USPAS Course
Accelerator Modeling Lab

LA-UR-14-24403

General Course Outline

Morning sessions

Monday
- Why build RF Linac for FEL
- Major elements of an RF linac
- Introduction to beam physics

Tuesday
- Photoinjector
- RF linac

Wednesday
- Transverse dynamics
- Longitudinal dynamics

Thursday
- Bunch compression
- Collective effects

Friday
- Final Exam

Afternoon sessions & lab

Monday
- How do we get the beams we need
  - describing the beam
- Two major modeling approaches
- Introduction to Elegant and SDDS

Tuesday
- Matrix transform
- Acceleration
- Emittance damping

Wednesday
- Linac design
- Start design project

Thursday
- More Simulation Details
- Linac design project

Friday
- Design project presentation (one from each team)

Monday

Outline Monday

- Simulation codes
  - Installing what's needed for the course
  - Checking the installation
- Basics of modeling
  - What physics are included?
  - How do we represent the particle beam?
- Introduction to matrix codes
  - Strengths and limitations
  - Codes that use it
  - Analogies to other fields
  - Other codes: particle pushers, PICs, etc.
- A brief but **elegant** example

What's the difference between a plasma and a beam?

- Plasma
  - can be charged (net positive or negative), often net neutral
  - can be a mix of different types of particles
  - big spreads in particle speed (energy)
  - going every which way at once

- Beam
  - usually charged
  - usually the same species of particle
  - relatively small spread about the average speed
  - generally going in the same direction at about the same time

In “Reality”…

- The beam, formally, is defined by a vector of 6N values at a time t:
  - each particle in the beam has six coordinates (x, p_x, y, p_y, z, p_z)
  - there are N particles per beam, all assumed to have the same charge

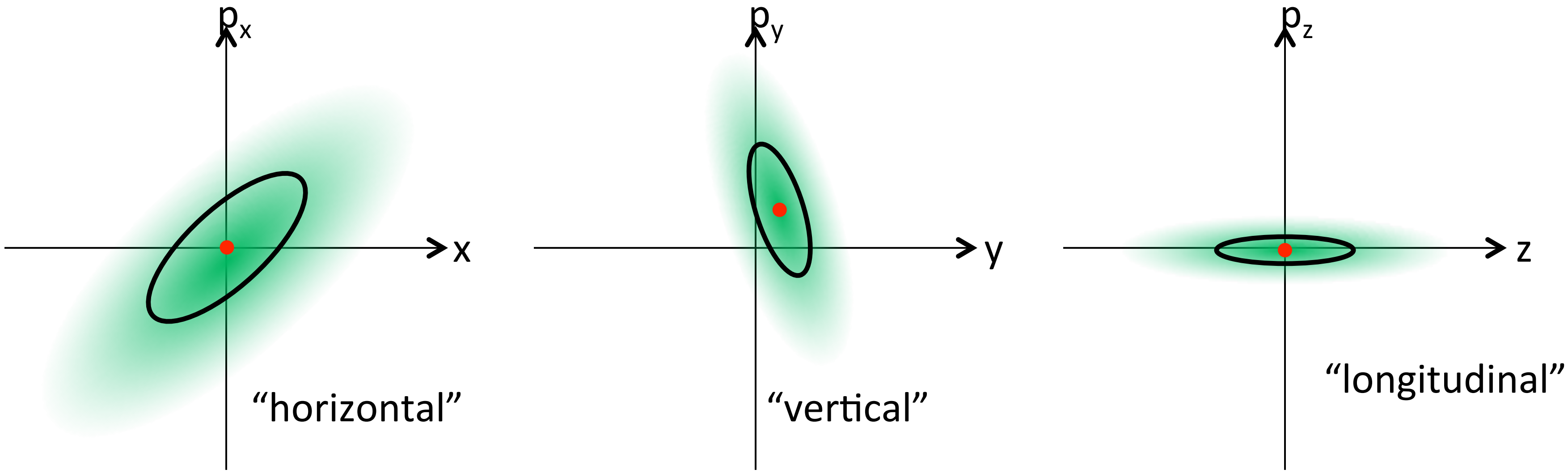
- We typically think of the beam as a collection of interacting particles in a 6-dimensional space
  - N assumed to be large enough so as to allow statistical descriptions
  - Calculations based on projections of the 6-d phase space onto a point, line, plane, or volume, yield properties such as emittance, spot size, etc.

- Linacs can induce unwanted as well as desired correlations, again visualized as projected onto various planes or n-dimensional volumes.
  - We can remove some of the unwanted ones, sometimes.

How do we describe the beam?

