PH1
Computational Ion Optics with IBSimu

Taneli Kalvas
taneli.kalvas@jyu.fi
FL114

University of Jyväskylä, Department of Physics

https://ibsimu.sourceforge.net/jss2015

Participants

Lecturer:

Taneli Kalvas, `taneli.kalvas@jyu.fi`

Course tutors:

Risto Kronholm, `risto.j.kronholm@student.jyu.fi`

Janne Laulainen, `janne.p.laulainen@student.jyu.fi`

Who else?

The course

- First week: Introductory part (1 ECTS)
  - Lectures 3×2 hours
  - Demonstrations 3×2 hours

- Second week: Main part (3 ECTS)
  - Lectures 2×2 hours
  - Demonstrations 5×2 hours
  - Homework!

Schedule

Schedule

Introductory part

Wed 5.8. 10–12, 14–16

Thu 6.8. 10–12, 14–16

Fri 7.8. 10–12, 14–16

Main part

Mon 10.8. 10–12, 14–16

Tue 11.8. 10–12, 14–16

Wed 12.8. 12–14

Thu 13.8. 12–14

Fri 14.8. 12–14

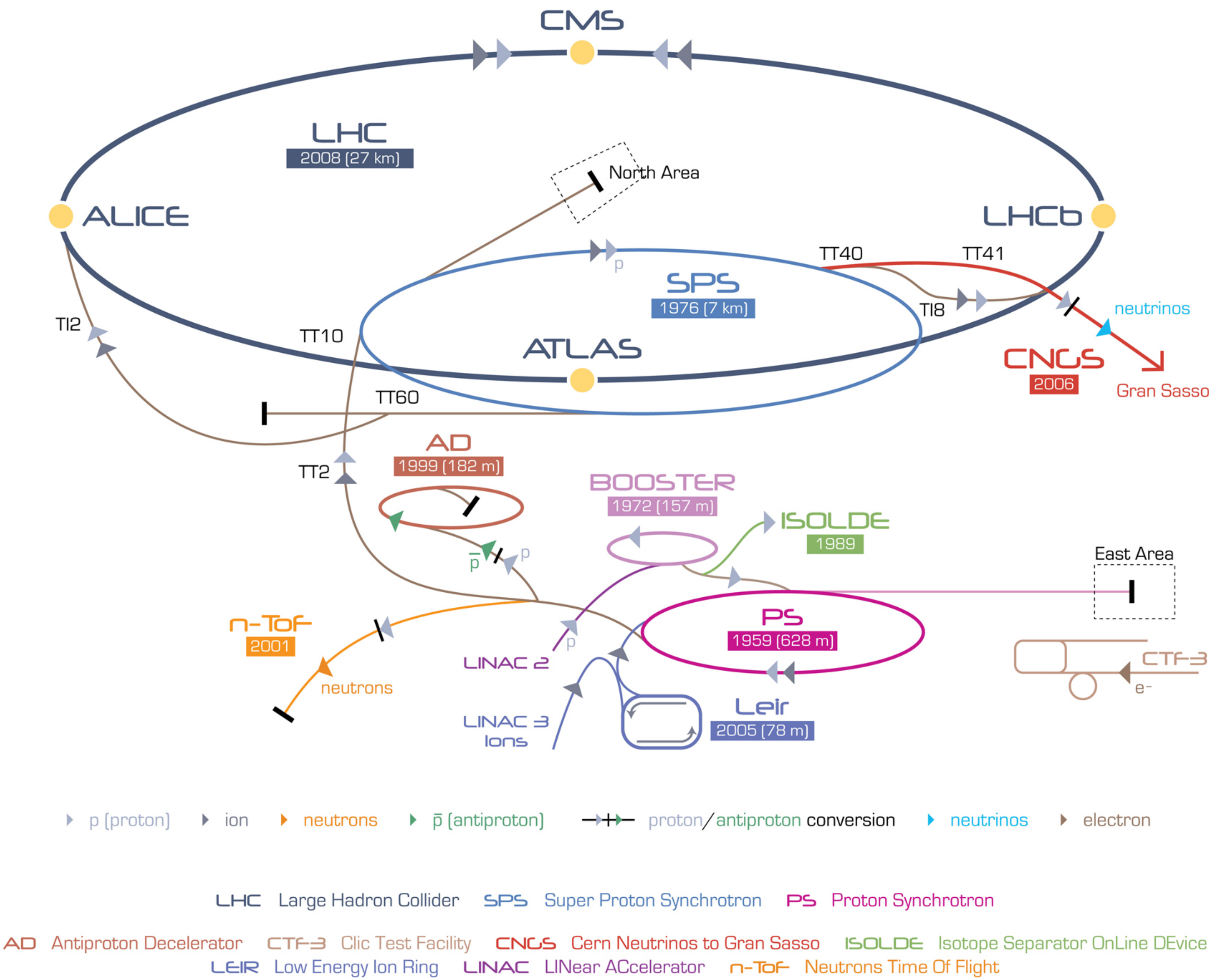
Contents

Introductory part

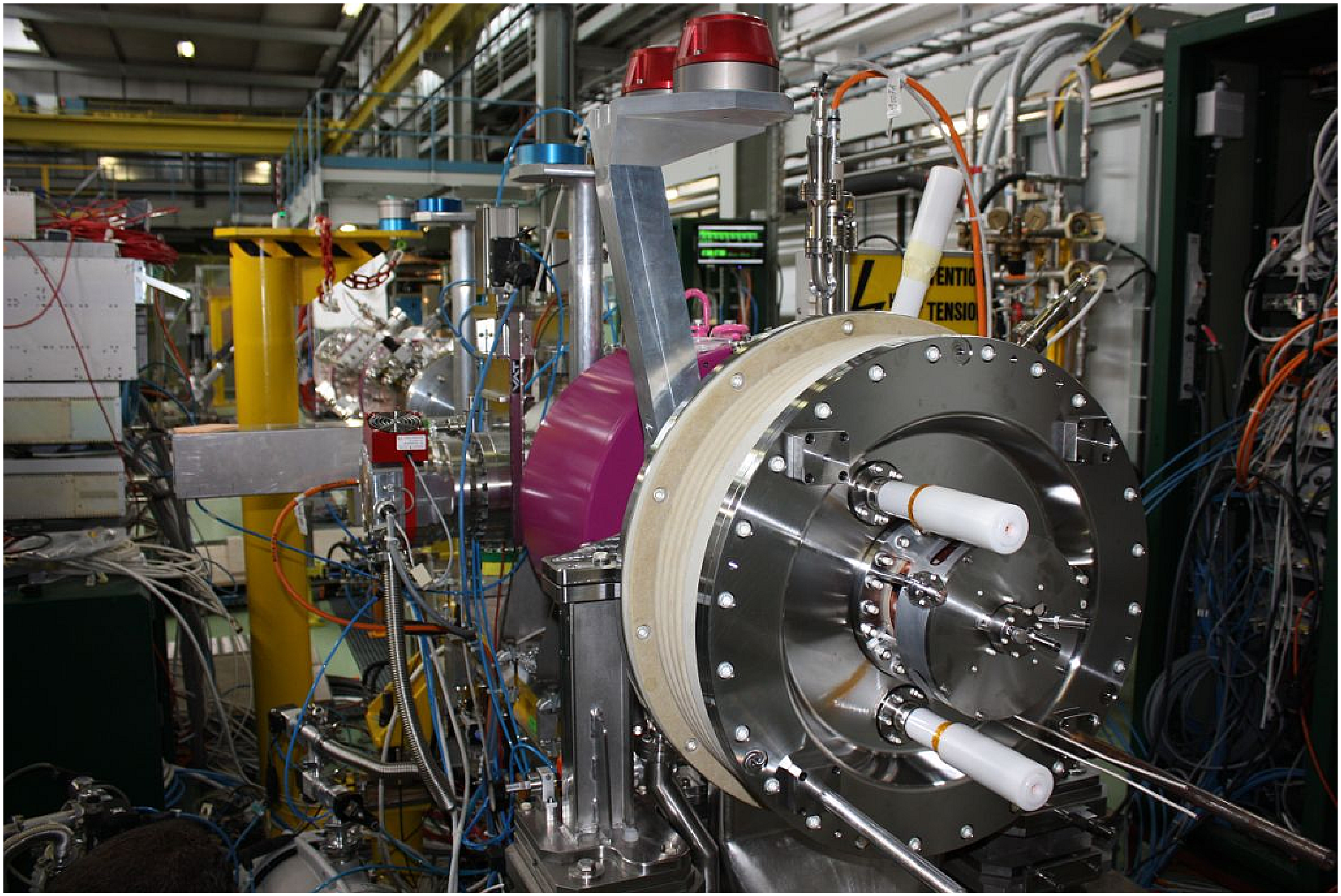
- Background: What and why?
- Emittance
- Computer programs for ion optics
- What does IBSimu do?
- Getting started with IBSimu
- Installing and using IBSimu on your own computers
- Reading the documentatation, code structure, contributing, git, versioning

Background

What and why?


CMS
LHC
2008 (27 km)
North Area
ALICE
LHCb
TT40
TT41
SPS
1976 (7 km)
TI8
neutrinos
TI2
TT10
ATLAS
CNGS
2006
Gran Sasso
TT60
AD
1999 (182 m)
TT2
BOOSTER
1972 (157 m)
ISOLDE
1989
East Area
PS
1959 (628 m)
n-ToF
2001
LINAC 2
CTF3
e–
neutrons
LINAC 3
Ions
Leir
2005 (78 m)
p (proton)
ion
neutrons
p̄ (antiproton)
proton/antiproton conversion
neutrinos
electron
LHC Large Hadron Collider
SPS Super Proton Synchrotron
PS Proton Synchrotron
AD Antiproton Decelerator
CTF3 Clic Test Facility
CNGS Cern Neutrinos to Gran Sasso
ISOLDE Isotope Separator OnLine DEvice
LEIR Low Energy Ion Ring
LINAC LINear ACcelerator
n-ToF Neutrons Time Of Flight

What and why?

