A Fast and Scalable Low Dimensional Solver for Charged Particle Dynamics in Large Particle Accelerators

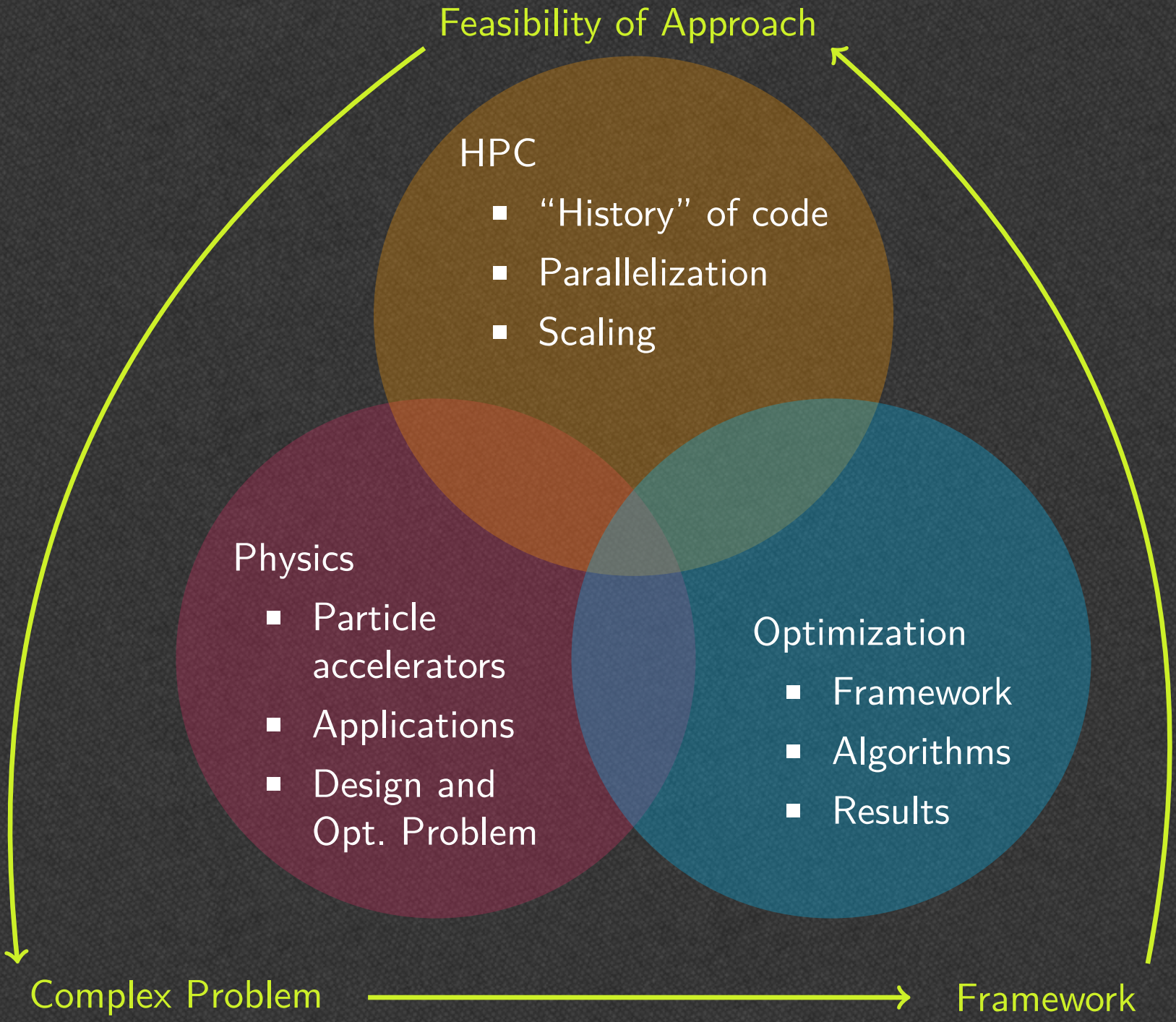
Yves Ineichen[1,2,3], Andreas Adelmann[2], Costas Bekas[3], Alessandro Curioni[3], Peter Arbenz[1]

[1]ETH Zürich
Department of Computer Science
CH-8092 Zürich, Switzerland

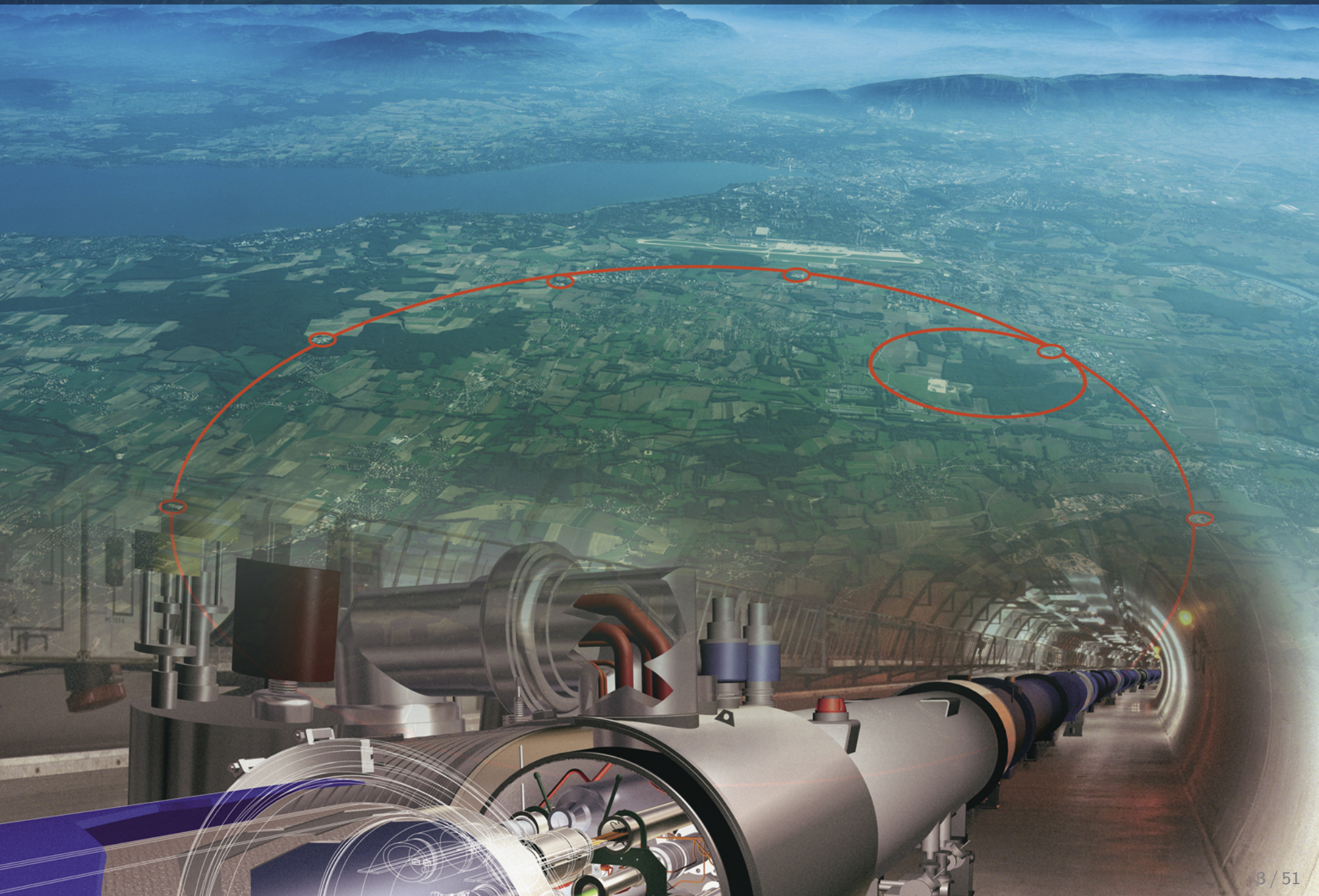
[2]Paul Scherrer Institut
Accelerator Modelling and Advanced Simulations
CH-5234 Villigen, Switzerland

[3]IBM Research-Zurich
CH-8803 Rüschlikon, Switzerland

17th June 2012

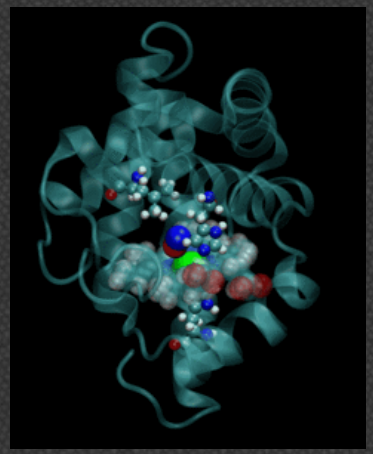

Feasibility of Approach
HPC
"History" of code
Parallelization
Scaling
Physics
Particle accelerators
Applications
Design and Opt. Problem
Optimization
Framework
Algorithms
Results
Complex Problem
Framework

Highly complex machines

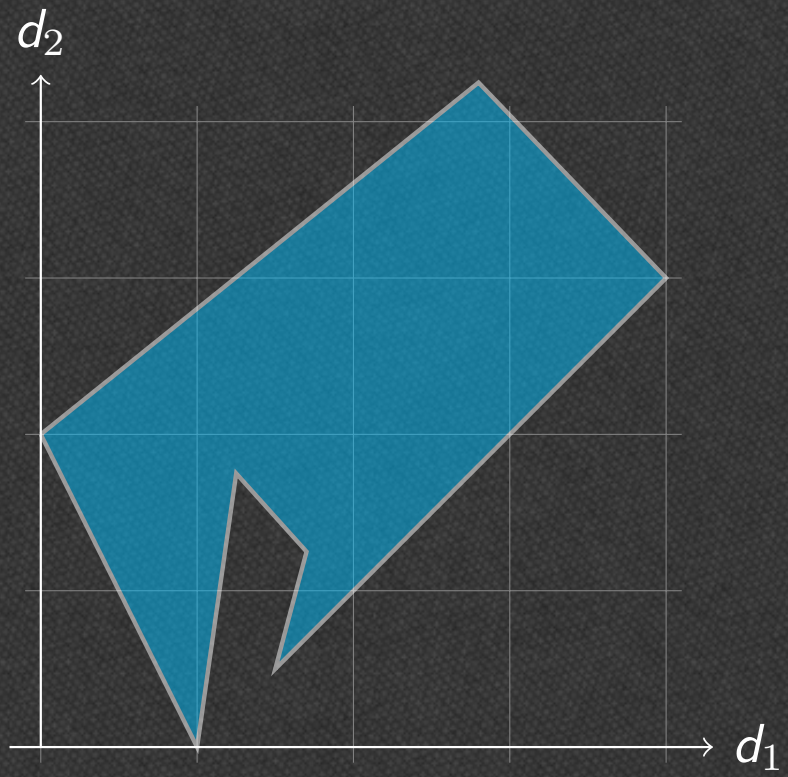


Applications

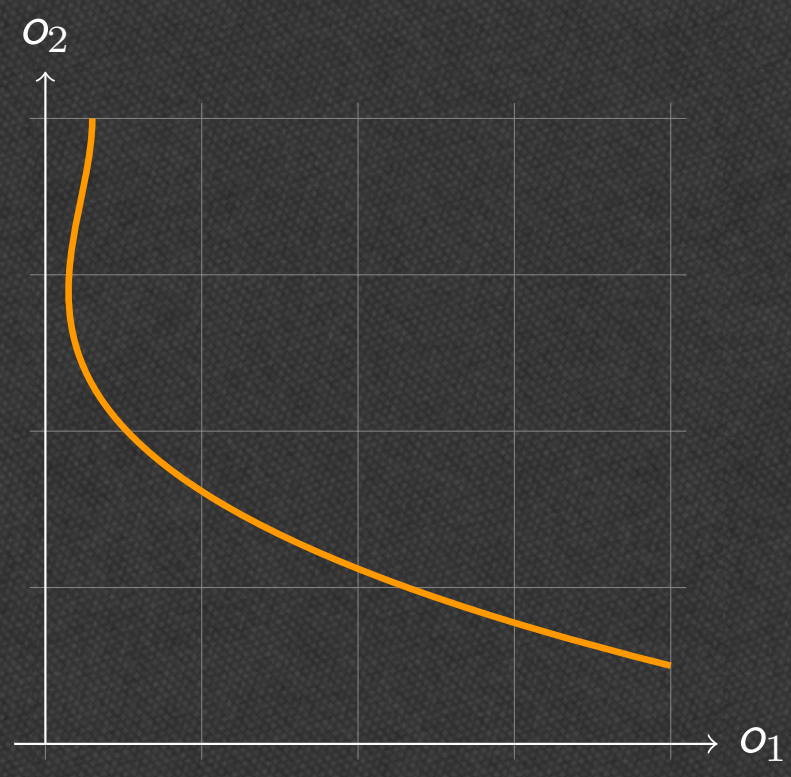
- particle physics
- medical treatment
- material science
- life science