- 0^{th} order parameters
  - describe the beam's centroid, or average location, in 6-d phase space and total charge: Q, $(\langle x\rangle, \langle p_x\rangle, \langle y\rangle, \langle p_y\rangle, \langle z\rangle, \langle p_z\rangle)$
  - spatial coordinates measured relative to the "ideal" trajectory along the linac
- 1^{st} order parameters: spot sizes, bunch length, energy spread
  - describe the beam's RMS size along one axis in 6-d phase space: $\sigma_x, \sigma_{px}, \sigma_y, \sigma_{py}, \sigma_z, \sigma_{pz}$
- 2^{nd} order parameters: correlations, quality
  - the beam's correlation along two co-axes are what we usually think of as divergences, e.g. $\langle x\cdot p_x\rangle$, or chirps, e.g. $\langle z\cdot p_z\rangle$
  - other correlations describe "skews", e.g. $\langle x\cdot z\rangle$ or dispersive effects, e.g. $\langle p_x\cdot p_z\rangle$
  - describe the beam's density-weighted area on a given plane, e.g. horizontal emittance $\varepsilon_{n,x}$
- Brightness
  - A measure of the beam's quality as a whole; definitions vary according to application, but usually include charge, duration, and transverse emittance
  - Can think of brightness as beam's charge density in 5- or 6-d space, depending on definition

3-plane projections



We can **project** the 6-d beam onto three planes: $(x\text{-}p_x)$ $(y\text{-}p_y)$ and $(z\text{-}p_z)$

We can plot the **location** of the beam centroid in each plane:
$(\langle x\rangle,\langle p_x\rangle)$ $(\langle y\rangle,\langle p_y\rangle)$ $(\langle z\rangle,\langle p_z\rangle)$

We can define an **rms ellipse** describing the beam in each plane; from this we can calculate 1st and 2nd-order parameters

Alternate Coordinate Representations

- Momentum vs. angle

$$\left(x, p_x, y, p_y, z, p_z\right) \Leftrightarrow (x, x', y, y', z, p)$$

$$p_x = \frac{p\,x'}{\sqrt{x'^2 + y'^2 + 1}} \qquad x' = {p_x}\big/{p_z}$$

$$p_y = \frac{p\,y'}{\sqrt{x'^2 + y'^2 + 1}} \qquad y' = {p_y}\big/{p_z}$$

$$p_z = \frac{p}{\sqrt{x'^2 + y'^2 + 1}} \qquad p = \sqrt{p_x^2 + p_y^2 + p_z^2}$$

If x’, y’ << 1, we can use paraxial approximations to model transport

Alternate Representations

- Momentum vs. energy

$$(x,x',y,y',z,p) \Leftrightarrow (x,x',y,y',z,E)$$

$$\begin{aligned} p &= \gamma m v & E &= \sqrt{p^2c^2 + m^2c^4} \\ &= \beta\gamma mc & &= mc^2\sqrt{1+\beta^2\gamma^2} \\ &= \frac{1}{c}\sqrt{E^2 - m^2c^4} & & \end{aligned}$$

- normalized momentum: $p = \beta\gamma mc \quad E = mc^2\sqrt{\not{p}^2 + 1}$
  - express p in units of mc $\quad = \not{p}mc$

Alternate Representations

- Absolute momentum / energy vs. fractional difference

$$(x, x', y, y', z, p) \Leftrightarrow (x, x', y, y', z, \delta)$$

$$p = \langle p \rangle (1 + \delta) \quad \delta = \frac{p}{\langle p \rangle} - 1$$

- Useful result: σ_δ = RMS fractional momentum (energy) spread in the beam

Alternate Representations

Beam as a function of time

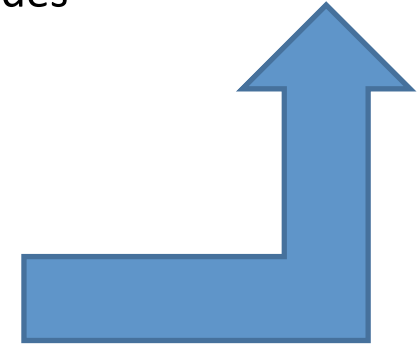
(x, p_x, y, p_y, z, p_z) at time t_i

- Treats time as the independent variable
- Consistent with a "natural" viewpoint of how the beam evolves
- Consistent with particle-in-cell modeling codes

Beam as a function of distance

(x, p_x, y, p_y, t, p_z) when the beam particle crosses the plane z_i

- Treats distance along the accelerator as the independent variable
- Works well with the "element" model of accelerators
- The viewpoint of most matrix-based codes



Even if our modeling code uses time as the independent variable, we often output the data in this format because it is consistent with:

- most of our diagnostics devices;
- most of our emittance and brightness definitions

What properties are important?

