

# J-PARC Main Ring 大強度運転時における 磁場リップルの影響

~GPU を使った粒子トラッキングシミュレーション~

高エネルギー加速器研究機構

J-PARC Main Ring 電磁石・電源グループ

栗本 佳典

# 目次

- 大強度陽子加速器J-PARC MRの性能と電磁石電源
- GPUを使った粒子トラッキングシミュレーション
- 結果と考察

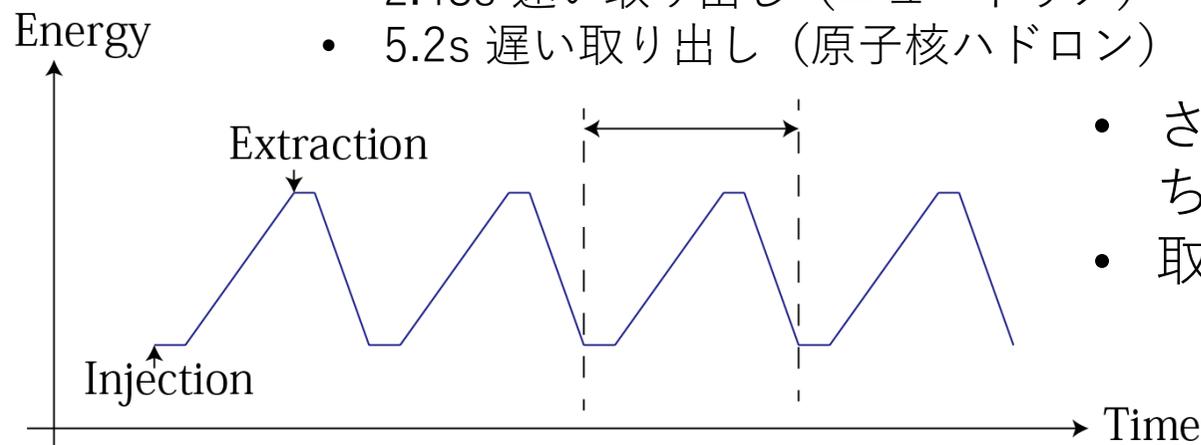
# Introduction

# J-PARC 30 GeV シンクロトロン (Main Ring)



周長	1568 m
入射エネルギー	3 GeV
取出しエネルギー	30 GeV
バンチ数	8
ビーム取出し周期	2.48 s (速取), 5.20 s (遅取)
ビーム強度	<b>500 kW (速取), 50 kW (遅取)</b>

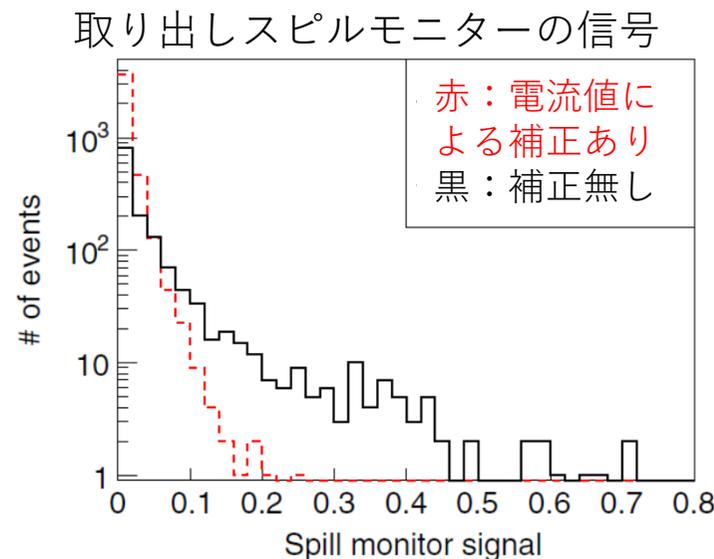
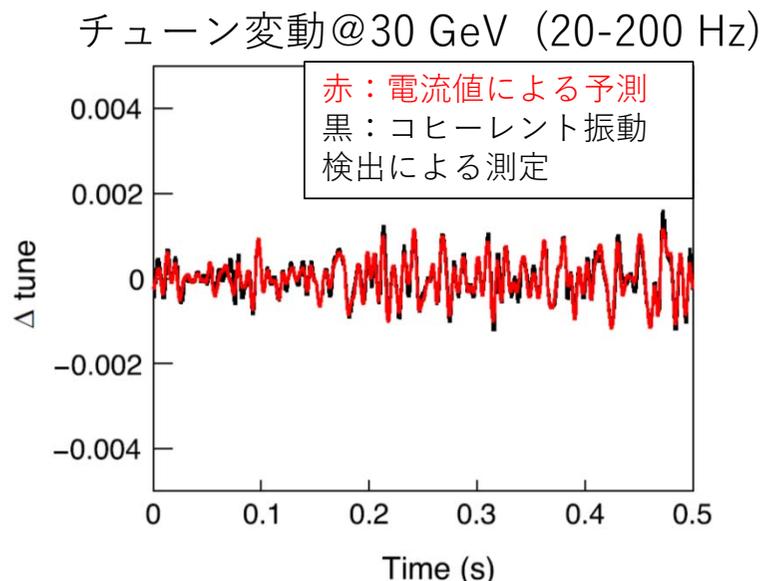
- 2.48s 速い取り出し (ニュートリノ)
- 5.2s 遅い取り出し (原子核ハドロン)



- さらなる**大強度化、ビームロス低減** (どちらも)
- 取り出しビーム平坦化 (遅い取り出し)

# 電源電流リップルと遅い取り出し

内藤大地さんらの論文 Phys. Rev. Accel. Beams **22**, 072802 (2019)



- Spill Dutyが3%から31%に改善

チューン (ベータatronチューン) : 四極磁場できまる横方向振動数/1ターン

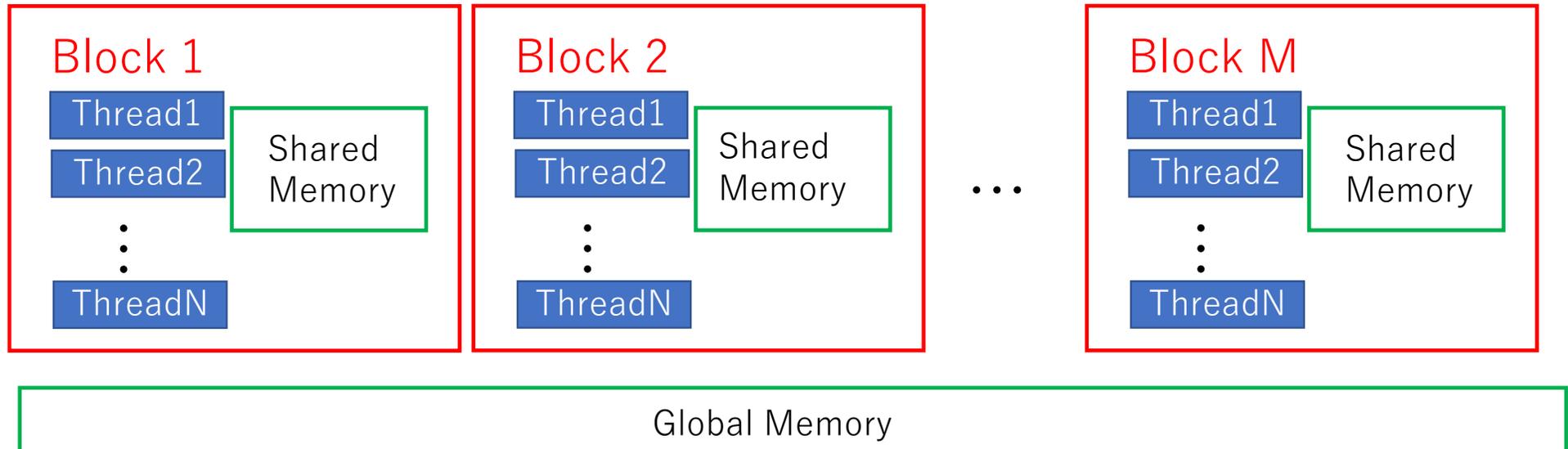
- チューン変動が電源電流から予測される値と一致した (200 Hz)
- 電流値によるチューン変動をFPGAを使ってリアルタイムで予測しチューンを補正した。
- これにより遅い取り出しビームの平坦性が改善することも分かった。



より大強度の速い取り出し運転での電源リップルの影響は？ 空間電荷効果の大きい入射エネルギー付近での粒子トラッキングシミュレーションが必要

# GPUを使った粒子トラッキングシミュ レーション

# GPUの構成



- ブロックとスレッドの多層構造
- 各スレッドに演算が割り当てられる。 **並列計算が可能**
- 各ブロックにはブロック内スレッドがアクセスできる共有メモリがある。 **高速だが小容量48~96KB**
- CUDA C/C++ というC++ベースのプログラミング言語を使用する。

# 粒子トラッキングシミュレーションの概要



単粒子力学：ある点からある点への写像は並列に計算可能



PIC シミュレーション：**写像には電荷の空間分布が必要**→完全に並列にはできない。

# 単粒子力学

## ハミルトニアン

$$H(x, p_x, y, p_y, \sigma, p_\sigma; s) = p_\sigma - (1 + hx) \sqrt{(1 + \delta)^2 - p_x^2 - p_y^2} - e \frac{A_s(x, y)}{p_0}$$
$$\approx \frac{p_x^2 + p_y^2}{2} + \frac{p_\sigma^2}{2\gamma_0^2} - hx - hx p_\sigma - \frac{p_x^2 + p_y^2}{2} p_\sigma - e \frac{A_s(x, y)}{p_0}$$

$$p_\sigma = \frac{E - E_0}{\beta_0 p_0 c}, \quad \sigma = s - \beta_0 t, \quad \delta = \frac{p - p_0}{p_0} \approx p_\sigma - \frac{1}{2\gamma_0^2} p_\sigma^2,$$

ハミルトン方程式が解けない場合は  $H = H_0 + V$  ( $H_0$  および  $V$  単体では解ける) とハミルトニアンを分割して、以下のシンプレクティック変換を行う。(  $L$  はコンポーネントの長さ)

$$e^{iH_0 L/2} e^{iVL} e^{iH_0 L/2} \quad \text{または} \quad e^{iH_0 aL} e^{-iVbL} e^{-iH_0 cL} e^{iVbL} e^{-iH_0 aL}$$

$$a = \frac{1}{2} \left( 1 - \frac{1}{\sqrt{3}} \right), \quad b = \frac{1}{2}, \quad c = \frac{1}{\sqrt{3}}$$

実装済み：偏向、四極、六極、薄い多極成分、ドリフト

# 空間電荷効果

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\varphi(x, y, z) = -\frac{\rho(x, y, z)}{\epsilon_0} \quad \longrightarrow \quad g(z) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)u(x, y) = -\frac{g(z)f(x, y)}{\epsilon_0}$$

2D 近似  
 $\sigma_x, \sigma_y \ll \sigma_z$       (長いビーム)

