Fast Space Charge Calculations with a Multigrid Poisson Solver & Applications

DESY, Hamburg, April 26, 2005

Gisela Pöplau
Ursula van Rienen
Rostock University

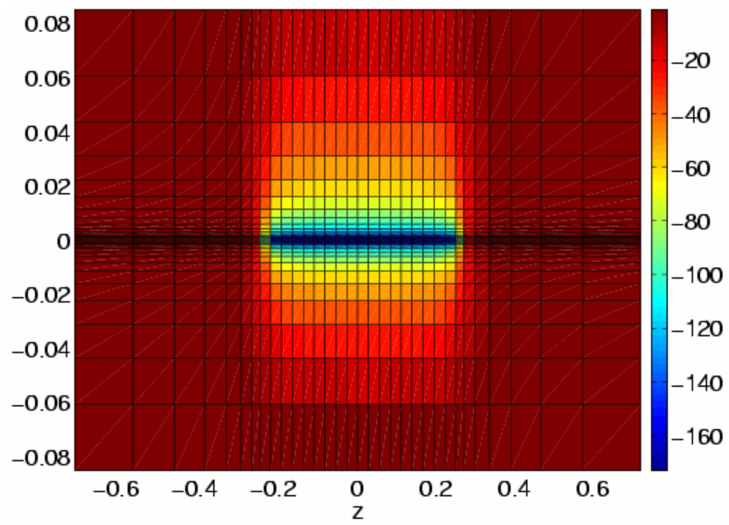
Bas van der Geer
Marieke de Loos
Pulsar Physics

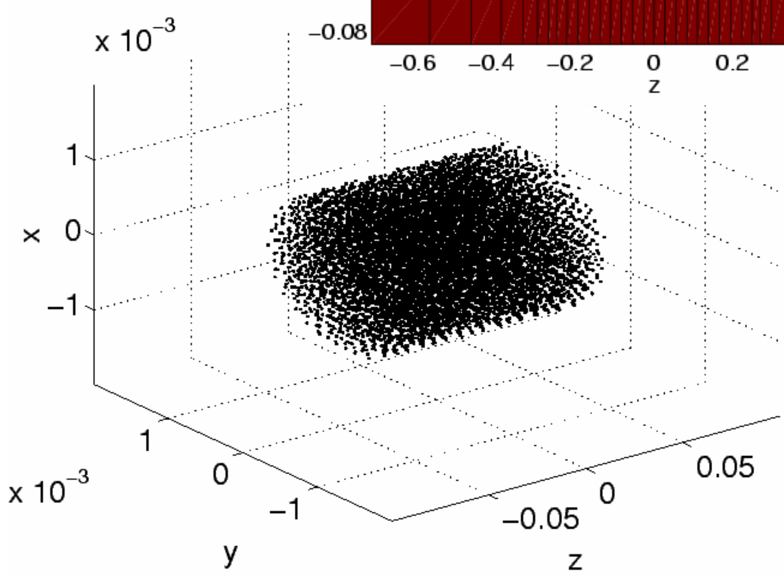
Pulsar Physics

The General Particle Tracer

Outline

- **Comparison of space charge calculations and Poisson solvers in use (not complete)**
- **Algorithm: 3D Space charge calculation with multigrid Poisson solver**
- **Real application:**
  - **Simulation of the DESY RF gun**
- **Algorithmic aspects:**
  - **Convergence studies for test cases (cylindrically shaped bunches)**





Computation of Space-Charge Fields

Due to Hockney, Eastwood

Particle-Mesh Method

- Solve Poisson´s equation

$$-\Delta V = \frac{\rho}{\varepsilon_0}$$

- Good accuracy for „smooth“ particle distributions
- Fast with best solver - $O(N)$:
  - Adaptive meshing
  - Adaptive multigrid Poisson solver

Particle-Particle Method

$$E(r_j) = \frac{1}{4\pi\varepsilon_0} \sum_{l=1}^{N} q_l \frac{r_j - r_l}{\| r_j - r_l \|^3}$$

- No mesh required
- Straightforward summation $O(N^2)$

Particle-Particle Particle-Mesh Method

3D Particle-Particle Method

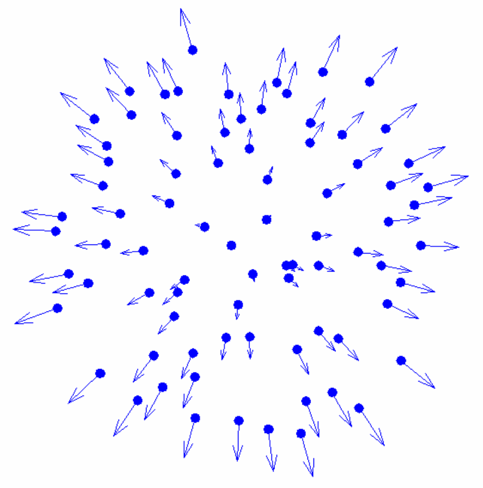
- *e. g. GPT*

↑Not limited by large energy spread

↑Straight forward summation: simple implementation

↓Granularity effects

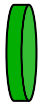
↓Numerical effort (N^2)



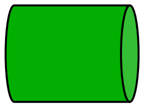
Graphics: Pulsar Physics

- **Fast summation methods:**
  - 2D methods
  - Fast multipol methods
  - Fast summation based on NFFT (Potts, P., 2004)

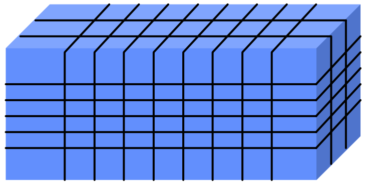
Particle-Mesh Method



Bunch in laboratory frame



Bunch in rest frame



Meshlines

ρ'

Charge density

$$-\nabla^2 V' = \rho' / \varepsilon_0$$

Poisson equation

$$\mathbf{E}' = -\nabla V' \quad \mathbf{B}' = 0$$

Interpolation

$$\{\mathbf{E}, \mathbf{B}\} = \mathcal{L}(\mathbf{E}')$$

Lorentz transformation to laboratory frame

- **Mesh-based electrostatic solver in rest-frame**
- **Bunch is tracked in laboratory frame**
- **Bunch in rest-frame is expanded by** $\gamma = 1 \big/ \sqrt{1 - v^2 / c^2}$

- **Solve Poisson equation:**

- **Transform E’ to E and B in laboratory frame**

Solve Poisson's Equation with FFT

- *e.g. ASTRA3D, IMPACT*

↑ Direct method

↑ No discretization of the Δ–operator

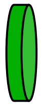
↓ Equidistant meshes

↓ Numerical effort $O(N \log N)$

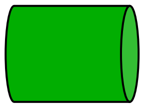
Application of NFFT (Potts&P., 2003):

- Multigrid: 29x faster for 64x64x64

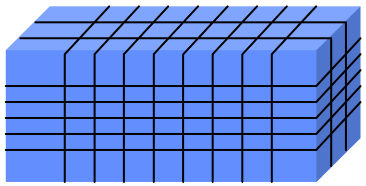
Particle-Mesh Method:
Adaptive Meshing & Multigrid



