
WIR SCHAFFEN WISSEN – HEUTE FÜR MORGEN

Sven Reiche :: SwissFEL :: Paul Scherrer Institute

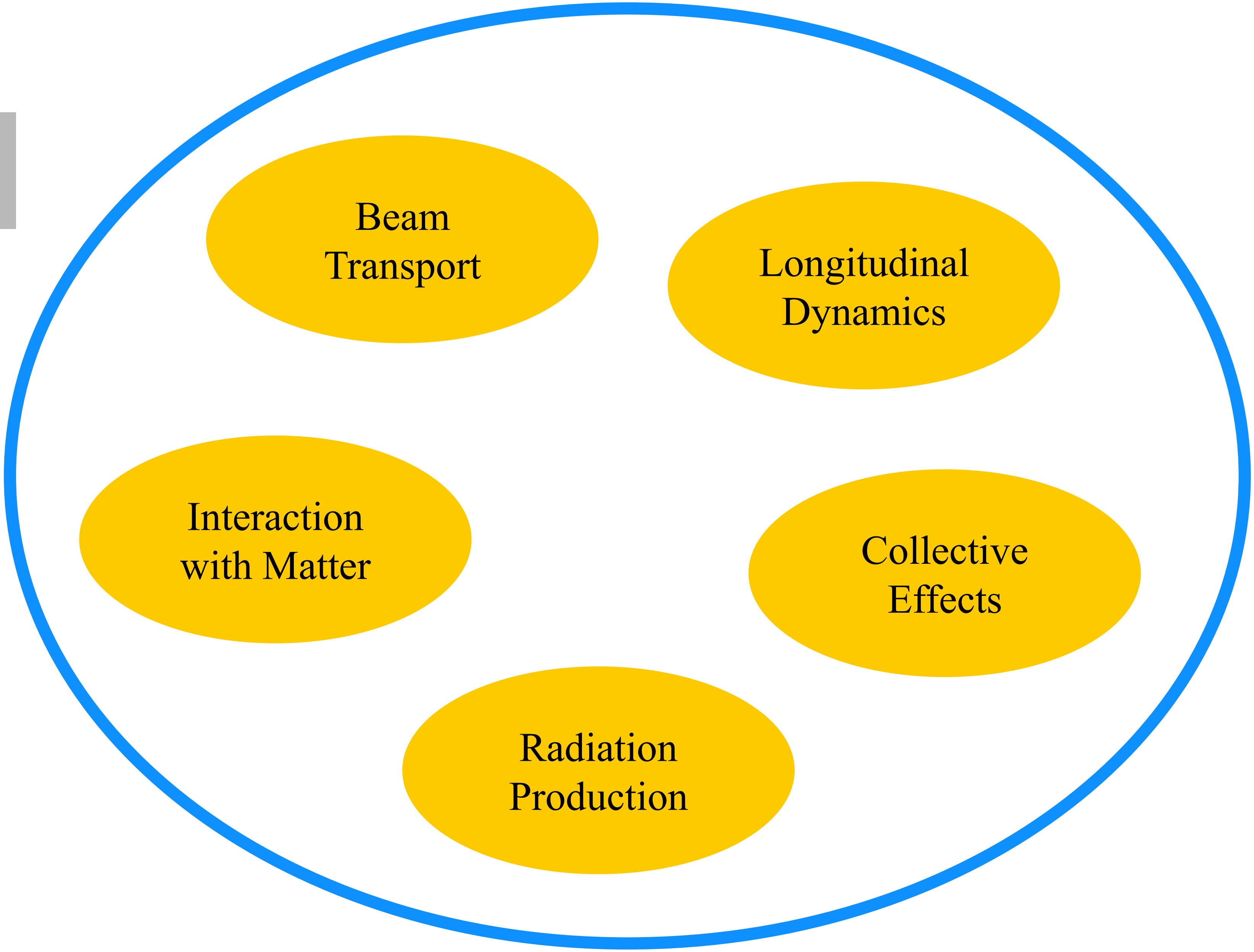
Simulation Codes

OMA School for Medical Accelerators, Pavia, June 2017

Overview

- Numerical Problems
- Numerical Modelling
- Codes and their Interface
- Conclusion

Building Blocks of Modelling Accelerators



Beam Transport

Longitudinal Dynamics

Interaction with Matter

Collective Effects

Radiation Production

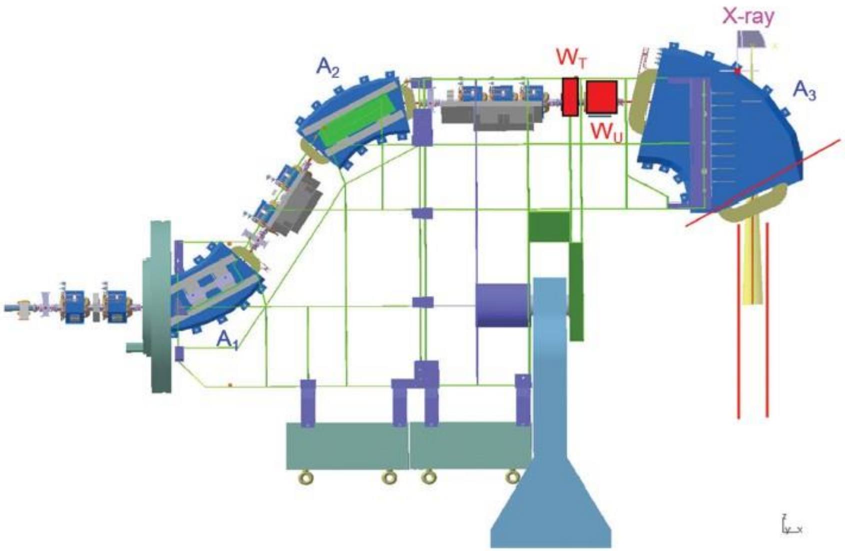
Beam Transport

Example: Gantry 3 Trajectories at PSI:

Trajectory Variations due to:

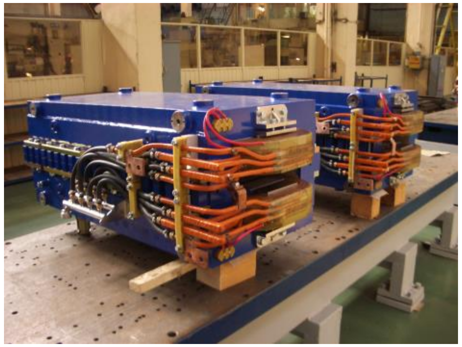
- Injection position
- Injection angles
- Energies

Gantry Layout

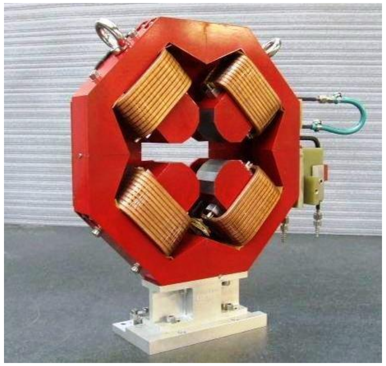


Typical Beam Transport Elements

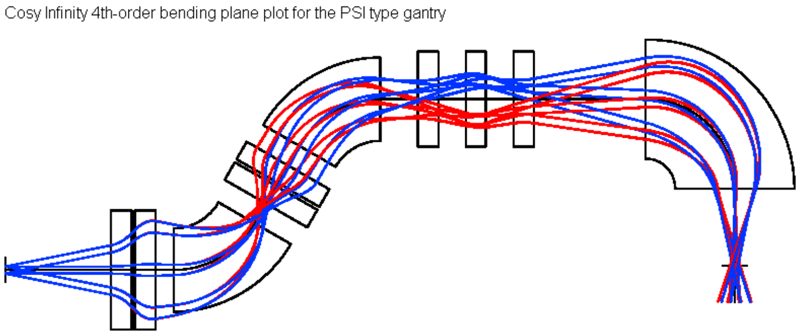
Dipole (Orbit)



Quadrupole (Beam Size)