- Usually … *all of them!*

- Generally, most attention paid to:
  - emittance
  - bunch length & charge (peak current)
  - spot size
  - energy spread

How do we get the beams we need?

- We assume we know the beam properties we need at the entrance of the undulator
  - 0th order: bunch charge, horizontal / vertical position, H/V entry angles, arrival time, average energy/momentum, etc.
  - 1st order: RMS sizes in x, x', y, y', t, E (spot size, bunch length, energy spread)
  - 2nd order: divergence, chirp, H/V emittance

- We can readily adjust many 0th and 1st order parameters within the linac, as well as most in-plane correlations (e.g. divergence)

- Charge and other 2nd order parameters (e.g. emittance)
  - the beam at the source is usually as good as it will ever be anywhere in the linac
    - generally can't add charge along the way
    - phase-space is conserved … or made worse
  - we can make some improvements in, say, emittance, but usually only at the expense of losing charge (e.g. aperturing)
  - some correlations can be "backed out" in the injector to help improve emittance, but generally this only works in simple, well-defined, quasi-linear situations

How do we adjust 0^{th}-order parameters?

- Charge (Q_b)
  - adjust the beam source to get more / less
  - can remove some along the way (deliberate = collimate; accidental = scrape)

- Horizontal / vertical position (<x>, <y>) and angle (<x'>, <y'>)
  - small dipole magnets
  - usually ignored until it's time for error-tolerancing studies

- Arrival time <t>
  - trajectory length adjustments (either via magnet strengths or via average energy at dispersive elements)
  - emission time at beam source
  - in principle, time of flight; but that's not very practical with electron machines

- Average energy <E> or momentum <p>, $\langle\beta\gamma\rangle$
  - accelerating structure gradient and / or phase

Linac Energy Gain

Single particle:

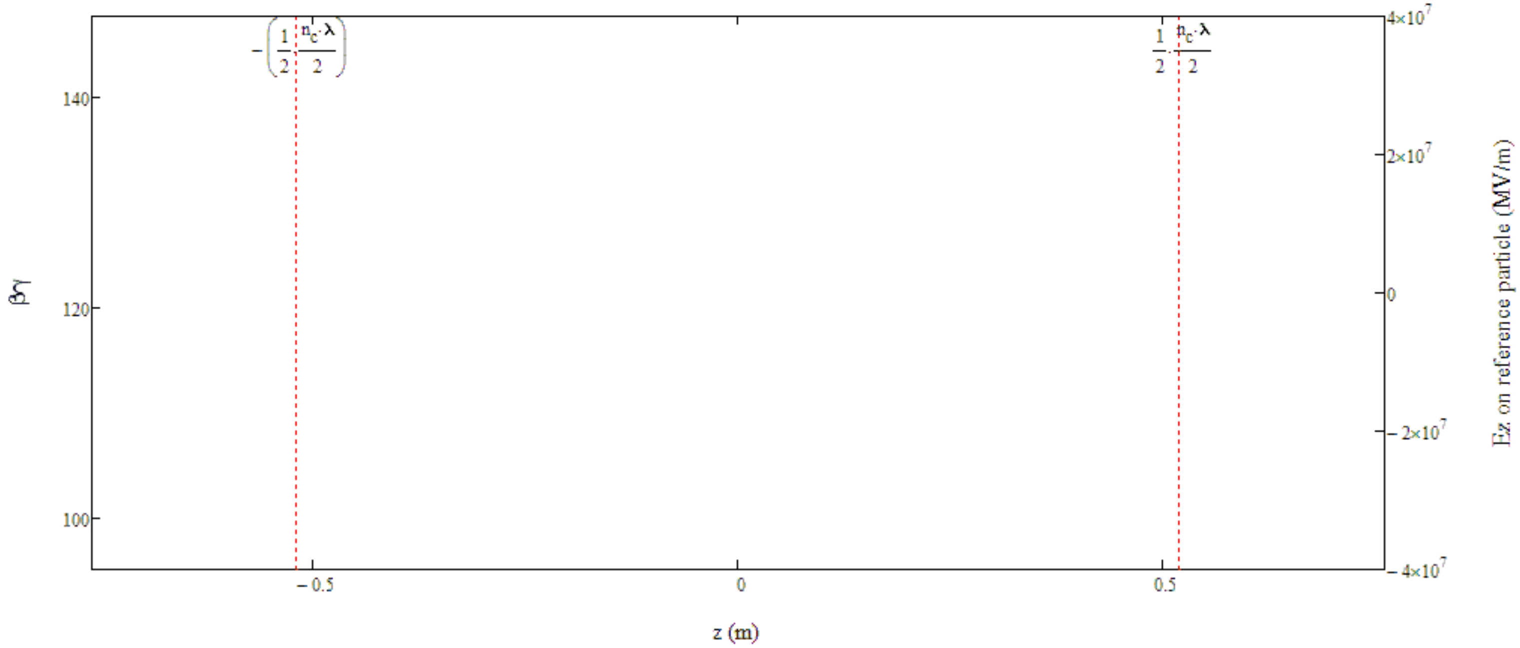
$$E_o(t,\phi) = E_i(t) + GL\cos(\omega t + \phi)$$