design, commissioning, and operation $\rightarrow$ non-trivial task



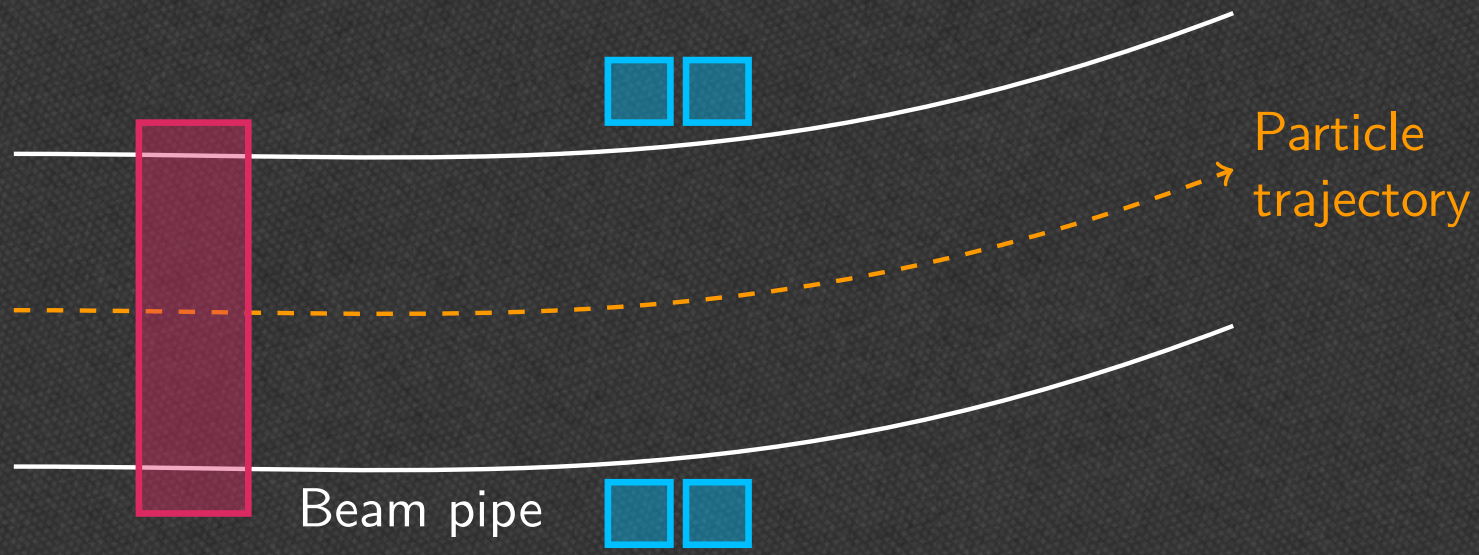
Design space



Objective space

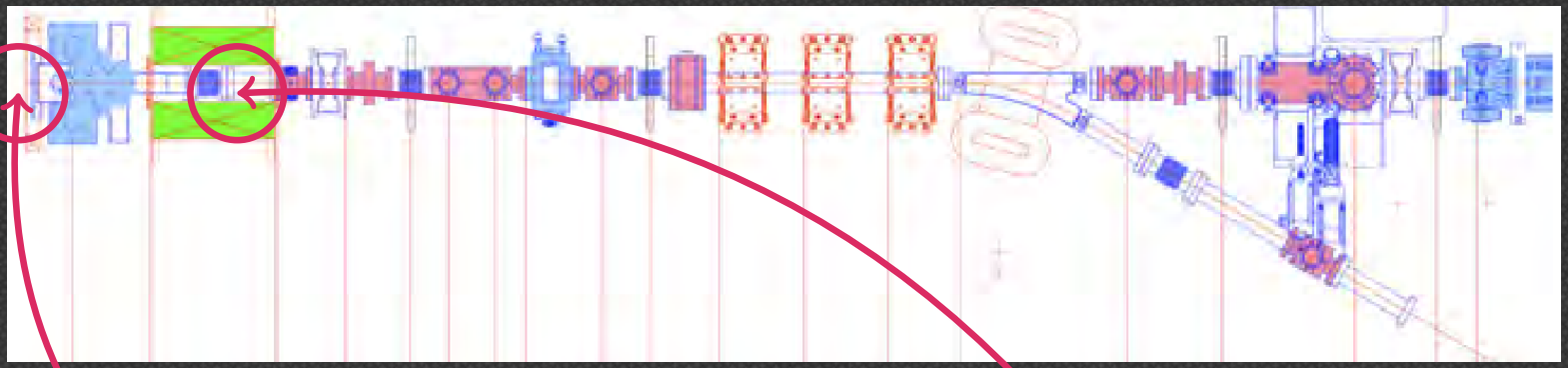
Accelerator Principles

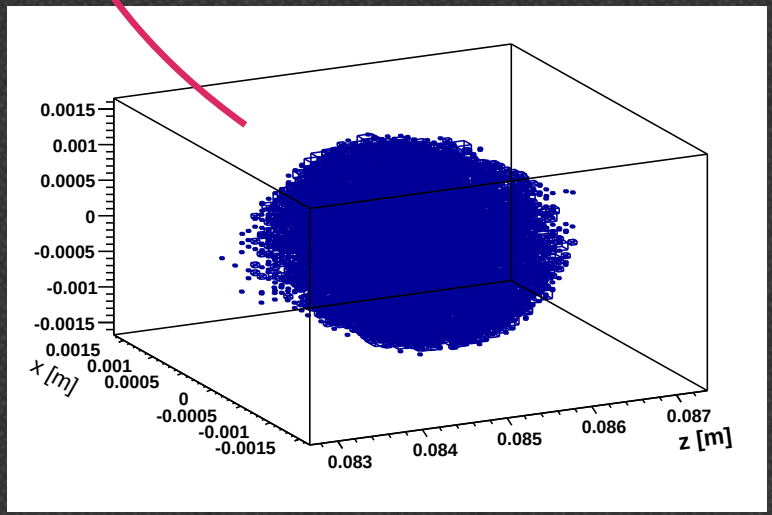
(acceleration / guiding / focusing)

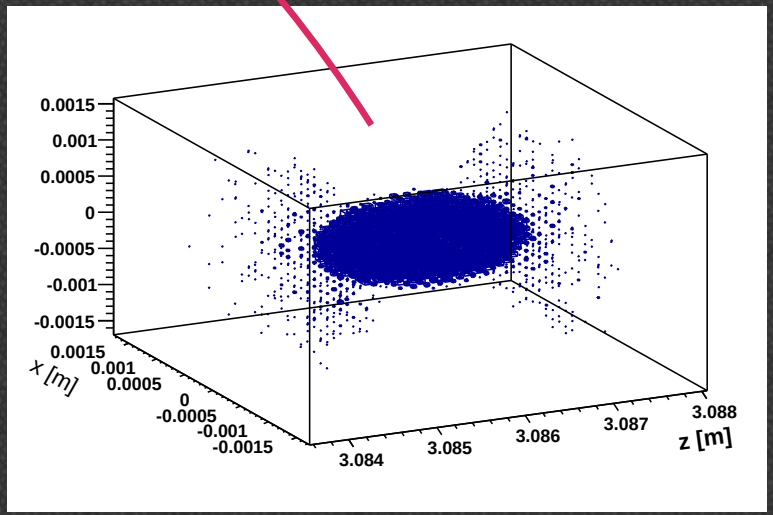


Focusing and guiding magnets

Time Evolution



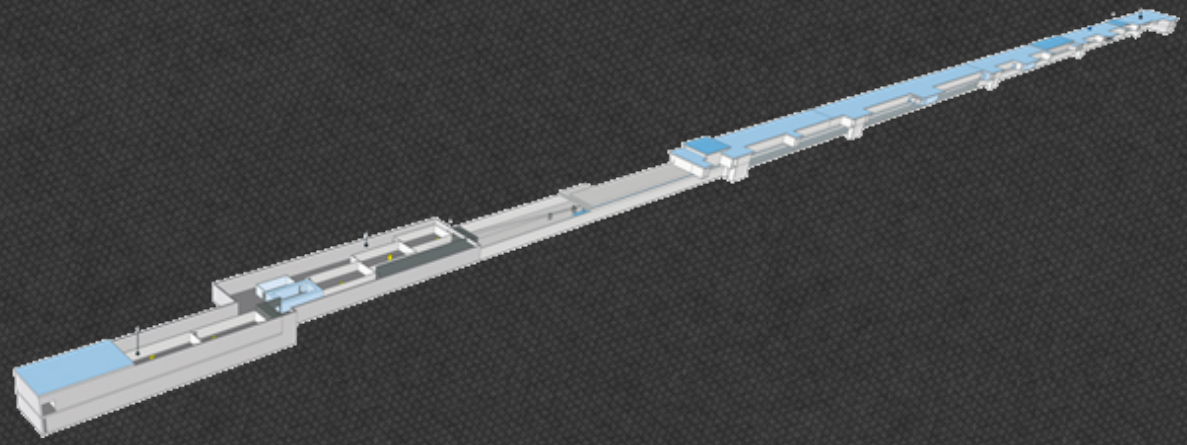




Example SwissFEL @ PSI

(machine / complexity / objectives and physical meaning)

SwissFel [1]: Switzerland's X-ray free-electron laser Project at PSI



- big project: $>$ 100 people, expensive, 700 m long
- 1 Ångström
- to reach target it is of crucial importance to attain "good" beam properties (e.g. narrow beam/small phase space volume)