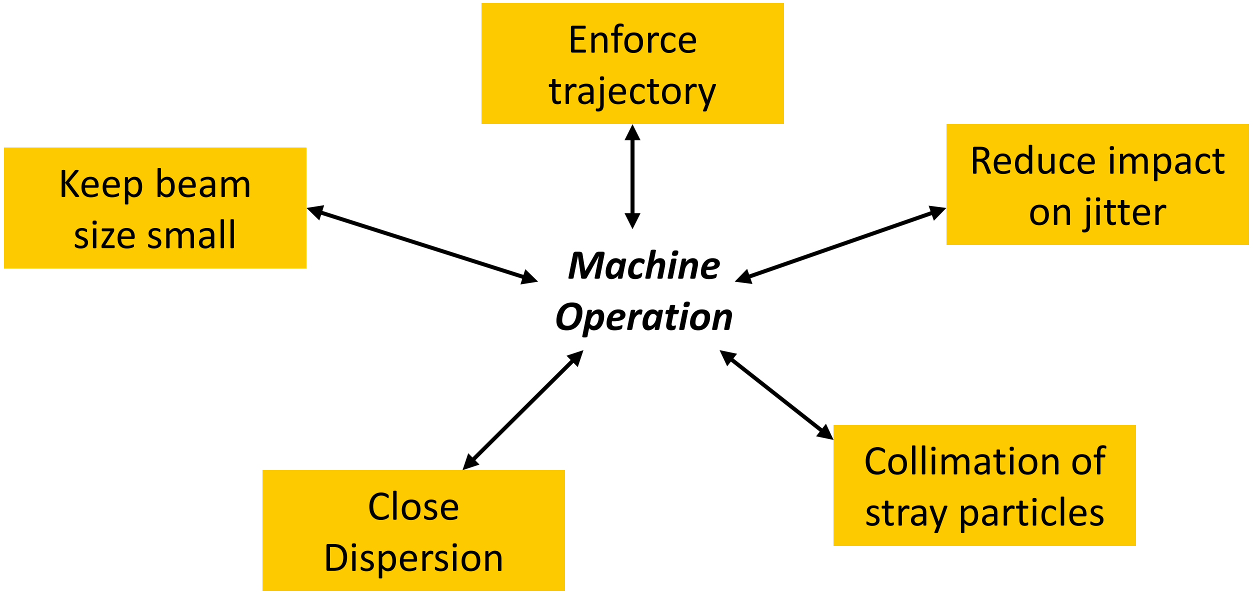


Trajectories with different angles and energies



Beam Transport Goals

Typical tasks for setting up transport with a given layout. Setup typically done with tracking code (e.g. MadX):



Enforce trajectory

Keep beam size small

Reduce impact on jitter

Machine Operation

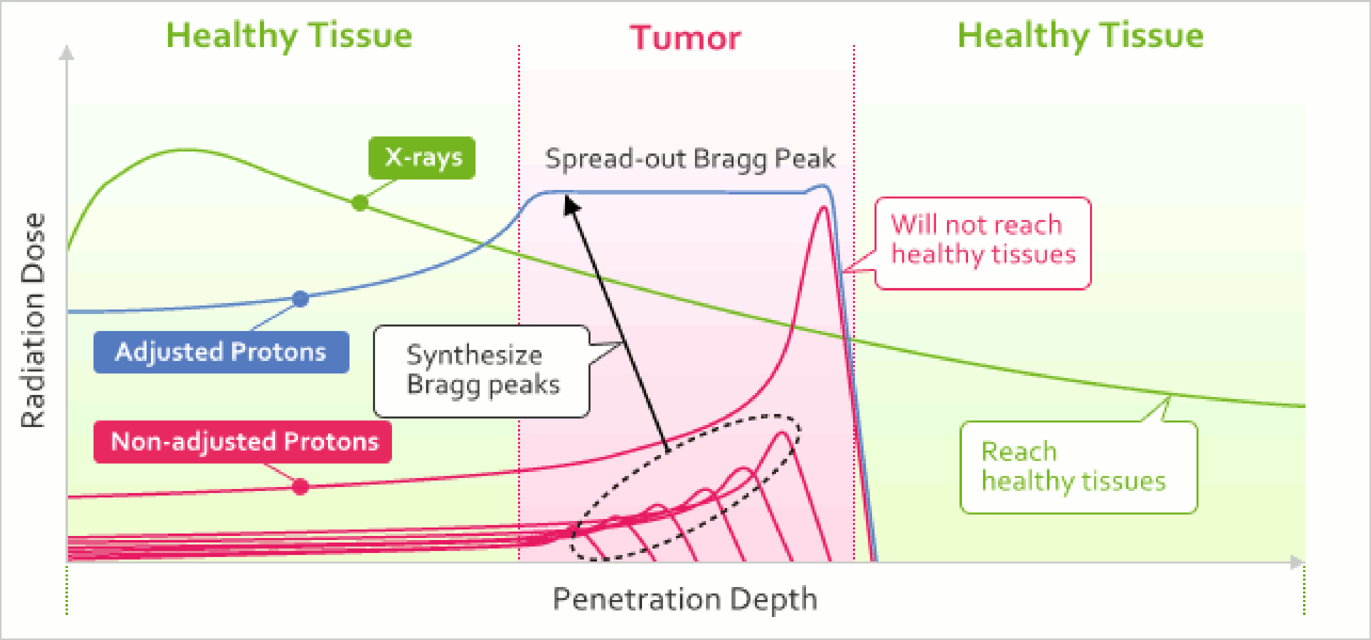
Close Dispersion

Collimation of stray particles

Reliable machine (medical accelerators) rely on having a good design optics. Challenging task for the design

Longitudinal Dynamics

Control on Particle Distribution in Energy



Stopping position (Bragg peak) depends on proton energy

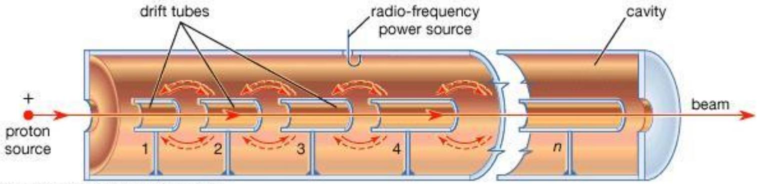
For best irradiation:

- Control on beam energy
- Control on beam intensity (current)

Controlling Energy (< 500 MeV Protons)

1. Controlling Acceleration

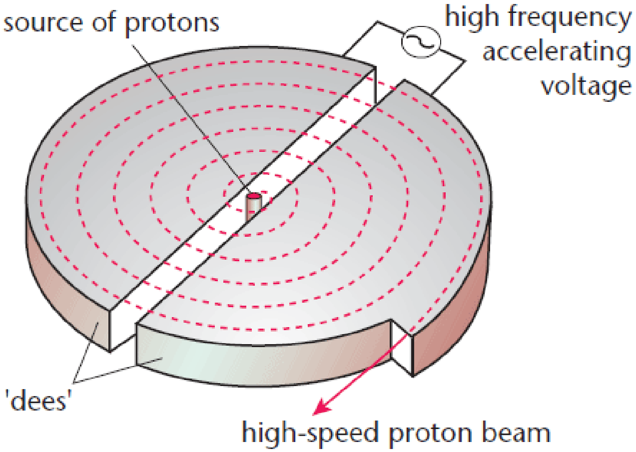
Linac



© 2007 Encyclopædia Britannica, Inc.

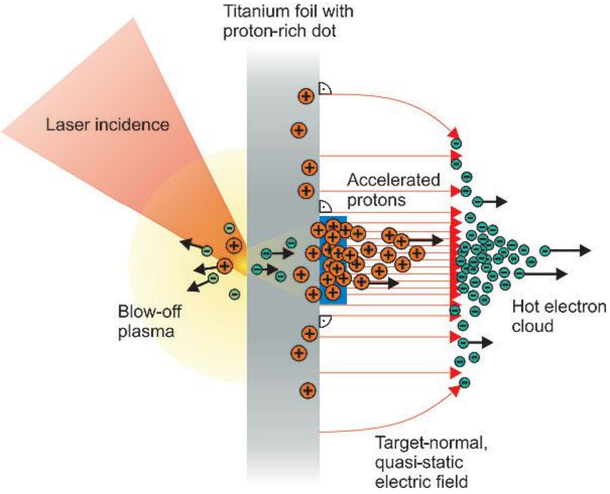
Inefficient for variable energy because time-of-flight of non-relativistic protons is encoded in layout

