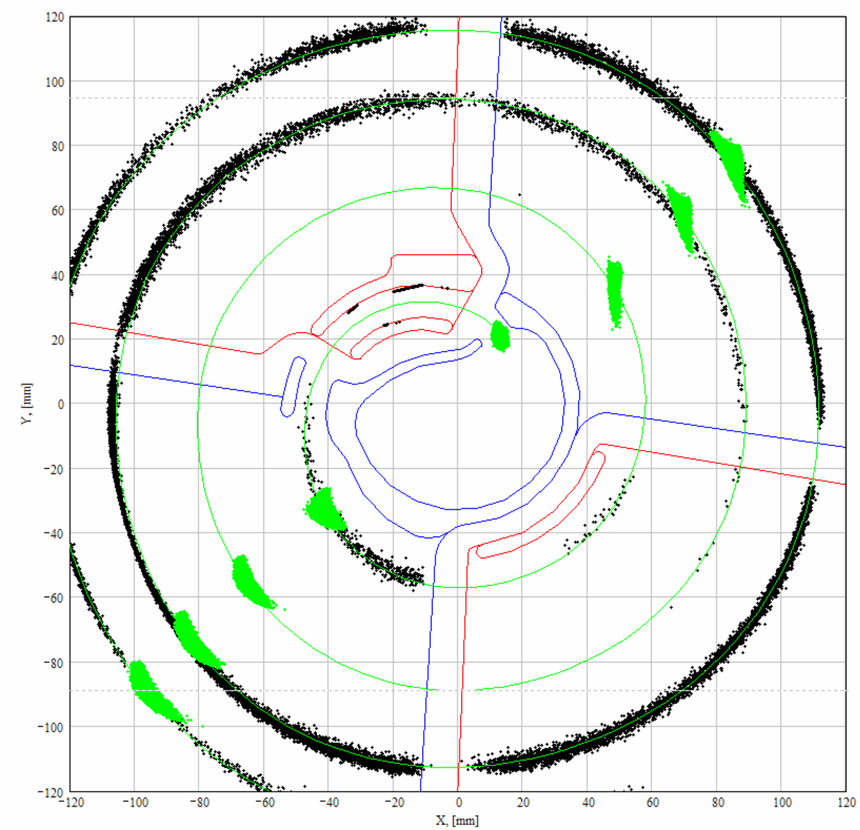
Tesla C1060

Number of particles	Time calculation		Rate, [x]
	CPU 2.5 GHz	GPU C 1060	
10,000	33 min. 36 sec.	44 sec.	45
100,000	5 h. 28 min. 12 sec.	5 min. 4 sec.	65
1,000,000	2 days 8 h. 27 min.	50 min. 17 sec.	67

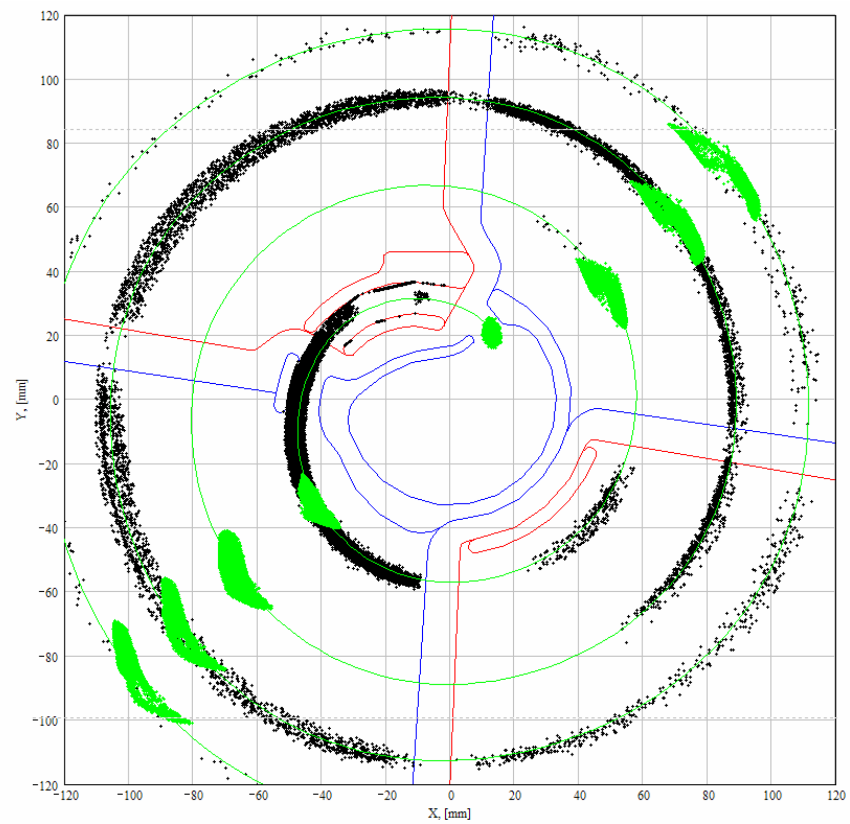
with space charge effect

НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

Space charge effect



I ~ 0
Losses 24%



I = 4 mA
Losses 94%

Conclusions

- Inexpensive technology.
- Increased performance 60-90 times gives a chance to perform the whole cyclotron computer modeling.
- Very careful programming is required that is the price for the obtained high performance of calculations.
- Application to other complicated simulation, related to the accelerator physics (beam halo etc), is possible.

НОЦ "ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ"
APPLIED PARALLEL COMPUTING E&R CENTER

Education and Research Center «Applied Parallel Compute»

http://parallel-compute.com
info@parallel-compute.com