[1] http://www.psi.ch/swissfel/

SwissFel [1]: Switzerland's X-ray free-electron laser Project at PSI

Calls for optimization of Injector

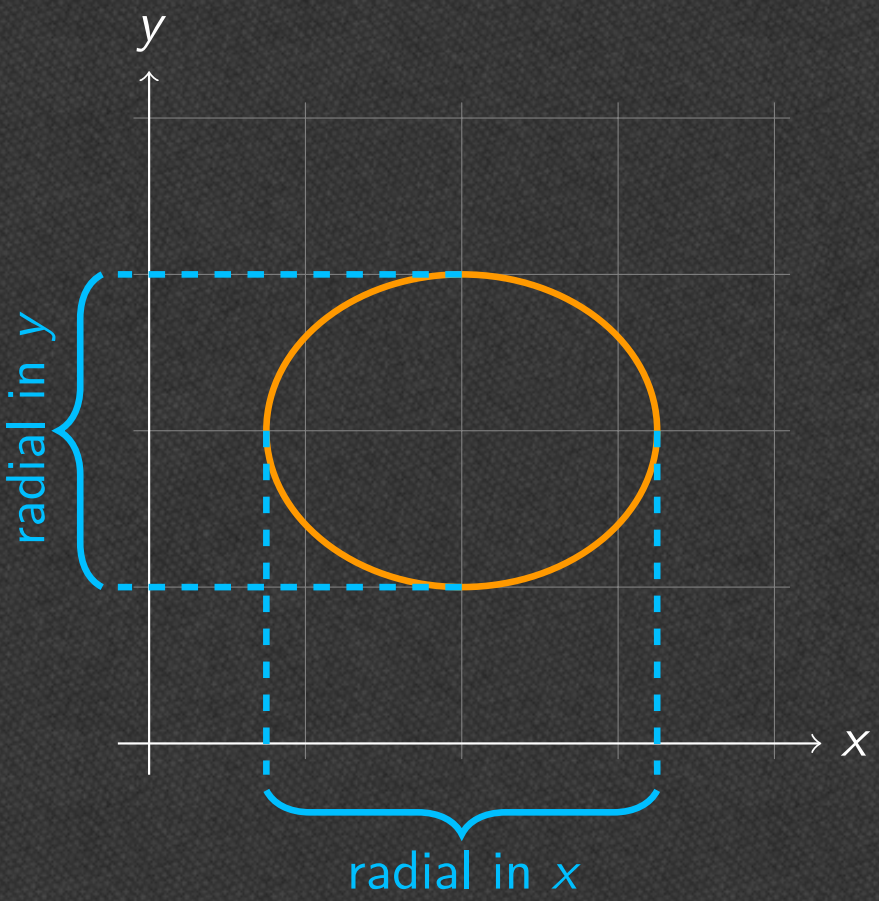
- Several conflicting objectives
- Key technology: multi-objective optimization

- big project: $>$ 100 people, expensive, 700 m long
- 1 Ångström
- to reach target it is of crucial importance to attain "good" beam properties (e.g. narrow beam/small phase space volume)

[1] http://www.psi.ch/swissfel/

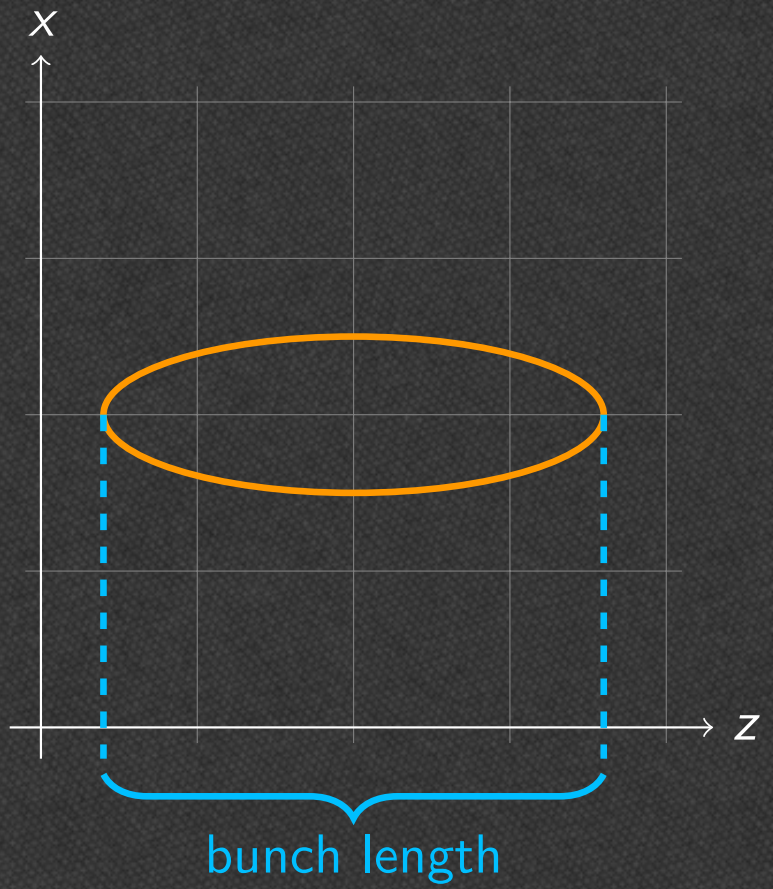
A Selection of Objectives

- transverse dimension
- bunch length
- 6D phase space
- energy spread
- emittance



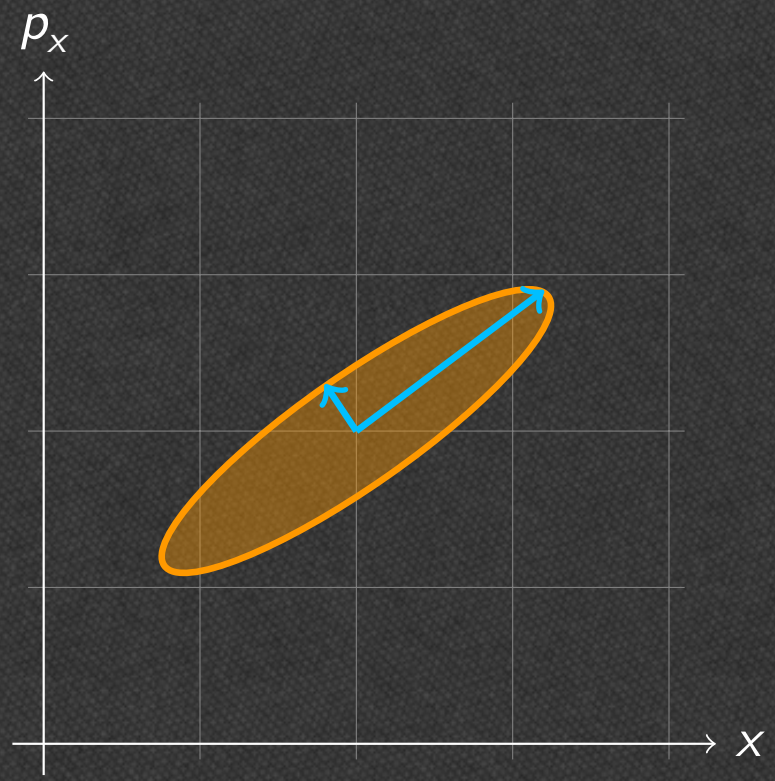
A Selection of Objectives

- transverse dimension
- bunch length
- 6D phase space
- energy spread
- emittance



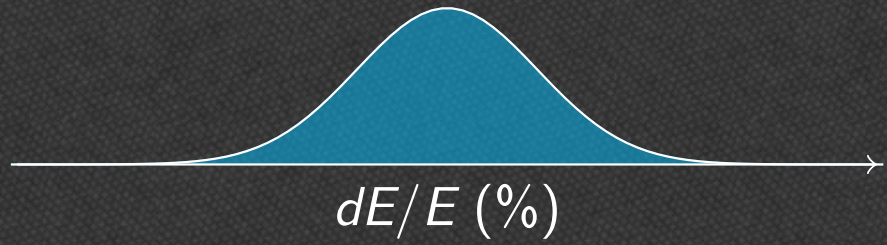
A Selection of Objectives

- transverse dimension
- bunch length
- 6D phase space
- energy spread
- emittance



A Selection of Objectives

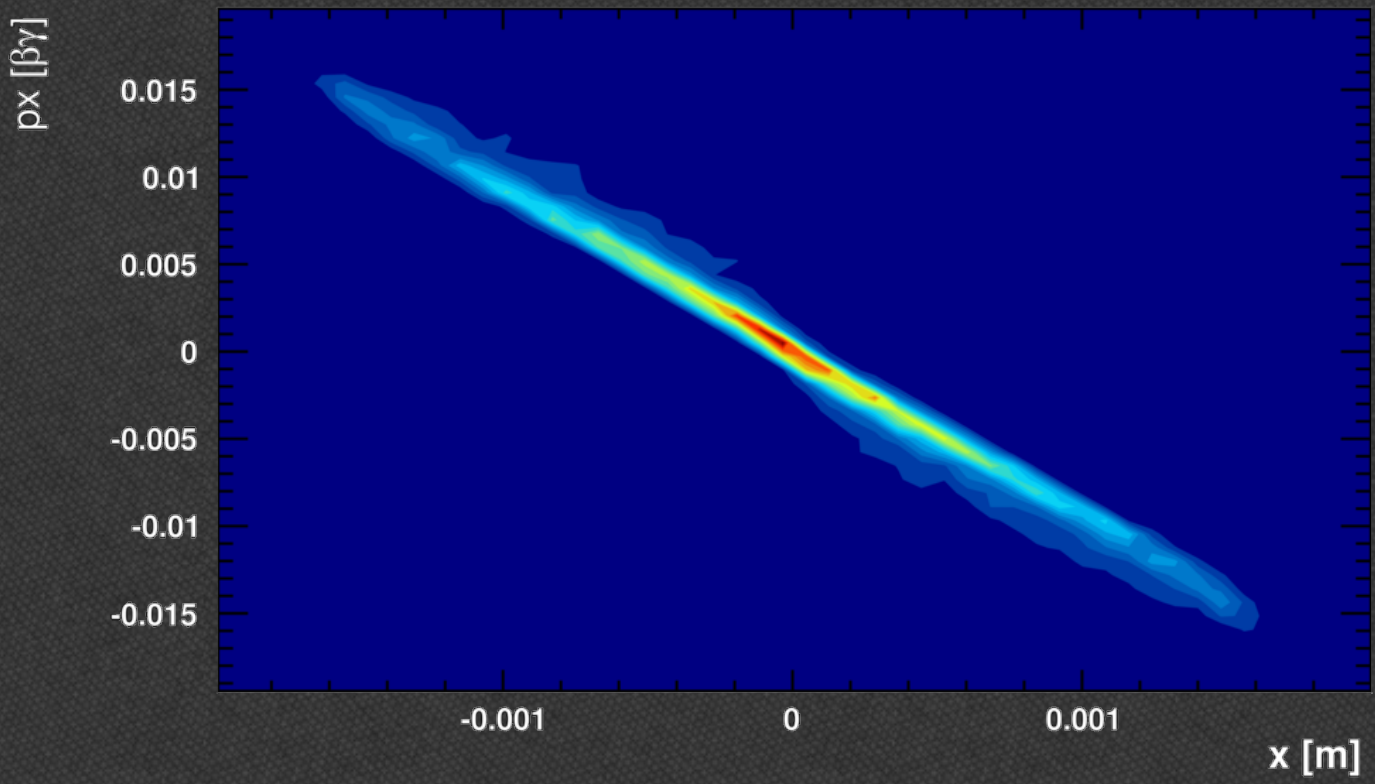
- transverse dimension
- bunch length
- 6D phase space
- energy spread
- emittance



→ energy distribution in the beam

→ small spread → particles have similar energy

Emittance



- volume of phase space
- small emittance:
  - particles confined to a small volume in configuration space
  - particles have same momenta spread

Design Variables

Lots of elements with “knobs” to turn/tune:

- EM field strength
- acceleration voltage
- radio frequency
- radio frequency phase
- ...

Typically > 10 design variables

Maxwell's Equation in the Electrostatic approximation

1,2 or 3D Field Maps & Analytic Models $(\mathbf{E}, \mathbf{B})_{\text{ext}}$

Poisson Problem s.t. BC's

$$\Delta \phi'_{sc} = -\frac{\rho'}{\epsilon_0}$$

$$\rightarrow \quad (E, B)_{sc}$$

Electro Magneto Optics

$$\mathbf{H} = \mathbf{H}_{\text{ext}} + \mathbf{H}_{\text{sc}}$$

N-Body Dynamics

If $\mathbf{E}(\mathbf{x}, t)$ and $\mathbf{B}(\mathbf{x})$ are known, the equation of motion can be integrated:

- Boris-pusher (adaptive version soon!)
- Leap-Frog
- RK-4

Forward solver is expensive to run..