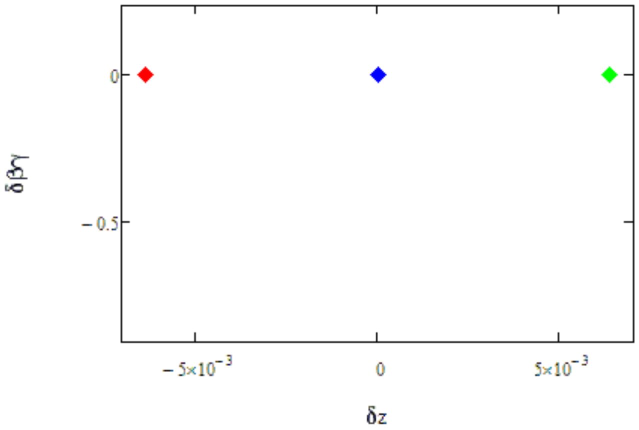
G = average on-crest gradient (MV/m)
L = length of linac structure (m)
ω = linac RF frequency
ϕ = phase of RF relative to t_0
t = time a particle arrives relative to t_0
t_0 = time the center of the bunch arrives

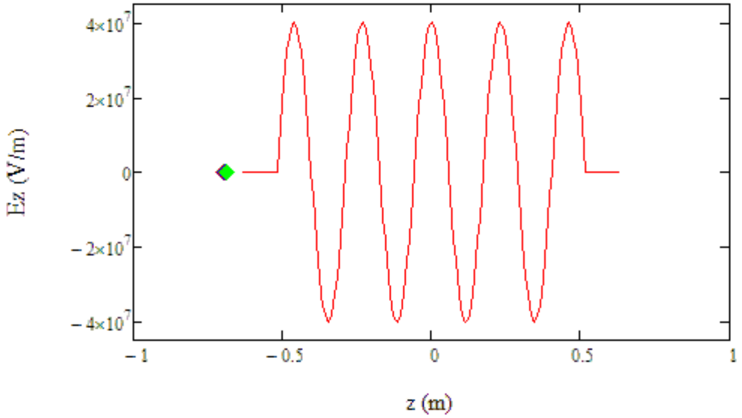
Repercussions:

- Nonlinear curvature in longitudinal phase space comes from sine wave
- Longer bunches, all else being equal, get bigger energy spreads

Animation: On-crest in standing-wave linac tank







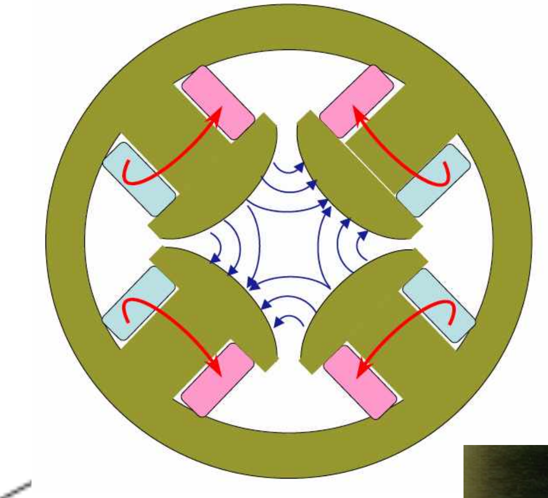
How do we adjust 1st-order parameters?