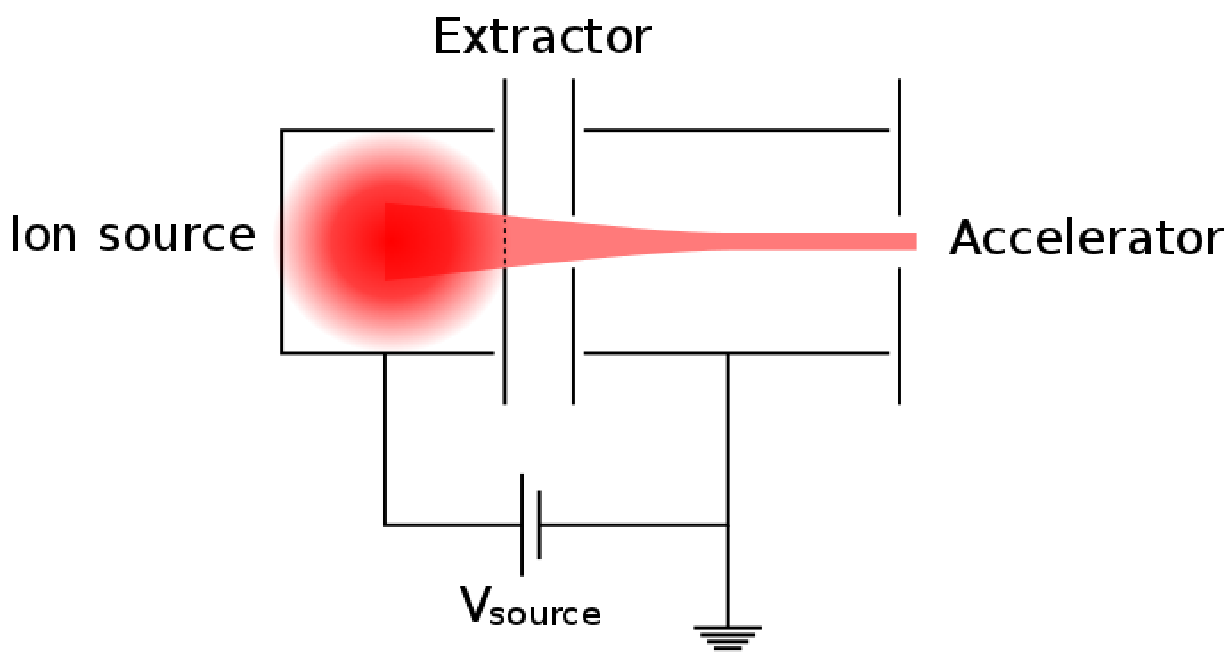


Linac4 ion source and LEBT, Copyright CERN

Taneli Kalvas

Basic beam extraction and transport

The extractor takes the plasma flux $J = \frac{1}{4}qn\bar{v}$ and forms a beam with energy $E = q(V_{\mathrm{source}} - V_{\mathrm{gnd}})$ transporting it to the following application.



Simple?

Extraction complications

- Plasma-beam transition physics
  - Plasma parameters: density, potential, temperature, etc
  - Beam intensity, quality, uniformity, species
- Application requirements for beam spatial and temporal structure
  - Need for focusing, chopping, etc
- Space charge
- Practical engineering constraints
  - Space for diagnostics, pumping, etc
  - Materials, power supplies, money

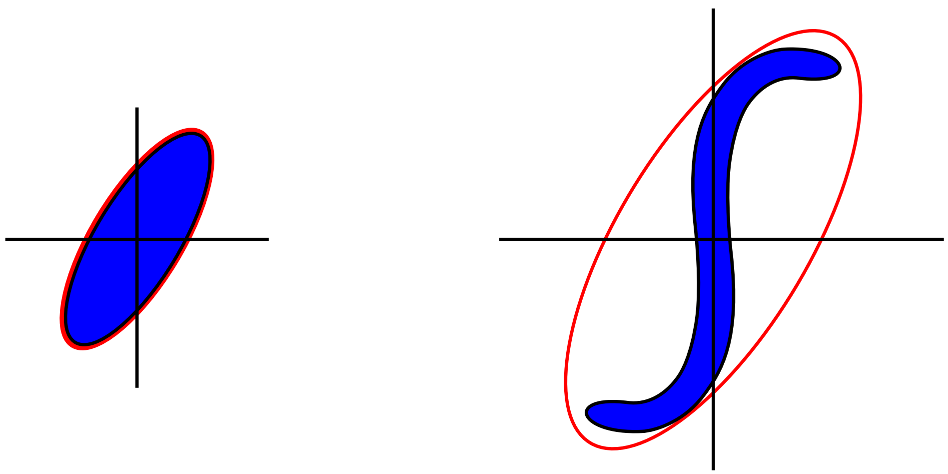
Emittance

Taneli Kalvas

Emittance

Traditionally the emittance is defined as the 6-dimensional volume limited by a countour of particle density in the (x, p_x, y, p_y, z, p_z) phase space. This volume obeys the Liouville theorem and is constant in conservative fields.

With practical accelerators a more important beam quality measure is the volume of the elliptical envelope of the beam bunch. This is not conserved generally — only in the case where forces are linear.



Taneli Kalvas

Transverse emittance

The transverse emittances are 4 and 2-dimensional reductions of the 6-dimensional definition, usually assuming that p_z is constant and replacing p_x with $x' = p_x/p_z$ and p_y with $y' = p_y/p_z$. The 2D emittance ellipse then becomes

$$\gamma x^2 + 2\alpha x x' + \beta x'^2 = \epsilon_x,$$

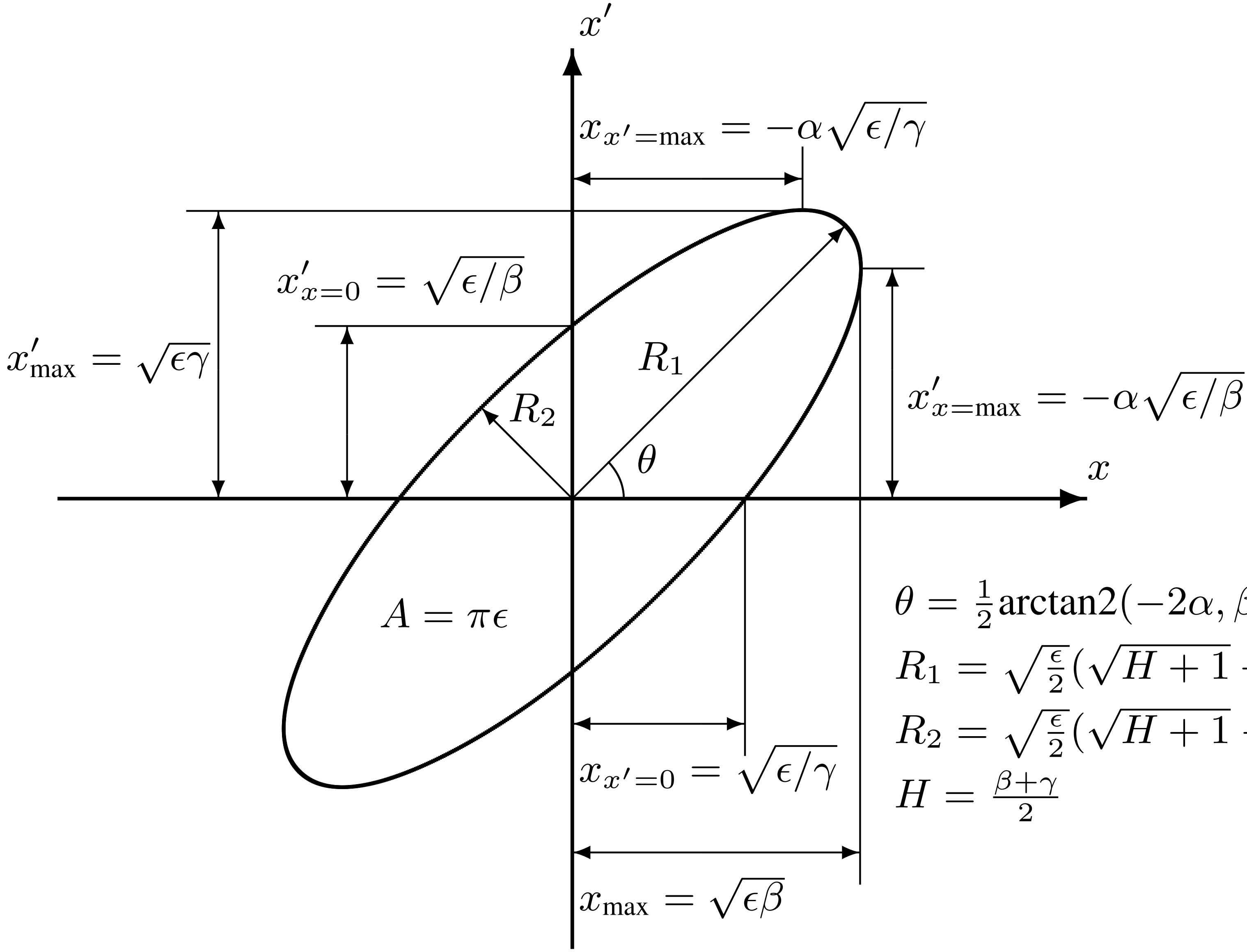
where scaling

$$\beta\gamma - \alpha^2 = 1$$

is chosen. The ϵ_x is the product of the half-axes of the ellipse (A/π) and α, β and γ are known as the Twiss parameters defining the ellipse orientation and aspect ratio.

Because of the connection between the area of the ellipse and ϵ there is confusement on which is used in quoted numbers. Sometimes π is included in the unit of emittance (π mm mrad) to emphasize that the quoted value is not the area, but the product of half-axes as defined here.

Ellipse geometry



$$\theta = \tfrac{1}{2}\,\mathrm{arctan2}(-2\alpha, \beta - \gamma)$$
$$R_1 = \sqrt{\tfrac{\epsilon}{2}}(\sqrt{H+1} + \sqrt{H-1})$$
$$R_2 = \sqrt{\tfrac{\epsilon}{2}}(\sqrt{H+1} - \sqrt{H-1})$$
$$H = \tfrac{\beta+\gamma}{2}$$

 Taneli Kalvas

Emittance envelope

How to define the "envelope"?