Forward solver is
expensive to run..

and we require a massive number
of forward solves per optimization

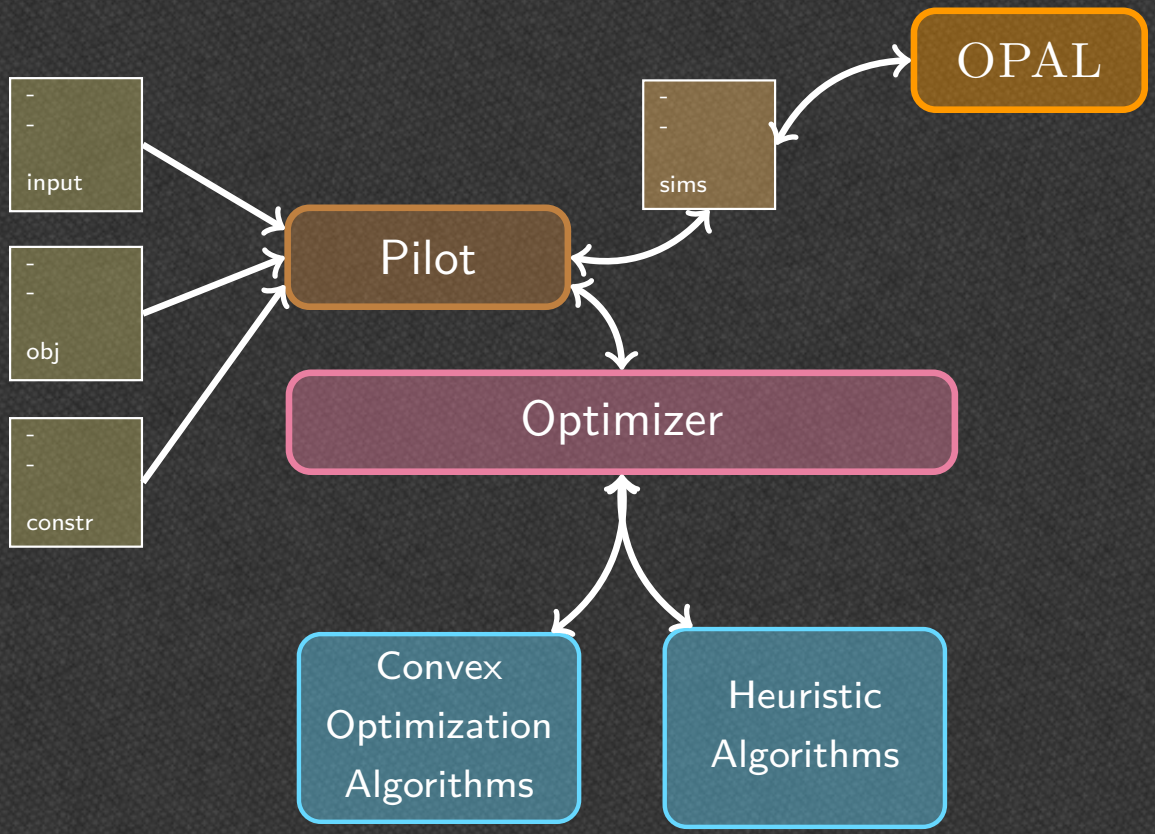
Multi-Objective Optimization Framework

- large multi-objective optimization problems
- conflicting goals
- expensive models
- ...

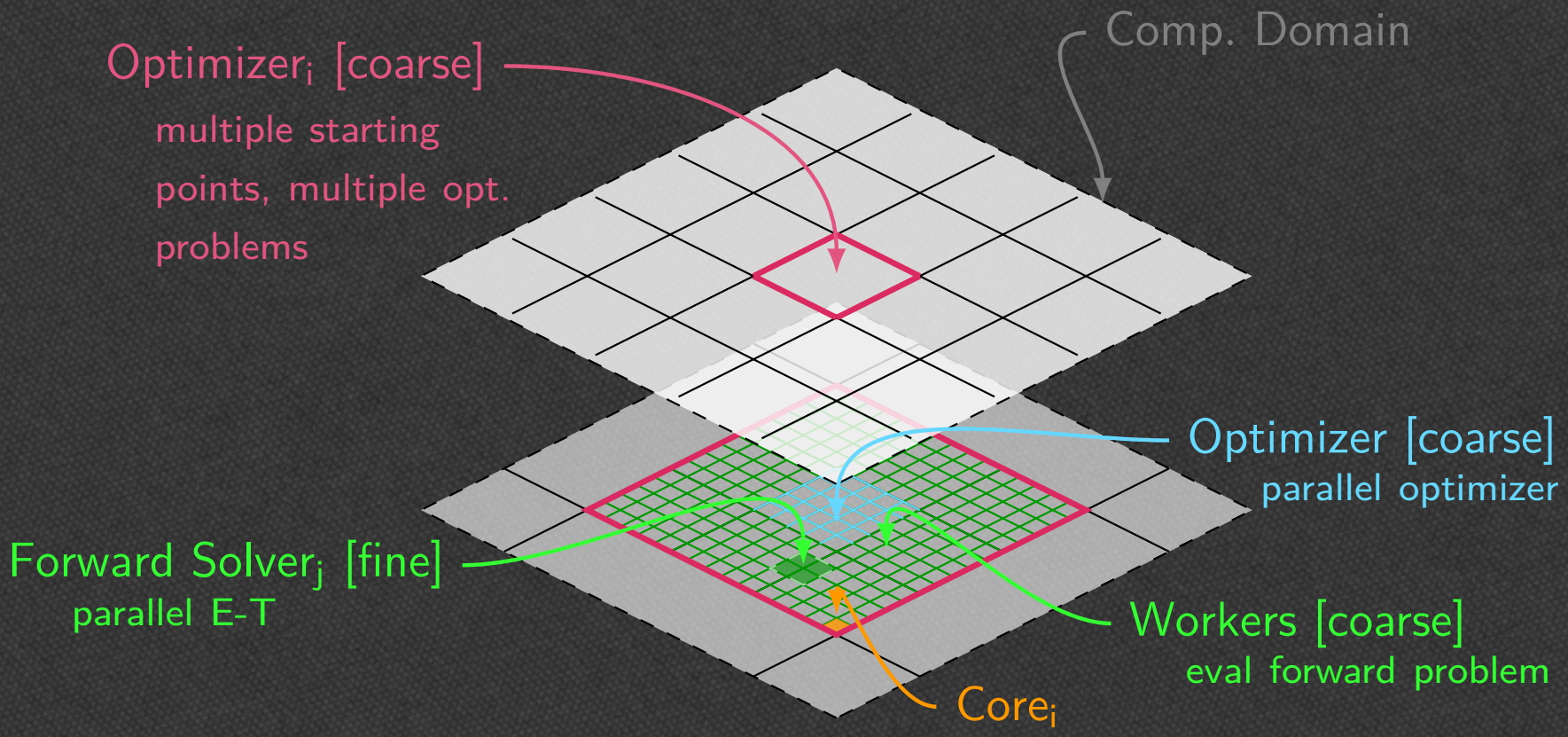
common settings for many applications

- structur analysis
- scheduling
- design
- ...

Multi-Objective Optimization Framework



Master/Worker Model



Issues with M/W Communication Model

- Exponential increase in hot-spot latency with increasing # cores
- Increasing congestion on links surrounding the master
- Gets worse with increasing message sizes

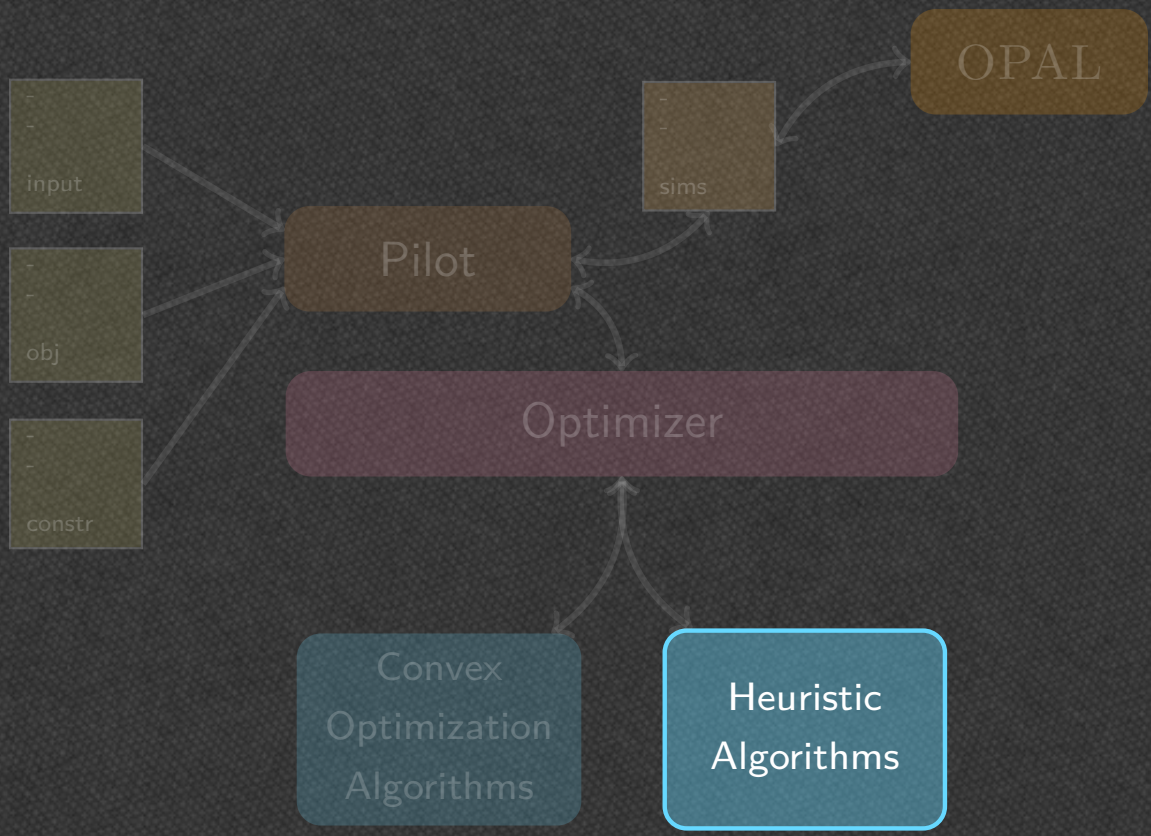
Flat network topologies not optimal

Adapt number of masters depending on # cores

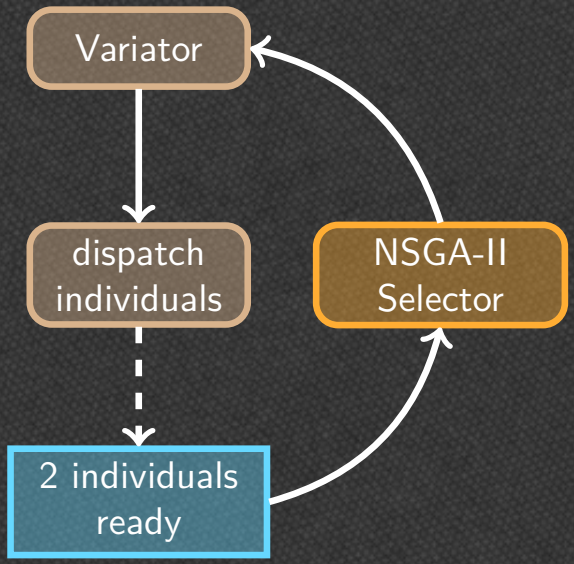
Place master taking network topology into consideration

Towards Message Passing for a Million Processes: Characterizing MPI on a Massive Scale Blue Gene/P, P. Balaji et. al.