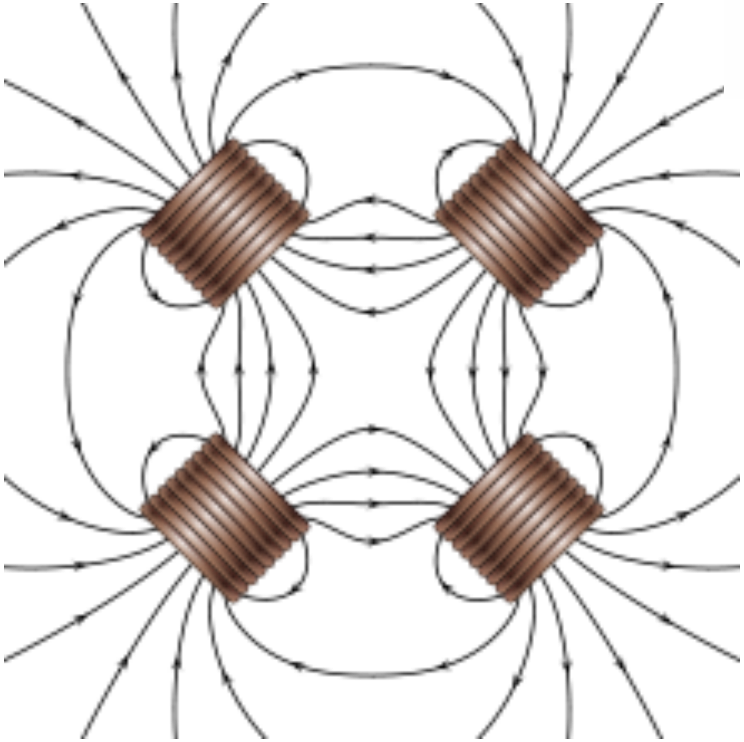
- Transverse RMS spot size (σ_x, σ_y) and angular spread ($\sigma_{x'}$, $\sigma_{y'}$)
  - magnetic lenses (quadrupoles) are the usual tools; BUT
  - other elements (accelerator tanks, dipoles, etc.) can also affect spot size and divergence

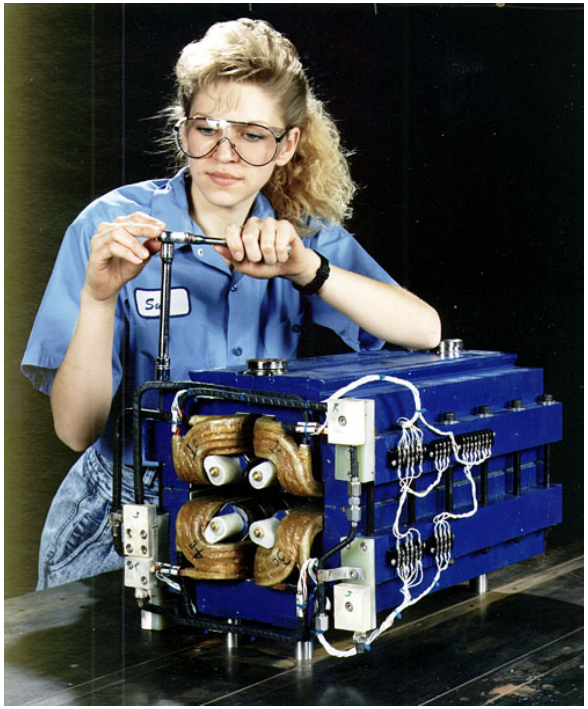
- bunch length / duration (σ_t)
  - adjust the beam source;
  - use dispersive elements combined with appropriate energy spread and t-p correlation

- energy (or momentum) spread (σ_δ or $\sigma_{\beta\gamma}$)
  - Almost every RF linac will have a correlation between bunch length (σ_t) and energy spread ($\sigma_{\beta\gamma}$): longer $\sigma_t \rightarrow$ larger $\sigma_{\beta\gamma}$
  - Adjust accelerating phase
  - Use harmonic linearizers to remove chirp, 2nd-order curvature terms
  - Use charge-dependent effects plus wakefields

Quadrupole Lens (Quad)