Cyclotron

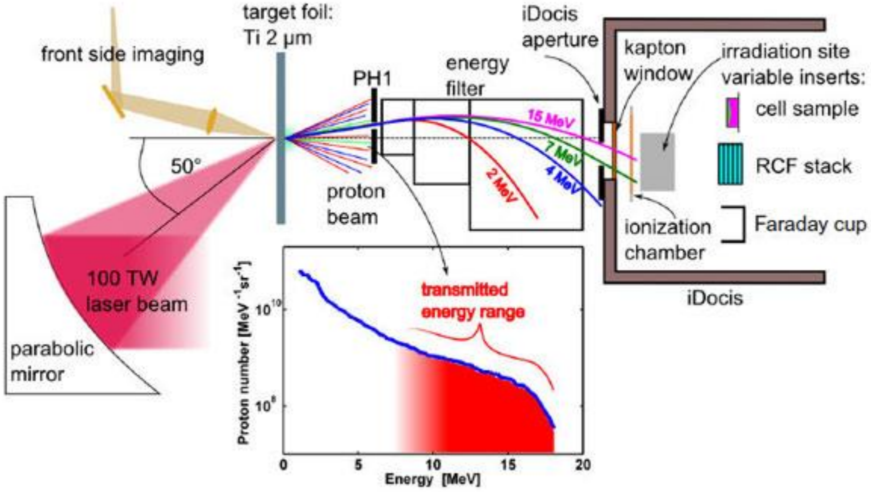


2. Filtering Energy (Collimation)

Laser-driven Source

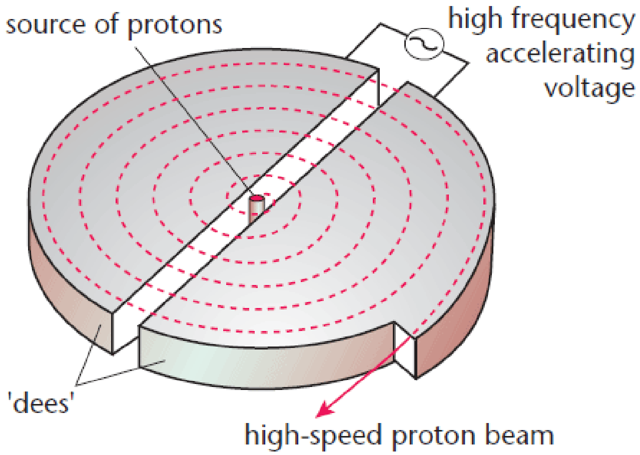


Filter

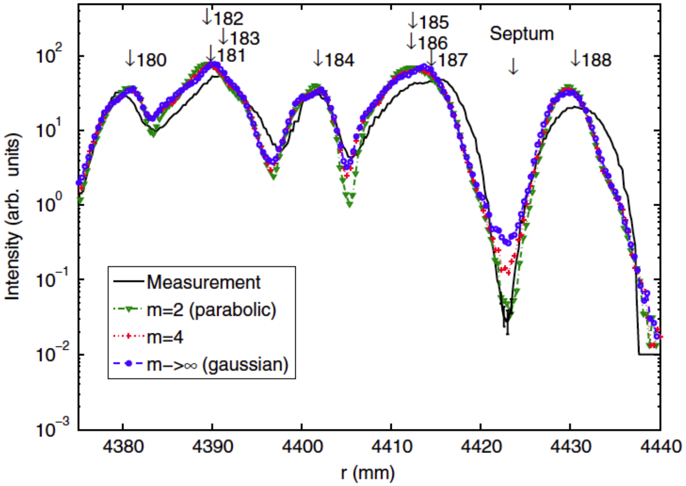


Pulse Structure of Cyclotron

- In cyclotron the proton source must be synchronized with high frequency
  - Due to finite length of bunch some particles are accelerated more or less than reference particles
  - Energy deviation results in different bending radius and thus a spread in the beam spot
  - Spiral spacing must be larger than the broadening of the spot
  - Transverse/energy collimation at exit needed
  - Shorter pulse duration of source reduces this effect but also reduces the average extraction current



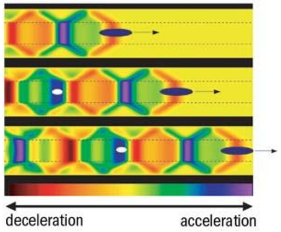
OPAL Simulation of PSI cyclotron



Collective Effect

Electrons and Protons are charged particles, which can interact with each other or the environment. In some case it can significantly alter the beam propagation

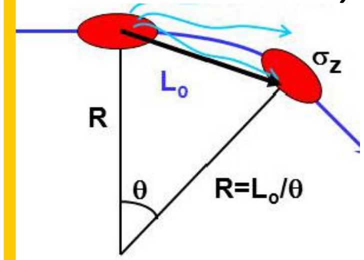
Wakefields



Distorted field due to boundary conditions:

- Finite conductance
- Changing apertures
- Coating (dielectric layer)

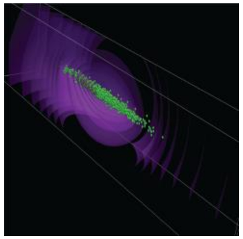
Coherent Synchrotron Radiation



Radiation in bends:

- Radiation can catch up
- At longer wavelength emission is coherent
- Kicks and energy losses

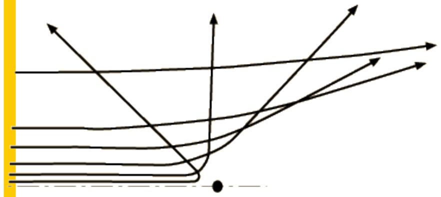
Space Charge



Very basic EM-Field

- Smooth beam
- Relativistic retardation must be included

Coulomb Scattering



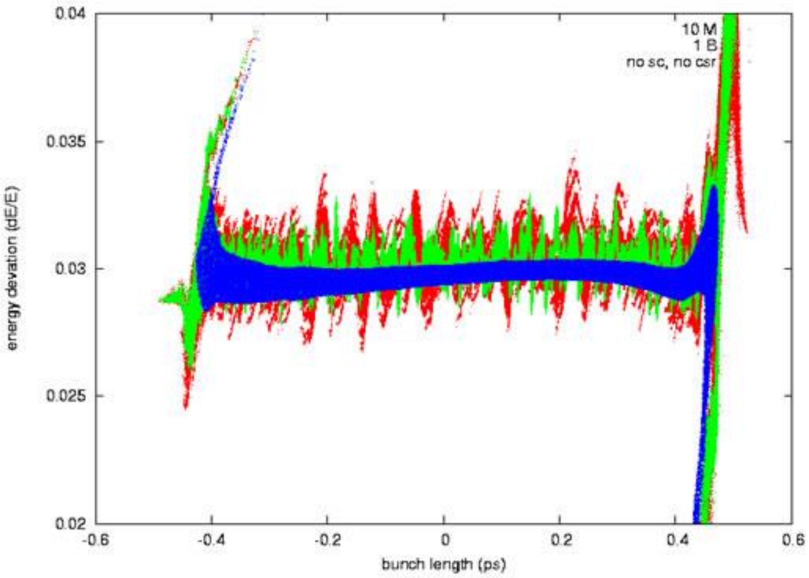
Close encounters:

- Strong, non-linear scattering
- Halo formation
- Diffusion process

Scaling Problem of Self-Interaction Modeling

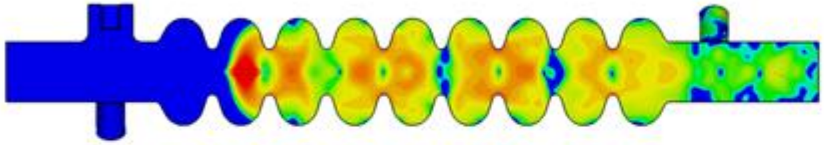
- Point-to-point interaction scales as N^2
- Stray Fields can have various forms and need typically a high resolution grid
- Requires expert code to handle the disadvantageous scaling and the overly high point charge of macro particle
- Normally (*non-intuitive*) smoothing, shielding, and filtering are required

Convergence study with IMPACT



[Courtesy by LBL]