Numerous algorithms exist for defining the ellipse from beam data. Often a minimum area ellipse containing some fraction of the beam is wanted (e.g. $\epsilon_{90\%}$). Unfortunately this is difficult to produce in a robust way.

A well-defined way for producing the ellipse is the rms emittance:

$$\epsilon_{\mathrm{rms}} = \sqrt{\langle x'^2 \rangle \langle x^2 \rangle - \langle xx' \rangle^2},$$

and similarly the Twiss parameters

$$\begin{aligned} \alpha &= -\frac{\langle xx' \rangle}{\epsilon}, \\ \beta &= \frac{\langle x^2 \rangle}{\epsilon}, \\ \gamma &= \frac{\langle x'^2 \rangle}{\epsilon}, \end{aligned}$$

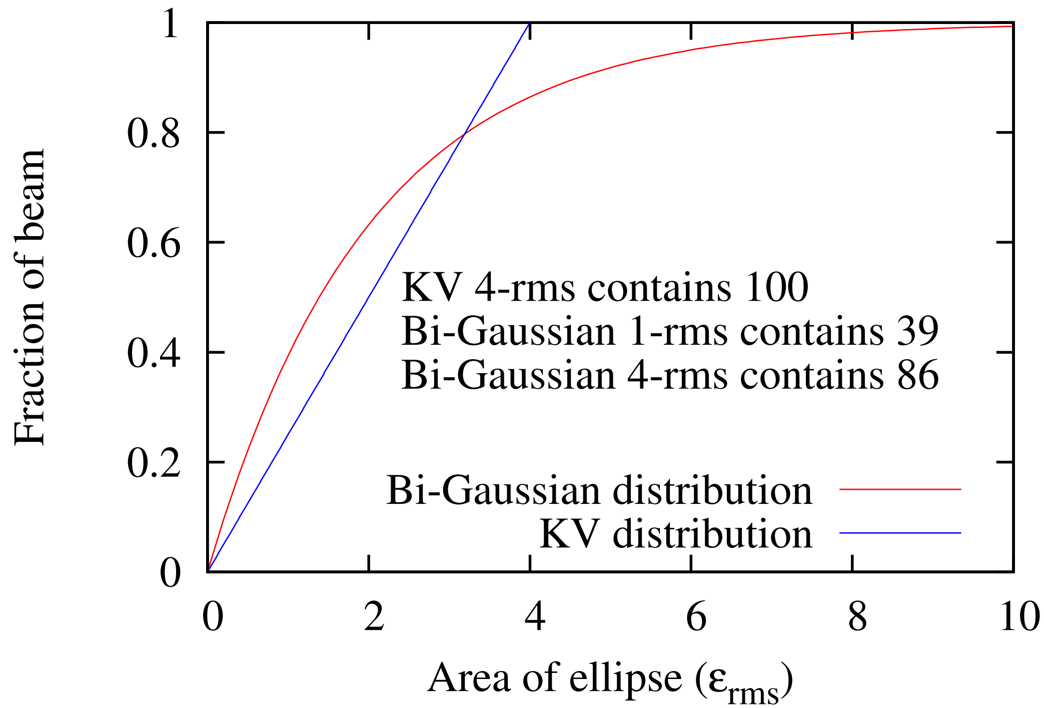
where

$$\begin{aligned} \langle x^2 \rangle &= \frac{\iint x^2 I(x,x')\mathrm{d}x\mathrm{d}x'}{\iint I(x,x')\mathrm{d}x\mathrm{d}x'}, \\ \langle x'^2 \rangle &= \frac{\iint x'^2 I(x,x')\mathrm{d}x\mathrm{d}x'}{\iint I(x,x')\mathrm{d}x\mathrm{d}x'}, \\ \langle xx' \rangle &= \frac{\iint xx' I(x,x')\mathrm{d}x\mathrm{d}x'}{\iint I(x,x')\mathrm{d}x\mathrm{d}x'}. \end{aligned}$$

Assuming $\langle x \rangle = 0$ and $\langle x' \rangle = 0$.

Meanining of rms emittance

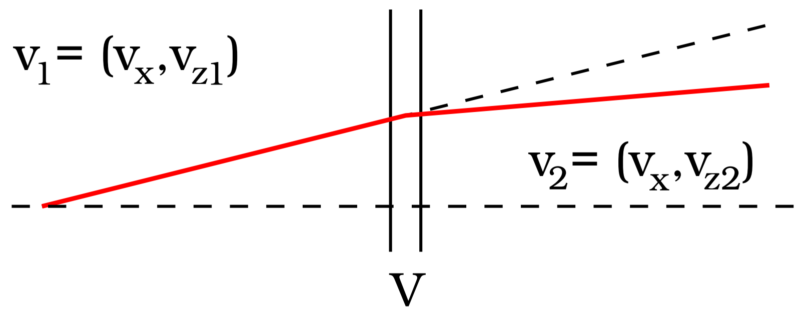
How much beam does the rms ellipse contain?



Depends on the distribution shape. For real simulated or measured distributions there is no direct rule.

Normalization of emittance

The transverse emittance defined in this way is dependent on the beam energy. If p_z increases, $x' = p_x/p_z$ decreases.



The effect is eliminated by normalizing the velocity to c, which gives

$$x'_n = \frac{p_x}{p_{z1}}\frac{v_{z1}}{c} = \frac{v_x}{c} = \frac{p_x}{p_{z2}}\frac{v_{z2}}{c}.$$

Normalized emittance is

$$\epsilon_n = \epsilon\frac{v_z}{c}$$

Emittance from plasma temperature

Assume circular extraction hole and Gaussian transverse ion distribution

$$I(x,x') = \frac{2}{\pi r^2}\sqrt{r^2 - x^2}\ \sqrt{\frac{m}{2\pi kT}}\exp\left(\frac{-m(x'v_z)^2}{2kT}\right).$$

The rms emittance can be integrated using the definition and normalized

$$\epsilon_{\text{rms,n}} = \frac{1}{2}\sqrt{\frac{kT}{m}}\frac{r}{c}.$$

Similarly for a slit-beam extraction

$$\epsilon_{\text{rms,n}} = \frac{1}{2}\sqrt{\frac{kT}{3m}}\frac{w}{c}.$$

Larger aperture $\Rightarrow$ more beam, weaker quality

Emittance from solenoidal B-field

If a circular beam starts from a solenoidal magnetic field (ECR) particles receive a azimuthal thrust of

$$v_\theta = r_0 \frac{qB}{2m},$$

when exiting the magnetic field. Far from solenoid the motion is cylindrically symmetric and

$$r' = \frac{v_r}{v_z} = \frac{v_\theta}{v_z} = \frac{qBr_0}{2mv_z}$$

The emittance of the beam is

$$\epsilon_{\text{rms}} = \frac{1}{4} r_0 r' = \frac{qBr_0^2}{8mv_z}$$

and normalized

$$\epsilon_{\text{rms,n}} = \frac{qBr_0^2}{8mc}$$

Low Energy Beam Transport

Beam line elements

Beam control happens with electromagnetic forces a.k.a. ion-optics.

Electrostatic elements are mostly used at the start of the beam line, but also the classic magnetic elements are used at low energies.

Electrostatic

- Diode (accel or decel gap)
- Einzel lens
- Dipole
- Quadrupole

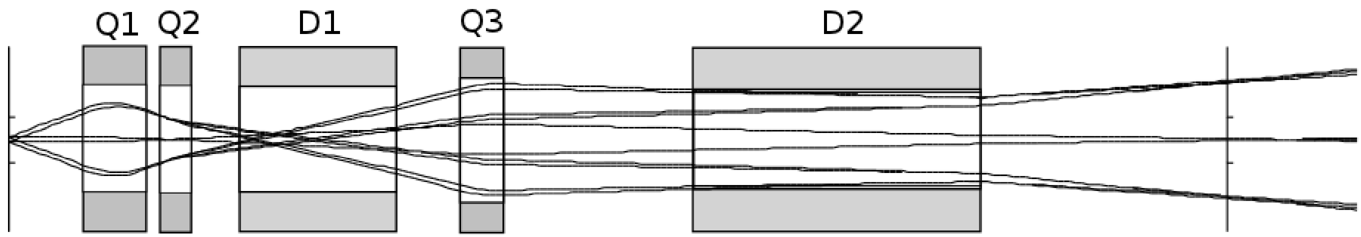
Magnetic

- Solenoid
- Dipole
- Quadrupole
- Other multipoles

Tools of trade

- Ion-optical software based on N^{th}-order approximation of trajectories (commonly used at higher energies)
- Electromagnetic field programs: POISSON SUPERFISH, FEMM, RADIA-3D, VECTOR FIELDS (OPERA), COMSOL MULTIPHYSICS, LORENTZ, etc. Some with and some without particle tracking capability.
- Specialized ion source extraction software.
- Many other specialized programs for modelling beam space charge compensation, bunching, cyclotron injection, collisional ion source plasmas, etc. with PIC-MCC type of methods.

Traditional transfer matrix optics



Treats ion-optical elements (and drifts) as black boxes with transfer matrices describing the effect to trajectories. In TRANSPORT $X = (x, x', y, y', l, \delta p/p)$

$$X_i(1) = \sum_j R_{ij} X_j(0) + \sum_{jk} T_{ijk} X_j(0) X_k(0) + \cdots$$

Ideal 1st order quadrupole:

$$R = \begin{pmatrix} \cos kL & \frac{1}{k}\sin kL & 0 & 0 & 0 & 0 \\ -k\sin kL & \cos kL & 0 & 0 & 0 & 0 \\ 0 & 0 & \cosh kL & \frac{1}{k}\sinh kL & 0 & 0 \\ 0 & 0 & k\sinh kL & \cosh kL & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Traditional transfer matrix optics

- Matrices based on analytic formulation, numerical integration of fields or fitting experimental/simulation data.
- The whole system can be described with one matrix:

$$R_{\text{system}} = R_N \cdots R_2 \cdot R_1$$

- Can also transport elliptical envelopes in addition to trajectories:

$$\sigma_1 = R\sigma_0 R^T, \text{ where}$$
$$\sigma = \epsilon \begin{pmatrix} \beta & -\alpha \\ \alpha & \gamma \end{pmatrix}$$

- Advantage: calculation is fast (automatic optimization, etc)
- May include additional space charge induced divergence growth for beam envelopes and/or rms emittance growth modelling for particle distributions.