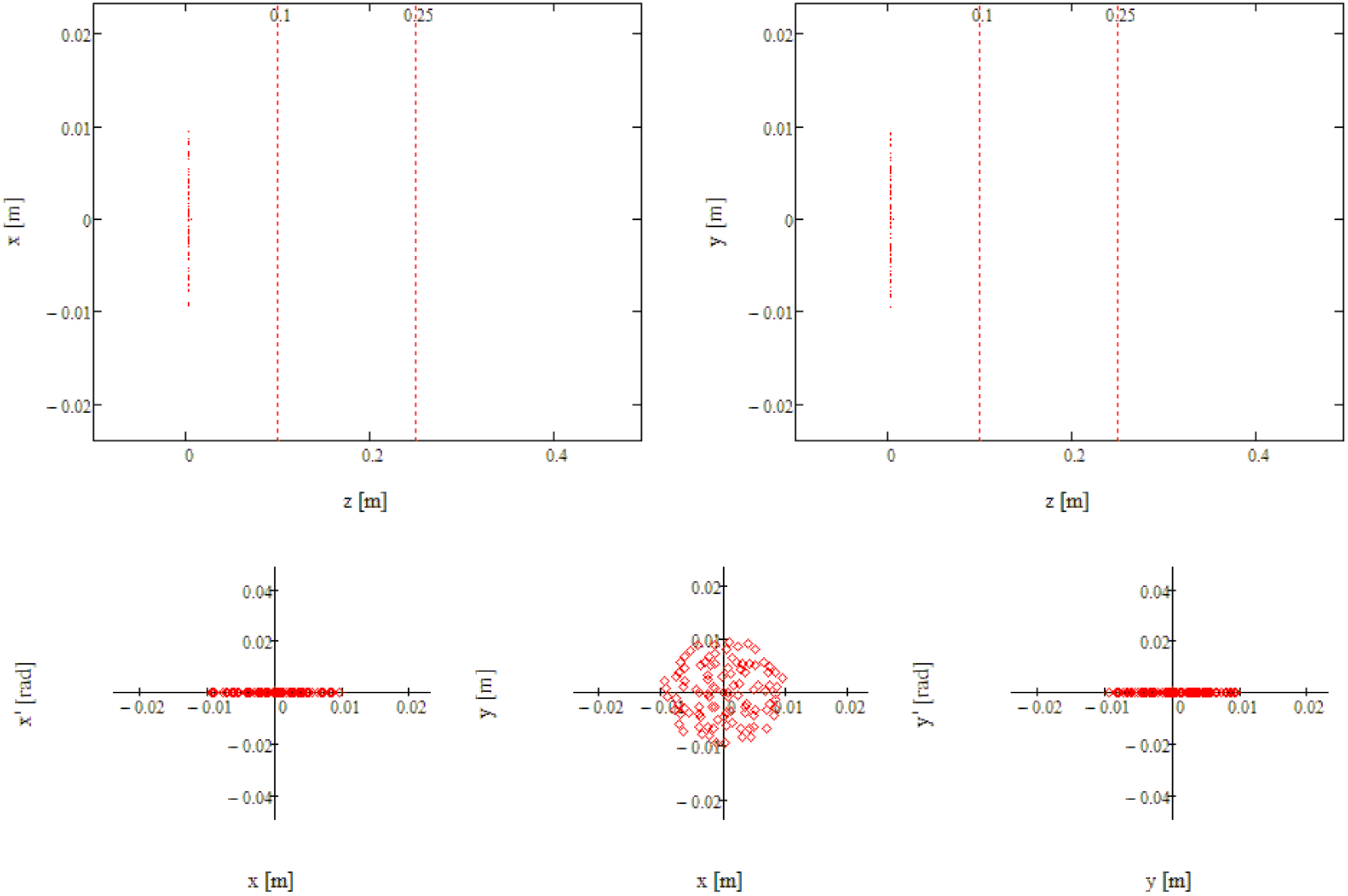
Quad properties

- Quads focus in one transverse plane, defocus in the other
- relative focusing strength depends on magnetic field gradient, length, beam energy