Wakefield for LCLS II cavity



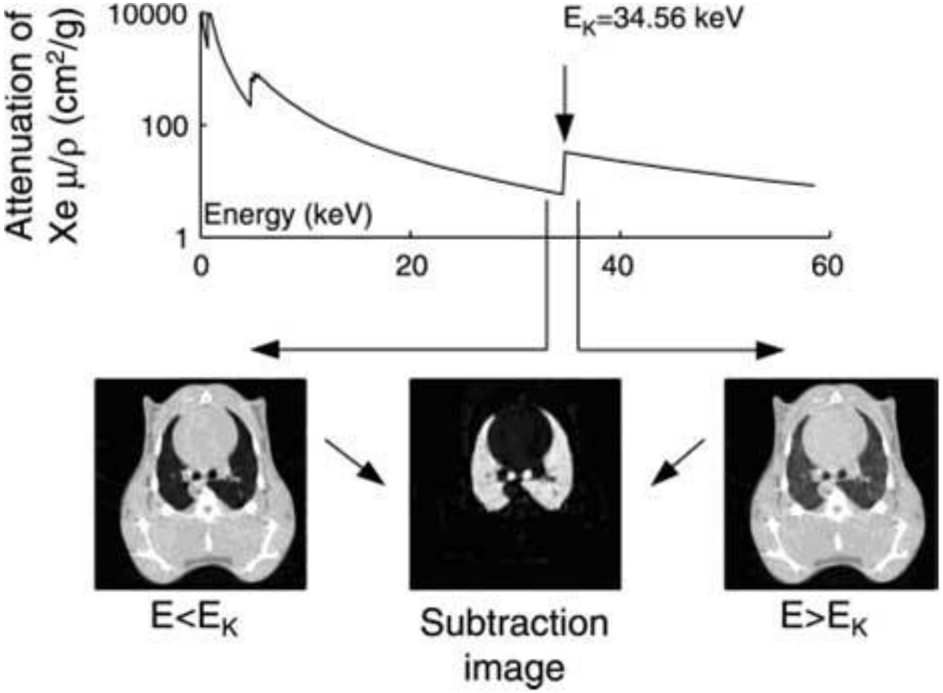
- Calculation with Dedicated Codes such as CST
- Needs ultra-fine grid to obtain single particle short range wake to use in tracking programs

[Courtesy by FNAL]

Radiation Production

- Medical application based on *tunable* X-ray radiation sources for K-edge subtraction imaging
- Radiation based on electrons in array of “mini”-bending magnets, called undulators or insertion devices, in 3rd and 4th generation light sources

Bronchography of Lung



3rd Gen. Light Source: SLS, Switzerland

4th Gen. Light Source: SACLA, Japan

Analogy between Electron Beam and Radiation

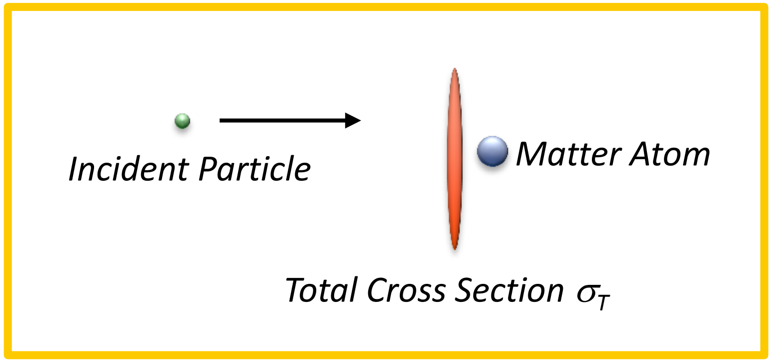
- Although developed independently, emittance dominated beams and radiation field propagate similar.

Parameter	Electron Beam	Radiation Field
Phase Space Area (size x divergence)	Emittance	Wavelength (for fundamental Gauss Mode)
Propagation	Twiss function: α, β	Rayleigh Length and Waist Position
Focusing	Quadrupoles	Lenses
Dispersion	Dipoles	Prism, Gratings

Interaction with Matter

Basic Behavior to Model:

- Particles are scattered
- Particles are loosing energy
- New Particles are generated
- Process is stochastic



Basic Process:

1. Characteristic Penetration Depth *(Full reaction cross section is equal to transverse size)*

$$\boxed{n_e \cdot \Delta z \cdot A \cdot \sigma_T = A}$$

Matter Density

Char. Depth

Trans. Size

2. Branching Chance into Different Reaction Channel

$$\boxed{1 = \frac{\sigma_1}{\sigma_T} + \frac{\sigma_2}{\sigma_T} + \frac{\sigma_3}{\sigma_T} + \cdots}$$

3. Scattering and Energy loss

$$\boxed{\sigma_1 = \iint \frac{\partial^2 \sigma_1}{\partial \Omega \partial \omega} d\Omega d\omega}$$

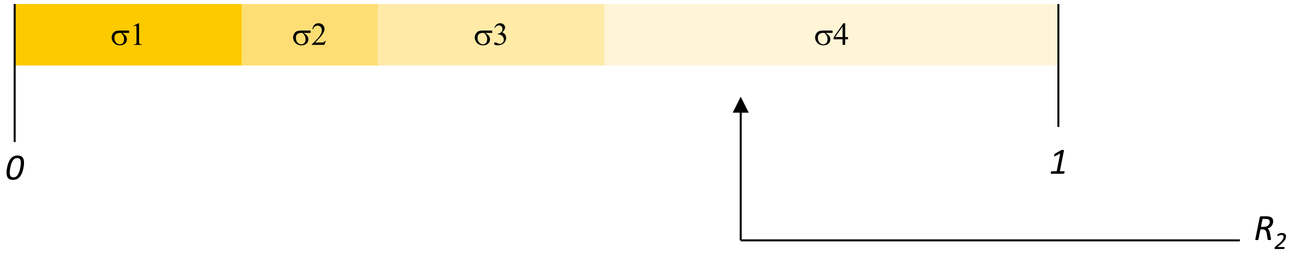
Probability function to scatter into solid angle and energy (Broglie-wave Package)

Monte-Carlo Simulation of Interaction

Stochastic nature makes Monte-Carlo algorithm most suitable:

1. Calculate free path length, typically ramping up linearly
   - R_1 is random number value from uniform distribution → $s = R_1 \Delta z$
   - More precise models allow for channeling or half-Gaussian random number

2. Select reaction branch
   - Random uniform number R_2 mapped into the various branching chances



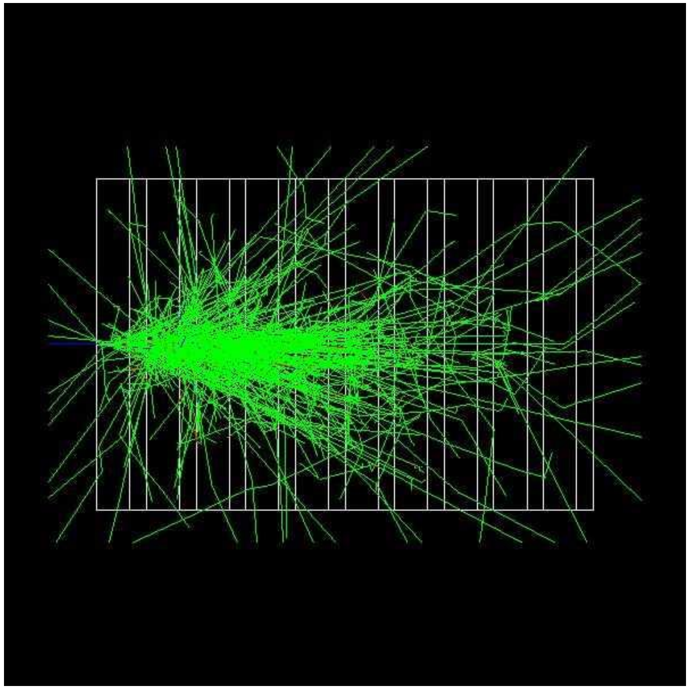
3. Scatter and Energy Loss
   - Invert differential cross check to map into uniform distribution and to derive angle and energy loss.

Material Properties

- The challenge is the calculation of the cross sections:
  - Dependence on Particle Energy
  - Available Reaction Channels
  - Possible secondary particles
- Needs huge library of material properties
- Possible Simplifications:
  1. Thin targets (one one event per particles)
  2. Simplified energy loss, using Bethe-Bloch
  3. Only Coulomb Scattering & Bremstrahlung