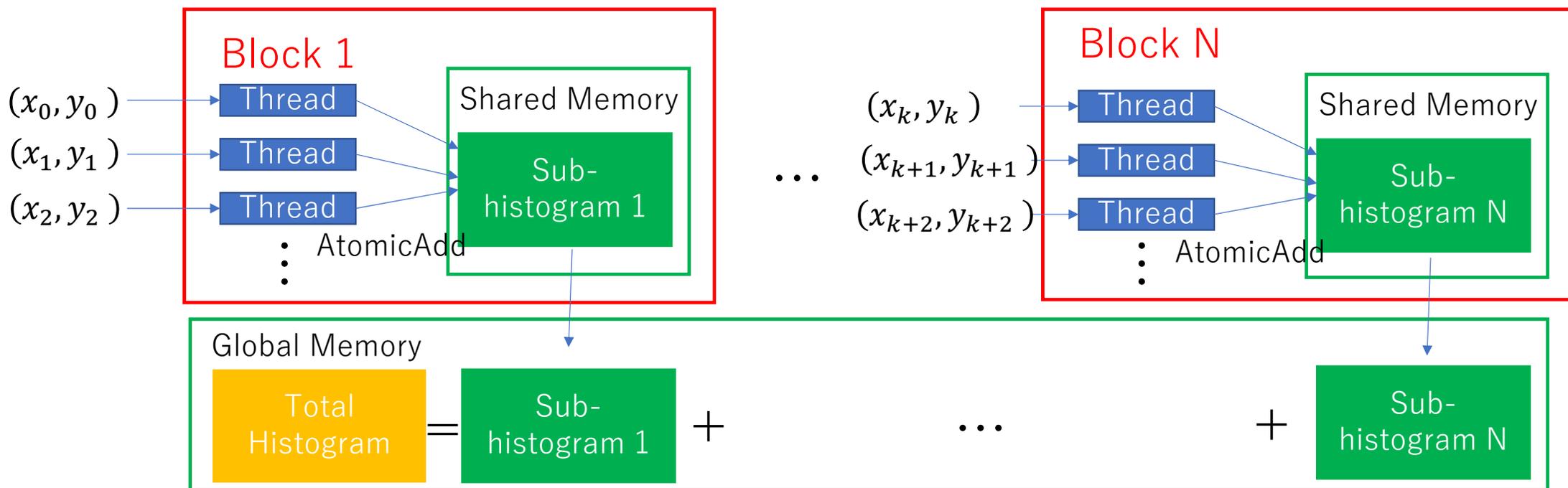
1. 電荷分布  $f(x, y)$  and  $g(z)$  の作成 ← もっとも時間コストが高い演算 (後述)

2. 境界条件<sup>a</sup>を含む二次元ポアソン方程式で  $u(x, y)$  を得る

3. 勾配  $-\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right) Ag(z)u(x, y) \times L$  を計算しキック       $A = \frac{e}{m_p \gamma_p^3 \beta_p^2 c^2}$        $L =$  移動距離

a. ダクトの形に応じて極座標と直交座標二種類のソルバーを実装。ソルバーはFFTと行列演算中心なので、GPUの並列計算が有効。

# 共有メモリを用いた電荷分布の作成



- 各ブロックの共有メモリにSub-histogramを割り当て、**複数のヒストグラムを並列に計算**し、あとで和をとる。
- 各スレッドが平行に共有メモリに電荷を足していくので、**他をブロックする和AtomicAdd関数**を使う。(CUDA LIBRARYにある。)

J-PARC MR の場合のシミュレーション

# シミュレーション条件

## 速い取り出しモードの運転条件

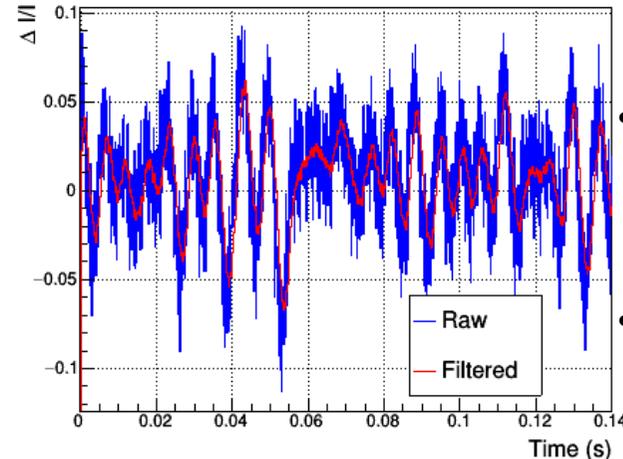
- 横方向チューン： $(v_x, v_y) = (21.35, 21.45)$
- バンチ内粒子数： $3 \times 10^{13}$  ppb
- マクロ粒子数：200000
- 加速電圧：基本波 160 kV 二倍高調波 110 kV
- 入射時バンチファクタ  $I_{\text{average}}/I_{\text{peak}}$ ：0.2
- 入射時横方向 $2\sigma$  エミッタンス： $16\pi$  mm · mrad
- アパーチャー： $60\pi$  mm · mrad
- コンポーネント数：約1600 / turn
- 空間電荷効果によるキック：1000カ所/turn
- エラーなしと電源リップルのみで比較



**2700 秒/10000 turns** using NVIDIA TESLA V100

**MRで使用しているのシミュレーション(CPU)の  
10倍以上の速度**

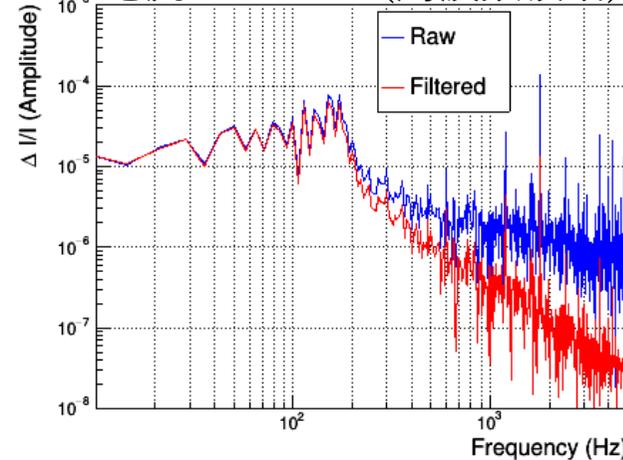
電流リップル (時間領域)



- 磁場リップルとして、**実際に測定された電源電流のリップル**を使用した。

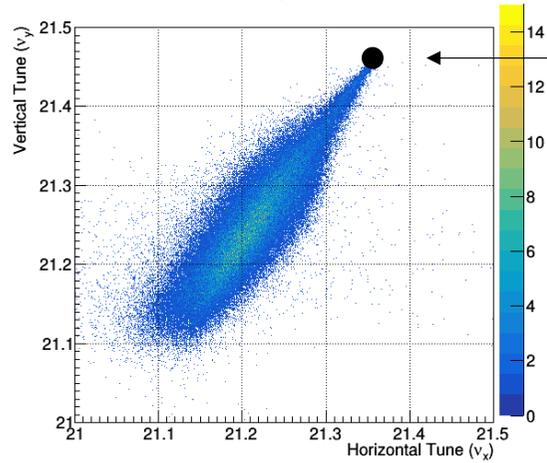
- ダクトで高周波磁場がシールドされる効果 (森田, 2015年会) をフィルタで模擬

電流リップル (周波数領域)



# シミュレーション結果

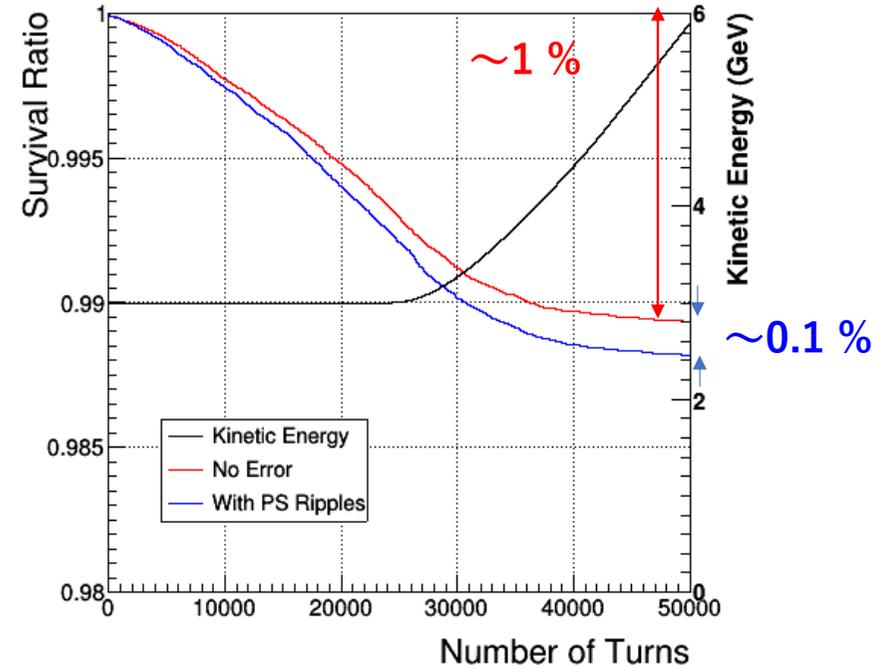
## チューン分布 $\nu_x$



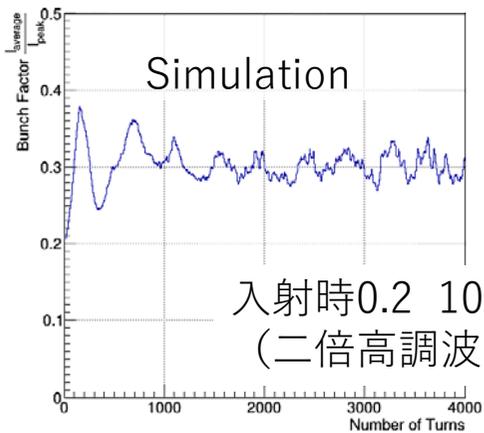
磁石による設定チューン  
(21.35, 21.45)

チューンの広がり  
 $\Delta\nu = 0.3-0.4$

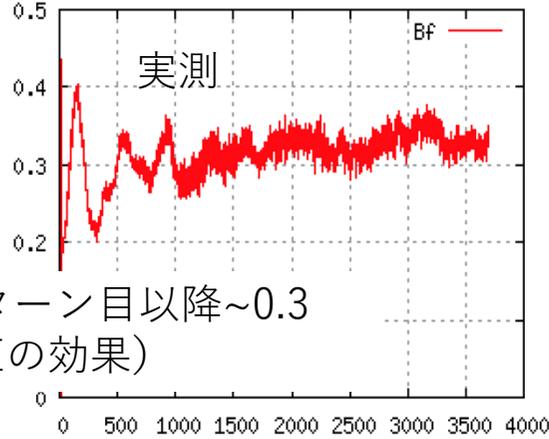
## ビーム生存率



## バンチファクタ $I_{\text{average}}/I_{\text{peak}}$



入射時0.2 1000ターン目以降 $\sim 0.3$   
(二倍高調波電圧の効果)