Bunch in laboratory frame



Bunch in rest frame



Meshlines

ρ' Charge density

$-\nabla^2 V' = \rho' / \varepsilon_0$ Poisson equation

$\mathbf{E}' = -\nabla V' \quad \mathbf{B}' = 0$ Interpolation

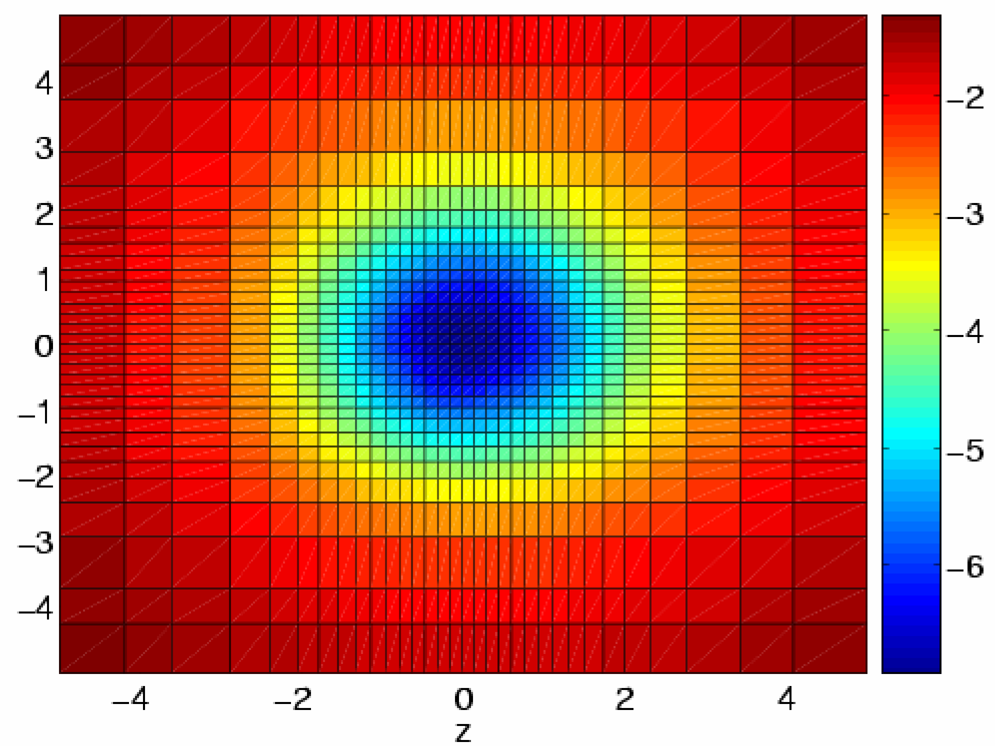
$\{\mathbf{E}, \mathbf{B}\} = \mathcal{L}(\mathbf{E}')$ Lorentz transformation to laboratory frame

- Mesh-based electrostatic solver in rest-frame
- Bunch is tracked in laboratory frame
- Bunch in rest-frame is expanded by $\gamma = 1 / \sqrt{1 - v^2 / c^2}$
- Optimal meshline positions follow beam density
- Trilinear interpolation to obtain charge density
- Solve Poisson equation:
  - Discretization: 7-point stencil
  - Multigrid, Multigrid preconditioned CG
  - Scales O(# nodes) in CPU time also on the adaptive mesh (see following slides!)
  - Adaptive meshing causes very high aspect ratios: $A_{mesh} = h_{max} / h_{min}$ (up to 3700)

    ! Problem for many Poisson solvers
  - Open boundary conditions
- 2nd order interpolation for the electrostatic field E’
- Transform E’ to E and B in laboratory frame

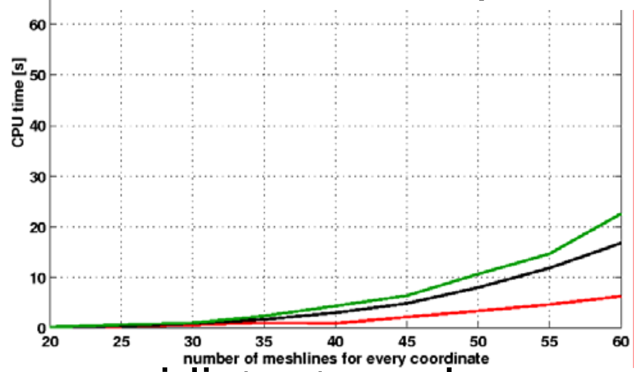
Why Multigrid?

- **Other Poisson solvers are much easier to implement**
- **They slow down considerably on non-equidistant meshes**

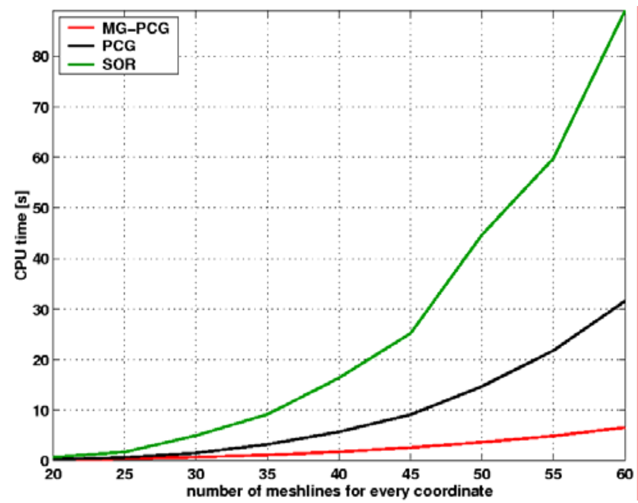


Discretization of a spherical bunch

bunch with Gaussian distribution in a sphere



equidistant mesh



non-equidistant mesh

SOR(ω)

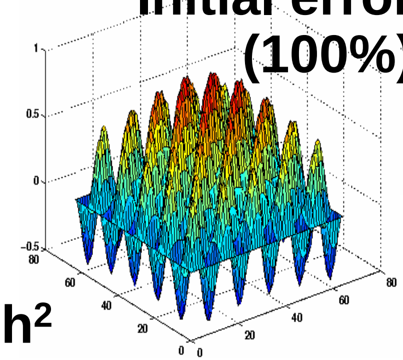
- *e.g. PARMELA_B*

↑Simple implementation

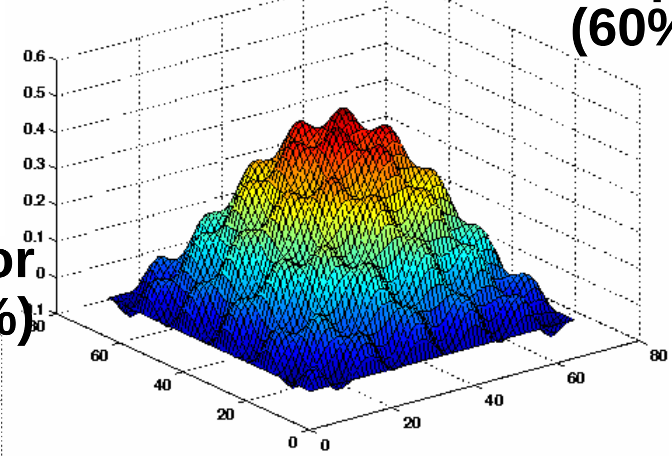
↑Convergence acceptable for

- low number of meshlines

Initial error (100%)



after 10 Gauss-Seidel steps (60%)

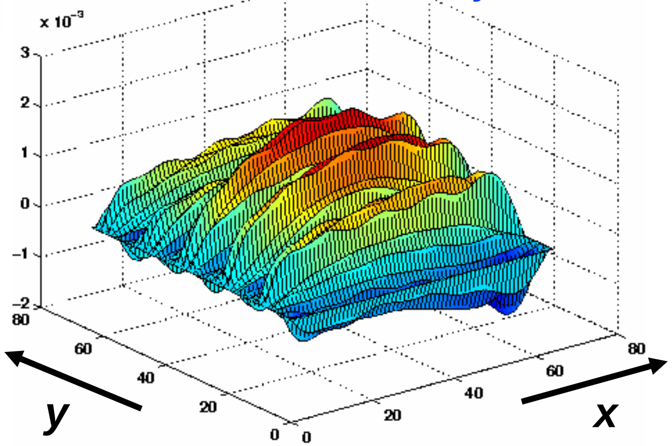


↓Convergence:

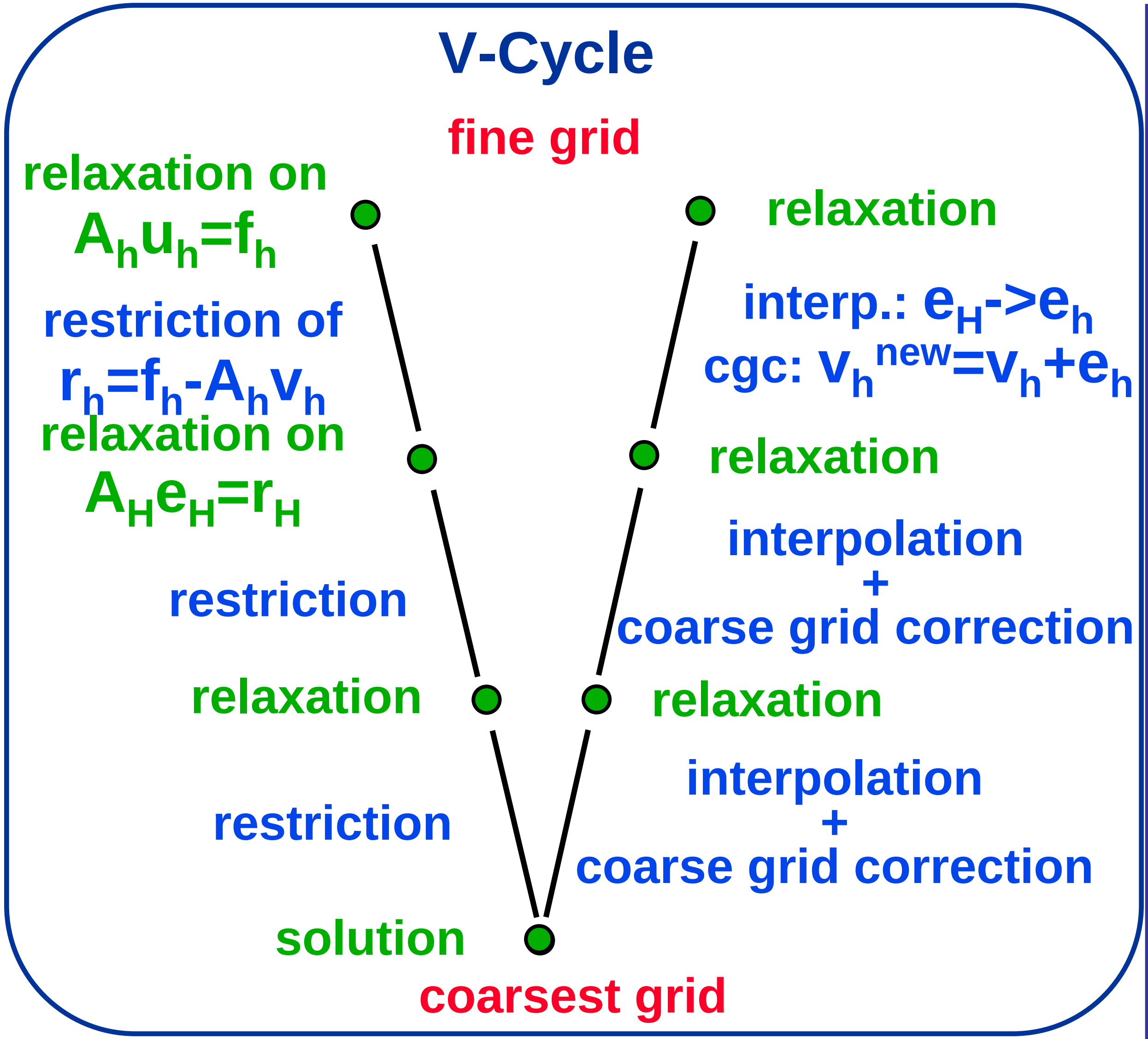
- $\omega=1$, **Gauss-Seidel: $1-\pi^2 h^2$**
- $\omega=\omega_{opt}$: **$1-\pi h$, problem: find ω_{opt}**
- **Slows down for meshes with large aspect ratios**

$$\lambda_1 \geq 1-\frac{3\pi^2}{2}\frac{\alpha^2 h^2}{1+2\alpha^2},\quad h_x = h_y = \frac{1}{N},\quad h_z = \alpha h_x,\quad A = \frac{1}{\alpha}$$

$h_y/10 = h_x$



Multigrid Technique



History:

1961 R. P. Fedorenko
- first MG

1972 A. Brandt
- adaptive grid refinement

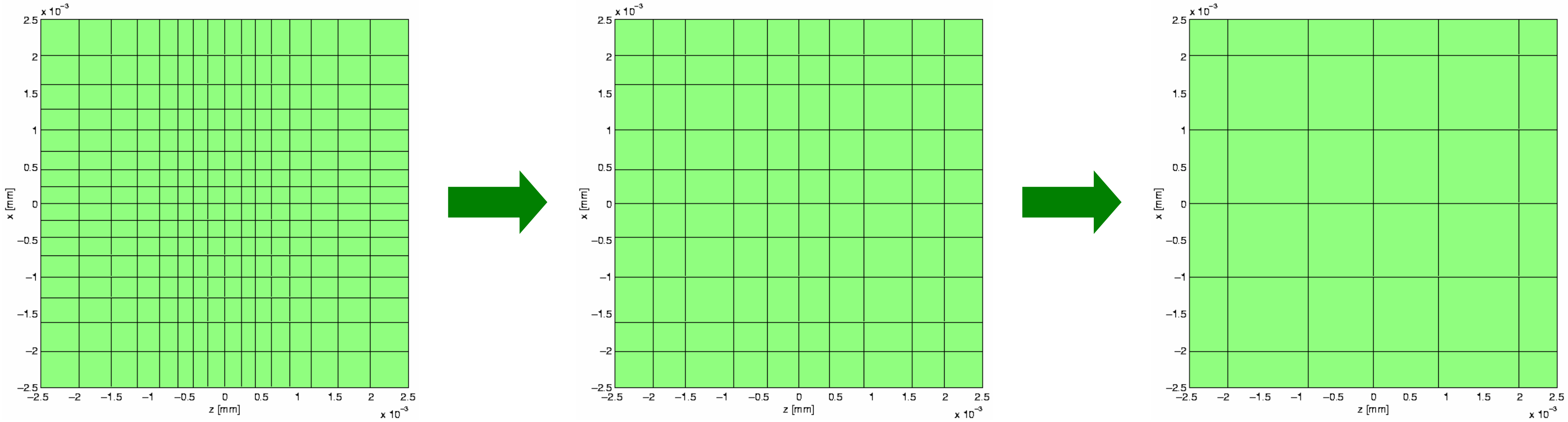
1976 W. Hackbusch
- First MG program
- New proofs for convergence

1985 W. Hackbusch
- First monograph

1989 U. Langer
- MG-PCG

The Multigrid Poisson Solver

- **The coarsening strategy is essential for a good convergence**
- **Objective of the coarsening is a sequence of coarser grids with a mesh spacing of descending aspect ratio**
- **The multigrid preconditioned conjugate gradient method (MG-PCG) accelerates convergence (Langer et al., 1989)**