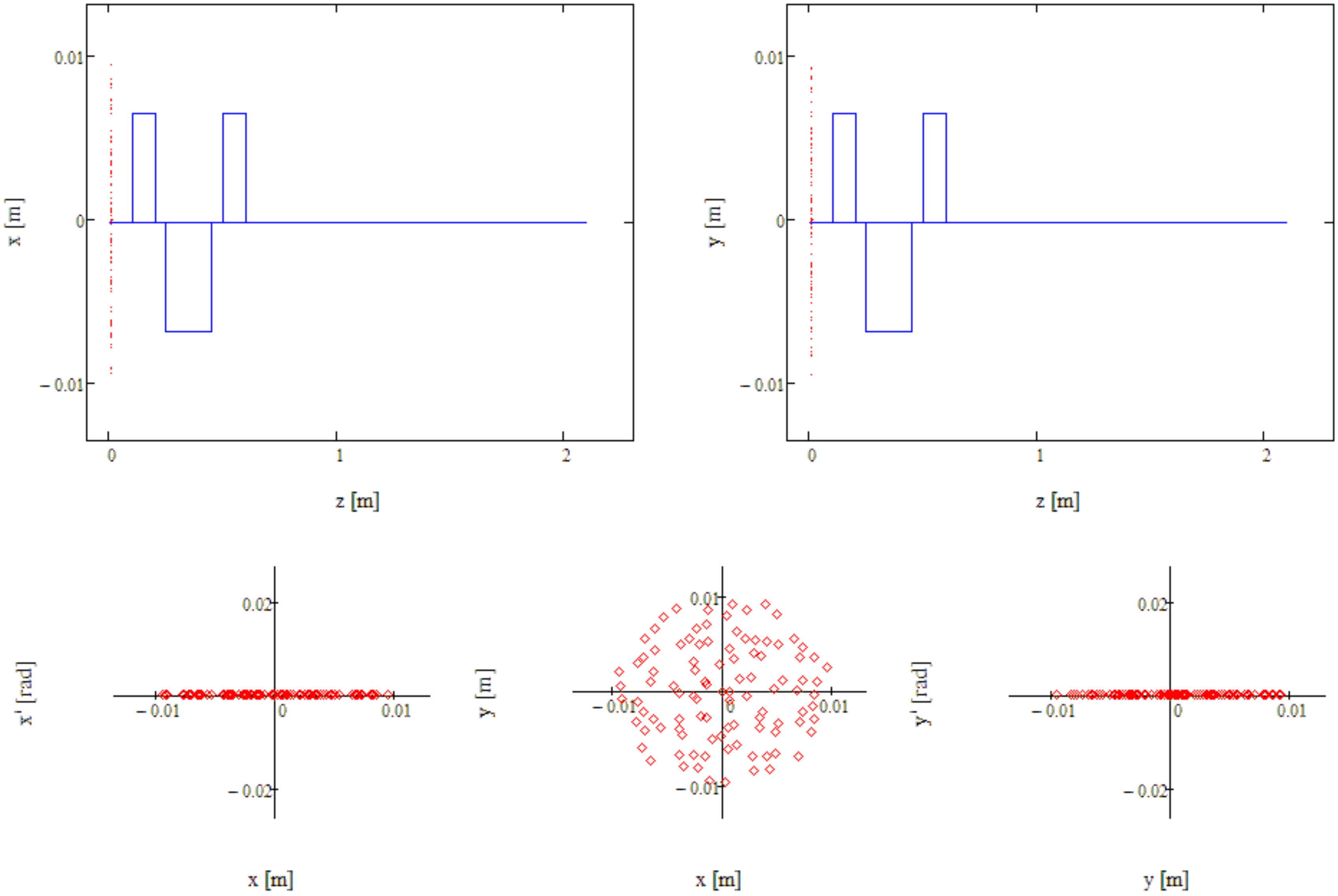
- Often-used (but confusing) convention:
  - an “F” quad focuses in the horizontal plane
  - a “D” quad focuses in the vertical plane
  - same piece of hardware, just powered in opposite polarities

- Other nomenclature
  - doublet: two quads, usually F-D or D-F of equal focusing strength
  - triplet: three quads, F-D-F, with the D having ~ 2x the focusing strength as the Fs