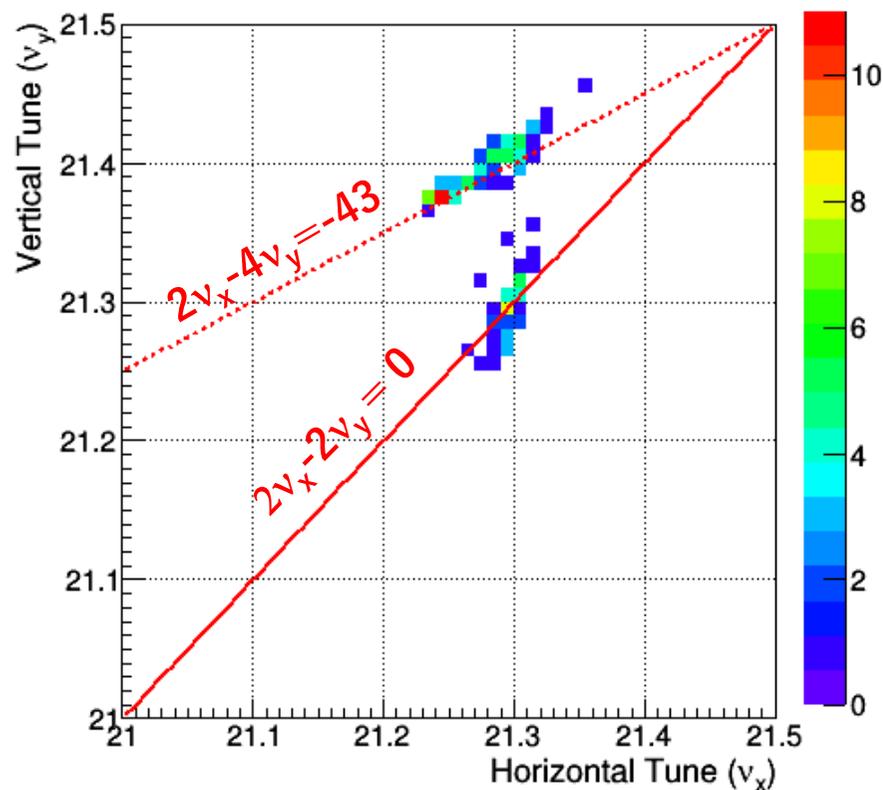


電源リップルの効果はビーム生存率で  
**0.1%**程度だが**一方でエラー無しです**  
**でに1%程度のビームロスがある。**

**➡** こちらの調査を優先すべき

# 損失粒子のチューン分布

損失直前のチューン分布 (10 turn 平均)



- 損失したマクロ粒子のIDを記録しておいて、もう一度同じプログラムを走らせ、軌道をターン毎に記録した。(エラーなしのとき)

損失する粒子は、非構造共鳴 $2v_x - 4v_y = -43$ と構造共鳴付近 $2v_x - 2v_y = 0$  にいることが分かった。

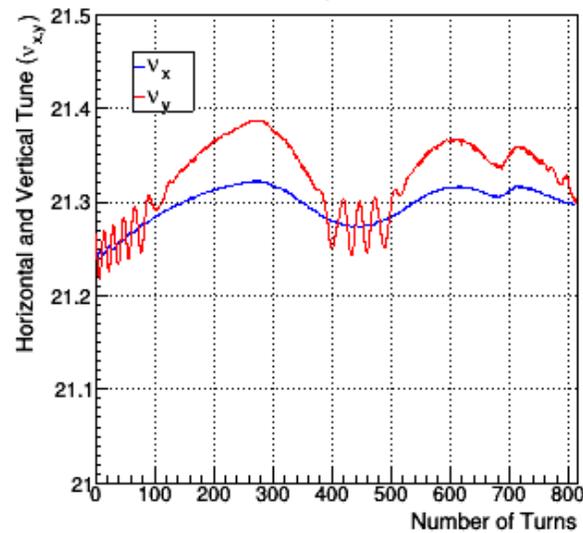
- シミュレーションに8極や12極の外部磁場は入れていないため、空間電荷分布がつくる共鳴である
- 非構造共鳴があるので、3回対称性を破るダクトの配置(菱形近似しているUFOダクト)をやめてみたが、結果は不変。空間電荷効果の計算箇所も3回対称。

# 損失粒子の詳細な運動

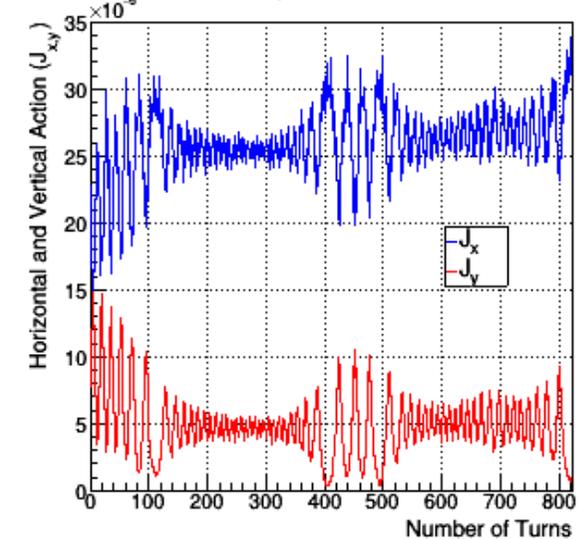
$$2\nu_x - 2\nu_y = 0$$

- 作用の時間変化に明確な結合共鳴の兆候がある。 $J_x$  と  $J_y$  が同じ量だけ、交換している。
- 構造共鳴のため、共鳴幅は広いと考えられる。
- 設定チューンを (21.35, 21.45) を右側 (21.45, 21.45) にずらすと、損失は大幅に増え、ほとんどがこの共鳴で損失するようになる。

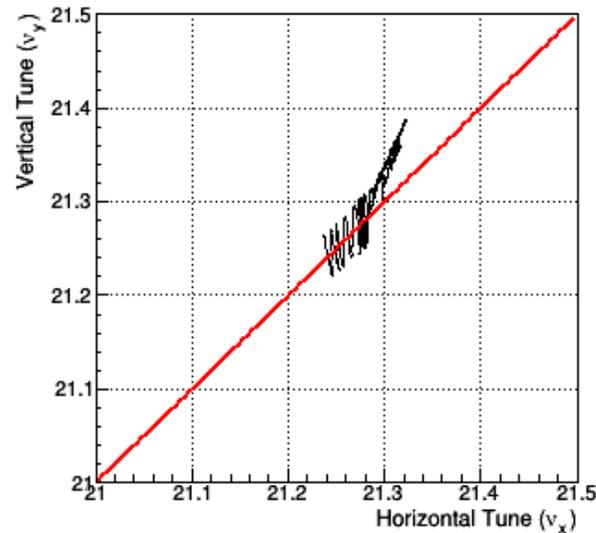
### チューン $\nu_x, \nu_y$ の時間変化



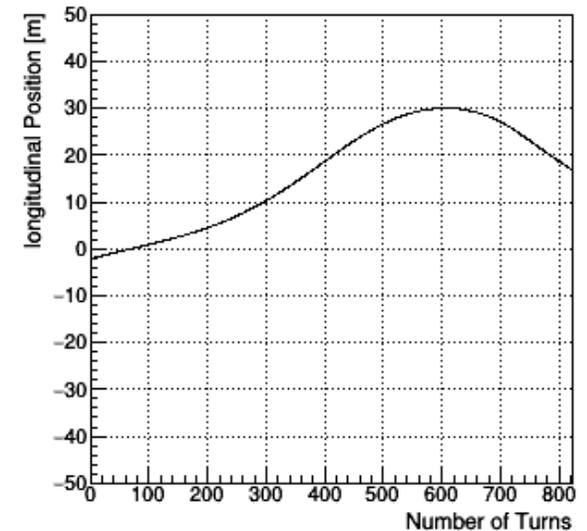
### 作用 $J_x, J_y$ の時間変化



### チューンダイアグラム上の軌跡



### 縦方向位置の時間変化

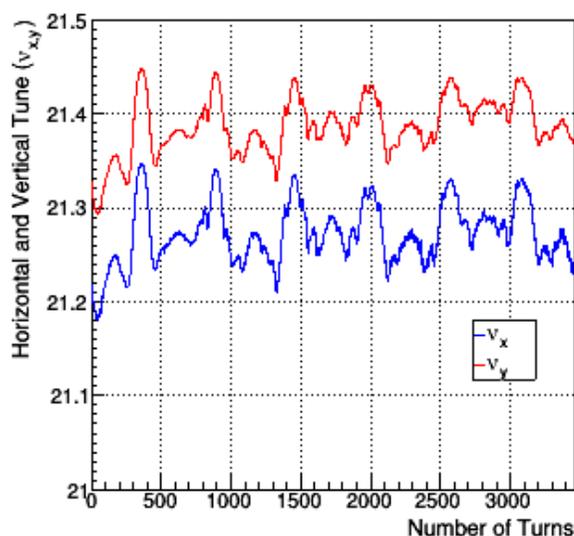


# 損失粒子の詳細な運動

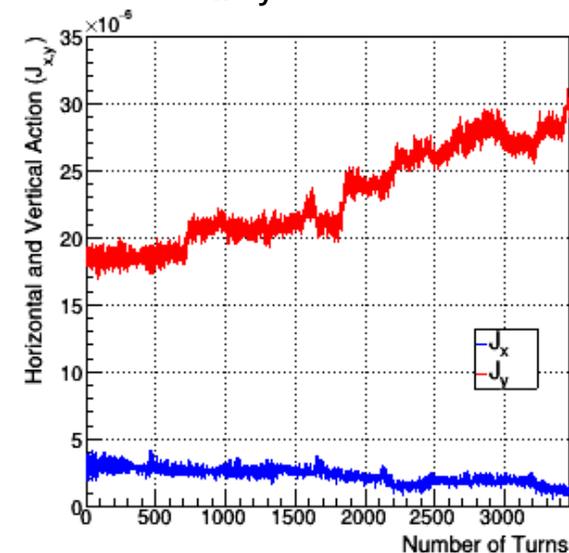
$$2\nu_x - 4\nu_y = -43$$

- 作用の時間変化に明確な共鳴の兆候はなく、じわじわエミッタンスが増えている。
- すべての粒子について似たような傾向。しかし、図は $J_x$ 増、 $J_y$ 減だが、逆の場合やどちらも増大の場合もある。
- 非構造共鳴のため、共鳴幅が狭く、短期間に起こる少量のエミッタンス変化を繰り返しているのか？
- なぜエラー無し非構造共鳴による損失が起きているのか、さらに調査が必要