Codes of this type

- TRANSPORT — One of the classics, up to 2nd or 3rd order calculation, no space charge
- COSY INFINITY — Up to infinite order, no space charge
- GIOS — Up to 3rd order, space charge of KV-beam
- DIMAD — Up to 3rd order, space charge of KV-beam
- TRACE-3D — Mainly linear with space charge of KV-beam
- PATH MANAGER (TRAVEL) — Up to 2nd order, more advanced space charge modelling for particle distributions (mesh or Coulomb)

Some of the codes are more suitable for low energies, choose carefully!

Differences to high energy transport

Now $v \ll c$ and J is large

- **Space charge** plays a major role
- Beam generated B-field is negligible.
- Several ion species
- Beam line elements often not well separated (no drift spaces in between).
- Complex electrostatic electrode shapes used.
- Nonlinear effects are significant!

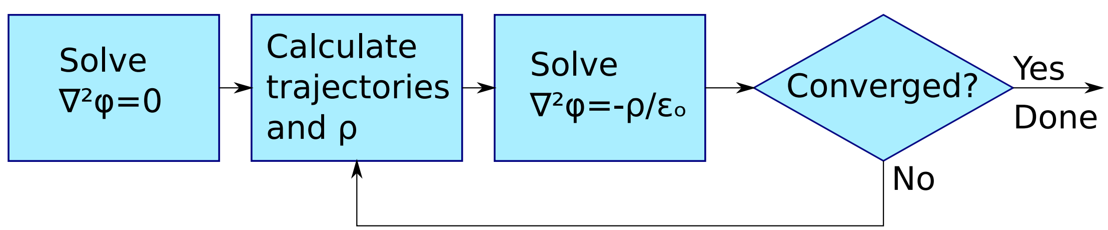
Traditional N^{th} order transfer matrix optics cannot be used (well) close to ion sources. More fundamental methods are needed.

Particle tracking codes

Particle tracking codes

Particle tracking codes for ion source extraction and LEBT systems:

- Calculation of electrostatic fields in electrode geometry including space charge effects (and possibly plasma).
- Calculation/importing of magnetostatic fields.
- Tracking of particles in the fields.
- Diagnostics and other supportive methods.



 Taneli Kalvas

Available codes of this type

- IGUN — Plasma modelling for negative and positive ions, 2D only
- PBGUNS — Plasma modelling for negative and positive ions, 2D only
- SIMION — Simple 3D E-field solver and particle tracer, low quality space charge modelling, no plasma
- KOBRA — More advanced 3D E-field solver, positive ion plasma modelling, PIC capability
- LORENTZ — State of the art 3D EM solver and particle tracer with a lot of capabilities, no plasma modelling
- IBSIMU — Plasma modelling for negative and positive ions, 1D–3D E-field solver

Ion Beam Simulator

IBSimu is an ion optical code package made especially for the needs of ion source extraction design. Using Finite Difference Method (FDM) in a regular cartesian mesh the code can model

- Systems of electrostatic and magnetic lenses
- High space charge beams (low energy)
- Positive and negative multispecies 3D plasma extraction

The code is made as a C++ library and is released freely under GNU Public Licence*.

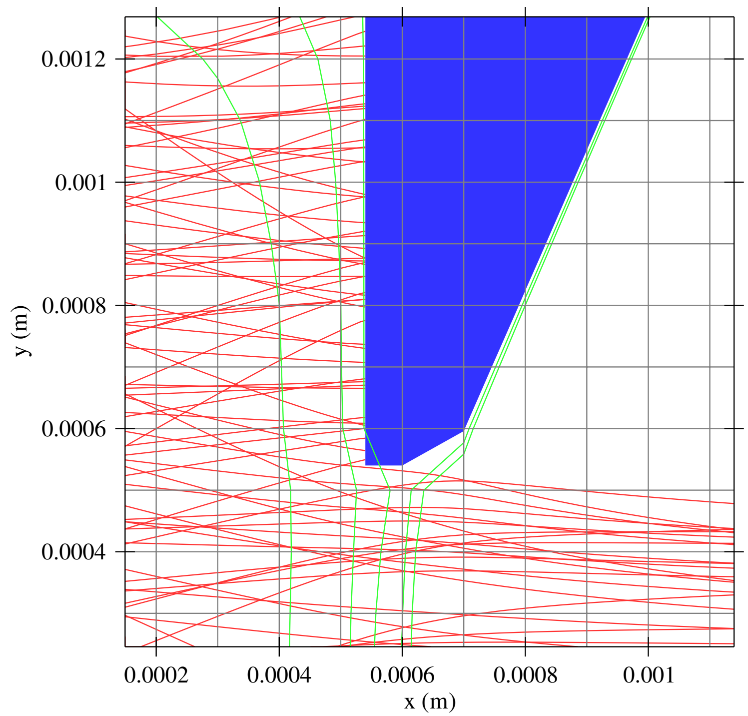
- Highly versatile and customizable.
- Can be used for batch processing and automatic tuning of parameters.

*) *http://ibsimu.sourceforge.net/*

Ion optics with FDM

Calculation is based on evenly sized square cartesian grid(s):

- Solid mesh (node type): vacuum, solid, near solid, neumann boundary condition, ...
- Electric potential
- Electric field
- Magnetic field
- Space charge density
- Trajectory density



 Taneli Kalvas

Electrostatic field solver

Poisson's equation

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0}$$

Finite Difference representation for vacuum node i:

$$\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{h^2} = -\frac{\rho_i}{\epsilon_0},$$

Neumann boundary node i:

$$\frac{-3\phi_i + 4\phi_{i+1} - \phi_{i+2}}{2h} = \frac{d\phi}{dx}$$

and Dirichlet (fixed) node i:

$$\phi_i = \phi_{\text{const}}$$

1D example

Solve a 1D system of length $L = 10$ cm, charge $\rho = 1 \cdot 10^{-6}$ C/m^3 and boundary conditions

$$\frac{\partial \phi}{\partial x}(x=0) = 0 \text{ V/m} \qquad \text{and} \qquad \phi(x=L) = 0 \text{ V}.$$

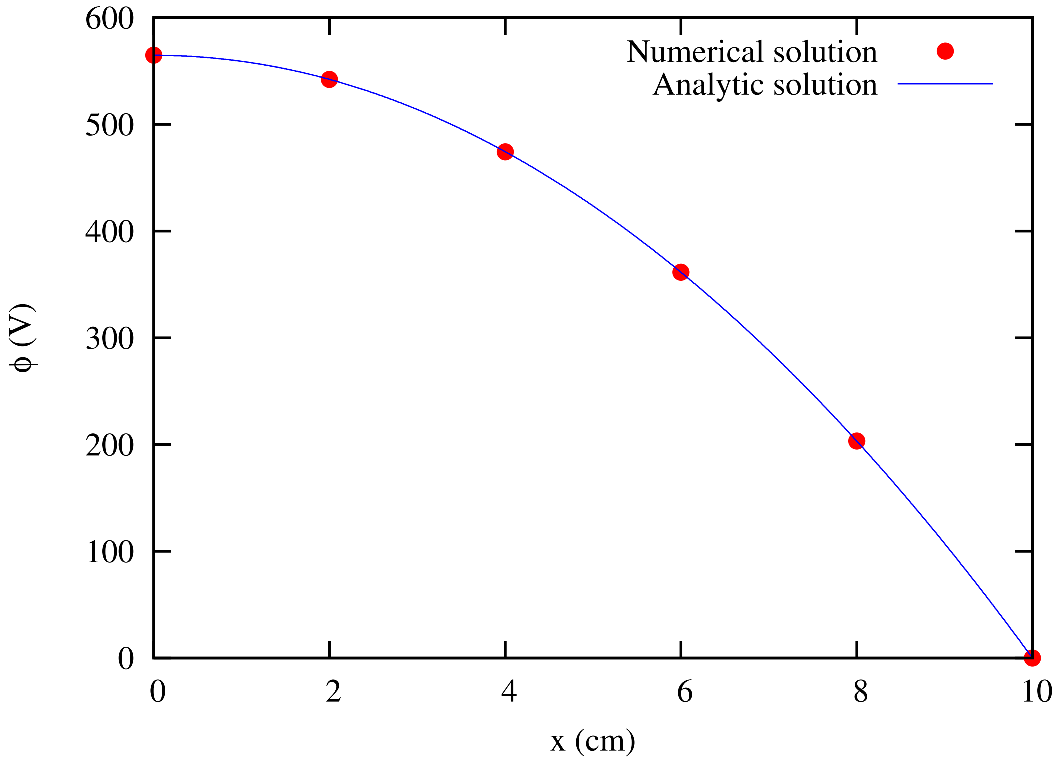
The system is discretized to $N = 6$ nodes. Problem in matrix form:

$$\begin{pmatrix} -3 & 4 & -1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \end{pmatrix} = \begin{pmatrix} 2h \frac{\partial \phi}{\partial x}(0) \\ -h^2 \frac{\rho}{\epsilon_0} \\ -h^2 \frac{\rho}{\epsilon_0} \\ -h^2 \frac{\rho}{\epsilon_0} \\ -h^2 \frac{\rho}{\epsilon_0} \\ \phi(L) \end{pmatrix}$$

Solving the matrix equation we get ...