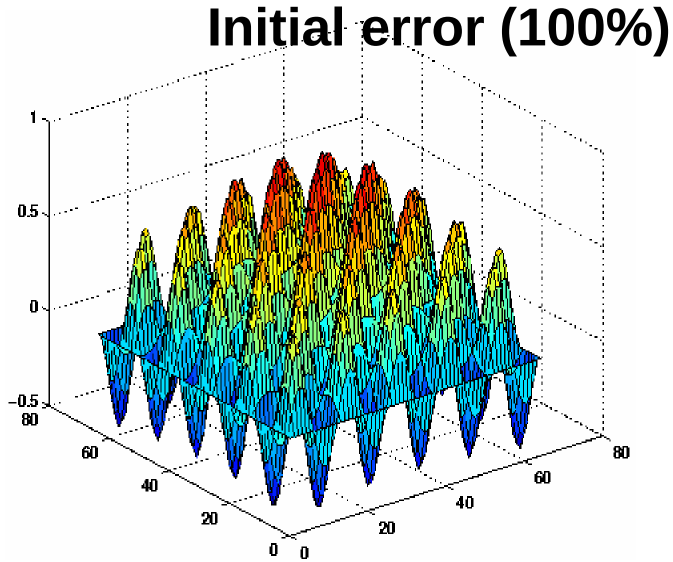
Multigrid & MG-PCG

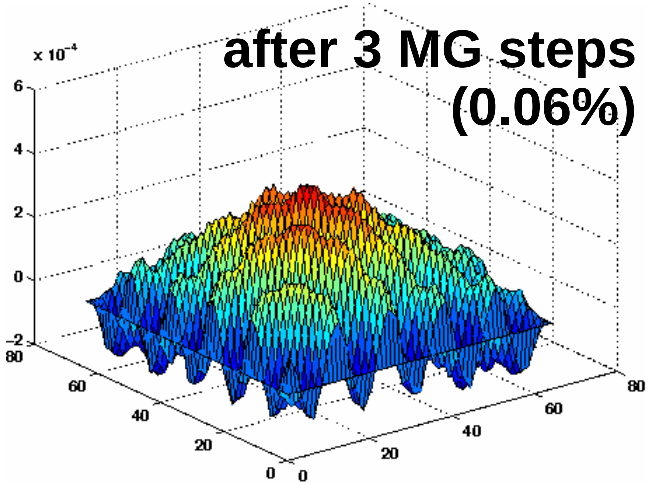
- *In new GPT release*

↑ **Convergence O(1) on non-equidistant grids**

↑ **Convergence O(1) on grids with high aspect ratios**

↓ **Implementation is more complicated**





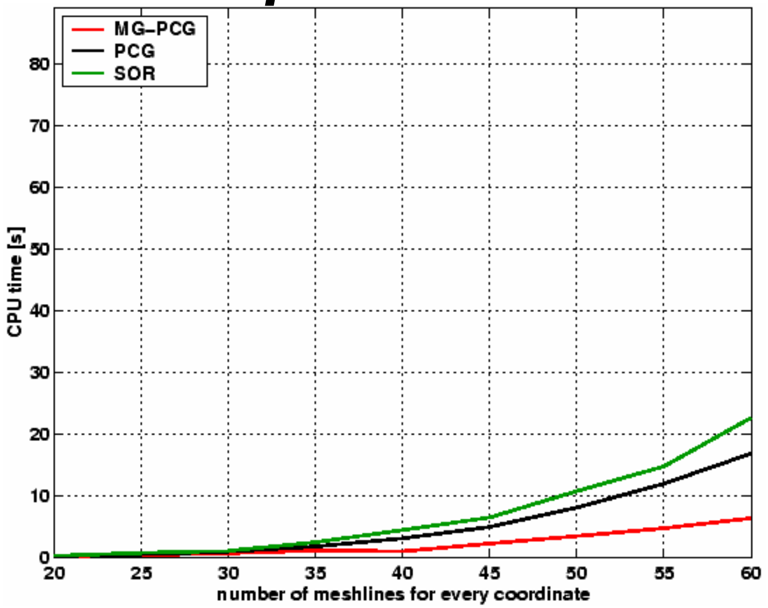
Preconditioned Conjugate Gradients

- ↑ Simple implementation
- ↑ Convergence better than for SOR
- ↓ Convergence depends on the condition number of the matrix: $$\mathrm{cond}_{\mathrm{A}} = \frac{\lambda_{\max}}{\lambda_{\min}} \leq N^2, \quad h = \frac{1}{N}$$
- ↓ Convergence
  - **O(N^2)**
  - **ILU-preconditioners: O(N)**
- ↓ Condition number increases for non-equidistant grids

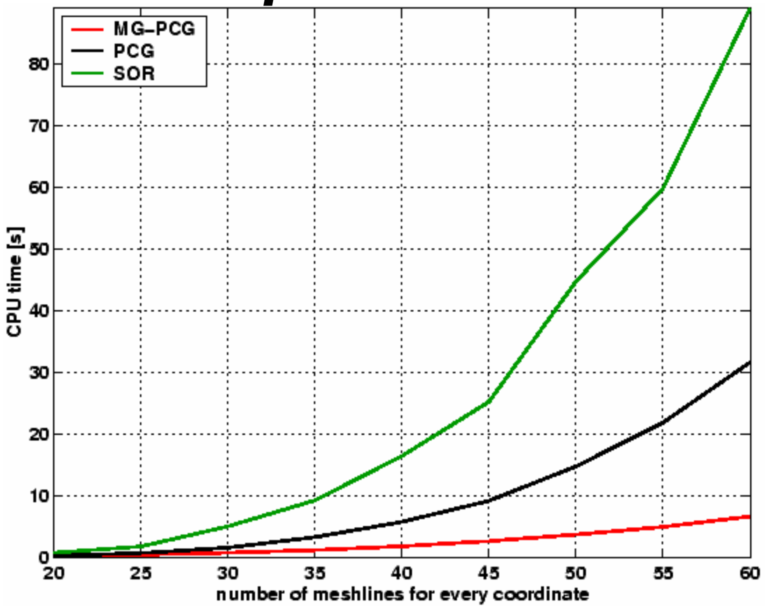
$$\mathrm{cond}_{\mathrm{A}} \leq \frac{1}{3}(2+\alpha^{-2})N^2, \quad h_x = h_y = \frac{1}{N}, \quad h_z = \alpha h_x, \quad A = \frac{1}{\alpha}$$

Gaussian Bunch in a Sphere

equidistant mesh



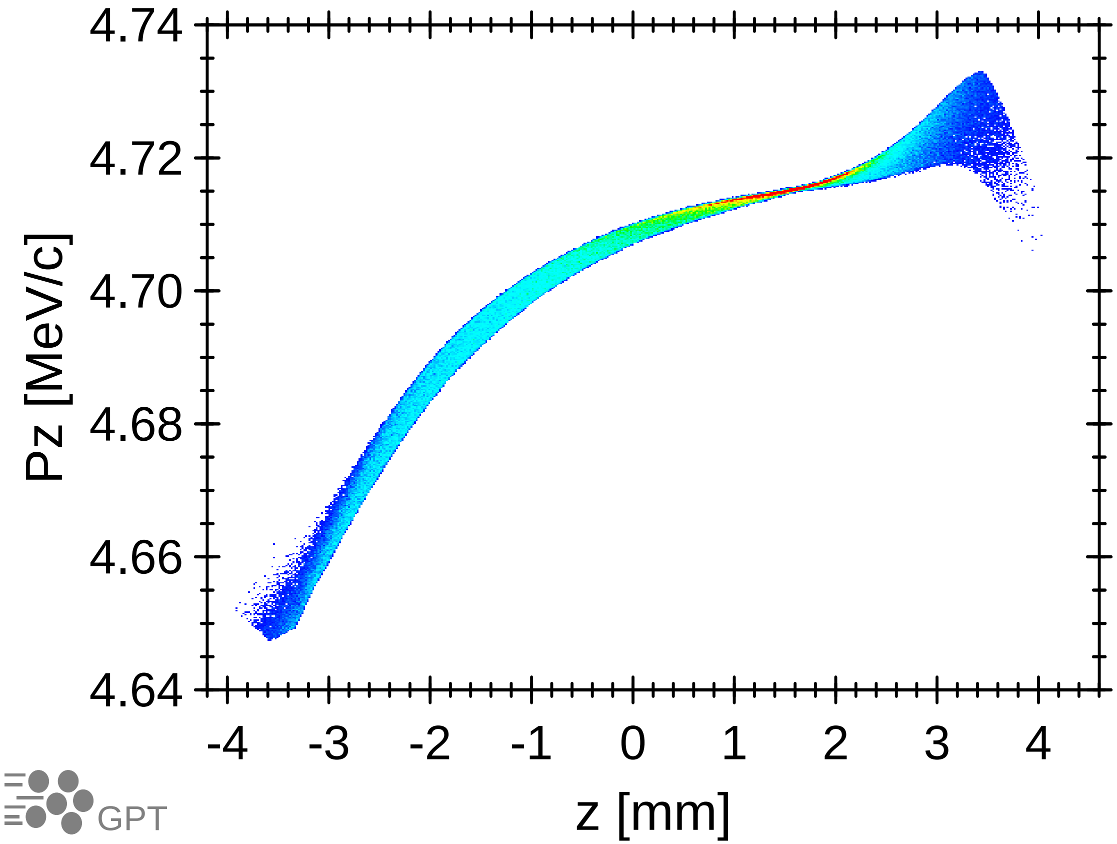
non-equidistant mesh



A_{mesh}=2.3 → 5.7

→ **Multigrid convergence remains on non-equidistant grids with high aspect ratio**

ASTRA – GPT comparison



Marieke de Loos

Bas van der Geer

Pulsar Physics

The Netherlands

www.pulsar.nl

TTF test case

- http://www.desy.de/s2e-simu/
  - XFEL: Benchmark S2E workshop, August 2003
  - 1 nC
  - First 0.25 m (rf-photogun only)