System is described just by material density and atomic number

GEANT4 Result for CERN Energy Calormeter



Four Phases of Numerical Simulation

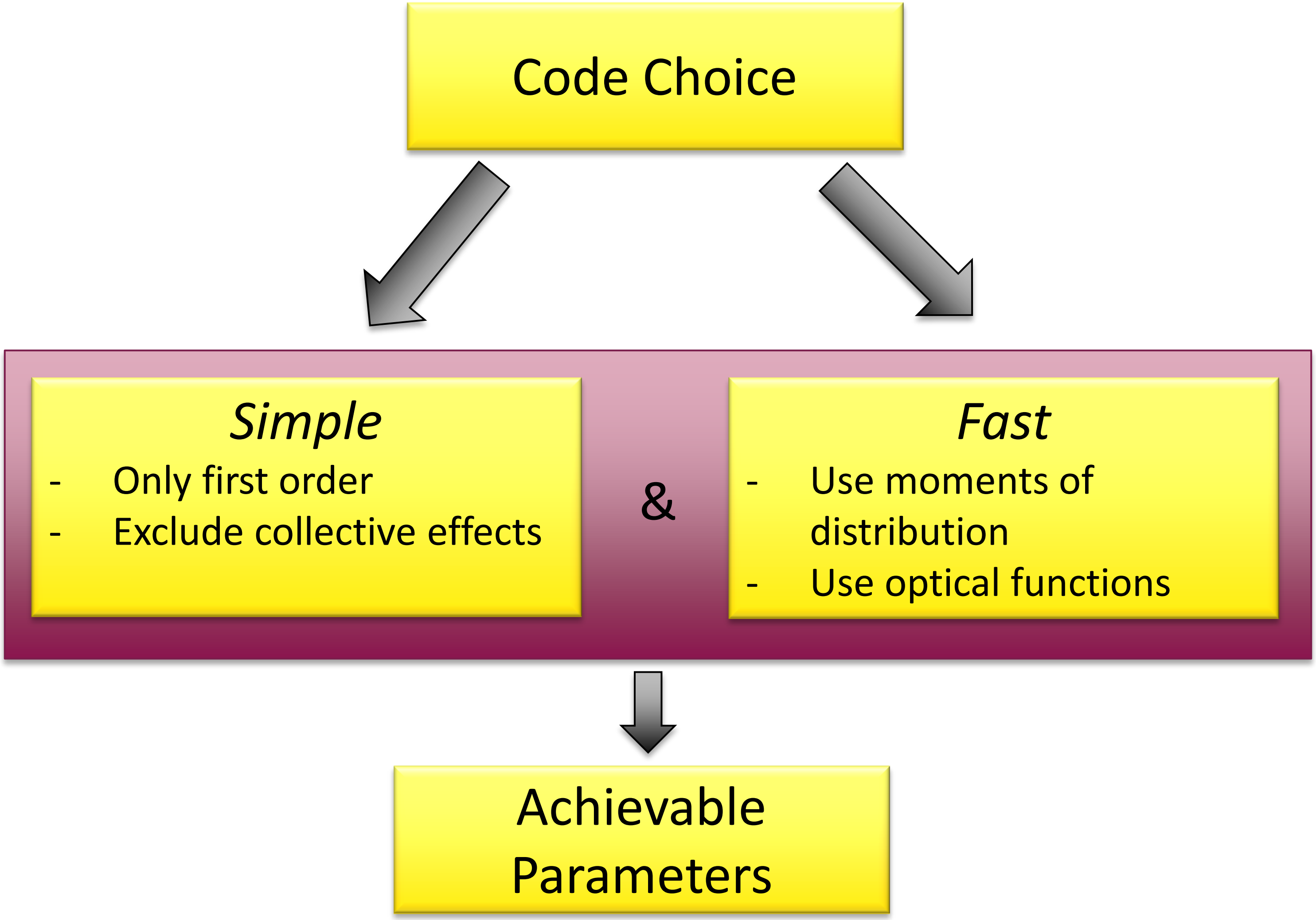
1st - Design

2nd - Optimization

3rd – Test/Benchmarking

4th – Production

Design Phase



Code Choice

Simple

- Only first order
- Exclude collective effects

&

Fast

- Use moments of distribution
- Use optical functions

Achievable Parameters

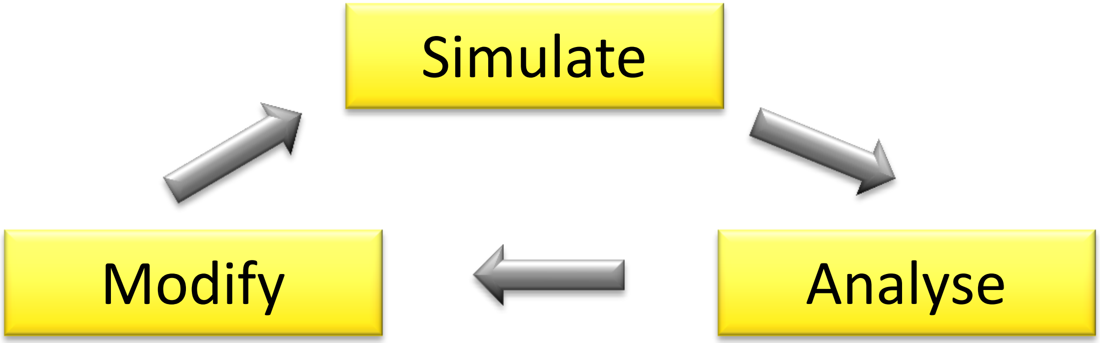
Example Codes: MadX (optics), LieTrack (long. Dynamics)

Optimization Phase

After the initial design a rather tedious task is the optimization.

Typical points to optimize are:

- Maximum reliability and safety
- Minimize sensitivity and jitters
- Stay within constraints (Budget, Foot Print, Energy Consumption)



Simulate

Modify

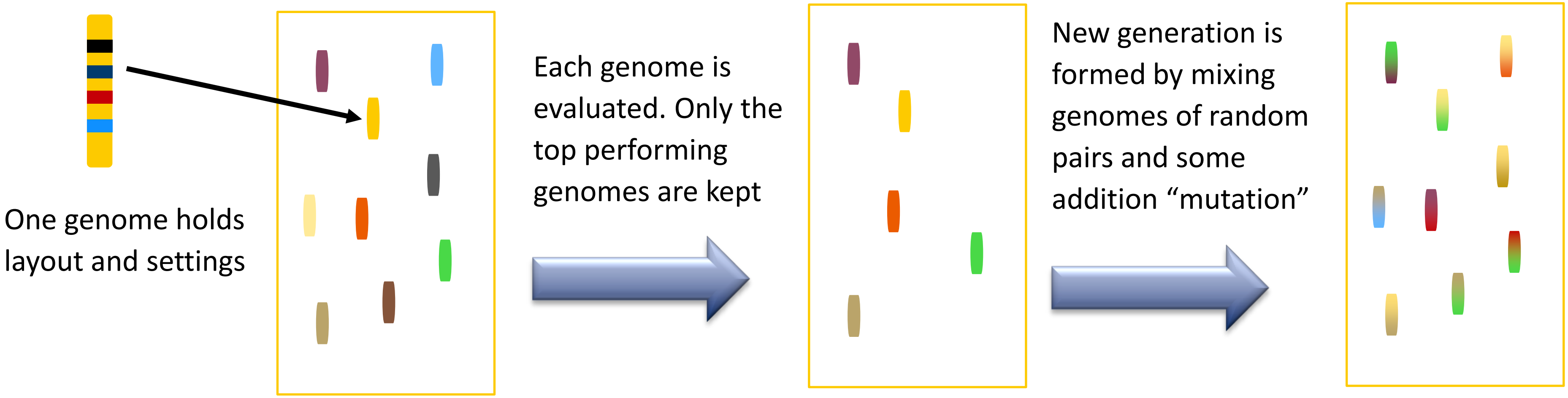
Analyse

The detail level in the simulation is much higher to not exclude effects(e.g. collective effects), which can degrade performance, rendering the found design useless.

Typically tracking programs with particle distribution are used

Genetic Optimizer

- If layout, simulation and analysis can be done in the same framework, then optimizer routines can be used for automatic optimization.
- Genetic optimizer are considered the most suitable for such complex optimizations



One genome holds layout and settings

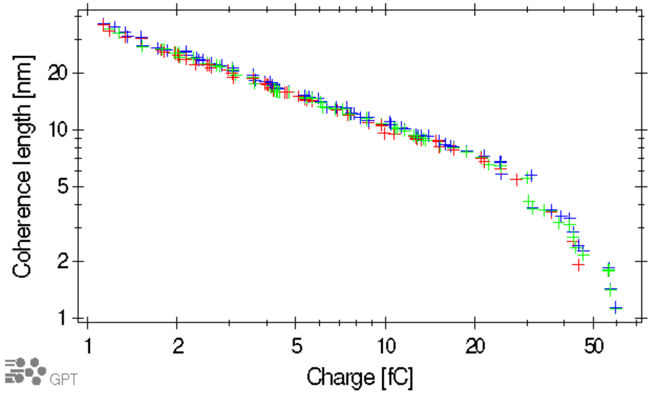
Generation is a set of various genomes with different value

Each genome is evaluated. Only the top performing genomes are kept

New generation is formed by mixing genomes of random pairs and some addition "mutation"

Repeat till genomes converge to hyperplane of multi-objectives (Pareto-front)