1D example

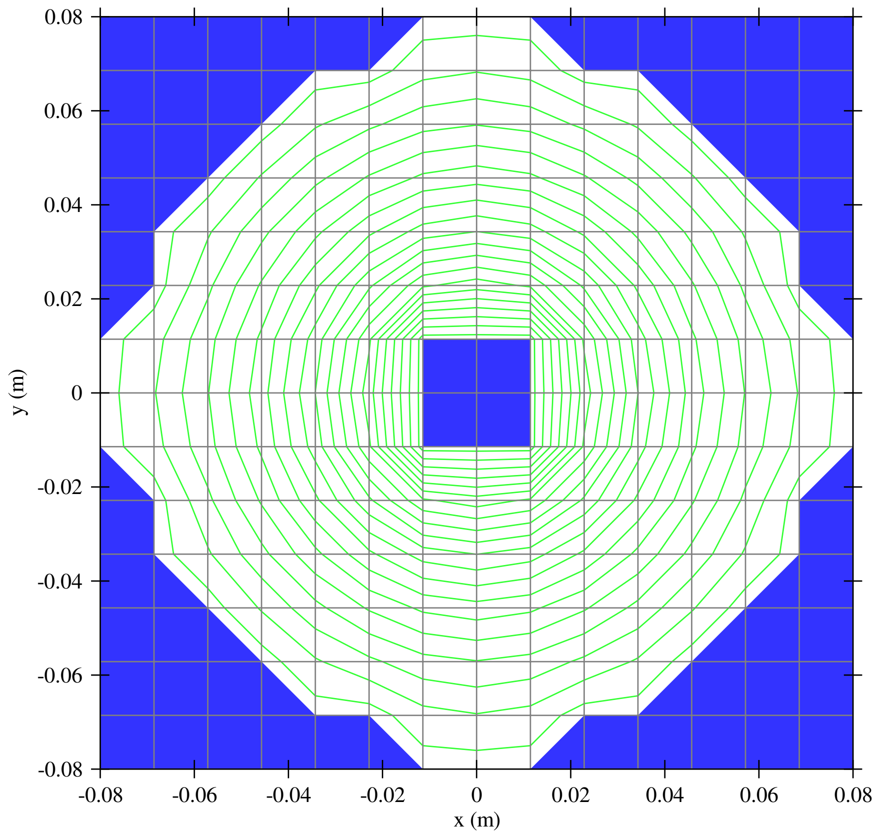
... perfect agreement with analytic result



but only because of flat charge distribution and boundaries defined exactly at node locations.

Jagged boundaries

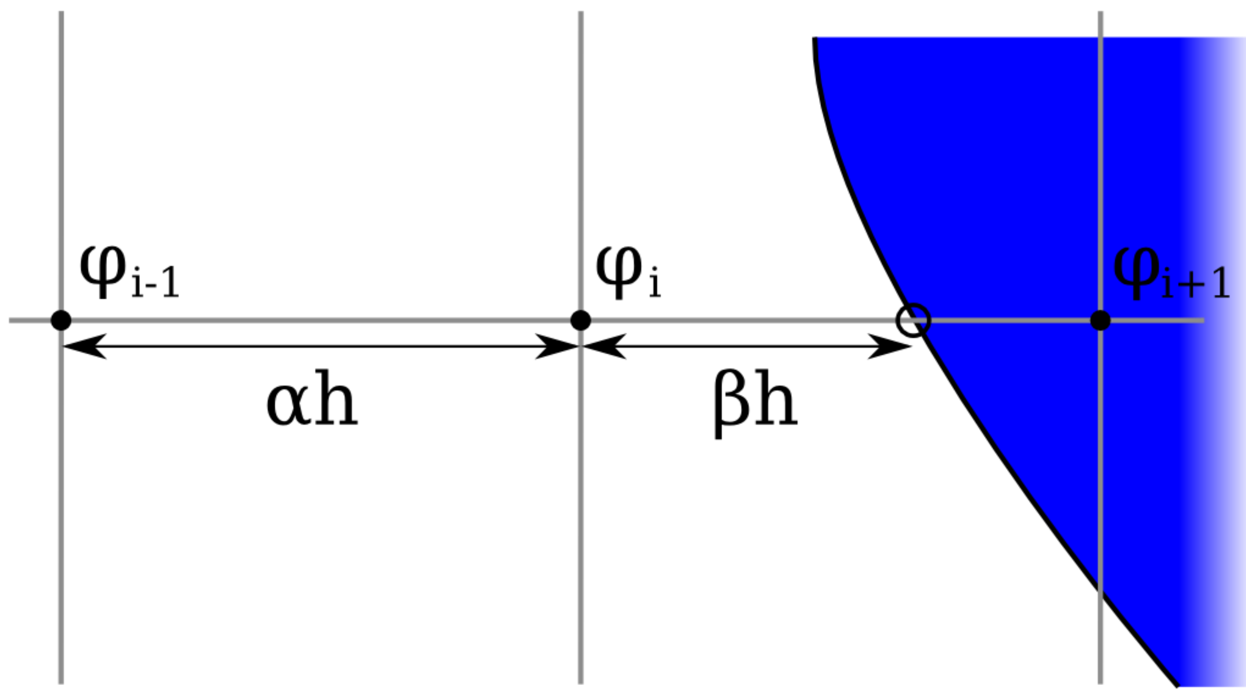
In higher dimensions basic FDM generally suffers from jagged boundaries (nodes don’t coincide with surfaces).



Smooth boundaries

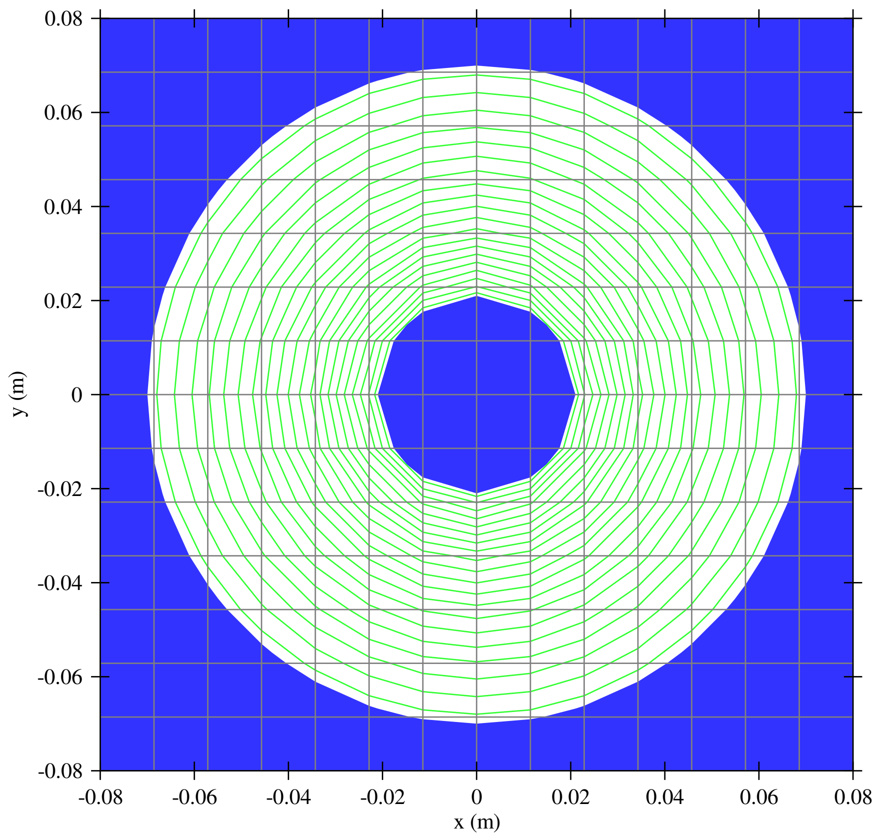
Derivatives in Poisson's equation featured with uneven distances

$$\frac{\beta\phi(x_0-\alpha h)-(\alpha+\beta)\phi(x_0)+\alpha\phi(x_0+\beta h)}{\frac{1}{2}(\alpha+\beta)\alpha\beta h^2}=-\frac{\rho(x_0)}{\epsilon_0}$$



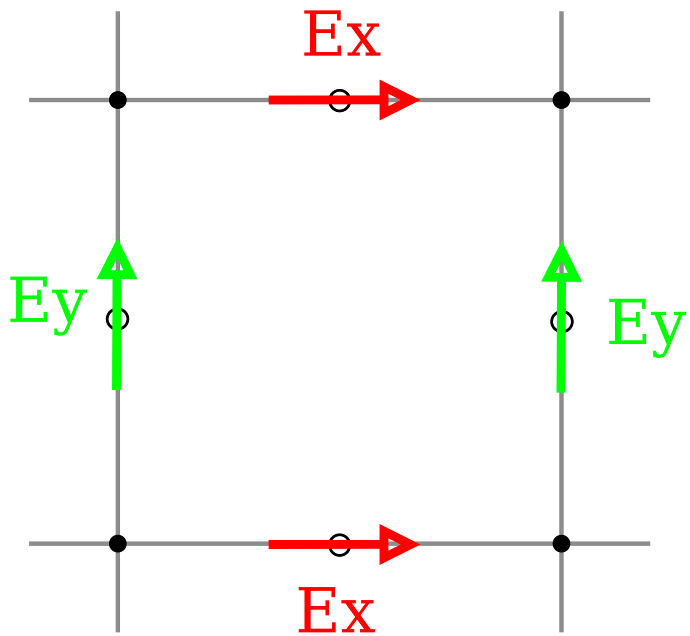
Smooth boundaries

A much better solution with smooth boundaries is achieved.



Electric field calculation

Electric field is calculated between the nodes simply by $E = \frac{V}{h}$.