Optimization Algorithms



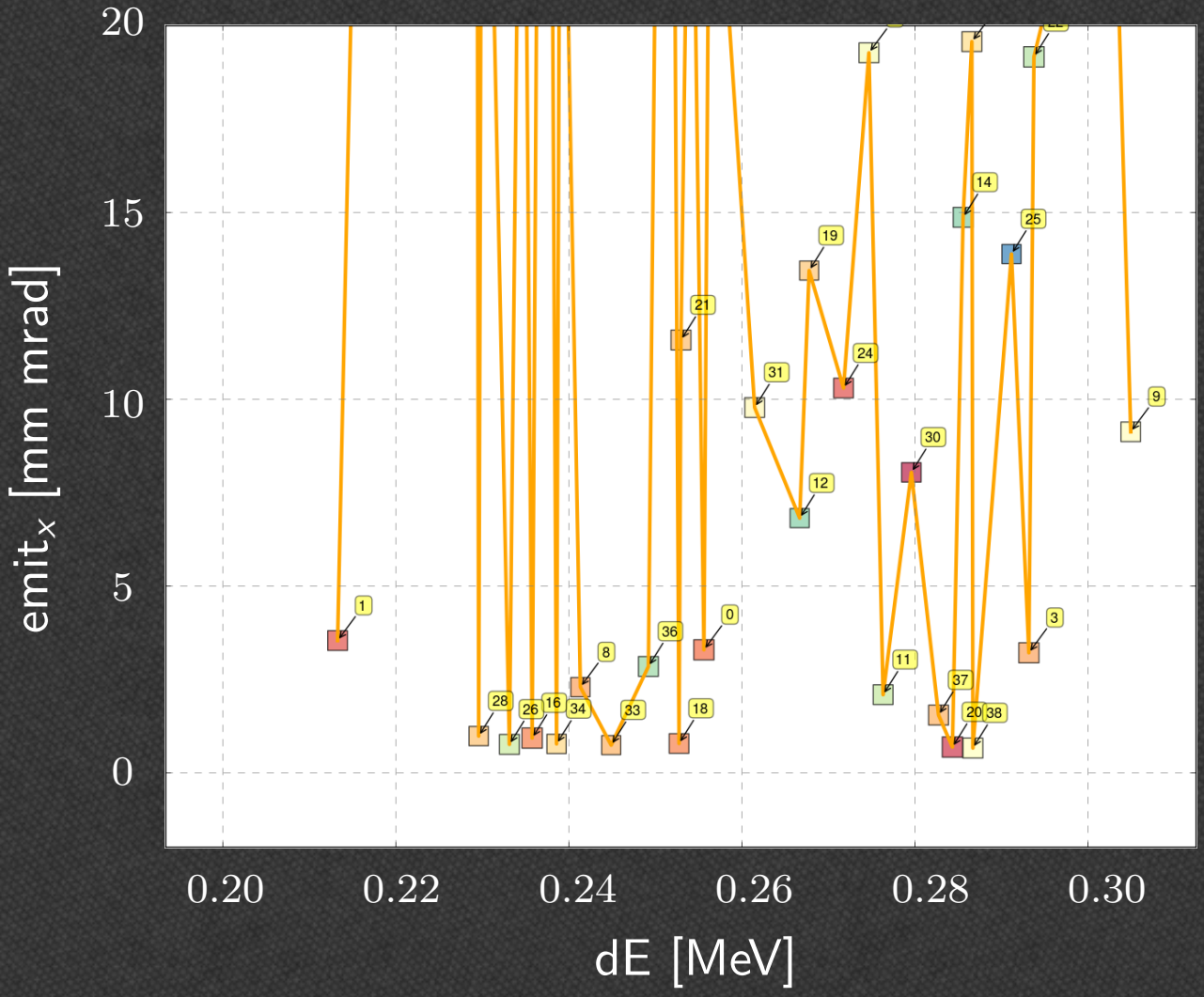
Genetic Algorithms



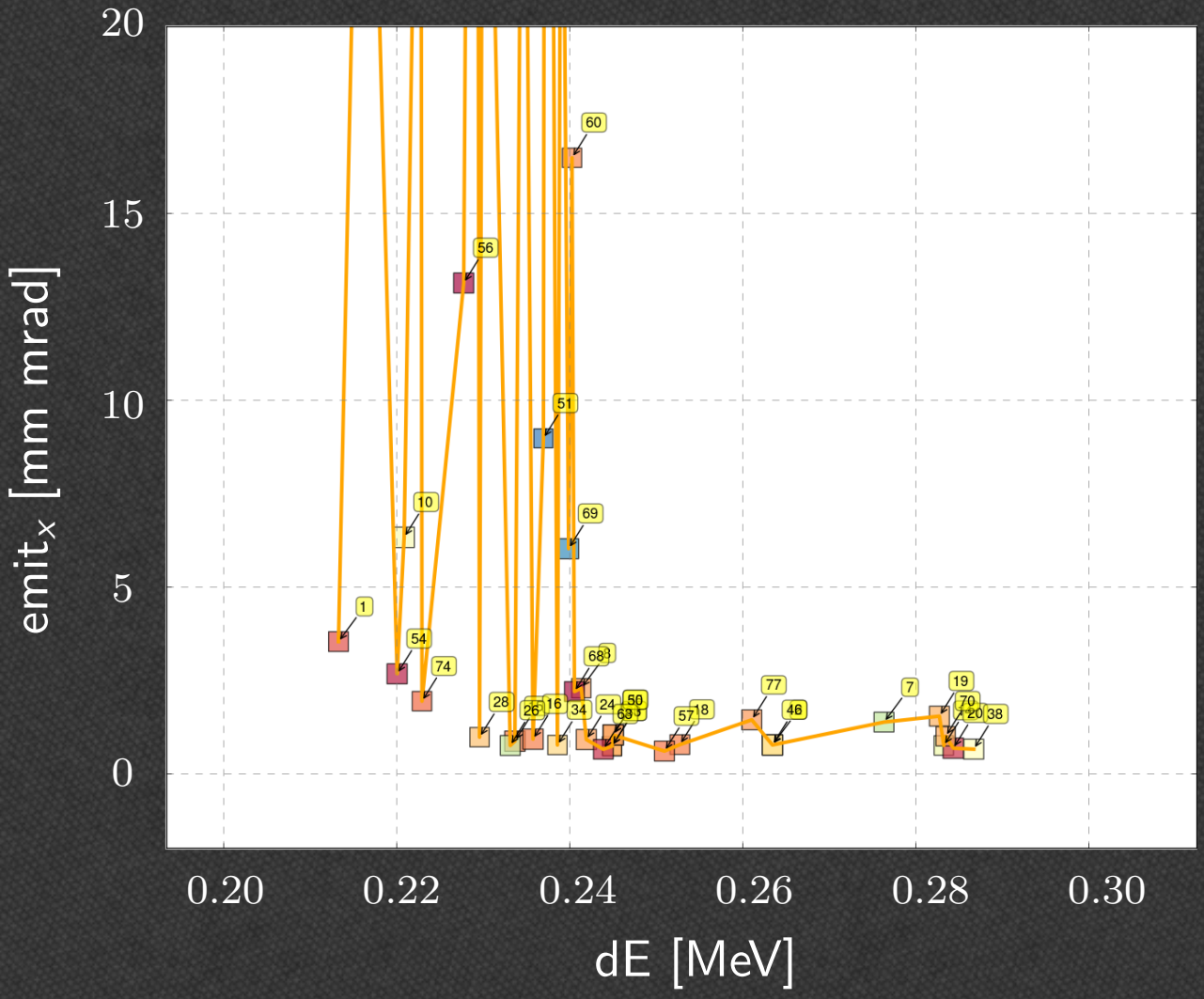
- PISA[1]
  - Finite state machine
  - NSGA-II selector
- "Continuous generations"
- Independent bit mutations
- Various crossover polices

[1] A Platform and Programming Language Independent Interface for Search Algorithms: `http://www.tik.ee.ethz.ch/pisa/`