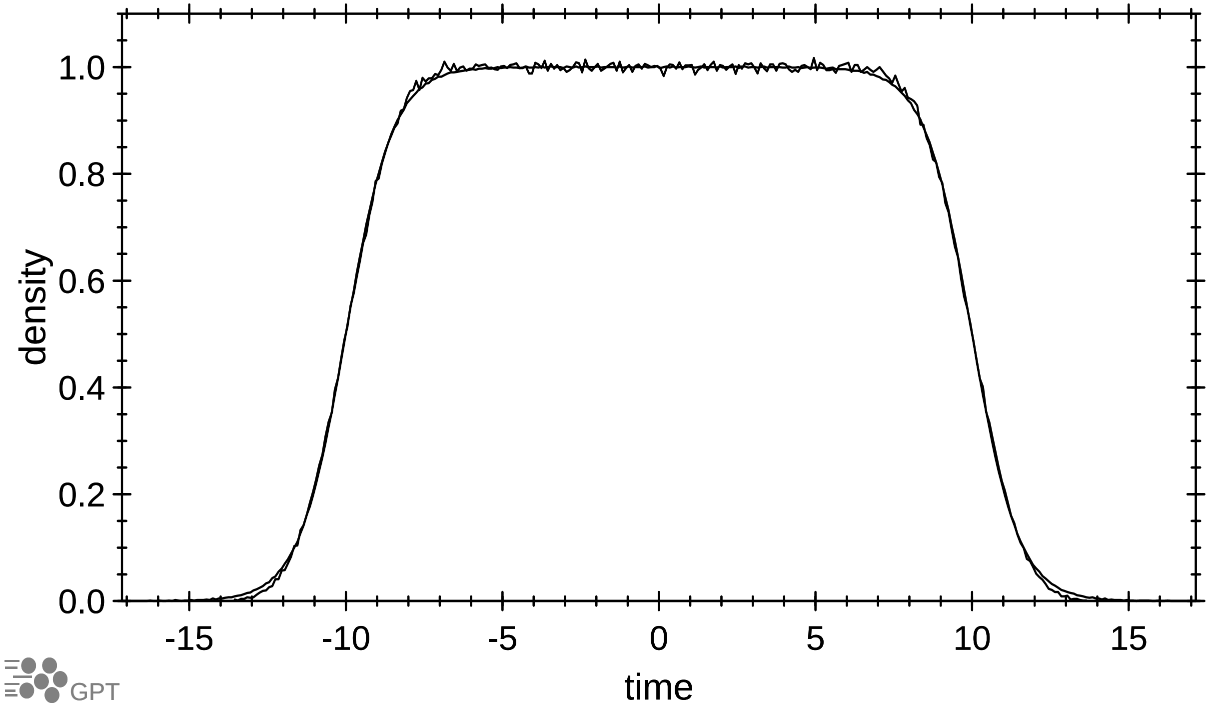
- Modeling limitations
  - 1D field-profiles (for ASTRA)
  - No Schottky effect (for GPT)

Physics: GPT vs. ASTRA

Code	GPT (2.7)	ASTRA (7-Dec-04)
Tracking	3D	3D
Space-charge	3D mesh	2D
Meshlines	Anisotropic	Non-equidistant in (z,r)
Image charges at cathode	Yes	Yes
Schottky effect	No	Yes
External field-maps	1D/2D/3D	1D
Positioning of components	6D	Position only
Bend-magnets	Yes	No
FEL	Yes	No
Longitudinal wakefields	No	No
CSR	No	No

Initial particle distribution

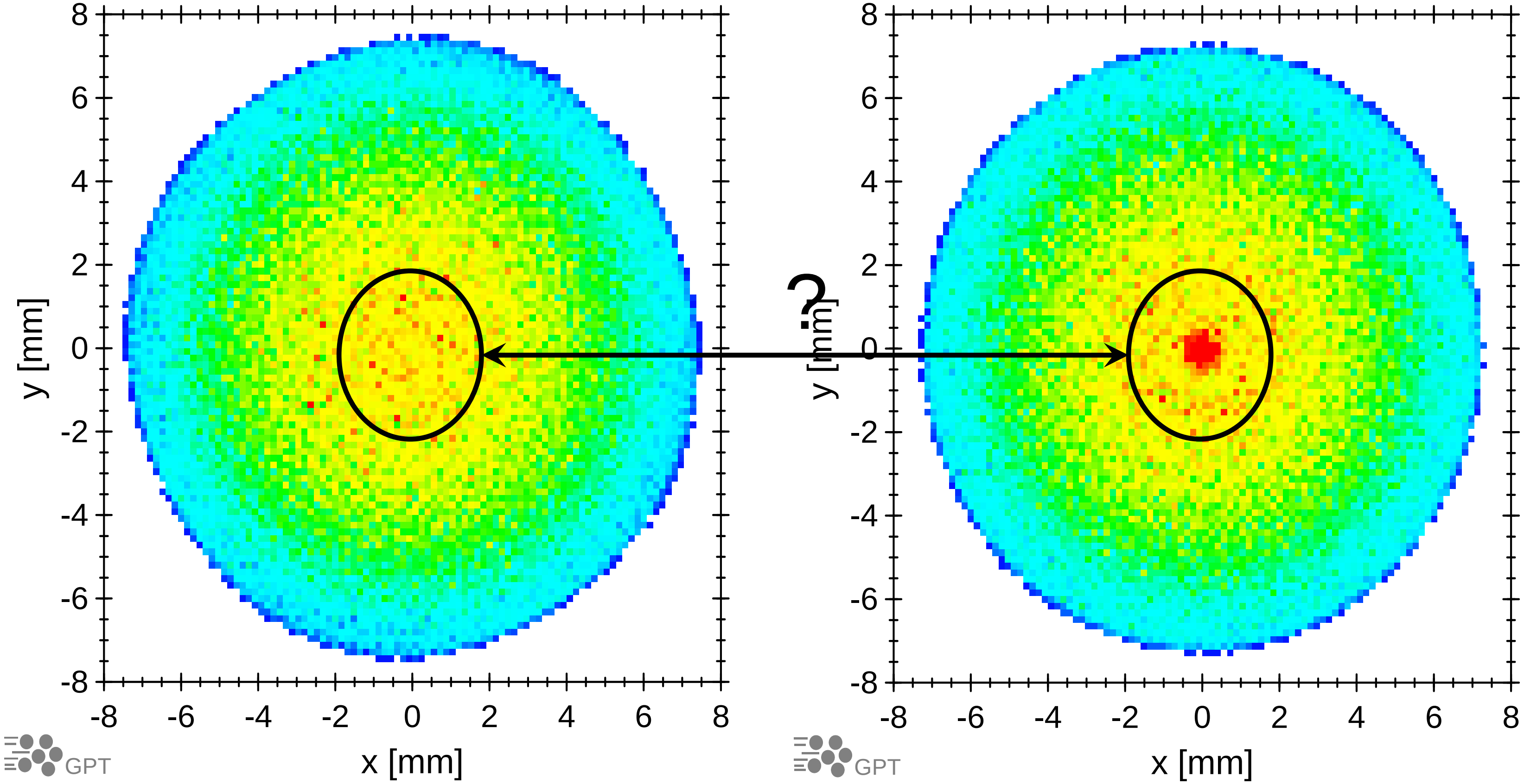
- **Hammersley (low noise) sequences**
- **'Plateau' distribution not in GPT**
  - **Quick fix: Convolution with Gaussian**
  - **Not so smooth, somehow less particles in tails**