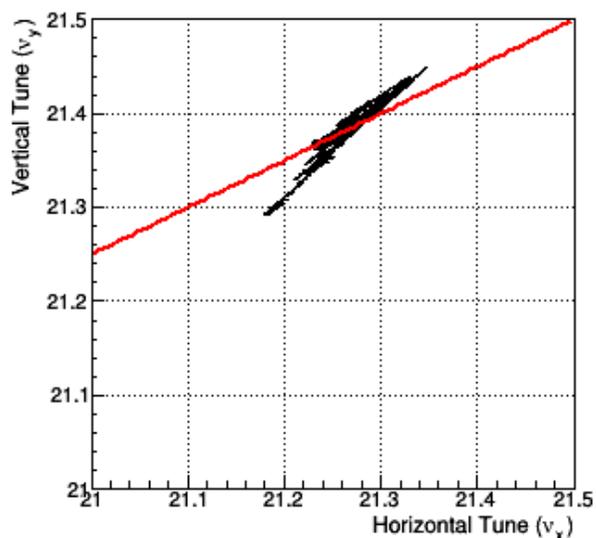
チューン $\nu_x, \nu_y$ の時間変化



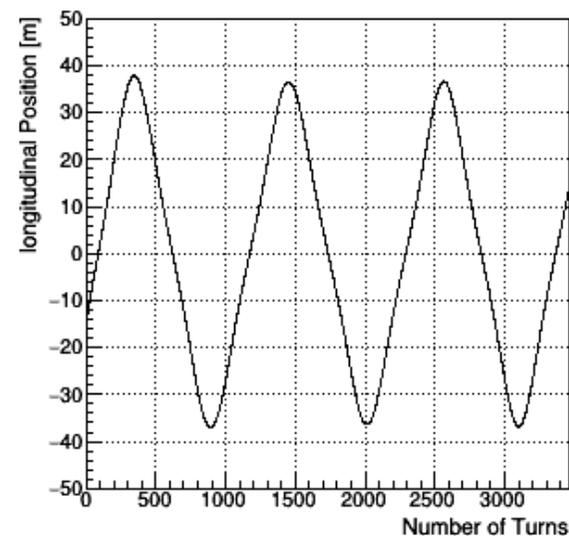
作用 $J_x, J_y$ の時間変化



チューンダイアグラム上の軌跡



縦方向位置の時間変化



# まとめ

- 電源による磁場リップルがJ-PARC MR大強度運転にどの程度影響するかをシミュレーションで調べた。
- GPUで動く空間電荷効果を含んだトラッキングシミュレーションコードを開発し、現行MRが使っているのCPUのものより、10倍以上速くなった。
- シミュレーションの結果、磁場リップルの存在はビーム損失にして0.1%しか寄与しないことが分かった。
- しかし、バンチ内陽子の電荷分布自身がつくる共鳴 $2v_x - 4v_y = -43$ および $2v_x - 2v_y = 0$ による1%の損失が存在する。

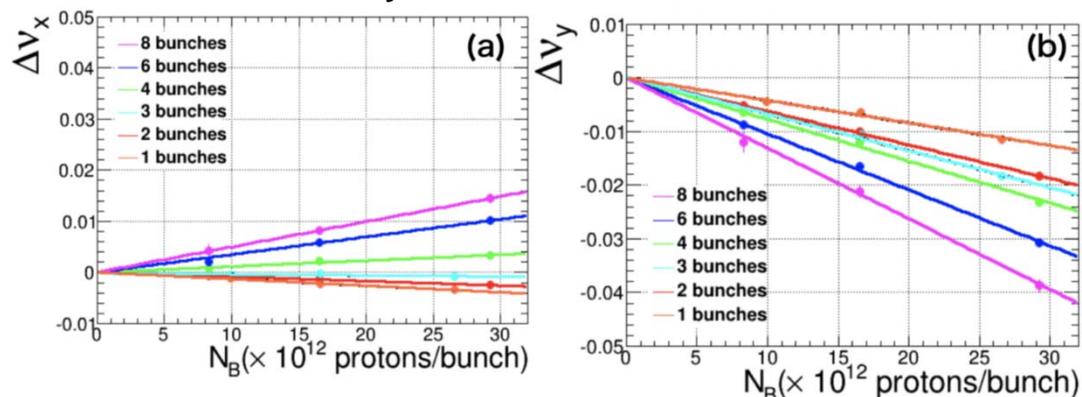
ただし、現時点ではあくまで一つのシミュレーションの結果に過ぎない。今後、より詳細な解析や、現存するトラッキングシミュレーションとの比較で、本結果の信頼度を高めていきたい。

Thanks to KEK大見さん、原研：發知さん、田村さん（アドバイス）、NVIDIA（GPU贈与）

Backup

# ダクト径のチューニング

コヒーレントチューンシフト測定  
A. Tobayashi IPAC 2019 MOPGW036



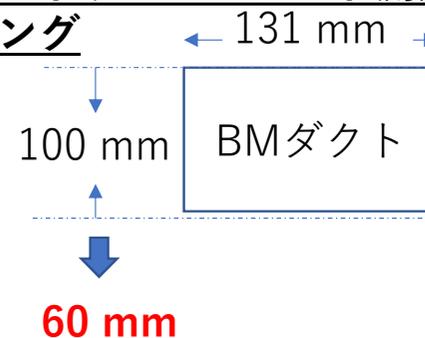
今回の空間電荷によるポテンシャル計算

$u_{self}/\gamma_p^2 + u_{image}$  を  $(u_{self} + u_{image})/\gamma_p^2$  で近似

→ 虚電荷の効果を小さく見積もっている

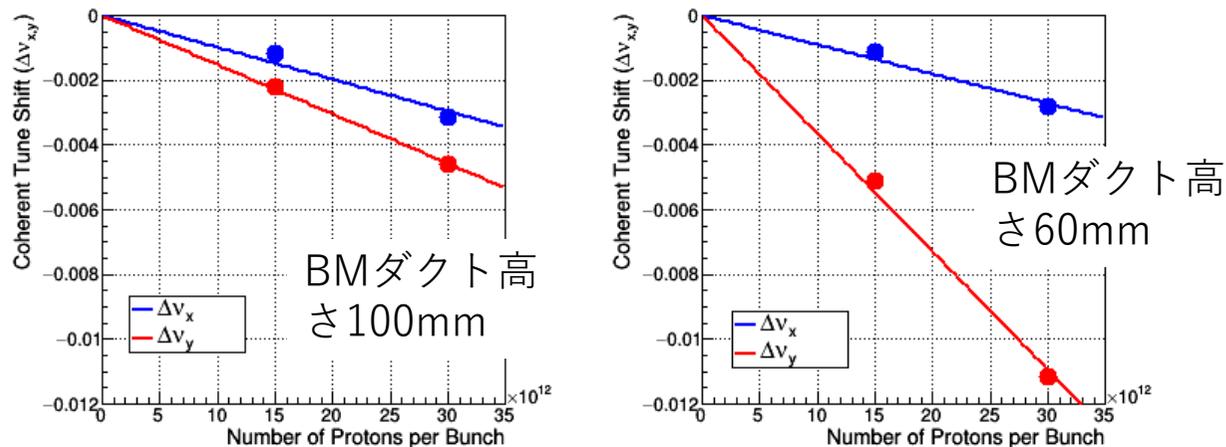
本来は、ダクト境界の解  $u_{self} + u_{image}$  と無限解  $u_{self}$  両方が必要。→ 今後実装予定。

コヒーレントチューンシフト測定によるチューニング



今回は、コヒーレントチューンシフトの測定がシミュレーションと合うように、偏向磁石用ダクト高さを変更した。

本シミュレーション (1 bunch) によるコヒーレントチューンシフト



# Particle Tracking with Space Charge Effect using GPU

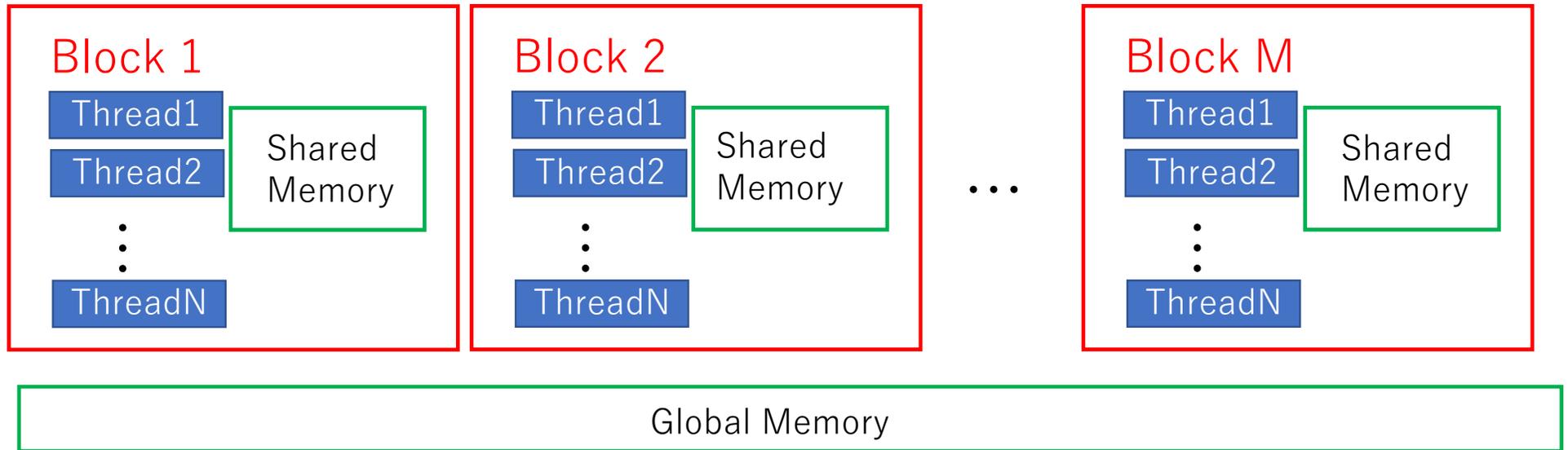
Yoshi Kurimoto

J-PARC MR Power Converter Group

# Motivation (Questions)

- What is the source of the beam loss ?
- How do rippling magnetic fields affect the beam loss ?
- Can we simulate the beam faster using GPU?