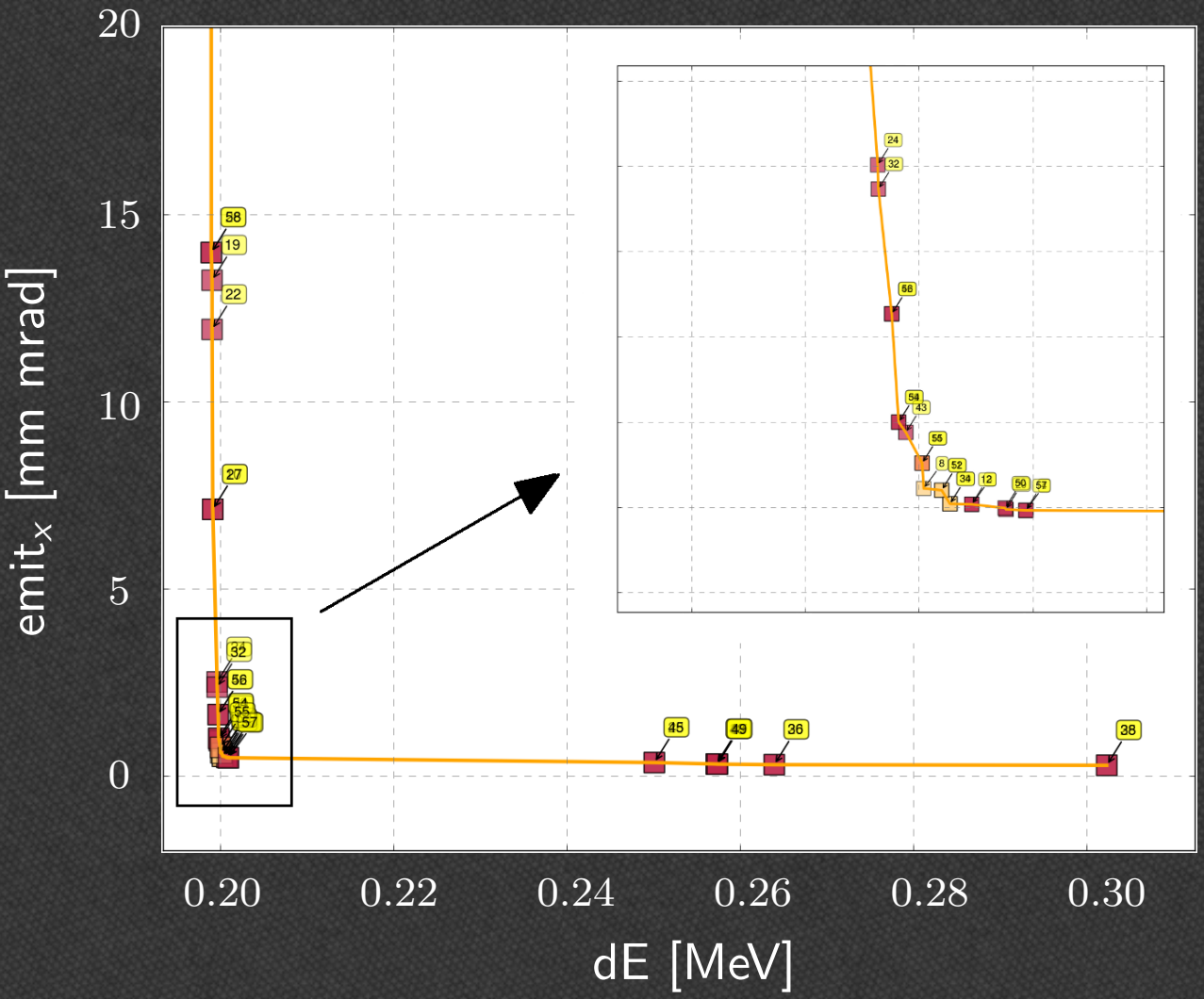
First Population



30th population

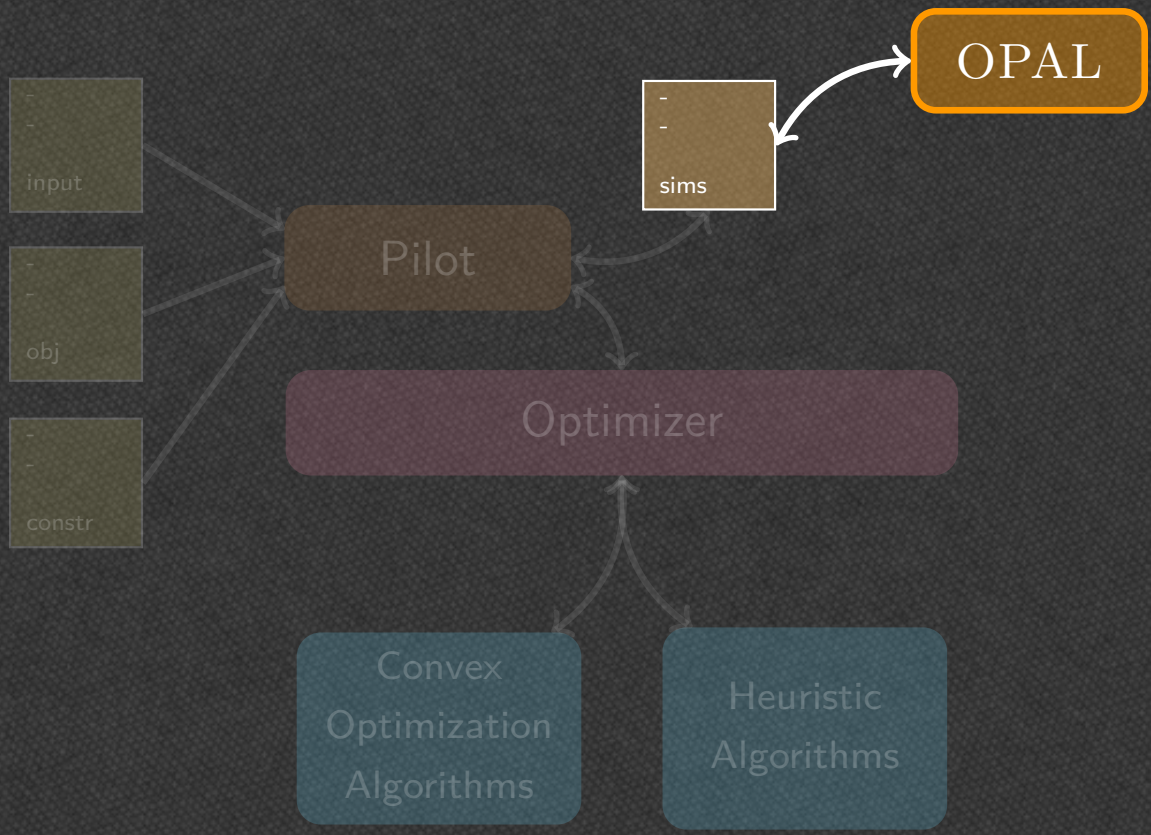


649th Population



Now scientist/operator can specify PREFERENCE

Forward Solver



Object Oriented Parallel Accelerator Library (OPAL)

OPAL is a tool for **precise** charged-particle optics in large accelerator structures and beam lines including 3D space charge.

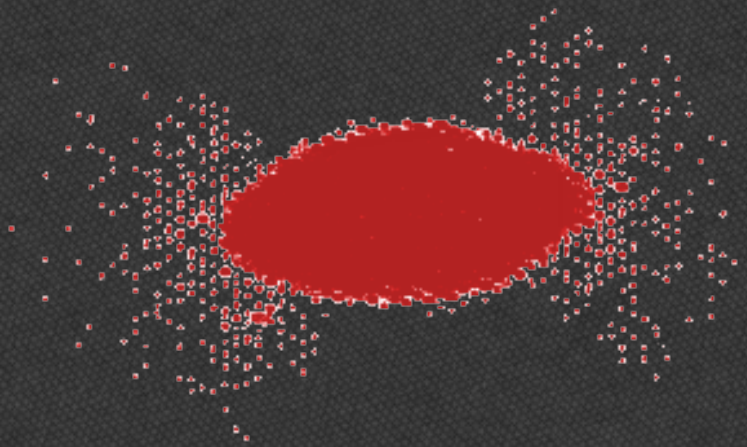
- built from the ground up as a parallel application
- runs on your laptop as well as on the largest HPC clusters
- uses the MAD language with extensions
- written in C++ using OO-techniques, hence OPAL is easy to extend
- nightly regression tests track the code quality

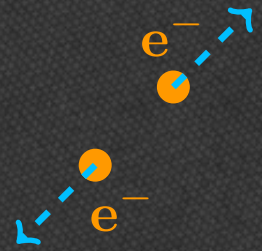
OPAL: `https://amas.psi.ch/OPAL`

Architecture

- **OPAL Object Oriented Parallel Accelerator Library**
- **IP^{2}L Independent Parallel Particle Layer**
- **Class Library for Accelerator Simulation System and Control**
- **H5hut for parallel particle and field I/O (HDF5)**
- **Trilinos http://trilinos.sandia.gov/**

3D Tracker

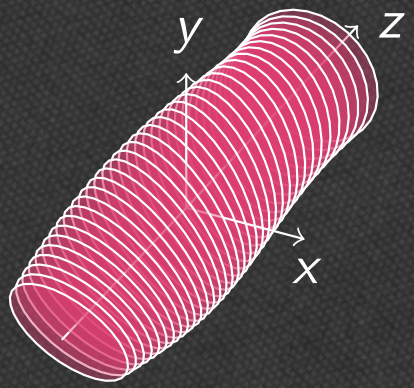




repulsive force of charged particles

- Huge # of macro particles (100'000 – 100'000'000)
- Computing space-charge is expensive
- Load balancing difficult
- Lots of synchronization points

Envelope Tracker



- # slices $\ll$ # macro particles
- Slices distributed in contiguous blocks
- Load imbalance of at most 1 slice
- Low number of synchronization points

Envelope Equations

Using a 5th order Runge-Kutta time stepper to update

- Slice radius:

$$\frac{d^2}{d^2 t} R_i = \ldots$$

- Slice energy:

$$E = \ldots$$

- Slice position:

$$\frac{d}{dt} z_i = \ldots$$

Model Implementation

```
1:  procedure ExecuteSliceTracker(t_n, slices)
2:      for all t_i ∈ t_1, ··· , t_{n−1}, t_n do
3:          for all s ∈ slices do
4:              getExternalFields(s)
5:          end for
6:          synchronizeSlices()
7:          calcSpaceCharge()
8:          for all s ∈ slices do
9:              timeIntegration(s)
10:         end for
11:     end for
12: end procedure
```

Model Implementation

```
1:  procedure ExecuteSliceTracker(t_n, slices)
2:      for all t_i ∈ t_1, ··· , t_{n−1}, t_n do
3:          for all s ∈ slices do
4:              getExternalFields(s)
5:          end for
6:          synchronizeSlices()
7:          calcSpaceCharge()
8:          for all s ∈ slices do
9:              timeIntegration(s)
10:         end for
11:     end for
12: end procedure
```

N-Body Space Charge

```
1: procedure ComputeSpaceCharge(slices)
2:     for all i ∈ my_slices do
3:         for all j ∈ all_slices do
4:             sm ← interaction(s_j, s_i)
5:         end for
6:         F_{l,i} ← longitudinalSpaceCharge(sm)
7:         F_{t,i} ← transversalSpaceCharge()
8:     end for
9: end procedure
```

Analytical Space Charge

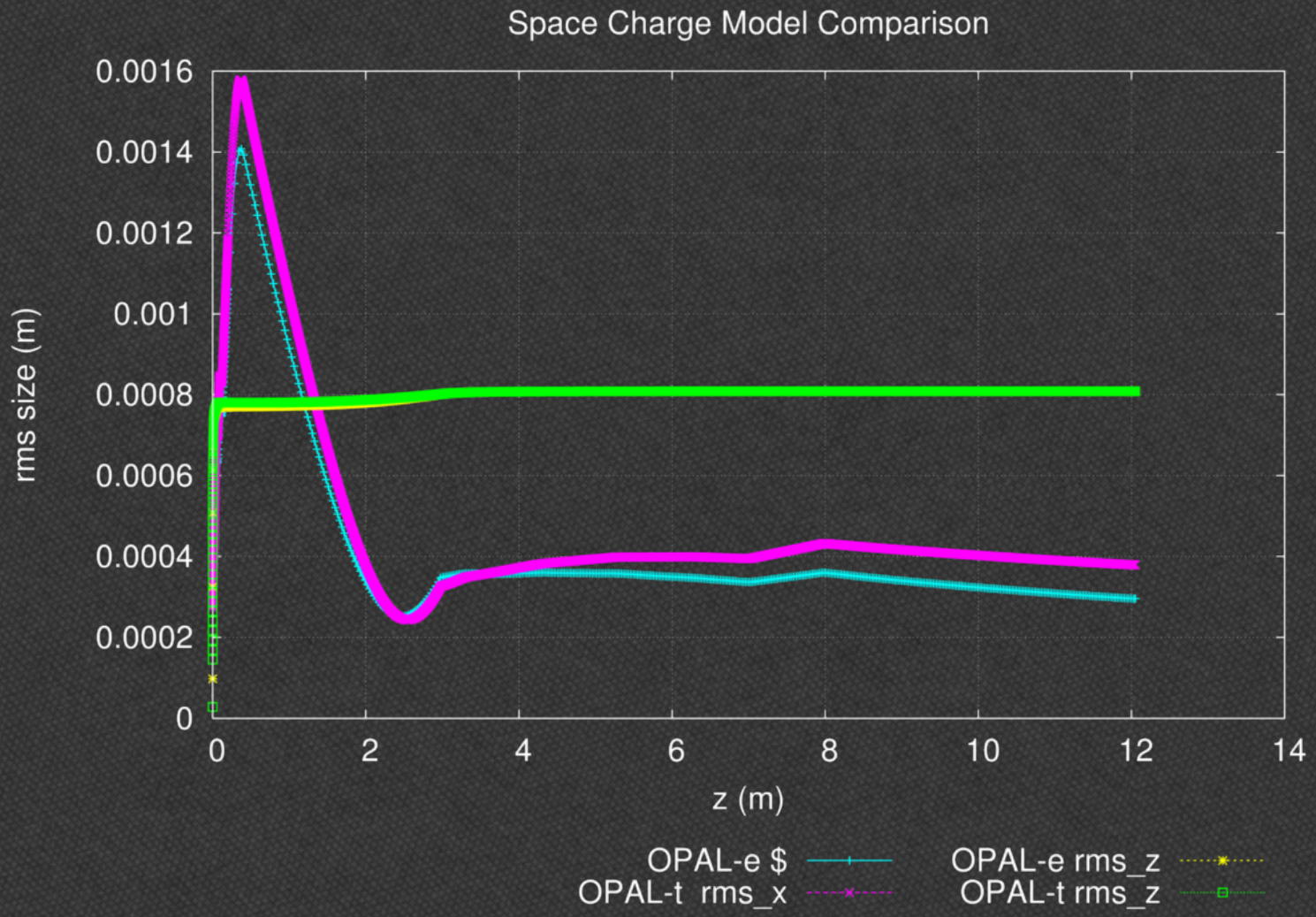
$$\text{Electrical Field} = \frac{Q}{2\pi\epsilon_0 R^2}\left[\sqrt{\left(1-\frac{z}{L}\right)^2+\left(\frac{R}{L}\right)^2}\right.$$
$$\left.-\sqrt{\left(\frac{z}{L}\right)^2+\left(\frac{R}{L}\right)^2}-\left|1-\frac{z}{L}\right|+\left|\frac{z}{L}\right|\right]$$

see HOMDYN, M. Ferrario, http://nicadd.niu.edu/fnpl/homdyn/manual.pdf

Complexity

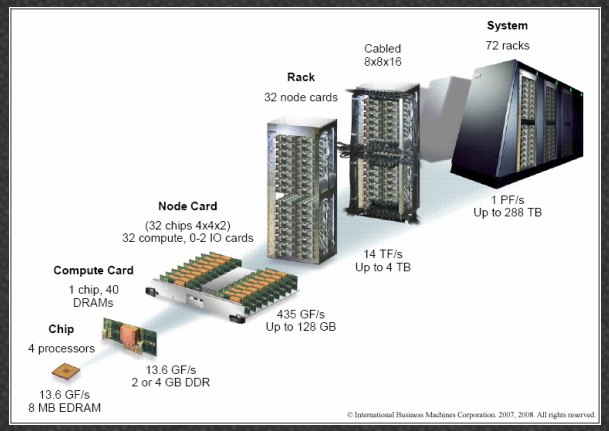
Phase	FLOPs	#Allreduce	size
getExternalFields()	$\mathcal{O}(m)$	–	
synchronizeSlices()	$\mathcal{O}(1)$	2	n
calcSpaceCharge()	$\mathcal{O}(n^2)$	2	1
calcSpaceCharge()*	$\mathcal{O}(1)$	2	1
timeIntegration()	$\mathcal{O}(n)$	–	

Typical $n = 10000$, $m = 150$



Important quantities withtin 5% margin w. r. t. 3D tracker

Computational Platform



Blue Gene/P[2]:

- VN mode
- Quad core 450 PowerPC (850 MHz)
- peak performance of 13.6 GFLOP/s (per node)
- 3D Torus network with bandwidth of 6 GB/s
- collective (tree) network has a bandwidth of 2 GB/s
  - round-trip worst case latency of 2.5 μs

Running first 2000 steps of SwissFel.