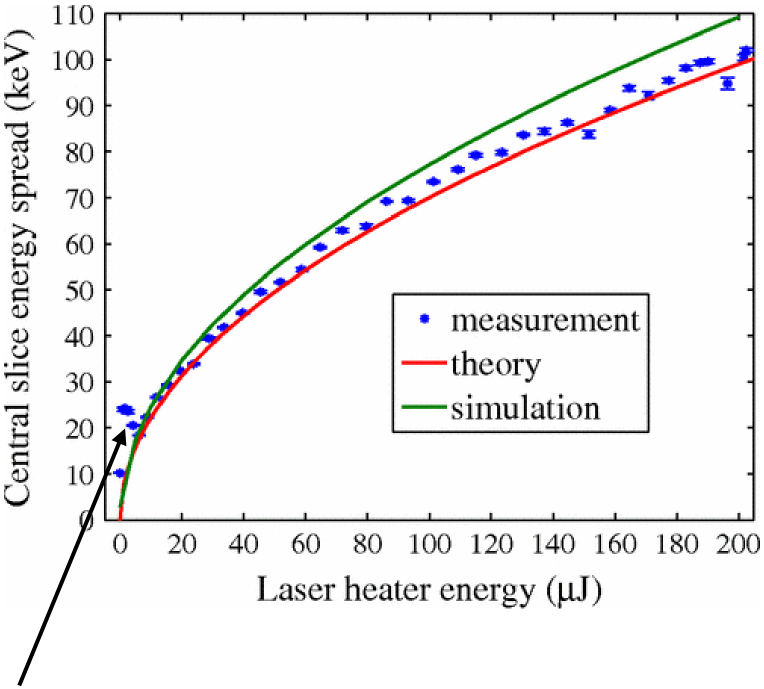
GTP-Code example for electron microscope



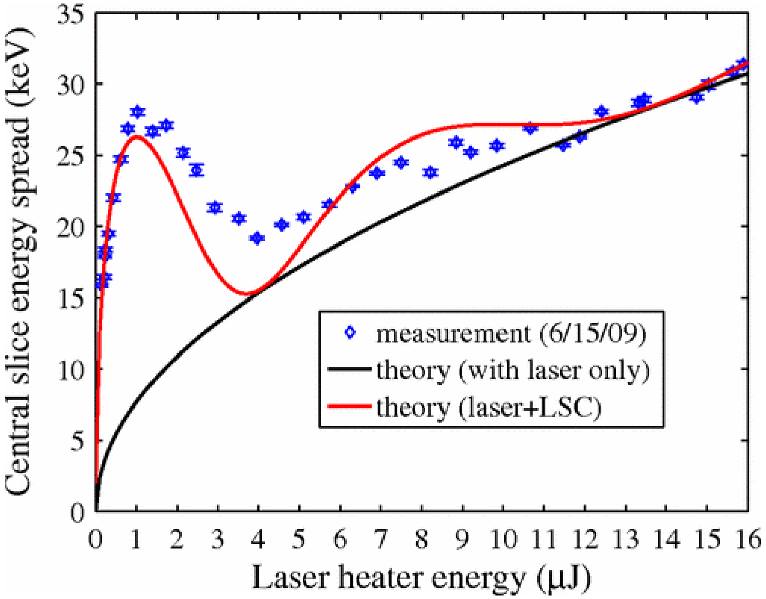
Benchmark Phase

- Often Prediction and Measurement deviates significant

Observation of enhanced energy spread at LCLS



Unexpected behaviour

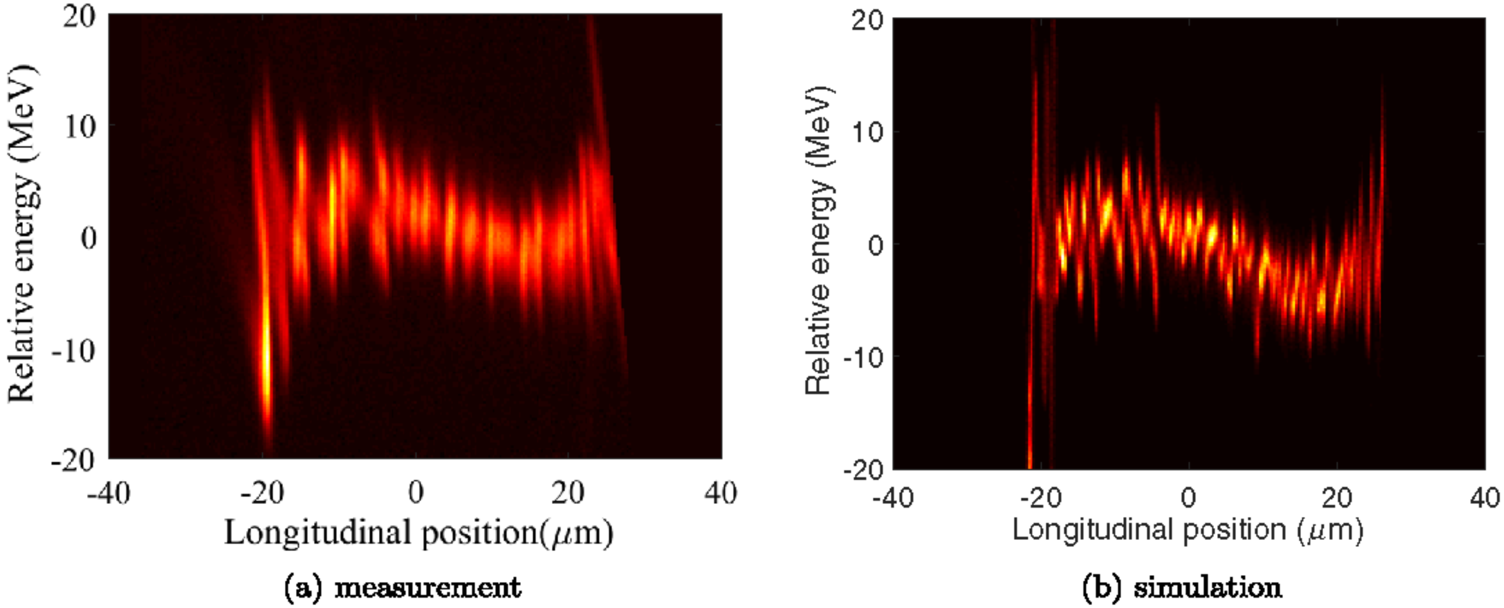


After some extensive 3D Model for space charge has been included

Benchmark (cont’)