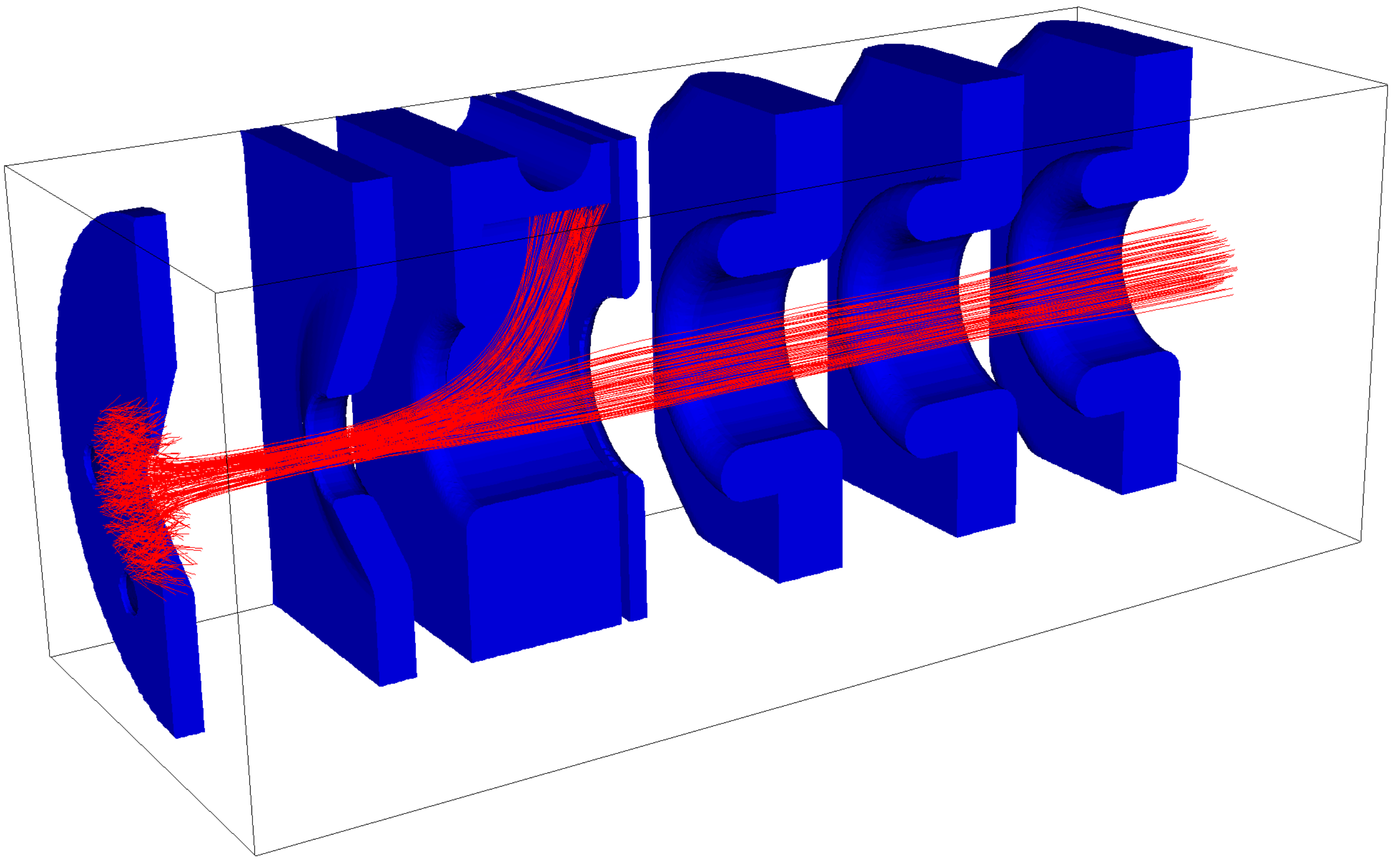


Electric field nodes between potential nodes.

Taneli Kalvas

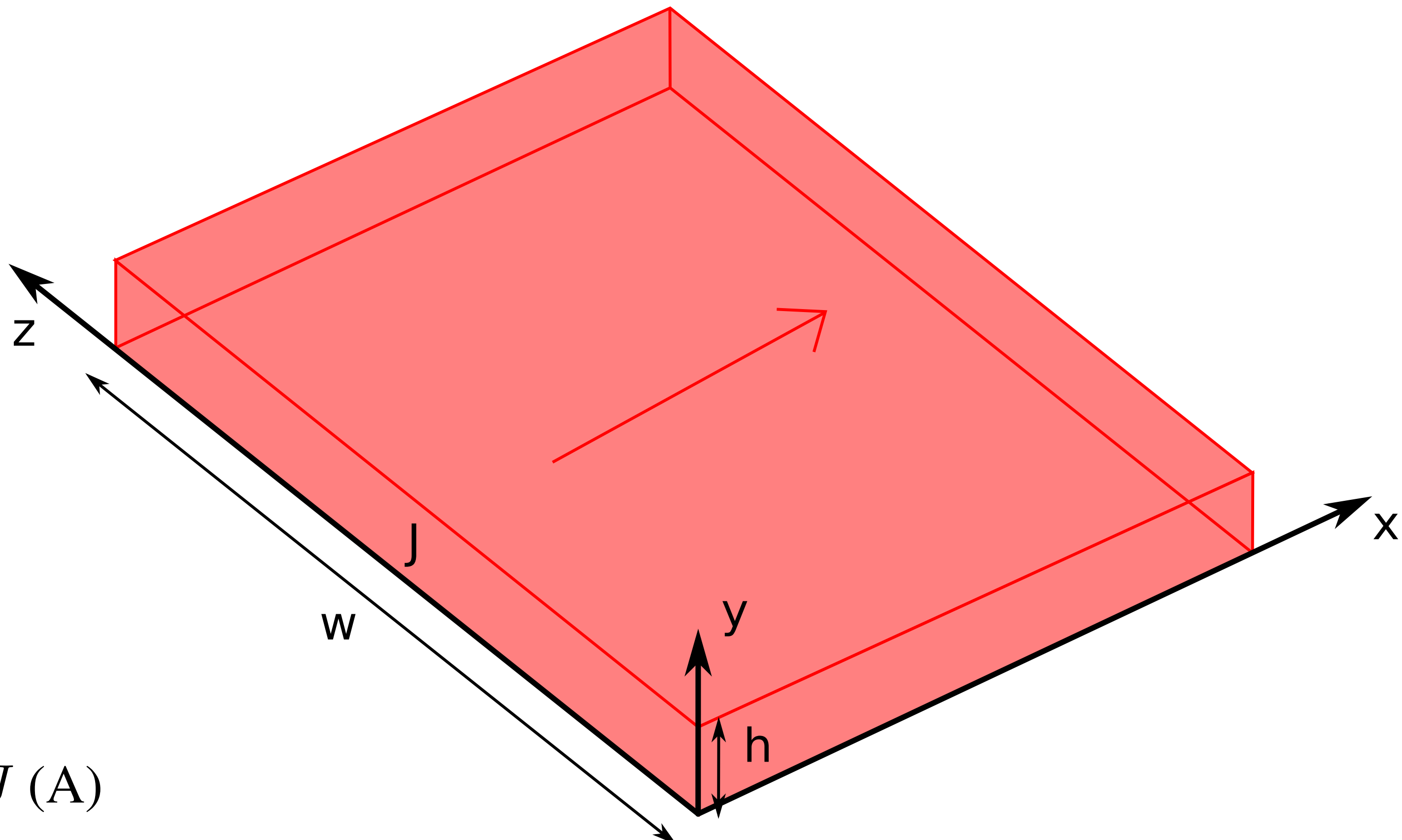
Problem geometries

3D: $\vec{E}$, $\vec{B}$, $\vec{x}$, $\vec{v}$



Problem geometries

Planar 2D: (E_x, E_y), B_z, (x, y), (v_x, v_y)

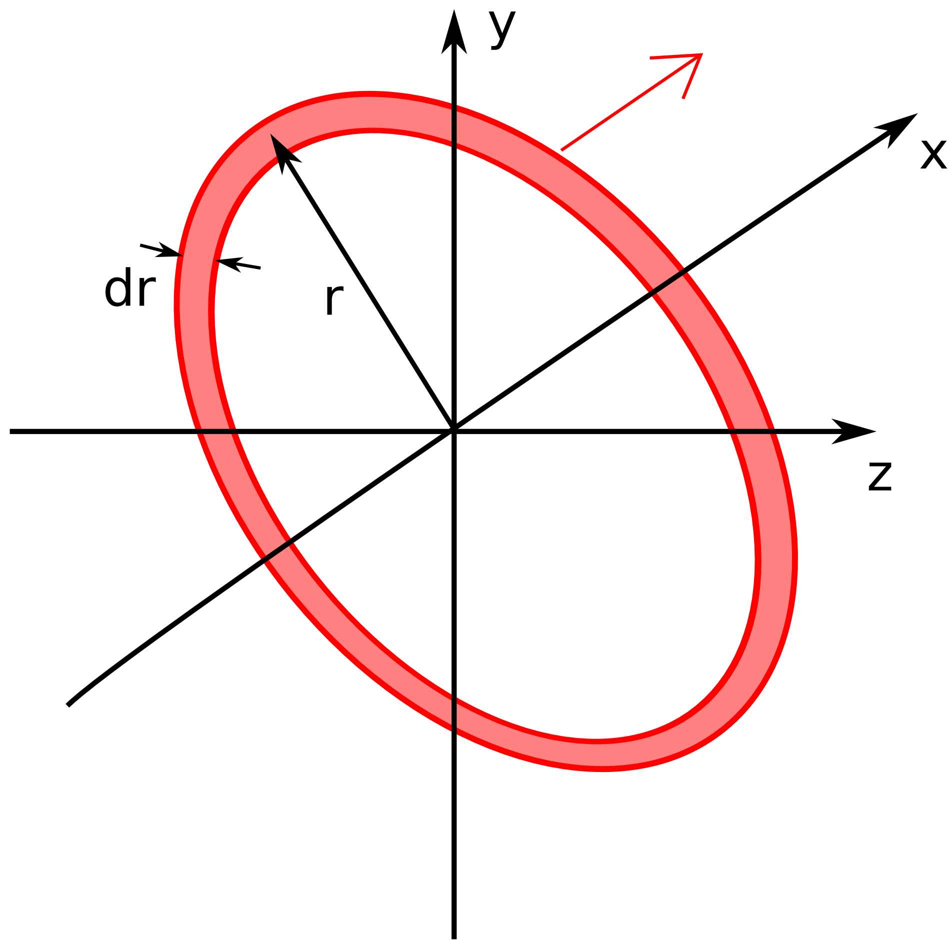


$I = whJ$ (A)

$\tilde{I} = hJ$ (A/m)

Problem geometries

Cylindrically symmetric: (E_x, E_r), (B_x, B_r, B_θ), (x, r), (v_x, v_r, ω)



$I = 2\pi r \mathrm{d}r J$ (A)

Trajectory calculation

Population of virtual particles is calculated with following properties:

- Charge: q
- Mass: m
- Current carried: I
- Time, position and velocity coordinates:
  - 2D: (t, x, v_x, y, v_y)
  - Cylindrical symmetry: $(t, x, v_x, r, v_r, \omega)$, $\omega = \frac{d\theta}{dt}$
  - 3D: $(t, x, v_x, y, v_y, z, v_z)$