# GPU



- Each operation can be assigned to each thread.
- **Execution of each thread can be parallelly done.**
- Threads in a common block can access the shared memory.
- Shared memory is limited (12288 float words) but **very fast**

# Example

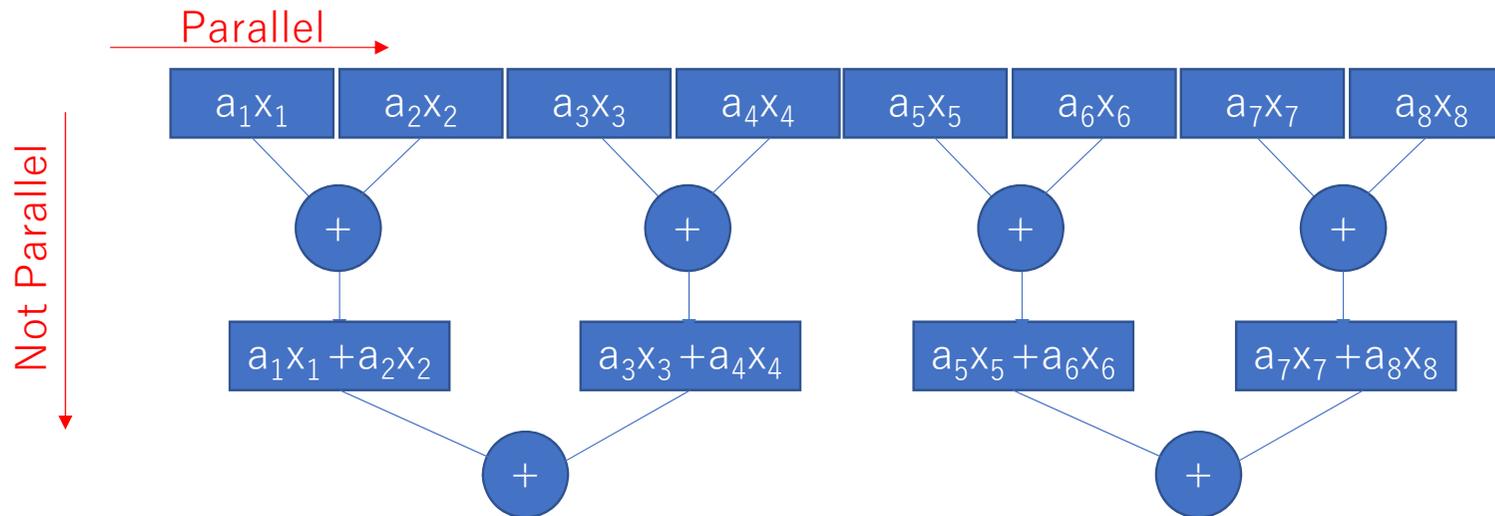
$$\sum_{i=1}^N a_i x_i = a_1 x_1 + a_2 x_2 + \dots + a_N x_N$$

Matrix multiplication, Fourier transformation ...

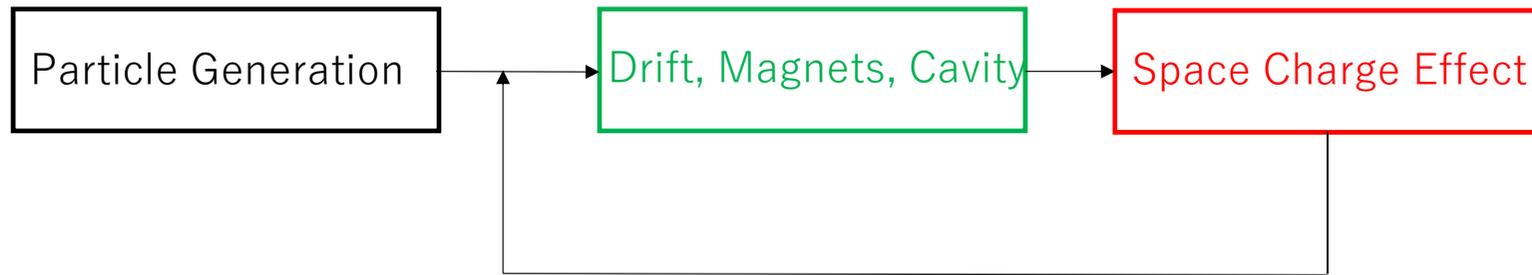
1. Each term can be calculated in parallel



2. Summation also can be partially parallel.



# Overview of the particle tracking simulation



Single Particle Mechanics : Maps between two locations can be calculated in parallel



PIC simulation : Maps requires spatial distributions of particles

# Single Particle Mechanics

# Hamiltonian

$$\begin{aligned} H(x, p_x, y, p_y, \sigma, p_\sigma; s) &= p_\sigma - (1 + hx) \sqrt{(1 + \delta)^2 - p_x^2 - p_y^2} - e \frac{A_S(x, y)}{p_0} \\ &\approx \frac{p_x^2 + p_y^2}{2} + \frac{p_\sigma^2}{2\gamma_0^2} - hx - hxp_\sigma - \frac{p_x^2 + p_y^2}{2} p_\sigma - e \frac{A_S(x, y)}{p_0} \end{aligned}$$

$$p_\sigma = \frac{E - E_0}{\beta_0 p_0 c}, \quad \sigma = s - \beta_0 t, \quad \delta = \frac{p - p_0}{p_0} \approx p_\sigma - \frac{1}{2\gamma_0^2} p_\sigma^2,$$

Then just have to solve the Hamilton equations. But not always solvable

# Symplectic Map

If a Hamilton equation about  $H$  can not be solved, The Hamiltonian is divided into two components like  $H = H_0 + V$  so that  $e^{iH_0s}$  and  $e^{iVs}$  can be exactly calculated (Hamilton Equations for  $H_0$  and  $V$  can be solved ) Then, a map between two components (Length : L) is approximately calculated as

$$e^{iH_0L/2} e^{iVL} e^{iH_0L/2}$$

or

$$e^{iH_0aL} e^{-iVbL} e^{-iH_0cL} e^{iVbL} e^{-iH_0aL}$$

$$a = \frac{1}{2} \left( 1 - \frac{1}{\sqrt{3}} \right), b = \frac{1}{2}, c = \frac{1}{\sqrt{3}}$$

# Bending Magnets

$$eA_s(x) = -hx - \frac{1}{2}h^2x^2$$

$$H \approx \underbrace{\frac{p_x^2 + p_y^2}{2} - \boxed{hxp_\sigma} + \boxed{\frac{1}{2}h^2x^2}}_{H_0} \underbrace{- \frac{p_x^2 + p_y^2}{2}p_\sigma + \boxed{\frac{p_\sigma^2}{2\gamma_0^2}}}_{V}$$

dispersion Weak focus                      slippage

- $e^{iH_0s}$

$$x(s) = x(0) \cos hs + \frac{p_x(0)}{h} \sin hs + \frac{p_\sigma(0)}{h} (1 - \cos hs)$$

$$p_x(s) = p_x(0) \cos hs - hx(0) \sin hs + p_\sigma(0) \sin hs$$

$$y(s) = p_y(0) s + x(0) \quad p_y(s) = p_y(0)$$

$$\sigma(s) = \sigma(0) - x(0) \sin hs + \frac{p_x(0)}{h} (\cos hs - 1)$$

$$-p_\sigma(0)s + \frac{p_\sigma(0)}{h} \sin hs$$

$$p_\sigma(s) = p_\sigma(0)$$

- $e^{iVs}$

$$x(s) = -p_x(0)p_\sigma(0)s + x(0)$$

$$y(s) = -p_y(0)p_\sigma(0)s + y(0)$$

$$\sigma(s) = \sigma(0) + \left( \frac{p_\sigma(0)}{\gamma_0^2} - \frac{p_x(0)^2 + p_y(0)^2}{2} \right) s$$

$$p_x(s) = p_x(0)$$

$$p_y(s) = p_y(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

# Quad Magnets

$$eA_s(x) = -\frac{1}{2}k_1(x^2 - y^2)$$

$$H \approx \underbrace{\frac{p_x^2 + p_y^2}{2} + \frac{1}{2}k_1(x^2 - y^2)}_{H_0} \underbrace{-\frac{p_x^2 + p_y^2}{2}p_\sigma + \frac{p_\sigma^2}{2\gamma_0^2}}_V$$

- $e^{iH_0s}$

$$x(s) = x(0) \cos \sqrt{k_1}s + \frac{p_x(0)}{\sqrt{k_1}} \sin \sqrt{k_1}s$$

$$p_x(s) = p_x(0) \cos \sqrt{k_1}s - \sqrt{k_1}x(0) \sin \sqrt{k_1}s$$

$$y(s) = y(0) \cosh \sqrt{k_1}s + \frac{p_y(0)}{\sqrt{k_1}} \sinh \sqrt{k_1}s$$

$$p_y(s) = p_y(0) \cosh \sqrt{k_1}s + \sqrt{k_1}y(0) \sinh \sqrt{k_1}s$$

$$\sigma(s) = \sigma(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

- $e^{iVs}$

$$x(s) = -p_x(0)p_\sigma(0)s + x(0)$$

$$y(s) = -p_y(0)p_\sigma(0)s + y(0)$$

$$\sigma(s) = \sigma(0) + \left( \frac{p_\sigma(0)}{\gamma_0^2} - \frac{p_x(0)^2 + p_y(0)^2}{2} \right) s$$

$$p_x(s) = p_x(0)$$

$$p_y(s) = p_y(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

# Drift Space

$$eA_s(x) = 0$$

$$H \approx \frac{p_x^2 + p_y^2}{2} - \frac{p_x^2 + p_y^2}{2} p_\sigma + \frac{p_\sigma^2}{2\gamma_0^2}$$

$H_0$

•  $e^{iH_0s}$

$$x(s) = x(0) + (1 - p_\sigma(0))p_x(0)s$$

$$y(s) = y(0) + (1 - p_\sigma(0))p_y(0)s$$

$$\sigma(s) = \sigma(0) + \left( \frac{p_\sigma(0)}{\gamma_0^2} - \frac{p_x(0)^2 + p_y(0)^2}{2} \right) s$$

$$p_x(s) = p_x(0)$$

$$p_y(s) = p_y(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

•  $e^{iVs}$

$$p_x(s) = -\frac{1}{2}k_2(x(0)^2 - y(0)^2)s + p_x(0)$$

$$p_y(s) = k_2x(0)y(0)s + p_y(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

$$x(s) = x(0)$$

$$y(s) = y(0)$$

$$\sigma(s) = \sigma(0)$$

# Sext Magnet

$$eA_s(x) = -\frac{1}{6}k_2(x^3 - 3xy^2)$$

$$H \approx \underbrace{\frac{p_x^2 + p_y^2}{2} - \frac{p_x^2 + p_y^2}{2}p_\sigma + \frac{p_\sigma^2}{2\gamma_0^2}}_{H_0} + \underbrace{\frac{1}{6}k_2(x^3 - 3xy^2)}_V$$

•  $e^{iH_0s}$

$$x(s) = x(0) + (1 - p_\sigma(0))p_x(0)s$$

$$y(s) = y(0) + (1 - p_\sigma(0))p_y(0)s$$

$$\sigma(s) = \sigma(0) + \left( \frac{p_\sigma(0)}{\gamma_0^2} - \frac{p_x(0)^2 + p_y(0)^2}{2} \right) s$$

$$p_x(s) = p_x(0)$$

$$p_y(s) = p_y(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

•  $e^{iVs}$

$$p_x(s) = -\frac{1}{2}k_2(x(0)^2 - y(0)^2)s + p_x(0)$$

$$p_y(s) = k_2x(0)y(0)s + p_y(0)$$

$$p_\sigma(s) = p_\sigma(0)$$

$$x(s) = x(0)$$

$$y(s) = y(0)$$

$$\sigma(s) = \sigma(0)$$

# Acceleration

- Cavity  $V_{RF}$  : Amplitude of RF voltage,  $f_{RF}$  : RF frequency,  $n$  : n-th order harmonics,  $\varphi_s$  : Synchrotron phase

$$\begin{aligned}x(s) &= x(0) & y(s) &= y(0) & \sigma(s) &= \sigma(0) \\p_x(s) &= p_x(0) & p_y(s) &= p_y(0) & p_\sigma(s) &= p_\sigma(0) + \frac{eV_{RF}}{m\gamma\beta c^2} (\sin\left(\frac{2\pi f_{RF}}{\beta c}\sigma(0) + \varphi_s\right) - \sin\varphi_s)\end{aligned}$$