- Often requires an ultra-high detail of modelling to get correct results

Final longitudinal phase space at LCLS, affected by micro-bunch instability



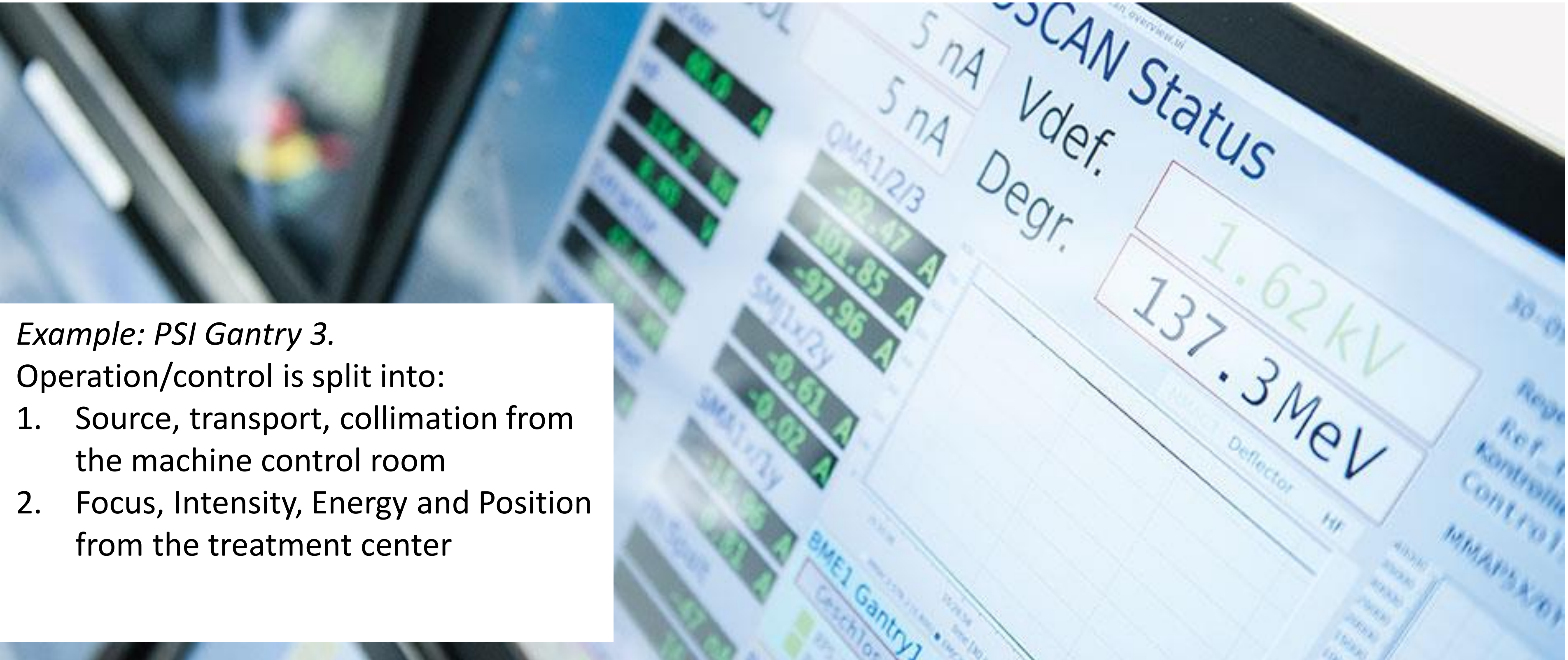
(a) measurement **(b) simulation**

To obtain results it required to run on the NERSC super computer with all electrons resolved and full 3D model of collective effects.

Prediction/Production Phase

Certificate of reliable operation is essential. This includes in particular the capability to change configuration based on the underlying model/codes.

This includes also some limitation in the user interface to avoid wrong settings of the machine, which could harm the patient.



Example: PSI Gantry 3.
Operation/control is split into:

1. Source, transport, collimation from the machine control room
2. Focus, Intensity, Energy and Position from the treatment center

Swiss Army Knife Code vs Expert Codes

All-purpose Code

- Consistent description of beam line
- Internal format of particle representation and radiation field
- Easier to learn using it
- Specific problems are solved with a simpler/reduced model (e.g. 1D Green’s function of CSR)
- Algorithm not fully optimized
  - Unneeded transport of full ensemble over through simple elements (e.g. drifts)
  - Inefficient Solver (e.g. particle-particle space charge)

Specialized Codes

- Algorithm are fully optimized, dedicated to specific problems
- Inconsistent description of the same beam line
- Conversion between particle formats needed
- Requires up- and down-sampling of particle distribution
- Semi-automated execution.

Code Interface – A Developer's Approach

Codes are coming in various forms:

Matlab Script

Windows Apps

Unix Command Line

Source Code

High Performance Code

Mostly written by physicists:

- Well tested due to own use
- Often inefficient code
- Heterogeneous input/output
- No interface with other codes

"Secondary" users have difficult times to use one or more codes, in particular the problem of magic numbers (unphysical constrants to adjust results)

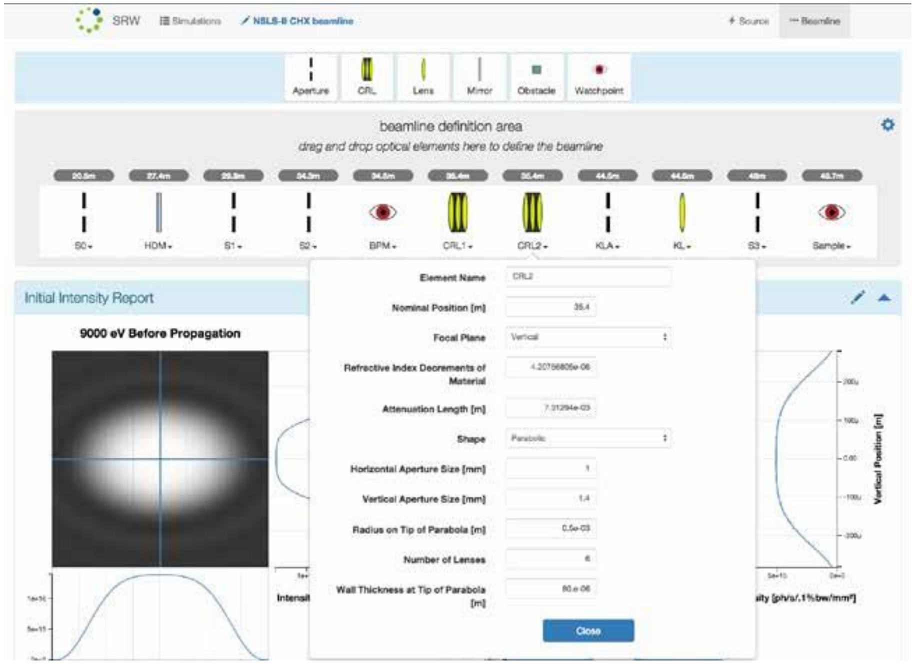
Code Interface – A User's Need

Developer	User
Can maximize efficiency of simulation	Need check for consistency of model and output.
Can program to expand code or to interface between codes	Needs a common interface or framework
Prefers Unix-based systems (also batch)	Windows-based system with GUI preferred
Can analyze raw data	Relies on post-processors

Very hard to obtain resources (funding) to transfer codes from expert level to user level

Only recently some business models are supported worldwide to do this work

Radiasoft open source framework →



Element Name: CRL2
Nominal Position [m]: 35.4
Focal Plane: Vertical
Refractive Index Decrements of Material: 4.20756805e-06
Attenuation Length [m]: 7.31294e-03
Shape: Parabolic
Horizontal Aperture Size [mm]: 1
Vertical Aperture Size [mm]: 1.4
Radius on Tip of Parabola [m]: 0.5e-03
Number of Lenses: 6
Wall Thickness at Tip of Parabola [m]: 80.e-06

Popular Codes - MadX

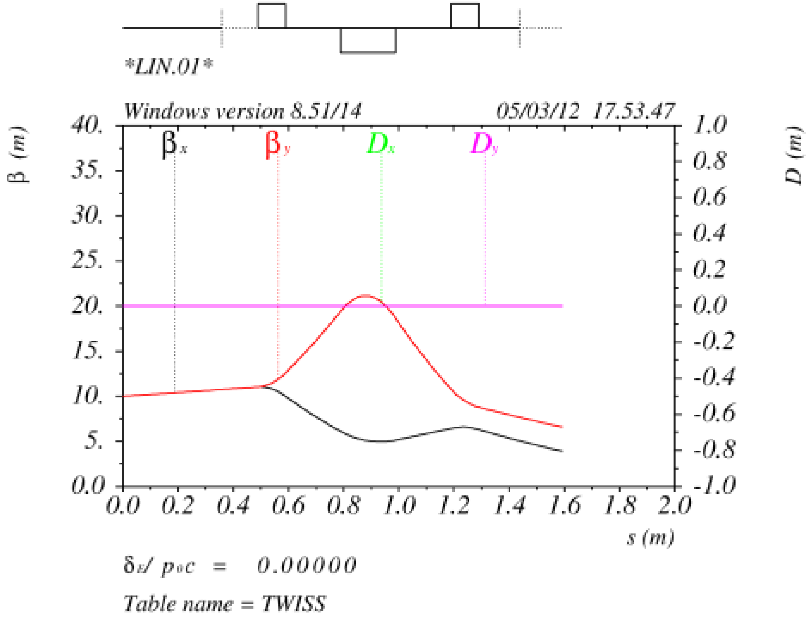
http://mad.web.cern.ch/mad/

The reference code for optics calculations.

Developed at CERN and has defined almost the standard on how to define a beamline.

Very steep learning curve and many hidden features, but is de-factor the reference for optics calculation.