Results

- **XY-projection**

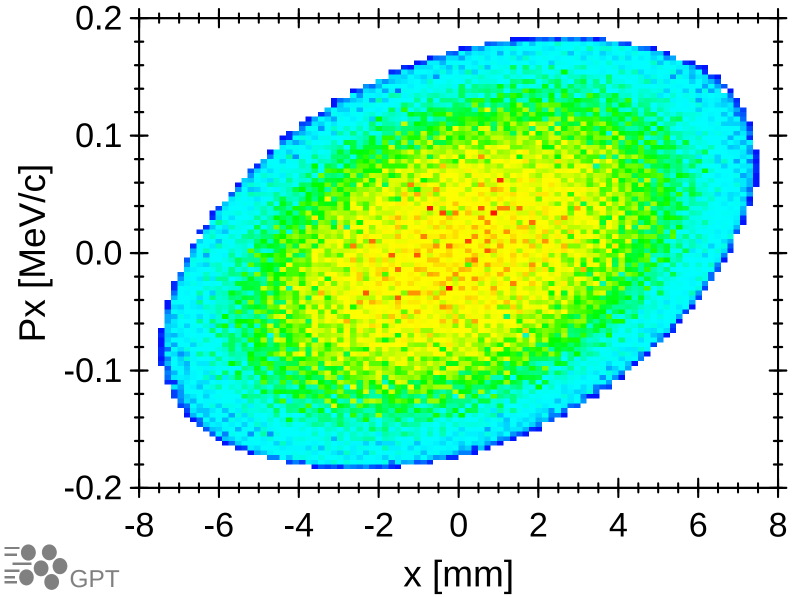
GPT **ASTRA**

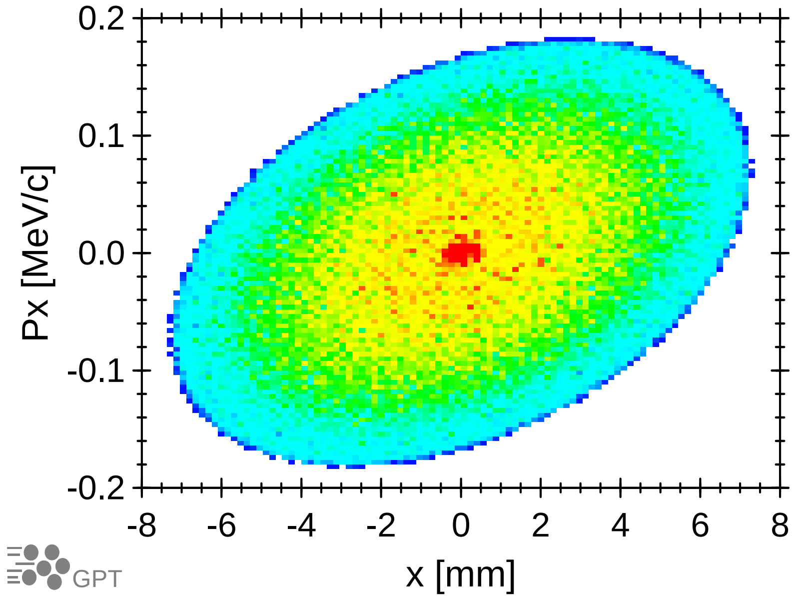


Results

- **Transverse phase-space**

GPT **ASTRA**

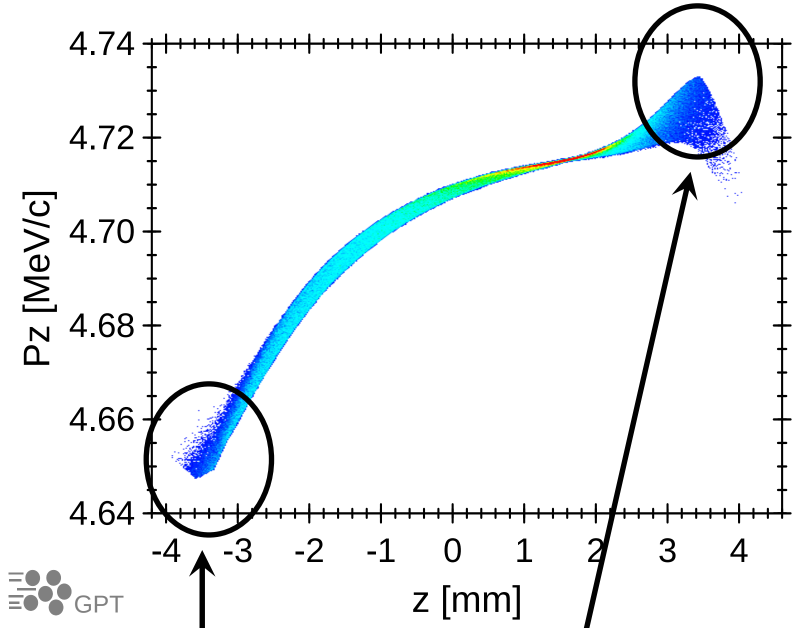




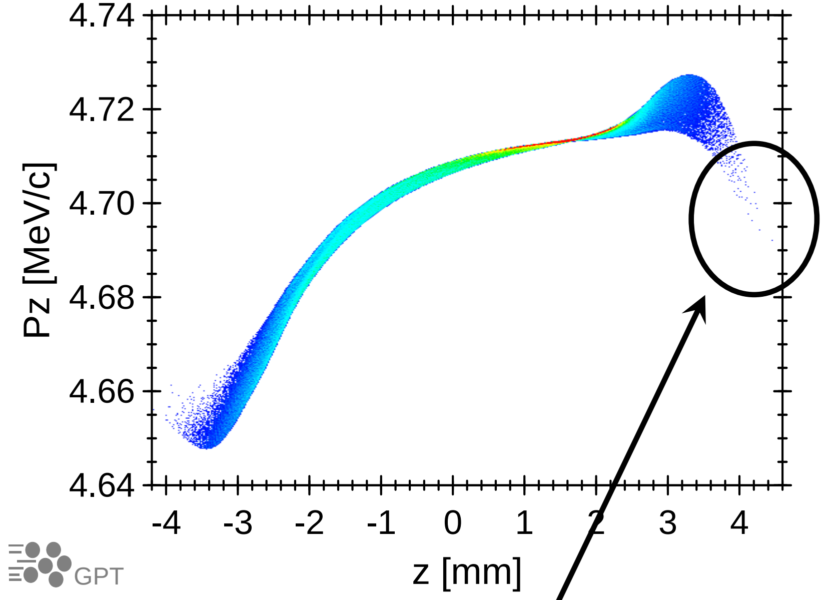
Results

- **Longitudinal phase space**

GPT



ASTRA



Limited number of meshlines
Especially at head and tail

More particles in tail
Also in initial distribution

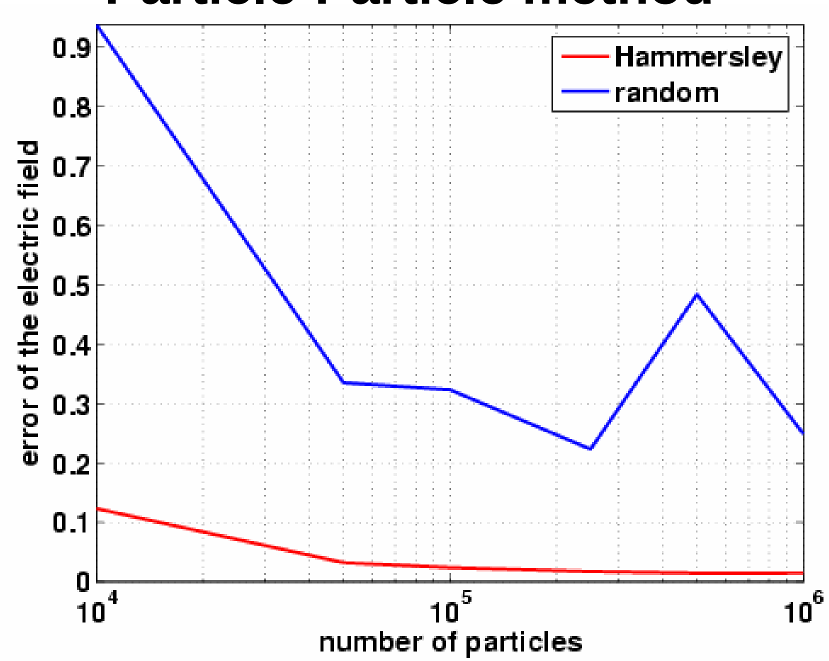
Conclusion

- Overall results are in excellent agreement
  - GPT is only 50% slower in 3D compared to ASTRA in 2D
- 3D Poisson solver works flawless
  - No errors, no crashes, no problems, just ‘plug-and-play’
- No big surprises to be expected for the next cavities
- But…
  - Better to have more meshlines at head and tail for TTF bunches
  - Our proposal: New meshing strategy based on previous solution

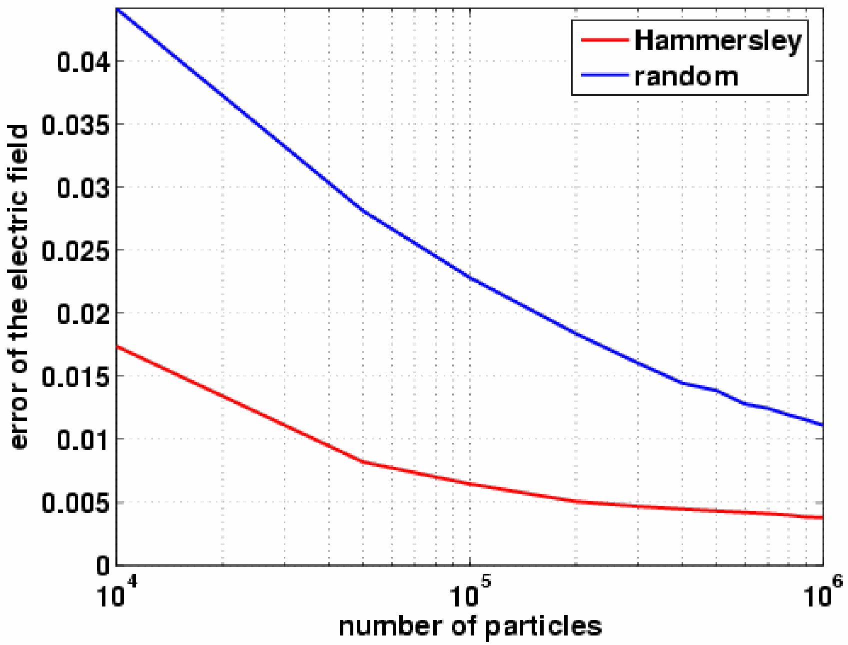
Particle Distribution

- Hammersley sequences are often used for the generation of particle distributions
- Advantage: avoid small distances between 2 particles (clustering)
- Experiment: uniformly distributed particles in a sphere