- Adiabatic Damping (turn by turn)

$$\begin{aligned}p_{x,n+1} &= p_{x,n} \left(1 - \frac{\Delta E}{m\gamma_n\beta_n^2}\right) \\p_{y,n+1} &= p_{y,n} \left(1 - \frac{\Delta E}{m\gamma_n\beta_n^2}\right) \\p_{\sigma,n+1} &= p_{\sigma,n} \left(1 - \frac{\Delta E \left(1 + \frac{1}{\gamma_n^2}\right)}{m\gamma_n\beta_n^2}\right)\end{aligned}$$

# Space Charge Effect

- Overview
- Histogram Making (Charge weighting)
- Poisson Solver
- Electric field

# Overview

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\varphi(x, y, z) = -\frac{\rho(x, y, z)}{\epsilon_0} \xrightarrow[\sigma_x, \sigma_y \ll \sigma_z]{\text{2D approximation}} g(z) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)u(x, y) = -\frac{g(z)f(x, y)}{\epsilon_0}$$

1. Make histogram  $f(x, y)$  and  $g(z)$  ← The most time-consuming part
2. Solve 2D Poisson equation  $u(x, y)$  with boundary conditions\*
3. Calculate gradient (kick)  $-\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right) Ag(z)u(x, y) \times L$       $A = \frac{e}{m_p \gamma_p^3 \beta_p^2 c^2}, L = \text{Moving Distance}$

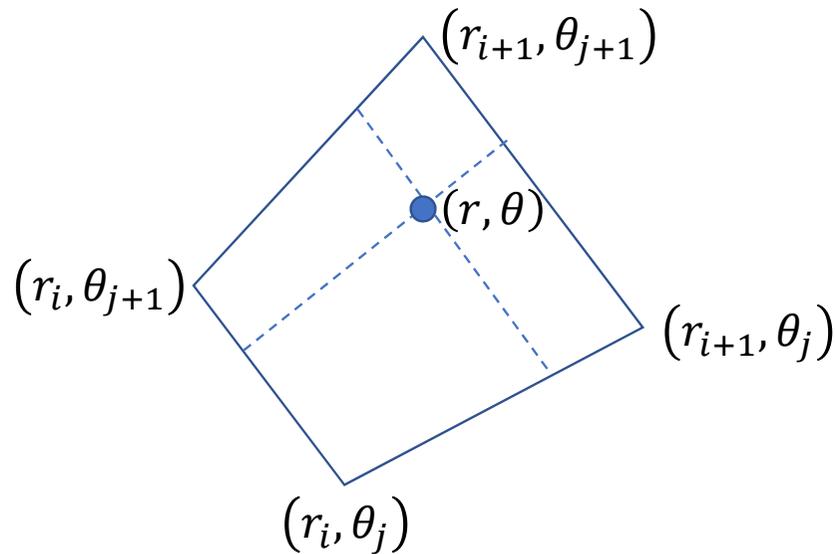
\* Using “polar” or “rectangular” boundary condition depending on the duct shape

# Space Charge Effect

- Overview
- Histogram Making (Charge weighting)
- Poisson Solver
- Electric field

# Making Histograms (1)

## Polar Coordinate



$$Q(r_i, \theta_j) += q \frac{(r_{i+1}^2 - r^2)(\theta_{j+1} - \theta)}{(r_{i+1}^2 - r_i^2)\delta\theta}$$

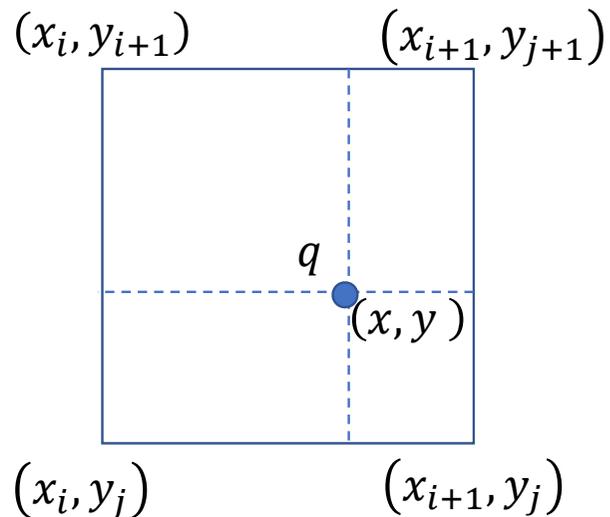
$$Q(r_{i+1}, \theta_j) += q \frac{(-r_i^2 + r^2)(\theta_{j+1} - \theta)}{(r_{i+1}^2 - r_i^2)\delta\theta}$$

$$Q(r_i, \theta_{j+1}) += q \frac{(r_{i+1}^2 - r^2)(-\theta_j + \theta)}{(r_{i+1}^2 - r_i^2)\delta\theta}$$

$$Q(r_{i+1}, \theta_{j+1}) += q \frac{(-r_i^2 + r^2)(-\theta_j + \theta)}{(r_{i+1}^2 - r_i^2)\delta\theta}$$

# Making Histograms (2)

## Cartesian Coordinate



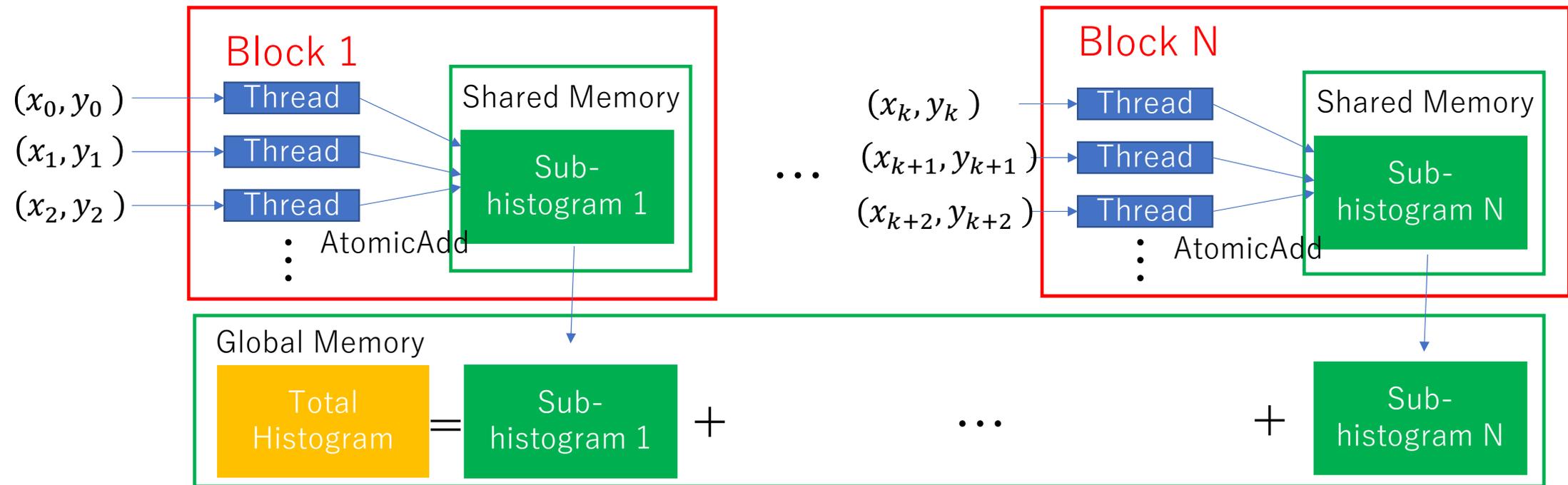
$$Q(x_i, y_j) += q \frac{(x_{i+1} - x)(y_{j+1} - y)}{\delta x \delta y}$$

$$Q(x_{i+1}, y_j) += q \frac{(-x_i + x)(y_{j+1} - y)}{\delta x \delta y}$$

$$Q(x_i, y_{j+1}) += q \frac{(x_{i+1} - x)(-y_j + y)}{\delta x \delta y}$$

$$Q(x_{i+1}, y_{j+1}) += q \frac{(-x_i + x)(-y_j + y)}{\delta x \delta y}$$

# Sub-histograms using shared memory



- Sub-histogram in each shared memory can be made in parallel.
- Each thread in a common block adds entries in shared sub-histogram. But use “exclusive add” operation (called atomicAdd function)

# Space Charge Effect

- Overview
- Histogram Making (Charge weighting)
- Poisson Solver
- Electric field

# Poisson Solver (Polar Coordinates)

2D Poisson Equation (Polar Coordinates)

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = f(r, \theta) \quad \text{Boundary Condition : } f(1, \theta) = 0$$



Discretization with  $\frac{\partial y}{\partial x} = \frac{y_{i+1} - y_i}{\Delta x}$

$$\frac{1}{r_i \Delta r} \left[ \left( r_i + \frac{\Delta r}{2} \right) \frac{U_{i+1,j} - U_{i,j}}{\Delta r} - \left( r_i - \frac{\Delta r}{2} \right) \frac{U_{i,j} - U_{i-1,j}}{\Delta r} \right] + \frac{1}{r_i^2} \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{\Delta \theta^2} = f_{i,j}$$

$$\text{where, } r_i = \left( i + \frac{1}{2} \right) \Delta r, \quad \theta_j = j \Delta \theta \quad \Delta r = \frac{1}{N + \frac{1}{2}}, \quad \Delta \theta = \frac{2\pi}{N}$$

# Poisson Solver (Polar Coordinates)

Using  $\mu_i = \frac{2r_i - \Delta r}{2r_i \Delta r^2}$ ,  $\gamma_i = \frac{1}{r_i^2 \Delta \theta^2}$ ,  $\delta = -\frac{2}{\Delta r^2}$ ,  $\tau_i = \frac{2r_i - \Delta r}{2r_i \Delta \theta^2}$  and

$$P = \begin{bmatrix} -2 & 1 & 0 & \cdots & 1 \\ 1 & 2 & -1 & \cdots & 0 \\ & \vdots & & \ddots & \vdots \\ & 0 & & \cdots & 1 & 2 & 1 \\ & 1 & & & 0 & 1 & 2 \end{bmatrix}$$



$$\mu_i U_{i-1,j} + \gamma_i (P + \delta I) U_{i,j} + \tau_i U_{i+1,j} = f_{i,j}$$