[2]Blue Gene/P is a trademark of the International Business Machines Corporation in the United States, other countries, or both.

“History”

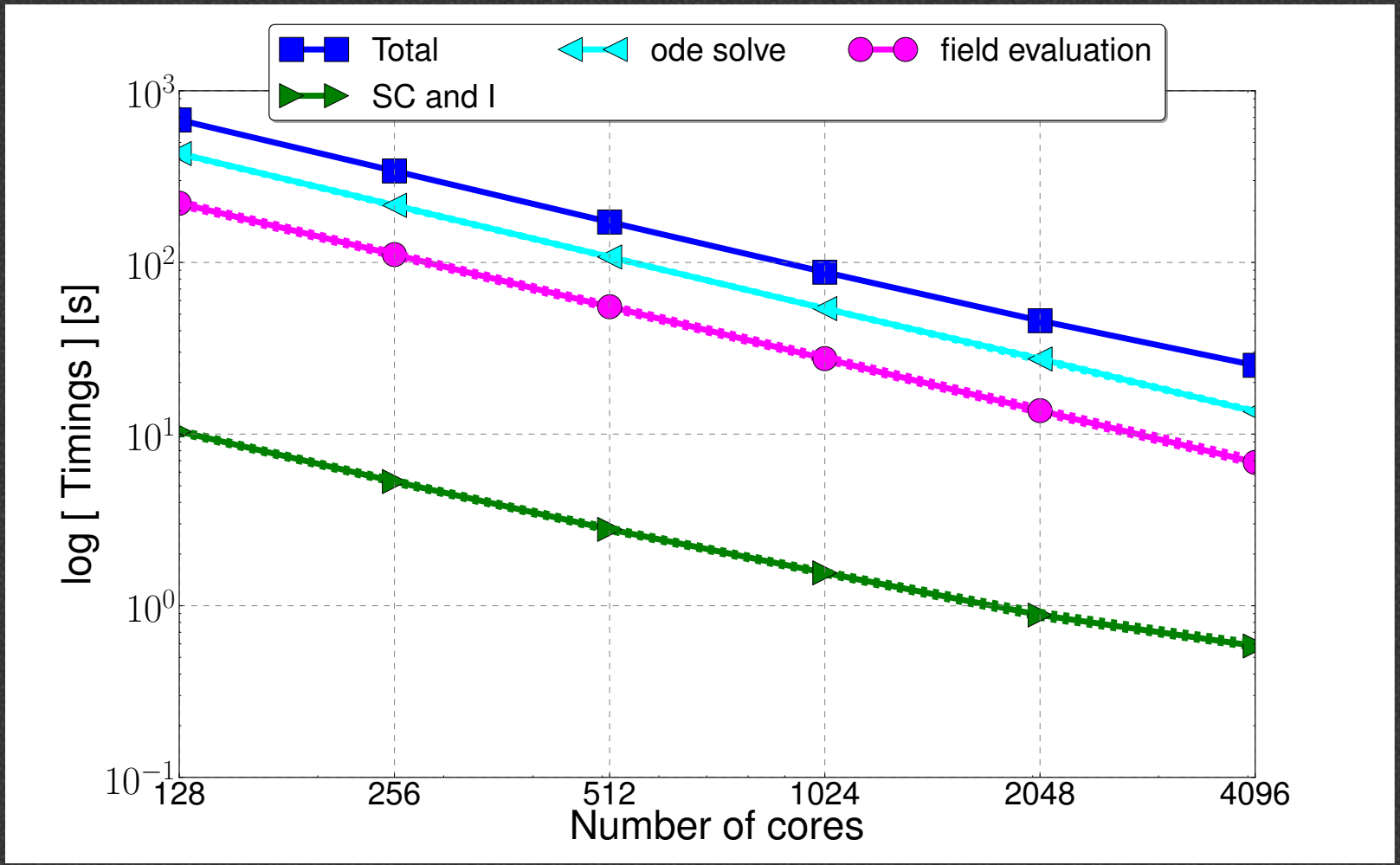
A stony road to achieve scalability..

Cores	Wall [s]	Comm [s]
1	321.46	–
2	167.93	3.67 (2.19%)
4	92.9	6.86 (7.43%)
8	57.1	9.08 (15.91%)

Run on FELSIM Cluster at PSI.

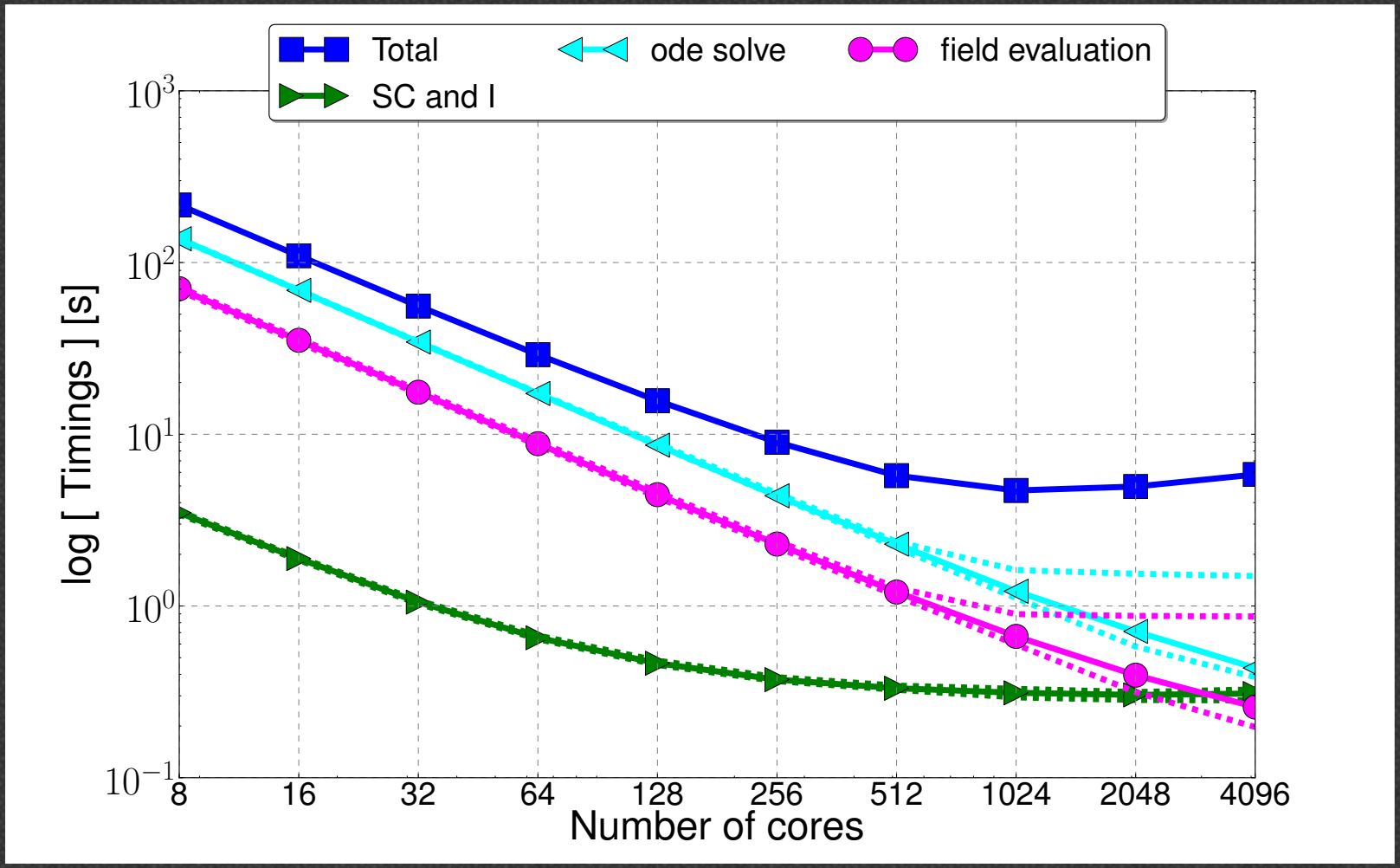
- proper load balancing
- profiling and benchmarking: serial parts (Amdahl’s law)
- one fix synchronization point per timestep

500'000 Slices

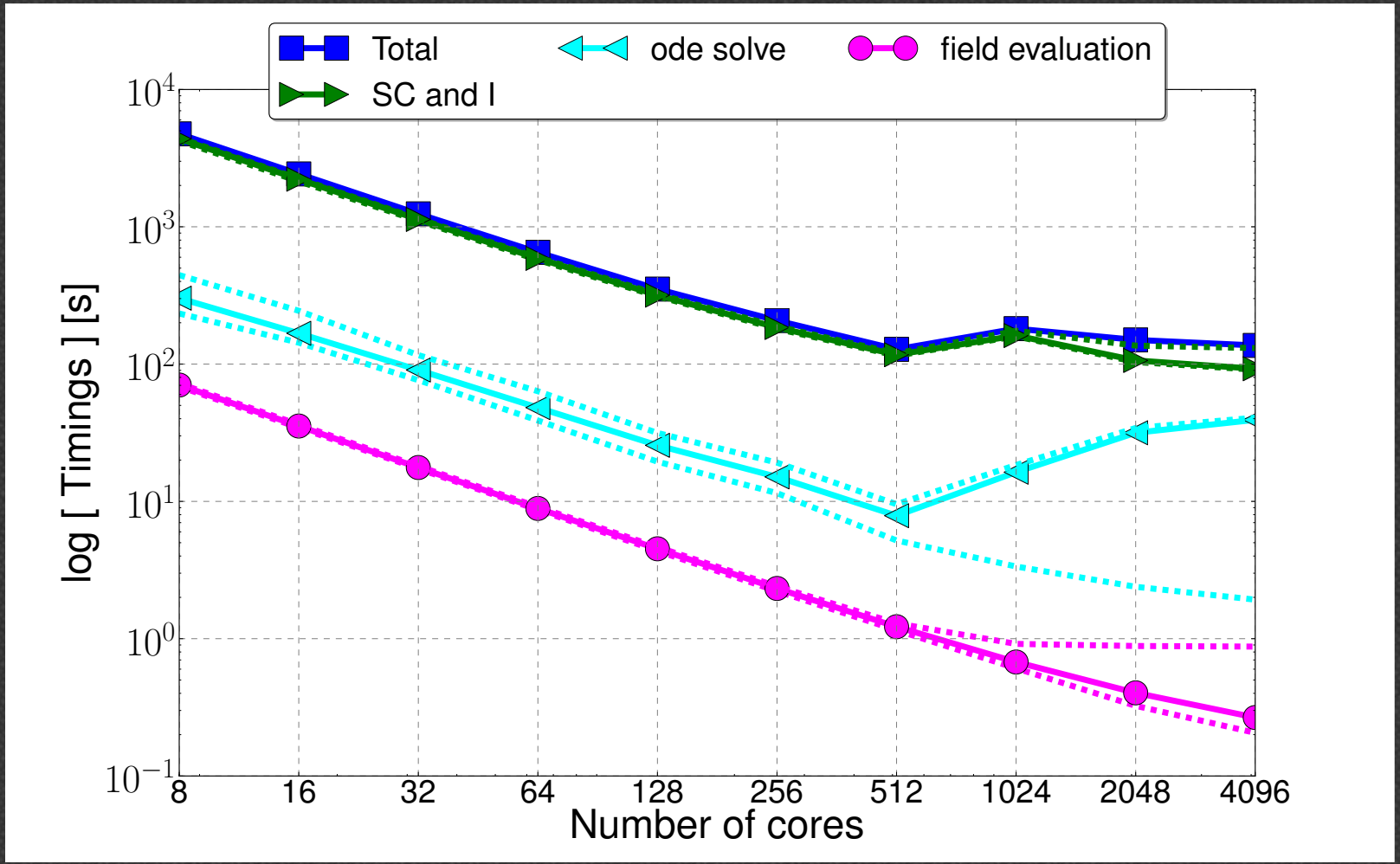


Analytical space-charge.