Particle-Particle method



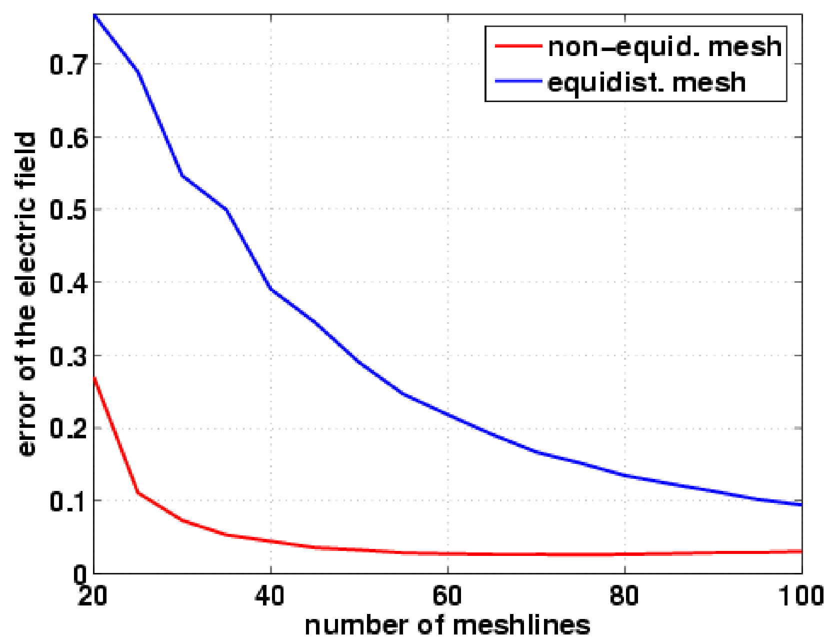
Particle-Mesh method



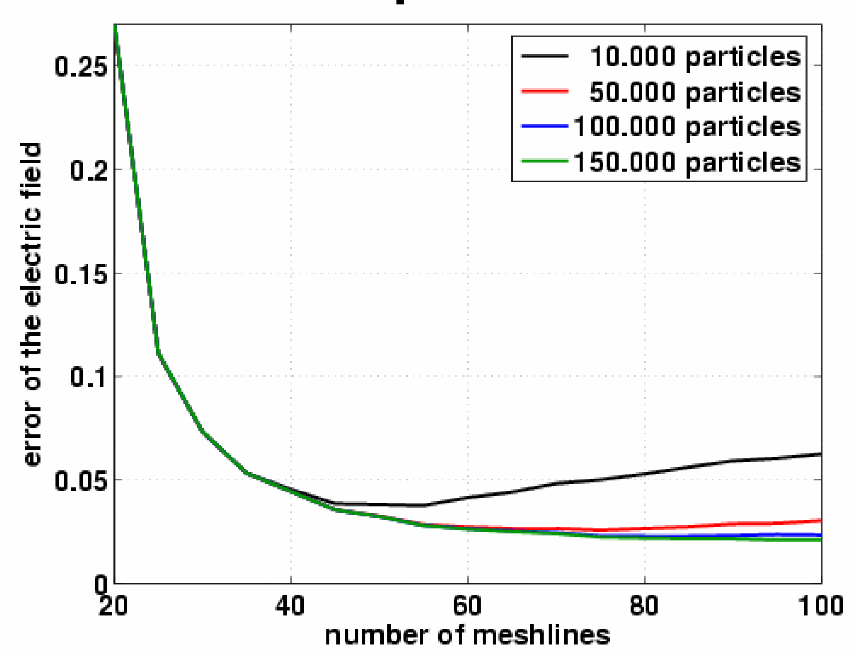
Solver Parameters

- **Numerical experiments with:**
  ‚cigar' shaped bunch with uniformly distributed particles

adaptive mesh

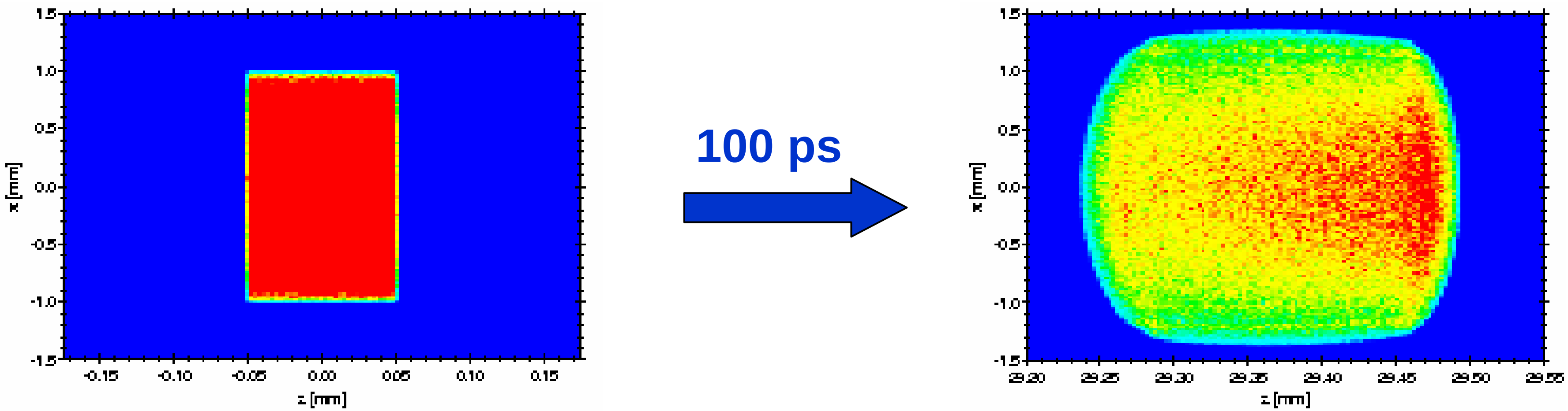


number of particles



Tracking Example

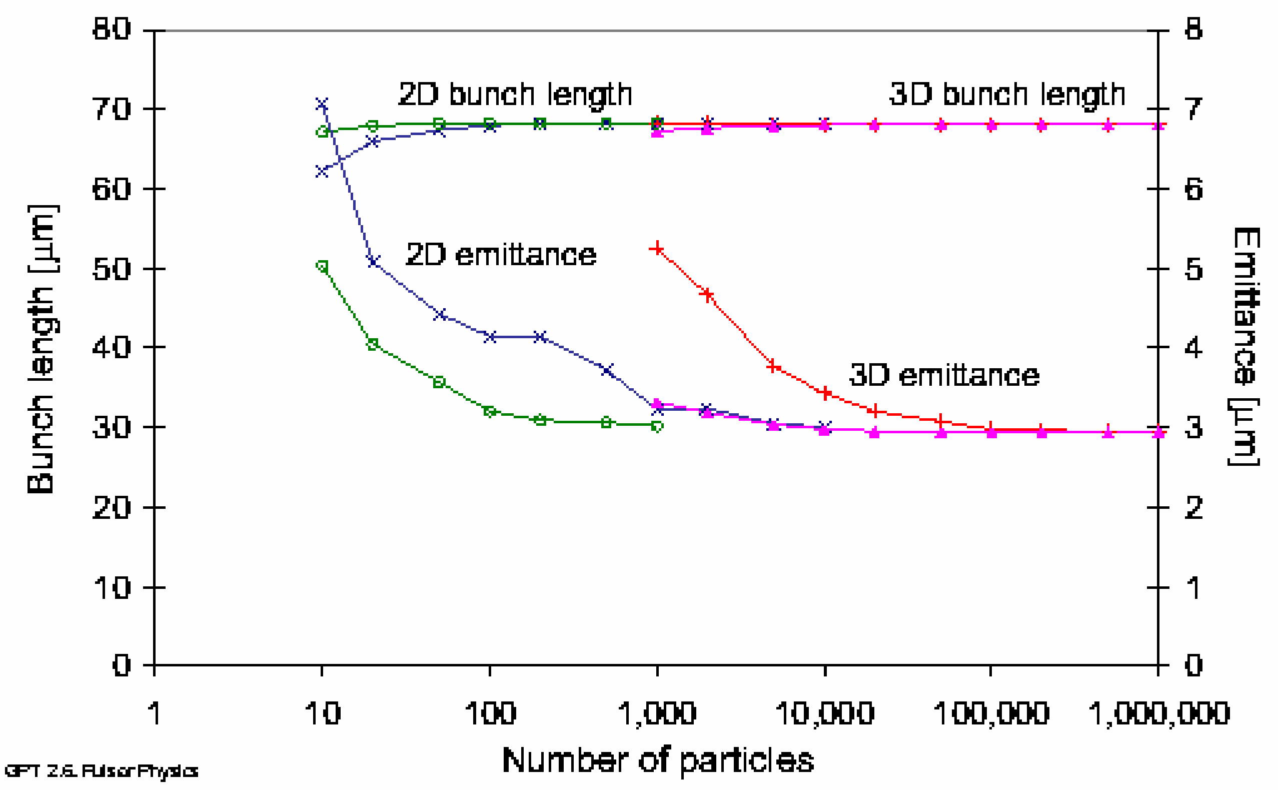
- Simulation with a hard-edged `pancake´ bunch:
  - Radius 1 mm, length 0.1 mm
  - Lorentz factor γ=5
  - Total charge 1 nC
  - Tracking time 100 ps