Here, a matrix Q is introduced

$$Q = \begin{pmatrix} \frac{1}{\sqrt{N}} \sqrt{\frac{2}{N}} \cos\left(\frac{2\pi(1-1)(2-1)}{N}\right) & \cdots & \sqrt{\frac{2}{N}} \cos\left(\frac{2\pi(1-1)(N/2-1)}{N}\right) & \frac{(-1)^{1-1}}{\sqrt{N}} & -\sqrt{\frac{2}{N}} \sin\left(\frac{2\pi(1-1)\left(\frac{N}{2}+2-1\right)}{N}\right) & \cdots & -\sqrt{\frac{2}{N}} \sin\left(\frac{2\pi(1-1)(N-1)}{N}\right) \\ \frac{1}{\sqrt{N}} \sqrt{\frac{2}{N}} \cos\left(\frac{2\pi(2-1)(2-1)}{N}\right) & \cdots & \sqrt{\frac{2}{N}} \cos\left(\frac{2\pi(2-1)(N/2-1)}{N}\right) & \frac{(-1)^{2-1}}{\sqrt{N}} & -\sqrt{\frac{2}{N}} \sin\left(\frac{2\pi(2-1)\left(\frac{N}{2}+2-1\right)}{N}\right) & \cdots & -\sqrt{\frac{2}{N}} \sin\left(\frac{2\pi(2-1)(N-1)}{N}\right) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{\sqrt{N}} \sqrt{\frac{2}{N}} \cos\left(\frac{2\pi(N-1)(2-1)}{N}\right) & \cdots & \sqrt{\frac{2}{N}} \cos\left(\frac{2\pi(N-1)(N/2-1)}{N}\right) & \frac{(-1)^{N-1}}{\sqrt{N}} & -\sqrt{\frac{2}{N}} \sin\left(\frac{2\pi(N-1)\left(\frac{N}{2}+2-1\right)}{N}\right) & \cdots & -\sqrt{\frac{2}{N}} \sin\left(\frac{2\pi(N-1)(N-1)}{N}\right) \end{pmatrix}$$

which satisfies  $QQ^T = I$ ,  $QPQ^T = \text{diag}(\sigma_i)_{i=1}^N$ ,  $\sigma_i = -4 \sin^2\left(\frac{(i-1)\pi}{N}\right)$

# Poisson Solver (Polar Coordinates)

Multiplying  $Q^T$  from the left, the equation becomes

$$\mu_i Q^T U_{i-1,j} + \gamma_i (Q^T P Q + \delta I) Q^T U_{i,j} + \tau_i Q^T U_{i+1,j} = Q^T f_{i,j}$$

Using  $\begin{pmatrix} V_{i,1} \\ V_{i,2} \\ \vdots \\ V_{i,N} \end{pmatrix} = Q^T \begin{pmatrix} U_{i,1} \\ U_{i,2} \\ \vdots \\ U_{i,N} \end{pmatrix}$  <sup>③</sup>, we obtain below

$$\begin{bmatrix} \gamma_1(\sigma_j + \delta) & \tau_1 & 0 & \cdots & 0 \\ \mu_2 & \gamma_2(\sigma_j + \delta) & \tau_2 & \cdots & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & & & \cdots & \mu_{N-1} & \gamma_{N-1}(\sigma_j + \delta) & \tau_{N-1} \\ 0 & 0 & & & 0 & \mu_N & \gamma_N(\sigma_j + \delta) \end{bmatrix} \begin{pmatrix} V_{1,j} \\ V_{2,j} \\ \vdots \\ V_{N,j} \end{pmatrix} = Q^T \begin{pmatrix} f_{1,j} \\ f_{2,j} \\ \vdots \\ f_{N,j} \end{pmatrix}$$

① Calculate  $Q^T(f_{1,j} \ f_{2,j} \ \cdots \ f_{N,j})^T$  using FFT

② Solve N linear equations about  $(V_{1,j} \ V_{2,j} \ \cdots \ V_{N,j})^T \ j = 1 \cdots N$

③ Calculate  $(U_{i,1} \ U_{i,2} \ \cdots \ U_{i,N})^T = Q(V_{i,1} \ V_{i,2} \ \cdots \ V_{i,N})^T \ i = 1 \cdots N$  using FFT

Using advantages of GPU

# Poisson Solver (Rectangular Coordinates)

2D Poisson Equation (Rectangular Coordinates)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

$$\text{Boundary Condition : } f(x, 0) = f(x, L_x) = f(y, 0) = f(y, L_y) = 0$$



Discretization with  $\frac{\partial y}{\partial x} = \frac{y_{i+1} - y_i}{\Delta x}$

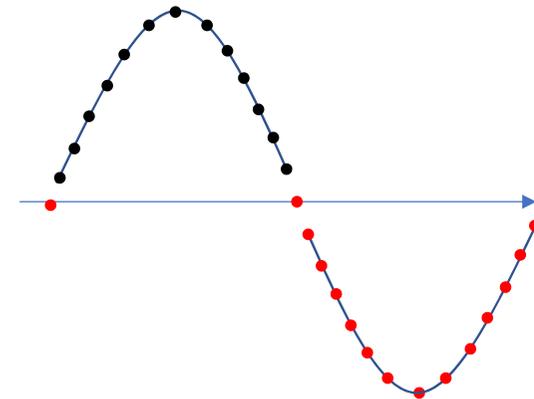
$$i, j = 1, \dots, m$$

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} = f_{i,j}$$

Odd Extension for x

$$U_{l,j} = (0, u_{1,j}, u_{2,j}, \dots, u_{m,j}, 0, -u_{m,j}, -u_{m-1,j}, \dots, -u_{1,j})$$

$$l = 0, \dots, 2(m+1) - 1$$



# Poisson Solver (Rectangular Coordinates)

$$\begin{aligned}
 DFT_x\left(\frac{U_{l-1,j} - 2U_{l,j} + U_{l+1,j}}{\Delta x^2}\right)_p &= u_{1,j} + e^{-i\frac{p}{2(m+1)}2\pi}(u_{2,j} - 2u_{1,j}) + e^{-i\frac{2p}{2(m+1)}2\pi}(u_{3,j} + u_{1,j} - 2u_{2,j}) \\
 &= 0 + \left(1 - 2e^{-i\frac{p}{2(m+1)}2\pi} + e^{-i\frac{2p}{2(m+1)}2\pi}\right)u_{1,j} \\
 &\quad + \left(e^{-i\frac{3p}{2(m+1)}2\pi} - 2e^{-i\frac{2p}{2(m+1)}2\pi} + e^{-i\frac{3p}{2(m+1)}2\pi}\right)u_{2,j} \dots \\
 &= \left(e^{i\frac{p}{2(m+1)}2\pi} + e^{-i\frac{p}{2(m+1)}2\pi} - 2\right)\left(0 + e^{-i\frac{p}{2(m+1)}2\pi}u_{1,j} + e^{-i\frac{2p}{2(m+1)}2\pi}u_{2,j} + \dots\right) \\
 &= -4 \sin^2 \frac{p\pi}{2(m+1)} \sum_{l=0}^{2(m+1)-1} e^{-i\frac{pl}{2(m+1)}2\pi} U_{l,j}
 \end{aligned}$$

Odd Extension for y

$$V_{l,l'} = (0, U_{l,1}, U_{l,2}, \dots, U_{l,m}, 0, -U_{l,m}, -U_{l,m-1}, \dots, -U_{l,1})$$

$$l' = 0, \dots, 2(m+1) - 1$$

$$DFT_y(DFT_x\left(\frac{V_{l-1,l'} - 2V_{l,l'} + V_{l+1,l'}}{\Delta x^2}\right)_p)_q = -\frac{4}{\Delta x^2} \sin^2 \frac{p\pi}{2(m+1)} \sum_{l=0}^{2(m+1)-1} \sum_{l'=0}^{2(m+1)-1} e^{-i\frac{pl}{2(m+1)}2\pi} e^{-i\frac{ql'}{2(m+1)}2\pi} V_{l,l'}$$

# Poisson Solver (Rectangular Coordinates)

$$\begin{aligned}
 DFT_y(DFT_x(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})_p)_q &\rightarrow -4 \left( \frac{1}{\Delta x^2} \sin^2 \frac{p\pi}{2(m+1)} + \frac{1}{\Delta y^2} \sin^2 \frac{q\pi}{2(m+1)} \right) \sum_{l=0}^{2(m+1)-1} \sum_{l'=0}^{2(m+1)-1} e^{-i \frac{pl}{2(m+1)} 2\pi} e^{-i \frac{ql'}{2(m+1)} 2\pi} V_{l,l'} \\
 &= -4 \left( \frac{1}{\Delta x^2} \sin^2 \frac{p\pi}{2(m+1)} + \frac{1}{\Delta y^2} \sin^2 \frac{q\pi}{2(m+1)} \right) DFT_y(DFT_x(V)_p)_q
 \end{aligned}$$

Odd Extension of  $f_{i,j}$

$$F_{l,0} = (0, 0, 0, \dots, 0, 0, 0, 0, \dots, 0)$$

$$F_{l,1} = (0, f_{1,1}, f_{2,1}, \dots, f_{m,1}, 0, -f_{m,1}, -f_{m-1,1}, \dots, -f_{1,1})$$

$$F_{l,2} = (0, f_{1,2}, f_{2,2}, \dots, f_{m,2}, 0, -f_{m,2}, -f_{m-1,2}, \dots, -f_{1,2})$$

$$F_{l,m} = (0, f_{1,m}, f_{2,m}, \dots, f_{m,m}, 0, -f_{m,m}, -f_{m-1,m}, \dots, -f_{1,m})$$

$$F_{l,m+1} = (0, 0, 0, \dots, 0, 0, 0, 0, \dots, 0)$$

$$F_{l,m+2} = (0, -f_{1,m}, -f_{2,m}, \dots, -f_{m,m}, 0, f_{m,m}, f_{m-1,m}, \dots, f_{1,m})$$

⋮

$$DFT_y(DFT_x(f)_p)_q \rightarrow \sum_{l=0}^{2(m+1)-1} \sum_{l'=0}^{2(m+1)-1} e^{-i \frac{pl}{2(m+1)} 2\pi} e^{-i \frac{ql'}{2(m+1)} 2\pi} F_{l,l'}$$

# Poisson Solver (Rectangular Coordinates)

$$DFT_y(DFT_x(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})_p)_q = DFT_y(DFT_x(f)_p)_q$$

$$\longrightarrow -4 \left( \frac{1}{\Delta x^2} \sin^2 \frac{p\pi}{2(m+1)} + \frac{1}{\Delta y^2} \sin^2 \frac{q\pi}{2(m+1)} \right) DFT_y(DFT_x(V)_p)_q = DFT_y(DFT_x(f)_p)_q$$

$$p \neq 0 \text{ or } q \neq 0 \quad DFT_y(DFT_x(V)_p)_q = - \frac{DFT_y(DFT_x(f)_p)_q}{4 \left( \frac{1}{\Delta x^2} \sin^2 \frac{p\pi}{2(m+1)} + \frac{1}{\Delta y^2} \sin^2 \frac{q\pi}{2(m+1)} \right)}$$

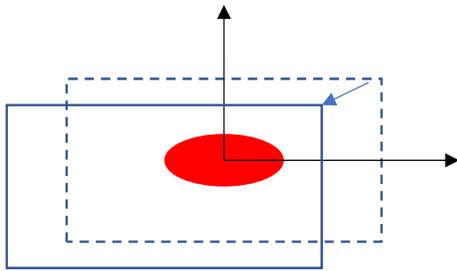
$$p = 0 \text{ and } q = 0 \quad DFT_y(DFT_x(V)_0)_0 = \sum_{l=0}^{2(m+1)-1} \sum_{l'=0}^{2(m+1)-1} V_{l,l'} = 0 \quad (V_{l,l'} \text{ is odd extension of } u_{i,j})$$

$$V_{l,l'} = - \frac{i DFT_y(i DFT_x(DFT_y(DFT_x(f)_p)_q))_{l,l'}}{4 \left( \frac{1}{\Delta x^2} \sin^2 \frac{p\pi}{2(m+1)} + \frac{1}{\Delta y^2} \sin^2 \frac{q\pi}{2(m+1)} \right)} \quad \leftarrow 4 \text{ FFTs using GPU}$$