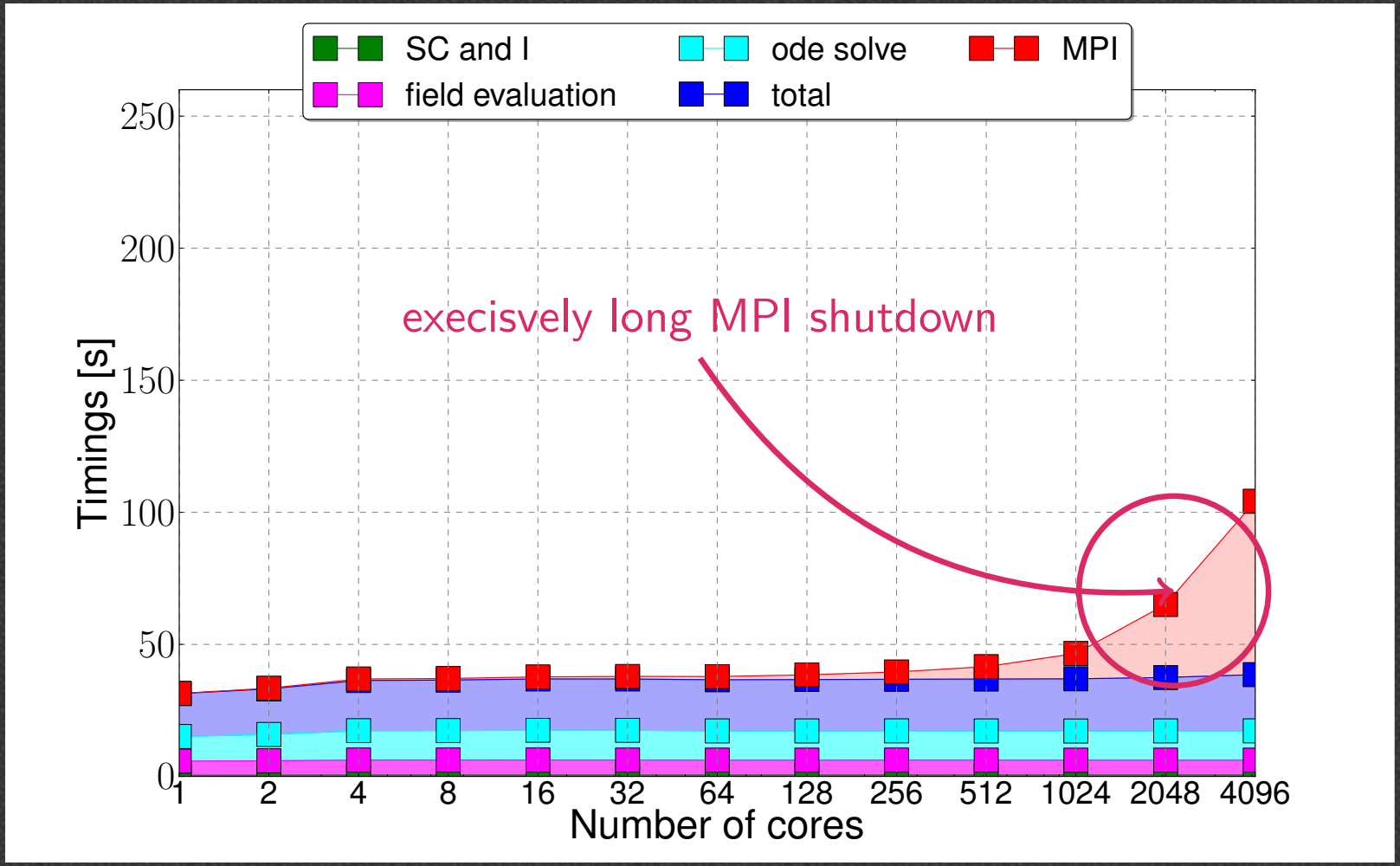
10'000 Slices: analytical Space-Charge



10'000 Slices: n^2 Space-Charge

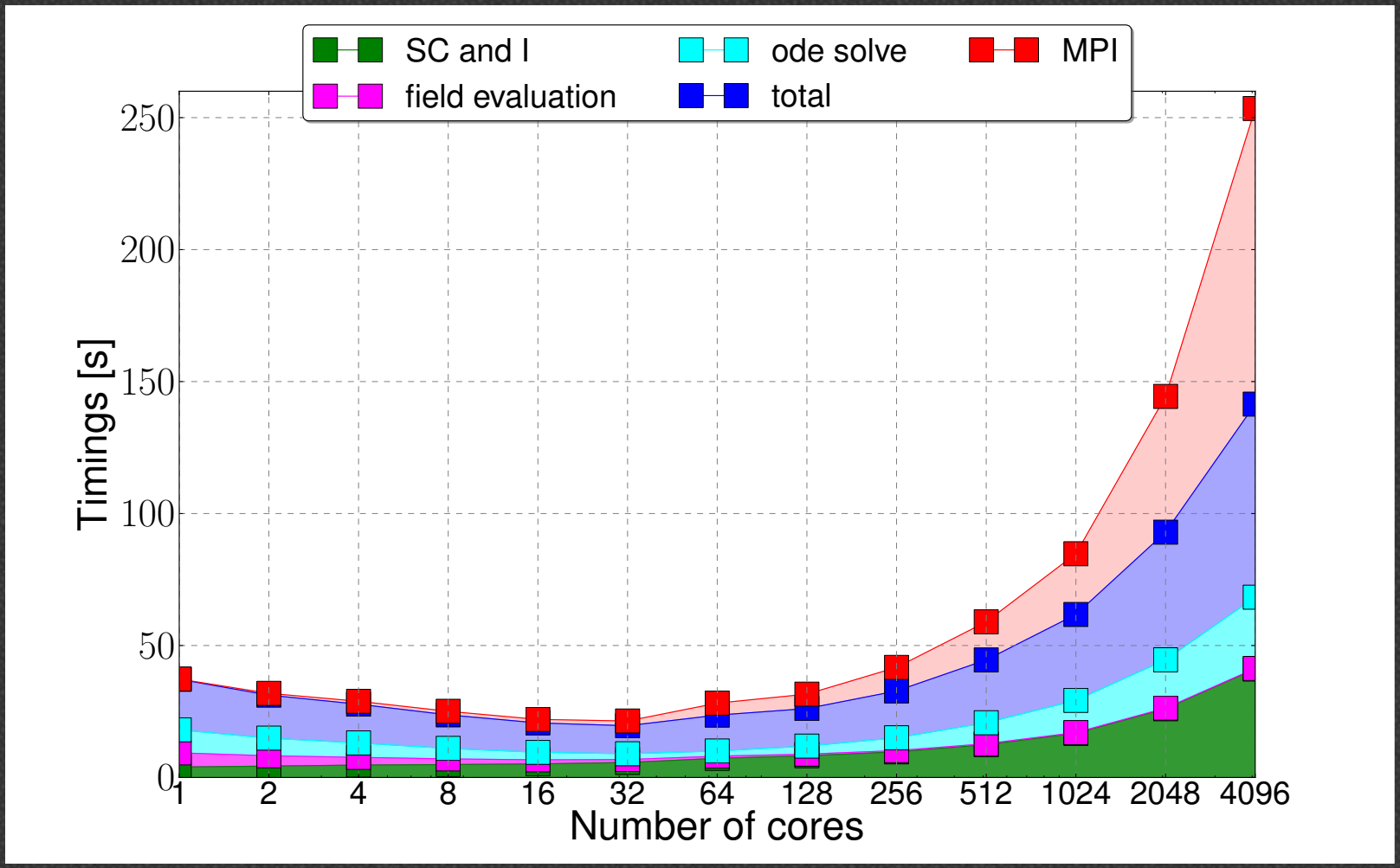


Analytical Weak Scaling



Problem scaling: n

n^2 Weak Scaling



Problem scaling: $\sqrt{n}$

Conclusions

→ Fast & scalable forward solver:

- introduced a new low-dimensional model
- almost 2 orders of magnitude reduction of runtime

→ Crucial cornerstone of framework

- becomes feasible with presented forward solver
- vary "resolution" (3D, n^2 sc., analytical sc.)
- achieve an acceptable parallel efficiency for one forward solve
- run as many parallel forward solves as possible

→ Helps in design/opartion of particle accelerators

Outlook

- Investigate other multi-objective optimization algorithms
- Run real optimization problem on massive number of cores
- Visualization of results

This project is funded by:

IBM Research – Zurich

Paul Scherrer Institut

Acknowledgements

- OPAL developer team
- SwissFEL team

Backup

Optimization Problem

$$\begin{aligned}
\min \quad & [\text{energy spread}, \text{emittance}] \\
\text{s.t.} \quad & \partial_t f(\mathbf{x}, \mathbf{v}, t) + \mathbf{v} \cdot \nabla_x f(\mathbf{x}, \mathbf{v}, t) + \frac{q}{m_0} \left(\mathbf{E}_{\text{tot}} + \mathbf{v} \times \mathbf{B}_{\text{tot}}\right) \cdot \nabla_v f(\mathbf{x}, \mathbf{v}, t) = 0 \\
& \nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}, \qquad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\
& \nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}, \qquad \nabla \cdot \mathbf{B} = \mathbf{0} \\
& \rho = -e \int f(\mathbf{x}, \mathbf{v}, t)\, d^3\mathbf{p}, \qquad \mathbf{J} = -e \int f(\mathbf{x}, \mathbf{v}, t) \mathbf{v}\, d^3\mathbf{p} \\
& \mathbf{E} = \mathbf{E}_{\text{ext}} + \mathbf{E}_{\text{self}}, \qquad \mathbf{B} = \mathbf{B}_{\text{ext}} + \mathbf{B}_{\text{self}} \\
& \dots
\end{aligned}$$

Envelope Equations

$$\frac{d^2}{d^2 t} R_i + \beta_i \gamma_i^2 \frac{d}{dt}(\beta_i R_i) + R_i \sum_j K_i^j = \frac{2c^2 k_p}{R_i \beta_i} \times$$

$$\left(\frac{G(\Delta_i, A_r)}{\gamma_i^3} - (1 - \beta_i^2) \frac{G(\delta_i, A_r)}{\gamma_i} \right) + \frac{4\varepsilon_n^{\text{th}} c}{\gamma_i} \frac{1}{R_i^3}$$

$$\frac{d}{dt} \beta_i = \frac{e_0}{m_0 c \gamma_i^3} \left(E_z^{\text{ext}}(z_i, t) + E_z^{\text{sc}}(z_i, t) \right)$$

$$\frac{d}{dt} z_i = c \beta_i$$