Trajectory calculation

Calculation of trajectories done by integrating the equations of motion

$$
\begin{aligned}
\frac{dx}{dt} &= v_x \\
\frac{dy}{dt} &= v_y \\
\frac{dz}{dt} &= v_z \\
\frac{dv_x}{dt} &= a_x = \frac{q}{m}(E_x + v_y B_z - v_z B_y) \\
\frac{dv_y}{dt} &= a_y = \frac{q}{m}(E_y + v_z B_x - v_x B_z) \\
\frac{dv_z}{dt} &= a_z = \frac{q}{m}(E_z + v_x B_y - v_y B_x)
\end{aligned}
$$

Trajectory calculation

... and in cylindrical symmetry:

$$
\begin{aligned}
\frac{dx}{dt} &= v_x \\
\frac{dr}{dt} &= v_r \\
\frac{dv_x}{dt} &= a_x = \frac{q}{m}(E_x + v_r B_\theta - v_\theta B_r) \\
\frac{dv_r}{dt} &= a_r + r\omega^2 = \frac{q}{m}(E_y + v_\theta B_x - v_x B_\theta) + r\omega^2 \\
\frac{d\omega}{dt} &= \frac{1}{r}\left(a_\theta - v_r\omega\right) = \frac{1}{r}\left(\frac{q}{m}(v_x B_r - v_r B_x) - 2v_r\omega\right),
\end{aligned}
$$

where $v_\theta = r\frac{d\theta}{dt} = r\omega$

Trajectory calculation

For relativistic particles, the equation of motion in 3D is

$$\gamma \frac{d\vec{v}}{dt} + \gamma^3 \frac{\vec{v}}{c^2} \left(\vec{v} \cdot \frac{d\vec{v}}{dt} \right) = \frac{q}{m} (\vec{E} + \vec{v} \times \vec{B}),$$

where $\gamma = 1/\sqrt{1 - v^2/c^2}$ is the relativistic gamma factor. The particle acceleration $d\vec{v}/dt$ is solved from the equation above and it is used in the defining system of equations.

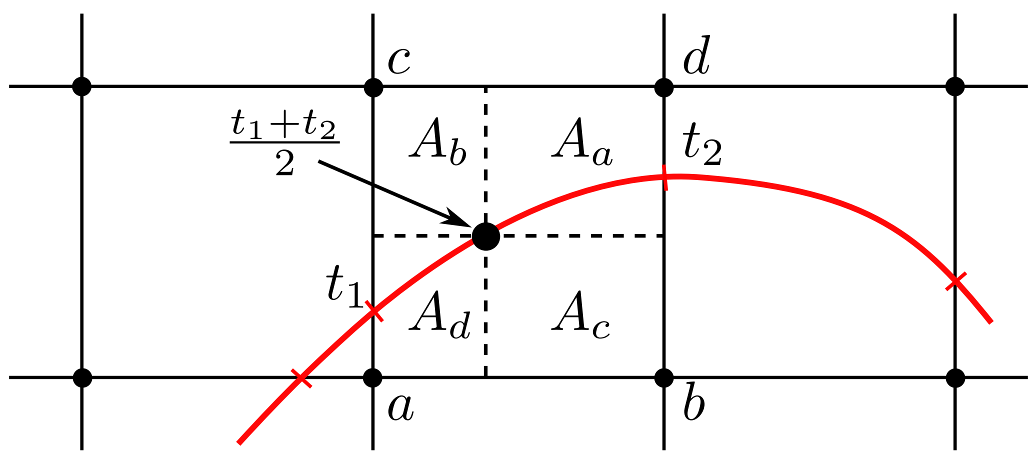
Space charge deposition

The space charge

$$Q = I\Delta t$$

is deposited inside each grid cell assuming that charge cloud is centered at the location, where the particle was at $t = \frac{1}{2}(t_1 + t_2)$. The nodes receive charge density $\rho_{i,j,k} = QS(\vec{r} - \vec{r}_{i,j,k})$ according to trilinear weighting function

$$S(x,y,z) = \begin{cases} \frac{1}{h^3}(1 - |x/h|)(1 - |y/h|)(1 - |z/h|) & |x| < h, |y| < h, |z| < h \\ 0 & \text{otherwise.} \end{cases}$$



 Taneli Kalvas

Plasma-beam interface

Ions are extracted from a plasma ion source

1. Full space charge compensation ($\rho_- = \rho_+$) in the plasma
2. No compensation in extracted beam (single polarity)

The boundary is often thought as a sharp surface known as the *plasma meniscus* dividing the two regions.

- Works as a thought model.
- In reality compensation drops going from plasma to beam in a transition layer with thickness $\sim \lambda_D \Rightarrow$ plasma sheath.
- E-field in extraction rises smoothly from zero.

Plasma flux

The plasma flux to a surface is

$$J = \frac{1}{4} q n \bar{v} = q n \sqrt{\frac{kT}{2\pi m}}$$

Extraction hole: ion beam samples plasma species with weight $\propto m^{-1/2}$.

Plasma flux sets the maximum current extractable

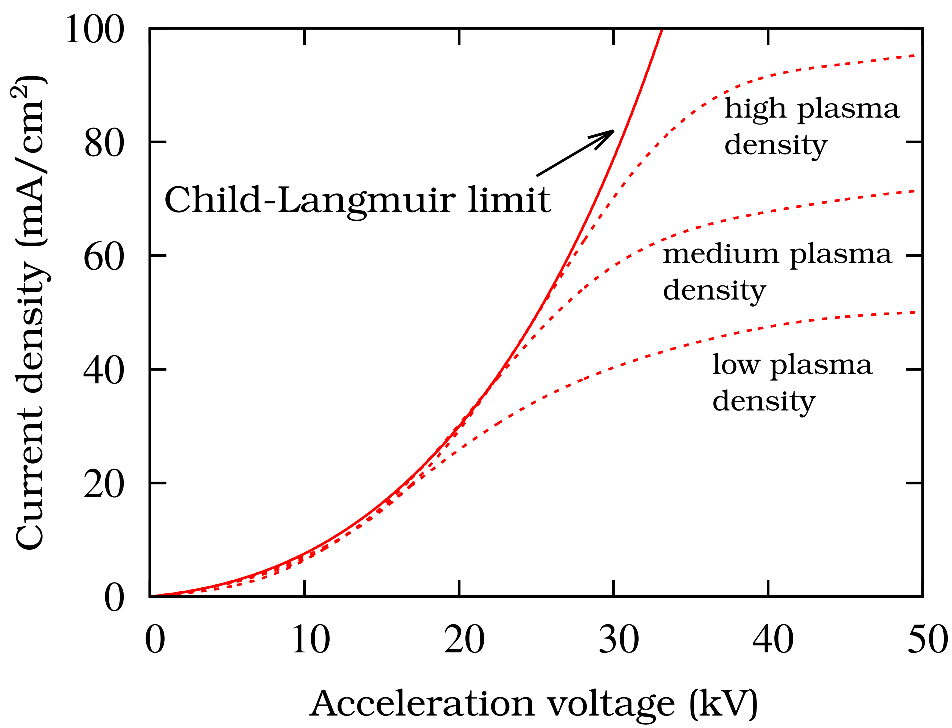
$$I = J A_{\text{meniscus}},$$

where the area of plasma meniscus $A_{\text{meniscus}} \neq A_{\text{aperture}}$ and therefore not quite constant. N-dimensional simulations needed for better estimates.

Child-Langmuir law

Ion beam propagation may also be limited by space charge. The 1D Child-Langmuir law gives the maximum current density for the special case where the beam is starting with $v_0 = 0$ (not plasma).

$$J = \frac{4}{9}\epsilon_0 \sqrt{\frac{2q}{m}} \frac{V^{3/2}}{d^2}.$$



 Taneli Kalvas

Thermal plasma sheath

Classic 1D plasma sheath theory: In an electron-ion plasma a positive plasma potential is formed due to higher mobility of electrons. Situation is described by Poisson equation

$$\frac{d^2U}{dx^2} = -\frac{\rho}{\epsilon_0} = -\frac{en_0}{\epsilon_0}\left[\sqrt{1-\frac{2eU}{m_i v_0^2}} - \exp\left(\frac{eU}{kT_e}\right)\right],$$

where the ions entering the sheath have an initial velocity

$$v_0 > v_{\mathrm{Bohm}} = \sqrt{\frac{kT_e}{m_i}}$$

or kinetic energy

$$E_0 > \frac{1}{2} m_i v_{\mathrm{Bohm}}^2 = \frac{1}{2} kT_e.$$

Model applies quite well for positive ion plasma extraction.

Positive ion plasma extraction model

Groundbreaking work by S. A. Self, *Exact Solution of the Collisionless Plasma-Sheath Equation*, Fluids **6**, 1762 (1963) and
J. H. Whealton, *Optics of single-stage accelerated ion beams extracted from a plasma*, Rev. Sci. Instrum. **48**, 829 (1977):

$$\frac{d^2U}{dx^2} = -\frac{\rho}{\epsilon_0} = -\frac{\rho_{\mathrm{rt}} + \rho_{\mathrm{e}}(U)}{\epsilon_0}$$

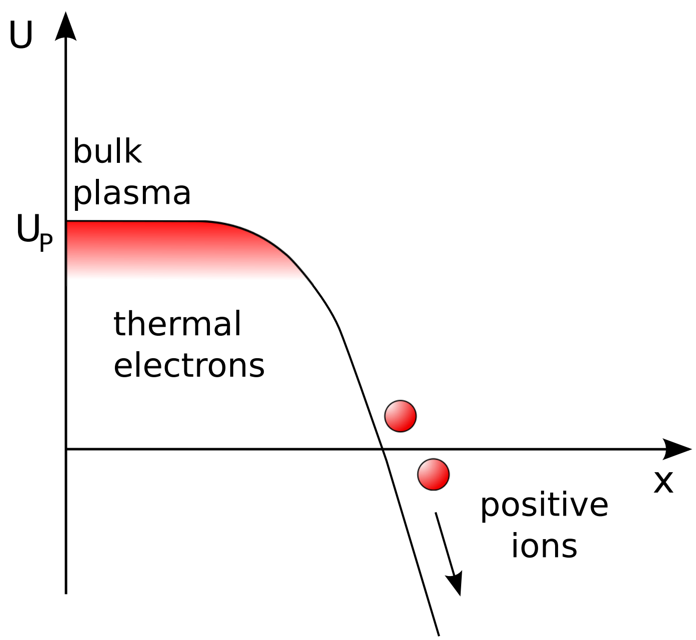
- Model has been used very successfully for describing positive ion extraction systems since.
- Assumptions: no ion collisions, no ion generation, electron density only a function of potential (no magnetic field).
- Take the model with a semiempirical approach and use it as a tool proving to yourself that it works for your case — don't take it for granted.