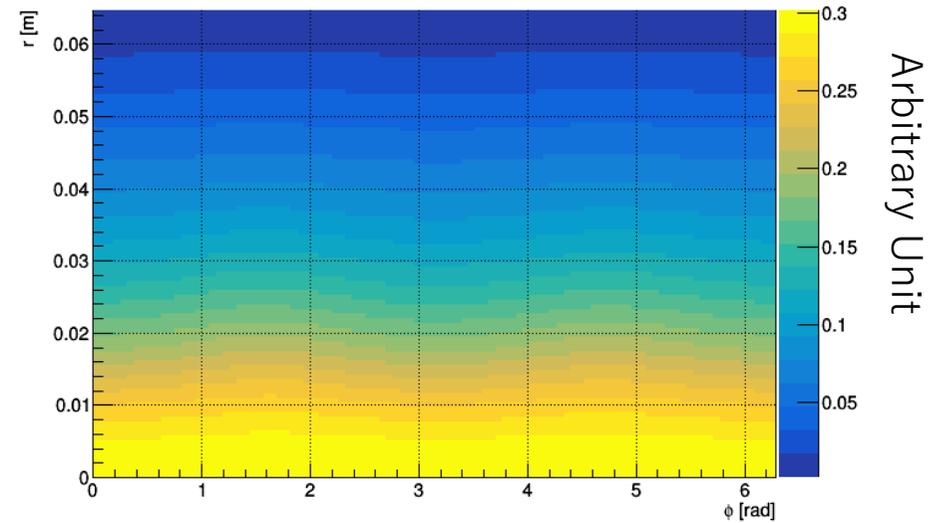
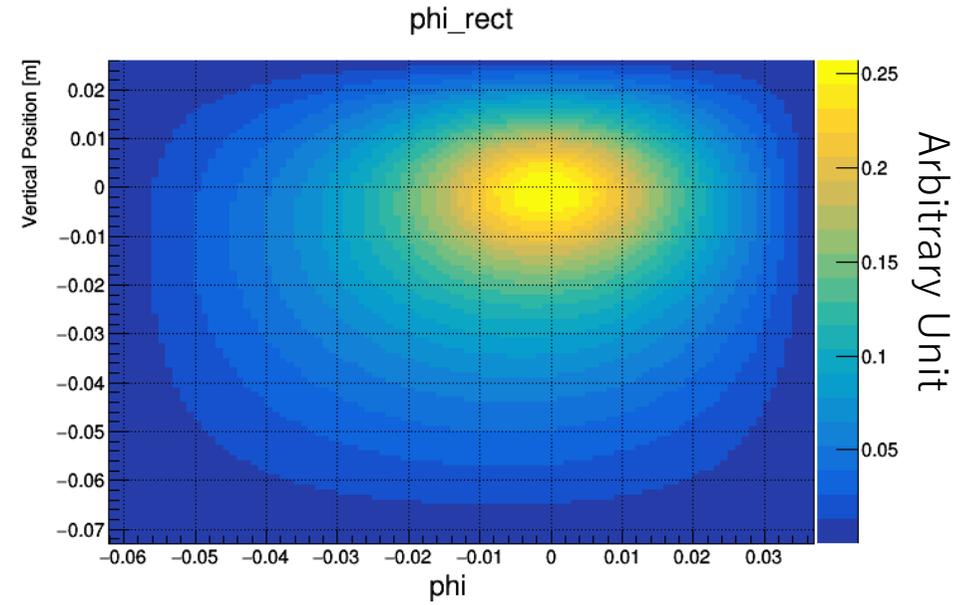
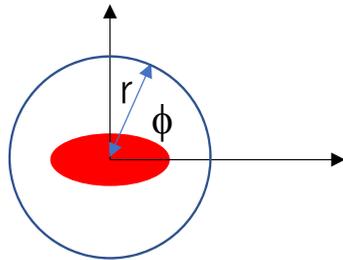
$$V_{l,l'} \rightarrow u_{i,j}$$

# Example

- Potential @ a collimator



- Potential @ a circular duct (R=0.065 m)

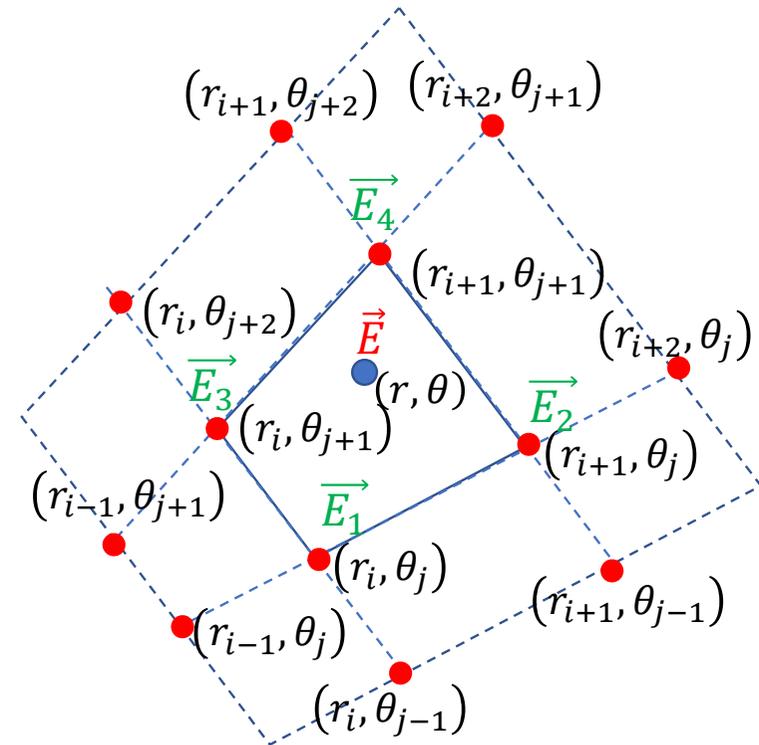


# Space Charge Effect

- Overview
- Histogram Making (Charge weighting)
- Poisson Solver
- Electric field

# Electric Field

## Polar Coordinate



$$\vec{E}_1: E_r(r_i, \theta_j) = \frac{\varphi(r_{i+1}, \theta_j) - \varphi(r_{i-1}, \theta_j)}{2\delta r} \quad E_\theta(r_i, \theta_j) = \frac{\varphi(r_i, \theta_{j+1}) - \varphi(r_i, \theta_{j-1})}{2\delta\theta}$$

$$\vec{E}_2: E_r(r_{i+1}, \theta_j) = \frac{\varphi(r_{i+2}, \theta_j) - \varphi(r_i, \theta_j)}{2\delta r} \quad E_\theta(r_{i+1}, \theta_j) = \frac{\varphi(r_{i+1}, \theta_{j+1}) - \varphi(r_{i+1}, \theta_{j-1})}{2\delta\theta}$$

$$\vec{E}_3: E_r(r_i, \theta_{j+1}) = \frac{\varphi(r_{i+1}, \theta_{j+1}) - \varphi(r_{i-1}, \theta_{j+1})}{2\delta r} \quad E_\theta(r_i, \theta_{j+1}) = \frac{\varphi(r_i, \theta_{j+2}) - \varphi(r_i, \theta_j)}{2\delta\theta}$$

$$\vec{E}_4: E_r(r_{i+1}, \theta_{j+1}) = \frac{\varphi(r_{i+2}, \theta_{j+1}) - \varphi(r_i, \theta_{j+1})}{2\delta r} \quad E_\theta(r_{i+1}, \theta_{j+1}) = \frac{\varphi(r_{i+1}, \theta_{j+2}) - \varphi(r_{i+1}, \theta_j)}{2\delta r}$$

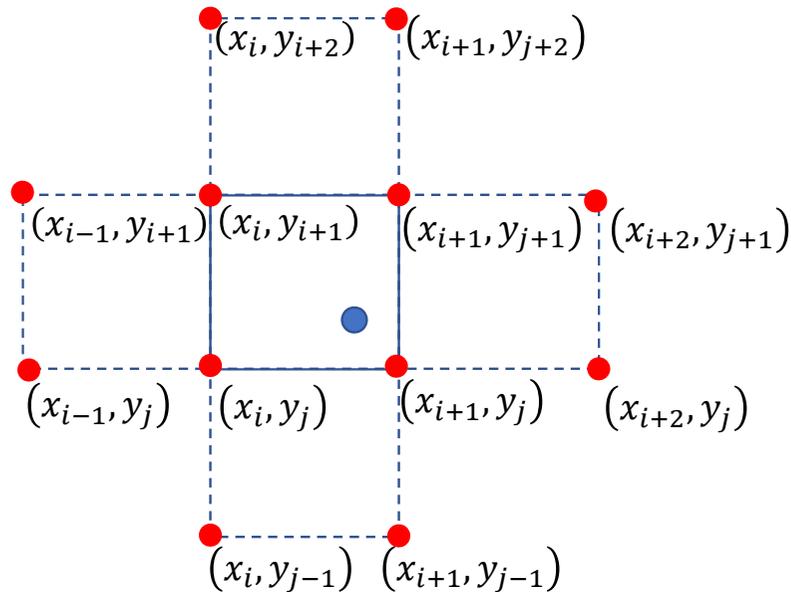
$$E_{r,\theta}(r, \theta) = \frac{E_{r,\theta}(r_i, \theta_j)(r_{i+1}-r)(\theta_{j+1}-\theta)}{\delta r \delta \theta} + \frac{E_{r,\theta}(r_{i+1}, \theta_j)(r-r_i)(\theta_{j+1}-\theta)}{\delta r \delta \theta} + \frac{E_{r,\theta}(r_i, \theta_{j+1})(r_{i+1}-r)(\theta-\theta_j)}{\delta r \delta \theta} + \frac{E_{r,\theta}(r_{i+1}, \theta_{i+1})(r-r_i)(\theta-\theta_j)}{\delta r \delta \theta}$$

$$E_x = E_r \cos \theta - \frac{1}{r} E_\theta \sin \theta$$

$$E_y = E_r \sin \theta + \frac{1}{r} E_\theta \cos \theta$$

# Electric Field

## Rectangular Coordinate



Interpolate potential using Bezier Surface

$$\varphi(x, y) = \sum_{m=0}^3 \sum_{n=0}^3 \varphi(x_{i+m-1}, y_{j+n-1}) \frac{3!}{m! (3-m)!} \left(\frac{x-x_i}{\delta x}\right)^m \left(\frac{x_{i+1}-x}{\delta x}\right)^{3-m} \\ \times \frac{3!}{n! (3-n)!} \left(\frac{y-y_j}{\delta y}\right)^n \left(\frac{y_{j+1}-y}{\delta y}\right)^{3-n}$$

$$\vec{E}(x, y) = -\text{grad } \varphi(x, y) = -\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \varphi(x, y)$$