Initial and final projections of the charge density of an expanding `pancake´ bunch into the (x,z)-plane. One million particles are used on a 65x65x65 mesh.

Test of Physical Quantities

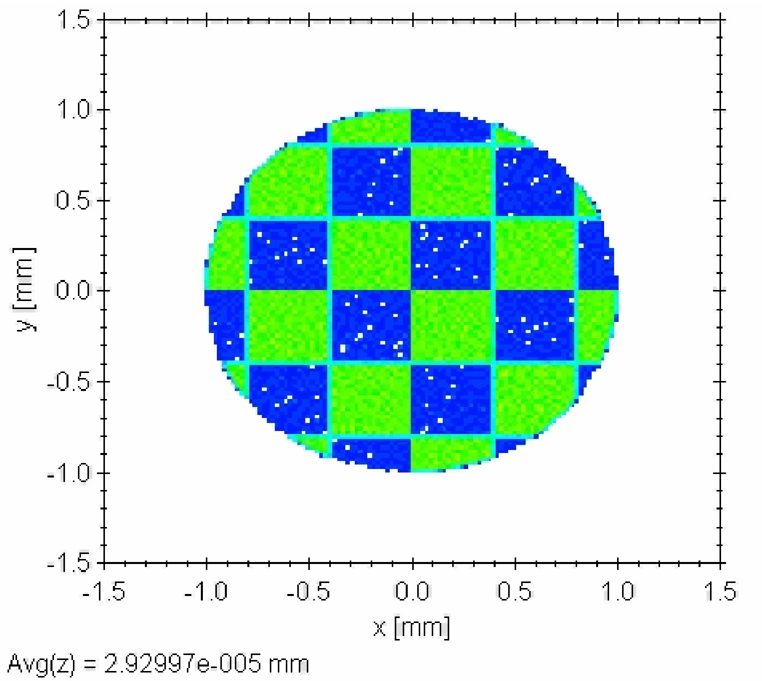
- **Comparison of the 3D model to the well-tested 2D model of GPT**
  - **Bunch length: perfect agreement**
  - **Emittance: perfect agreement with Hammersley sequences**



Summary

- **MG-PCG: stable and fast Poisson solver**
  - **Adaptive meshes with high aspect ratios possible**
  - **Tested in worst case scenarios (pancake-cigar-bunches)**
  - **Compared to well-established 2D methods and ASTRA**
- **3D space charge routine in new GPT release**
- **Investigation of 3D effects**
  - **Misalignments**
  - **Non-uniform emission from cathode**
  - **Bend magnets**
- **Further developments:**
  - **New meshing strategy for “TTF” bunches**

Thank you for your attention!



Movie:
Pulsar Physics

Preconditioned Conjugate Gradients

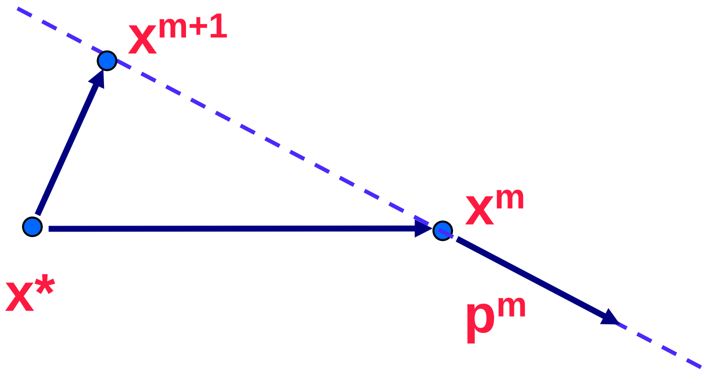
Solve: $Ax=b$

Preconitioned CG

Solve: $W^{-1}AW^{-1}Wx=W^{-1}b$

$Bu=v$

$cond_2(B)<cond_2(A)$



Choice of W:

- $diag(a_{ii})_{i=1,..N}$
- ILU

Start: x^0

$p^0=W^{-1}r^0$

$\rho^0=(p^0,r^0)$

m=0,1,2,.....

$a^m=Ap^m,\ \lambda=\rho/(a^m,p^m)$

new solution

$x^{m+1}=x^m+\lambda p^m$

new residual

$r^{m+1}=r^m-\lambda a^m$

new projection

$q^{m+1}=W^{-1}r^{m+1}$

$\rho^{m+1}=(q^{m+1},r^{m+1})$

$p^{m+1}=q^{m+1}+\rho_{m+1}/\rho_m p^m$

Multigrid Preconditioned CG

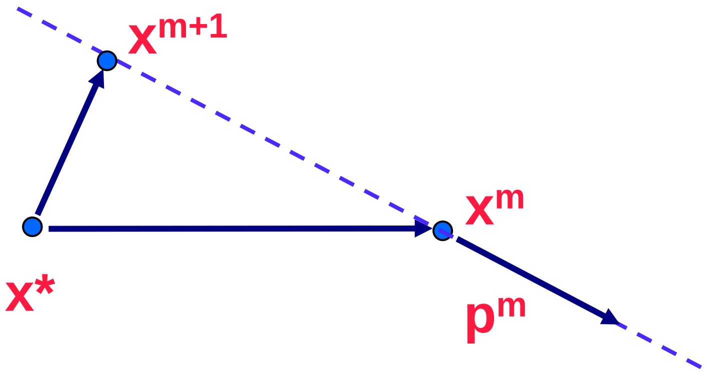
Solve: $Ax=b$

Preconitioned CG

Solve: $W^{-1}AW^{-1}Wx=W^{-1}b$

$Bu=v$

$cond_2(B)<cond_2(A)$



optimal: W=A ⇒ MG-PCG[9]

Start: x^0

$p^0=A^{-1}r^0$ **solve with MG**

$\rho^0=(p^0,r^0)$

$m=0,1,2,.....$

$a^m=Ap^m,\ \lambda=\rho/(a^m,p^m)$

new solution

$x^{m+1}=x^m+\lambda p^m$

new residual

$r^{m+1}=r^m-\lambda a^m$

new projection

$q^{m+1}=A^{-1}\, r^{m+1}$ **solve with MG**

$\rho^{m+1}=(q^{m+1},r^{m+1})$

$p^{m+1}=q^{m+1}+\rho_{m+1}/\rho_m p^m$

Numerical Effort

	Direct solver (FFT, equidistant)	Multigrid (equidistant+non-equidistant) V-cycle mit 2 pre- and 2 post-smoothing steps
2D: grid points: NxN step size: h=1/N	$10N^2 \log N$	$100/3\ N^2$ per iteration
3D: grid points: NxNxN step size: h=1/N	$15N^3 \log N$	$40\ N^3$ per iteration

Pancake-Bunch:

equidist.: N_z=1024, error=0.45 non-equidst.: N_z=128, error=0.15