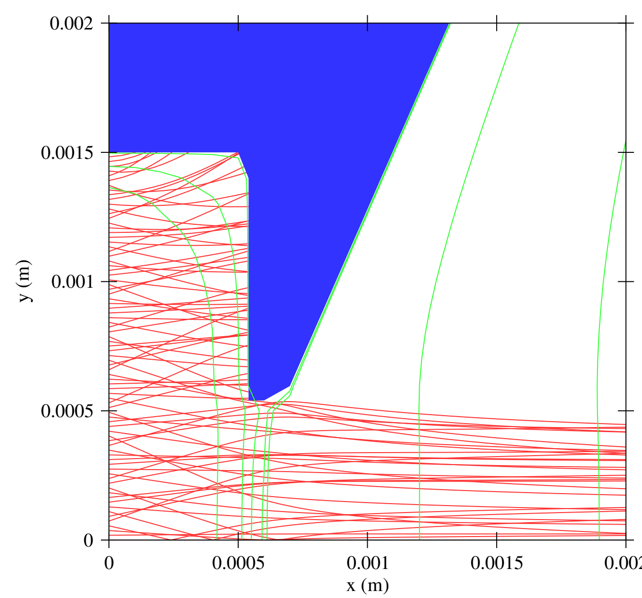
Positive ion plasma extraction model

Modelling of positive ion extraction

- Ray-traced positive ions entering sheath with initial velocity
- Nonlinear space charge term (analytic in Poisson's equation):

$$\rho_e = \rho_{e0} \exp\left(\frac{U - U_P}{kT_e/e}\right)$$



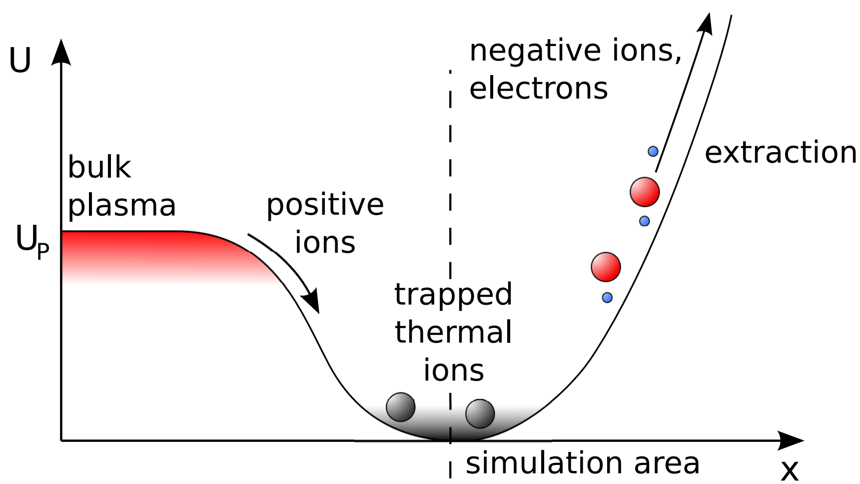


 Taneli Kalvas

Negative ion plasma extraction model

Modelling of negative ion extraction

- Ray-traced negative ions and electrons
- Analytic thermal and fast positive charges
- Magnetic field suppression for electrons inside plasma



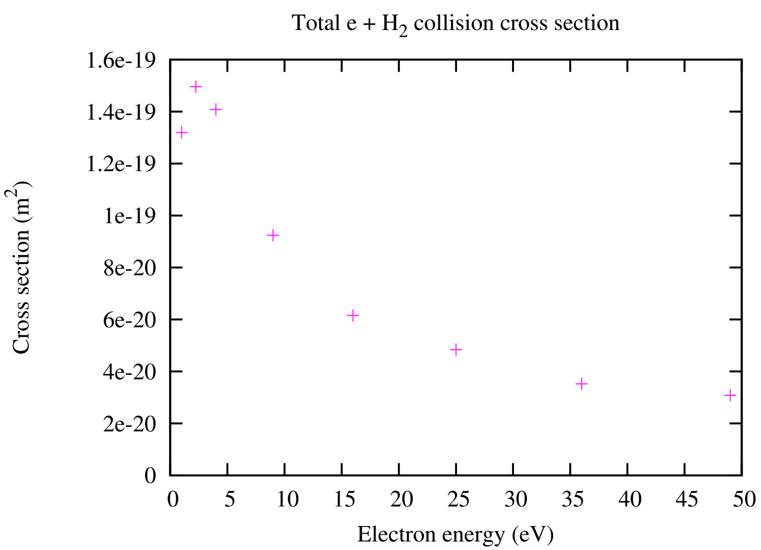
$$\rho_{\mathrm{th}} = \rho_{\mathrm{th0}} \exp\left(\frac{-eU}{kT_i}\right)$$

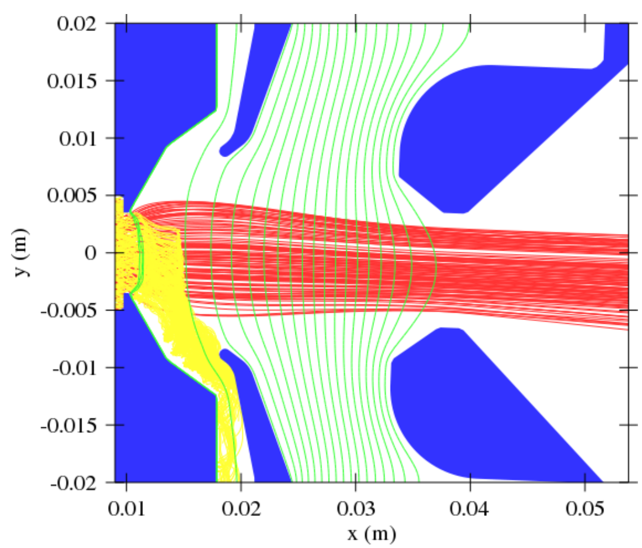
$$\rho_{\mathrm{f}} = \rho_{\mathrm{f0}} \left(1 + \mathrm{erf}\left(\frac{eU}{E_i}\right)\right)$$

Negative ion plasma extraction model

Magnetic field suppression for electrons inside plasma

- Electrons highly collisional until velocity large enough
- Ray-tracing simulations not capable of simulating diffusion
- Magnetic field does not dictate electron trajectories in plasma
  $\rightarrow$ B-field suppression is a sufficient approximation





Difficulties in modelling extraction systems

1. Amount of parameters fed to the model is quite large

- Extracted species: J_i, T_i, v_0
- Positive ion plasma model: T_e, U_P
- Negative ion plasma model: T_i, E_i/T_i, gas stripping loss of ions
- All: space charge compensation degree and localization in LEBT

Methods: educated guessing (literature data), plasma measurements and matching to beam measurements (emittance scans).

2. Effect of approximation in plasma model

- Very difficult to estimate
- Comparison to simulations containing more (and more accurate) physics

Getting started

Getting started

Practical work

- We will construct examples together at computer class
- We will read the manual
- We will (possibly) make errors and fix them

Environment

- Linux server at `ph1.phys.jyu.fi` is at your disposal
- Your own laptops
- PC-class Windows computers have `putty` and `Xming` server for graphics
  - Start `Xming`
  - Connect with `putty` with X11 forwarding
    (Tick Connection → SSH→ X11 → Enable X11 forwarding)

Getting started

Software

- IBSimu 1.0.6 (installation covered later during introduction)
- Something to edit DXF-files with (server has LibreCAD)
- Something to plot with (server has gnuplot)

Installation, versions, new features, etc

Installation/compilation

- Where to get and what to get
  - Sourceforge/Files: Releases
  - Git: Development versions (recommended)
- Tools
  - Release: g++, POSIX compilation environment
  - Git package needs GNU toolchain autotools!
- Dependencies
  - GNU Scientific Library, zlib
  - GTK+-3.0 (cairo, libpng, FontConfig, Freetype2)
  - Conditional: GtkGLExt, UMFPACK, CSG
- Compilation, optimization

Versions

Stable releases

- Public versions started from 1.0.0, currently 1.0.6 (Jun 15 2015)

Development versions

- IBSimu does not have unstable releases, development versions are available through git, a revision control system.
- Version stamp is given according to the last stable release with additional "dev" tag, currently 1.0.6dev.
- Git tags each revision with a hexadecimal string, which is printed along with a date on salution string, for example

  `Ion Beam Simulator 1.0.6dev (d938f8b, Mon Jun 15 12:27:35 2015 +0300)`

- Every now and then a development version is packaged on `http://ibsimu.sourceforge.net/download.html` page.

Versions

Development versions

- Development versions have bug fixes, new features, more documentation, etc.
- More probable to have bugs, changes in API
- Use is recommended
- Different development branches may exist (there used to be a `new_solver` branch). Users will be informed if necessary.

Git

Optimization and non-standard pkgconfig directory, edit `.bashrc`:

```
export CFLAGS="-O2"
export LD_LIBRARY_PATH="/home/tvkalvas/lib"
export PKG_CONFIG_PATH="/home/tvkalvas/lib/pkgconfig"
```

Fetch IBSimu, configure and compile:

```
> git clone git://ibsimu.git.sourceforge.net/gitroot/ibsimu/ibsimu
> cd ibsimu
> ./reconf
> ./configure --prefix=/home/tvkalvas
> make -j4
> make install
```

Fetch IBSimu, configure and compile:

```
> git pull
```

It might not be necessary to run the reconf and configure scripts. The need depends on what was updated.

```
> ./reconf
> ./configure --prefix=/home/tvkalvas
> make -j4
> make install
```

Bug reports, requests, contributions

Found a bug or want something that does not exist?

- Use email list
- I will usually make small updates quickly upon request
- Larger scale work can be discussed

Contributions

- If you are capable, you are welcome to contribute
- Please patch (even preliminary) work with `git diff`