It’s free



Popular Code - Elegant

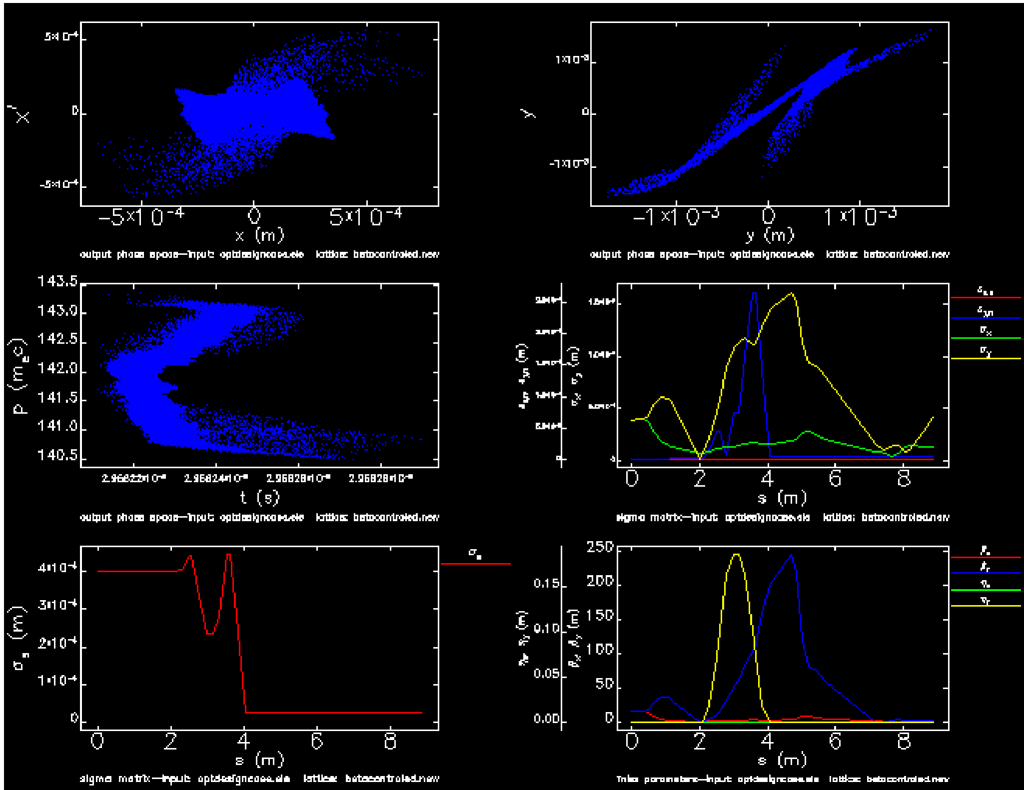
http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/elegant.html

Multi-Purpose particle tracker for linacs and storage rings.

Has a huge variety of beam line elements and support for many collective effects

Following MADX syntax for lattice (easy transition). Runs on many platforms and is extremely well supported. Output fileformat (SDDS) is a bit tedious.

It's free



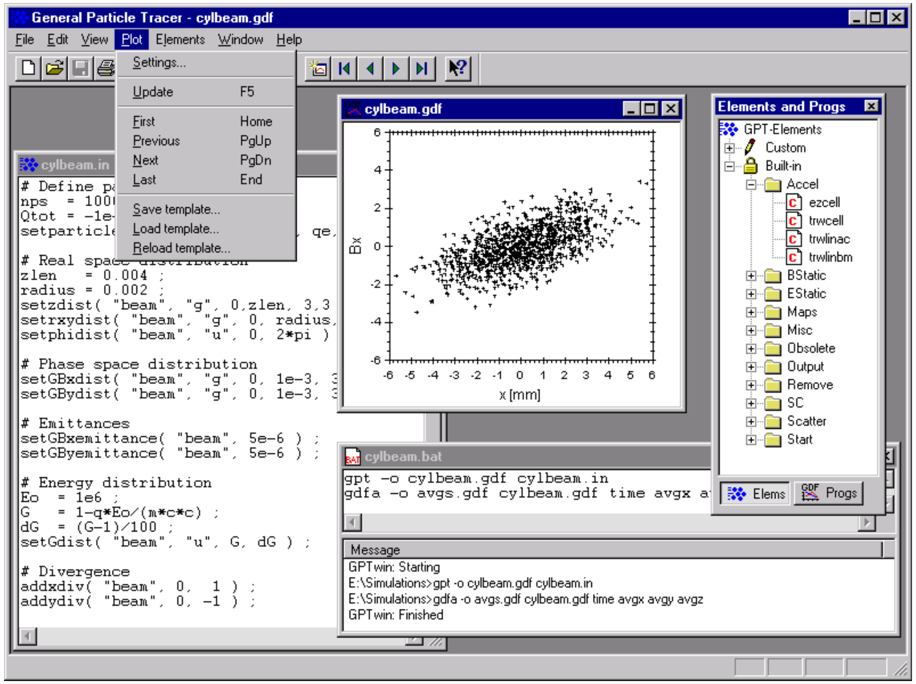
Popular Code - GTP

http://www.pulsar.nl/gpt/

Single environment with advanced 3D space charge and radiation solver.
Aimed to simplify set-up and execution of simulations.
Interface with Geant4

All-purpose code within one development framework to access all different physics models.

It's commercial



```
# Define pa
nps   = 100
Qtot = -1e-
setparticle

# Real space
zlen    = 0.004 ;
radius = 0.002 ;
setzdist( "beam", "g", 0,zlen, 3,3
setrxydist( "beam", "g", 0, radius,
setphidist( "beam", "u", 0, 2*pi )

# Phase space distribution
setGBxdist( "beam", "g", 0, 1e-3, 3
setGBydist( "beam", "g", 0, 1e-3, 3

# Emittances
setGBxemittance( "beam", 5e-6 ) ;
setGByemittance( "beam", 5e-6 ) ;

# Energy distribution
Eo   = 1e6 ;
G    = 1-q*Eo/(m*c*c) ;
dG   = (G-1)/100 ;
setGdist( "beam", "u", G, dG ) ;

# Divergence
addxdiv( "beam", 0,  1 ) ;
addydiv( "beam", 0, -1 ) ;
```

```
gpt -o cylbeam.gdf cylbeam.in
gdfa -o avgs.gdf cylbeam.gdf time avgx a
```

```
GPTwin: Starting
E:\Simulations>gpt -o cylbeam.gdf cylbeam.in
E:\Simulations>gdfa -o avgs.gdf cylbeam.gdf time avgx avgy avgz
GPTwin: Finished
```

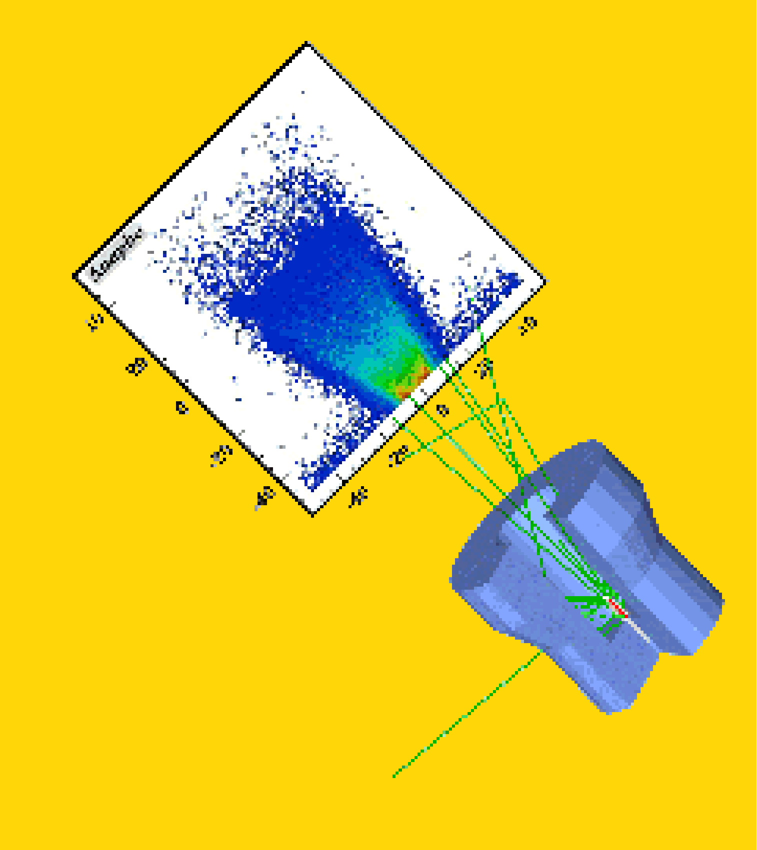
Popular Code – Geant4

http://geant4.cern.ch/

The reference code for interaction of relativistic particles with matter.
Has many extension for specific application (e.g. medical applications)

Very powerful but not much user-friendly. Models are coded into the source code.

It's free



Numerical codes are essential to design, optimize and operate medical accelerator

Various codes exists, which are dedicated to specific problems.

Using them can be challenging/tedious, mostly due to incompatible interfaces

A lot of codes, which are considered the reference, are for free.

Integrated solution/framework exist or can be developed/provided by companies, in particular those which are selling the accelerators (complete package).