Quad focusing demo – one quad



Quad focusing demo - triplet



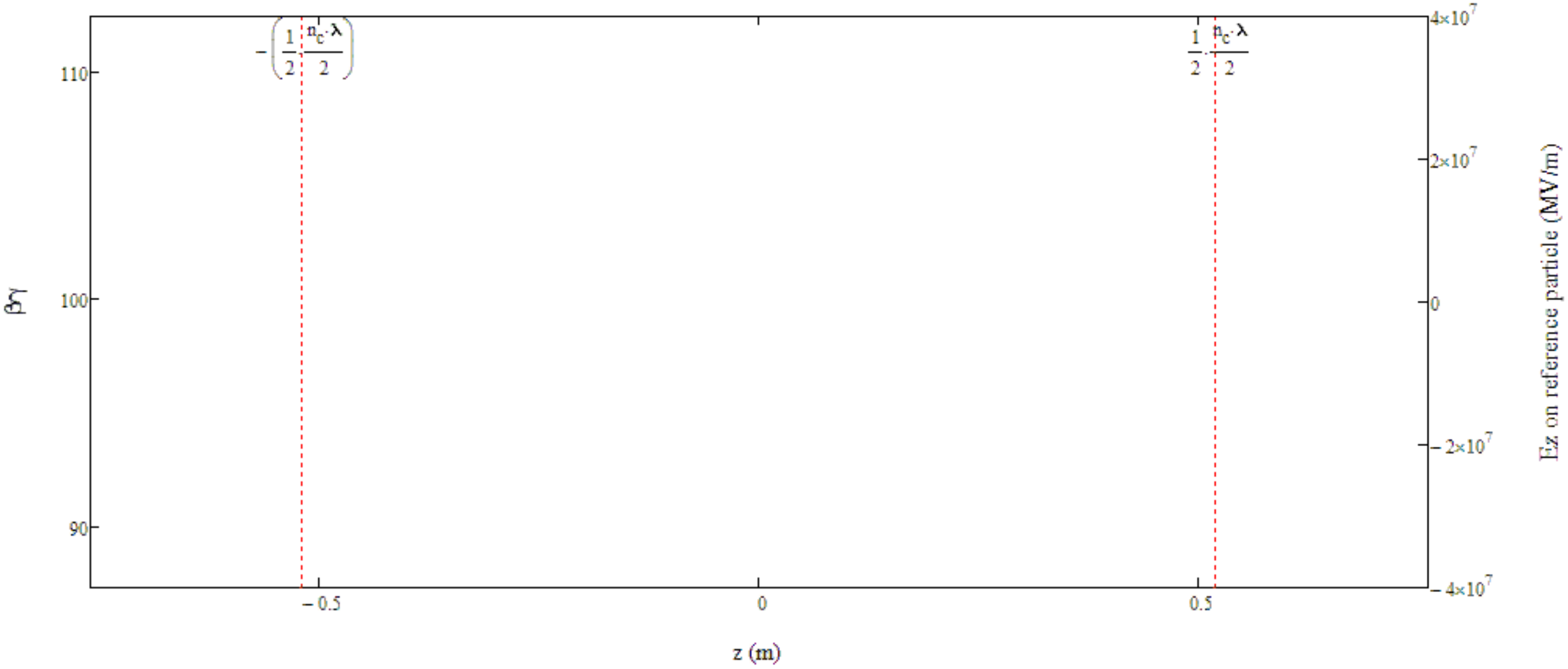
How do we adjust 2nd-order parameters?

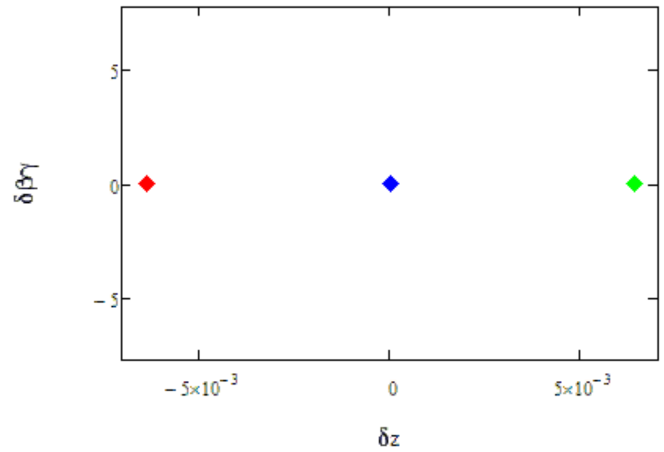
- Transverse correlations (divergences, e.g. $<x \cdot x'>$)
  - magnetic lenses (quadrupoles) are the usual tools; BUT
  - other elements (accelerator tanks, dipoles, etc.) can also affect this

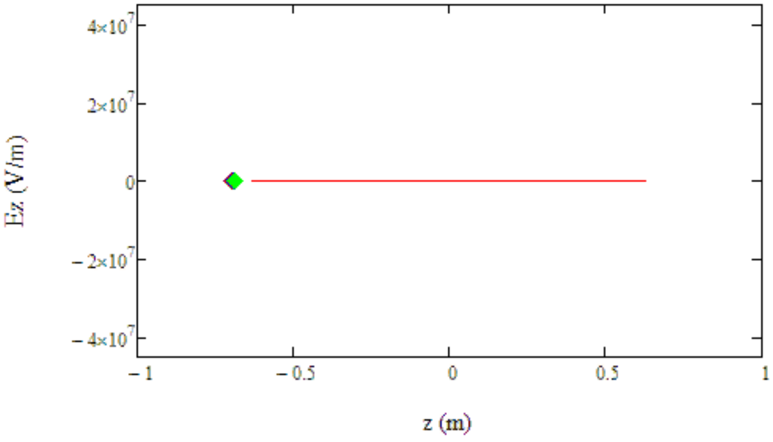
- Longitudinal correlations (energy chirp, e.g. $<t \cdot \beta\gamma>$)
  - adjust phase in accelerating structures;
  - use dispersive elements;
  - use harmonic RF structures to remove / modify higher-order terms

- transverse emittance
  - generally set at the source
  - "de-correlating" techniques such as emittance compensation, flat-beam transforms, can improve the emittance at or near the source
  - must take great care to not mess it up later, e.g. in bunch compressors

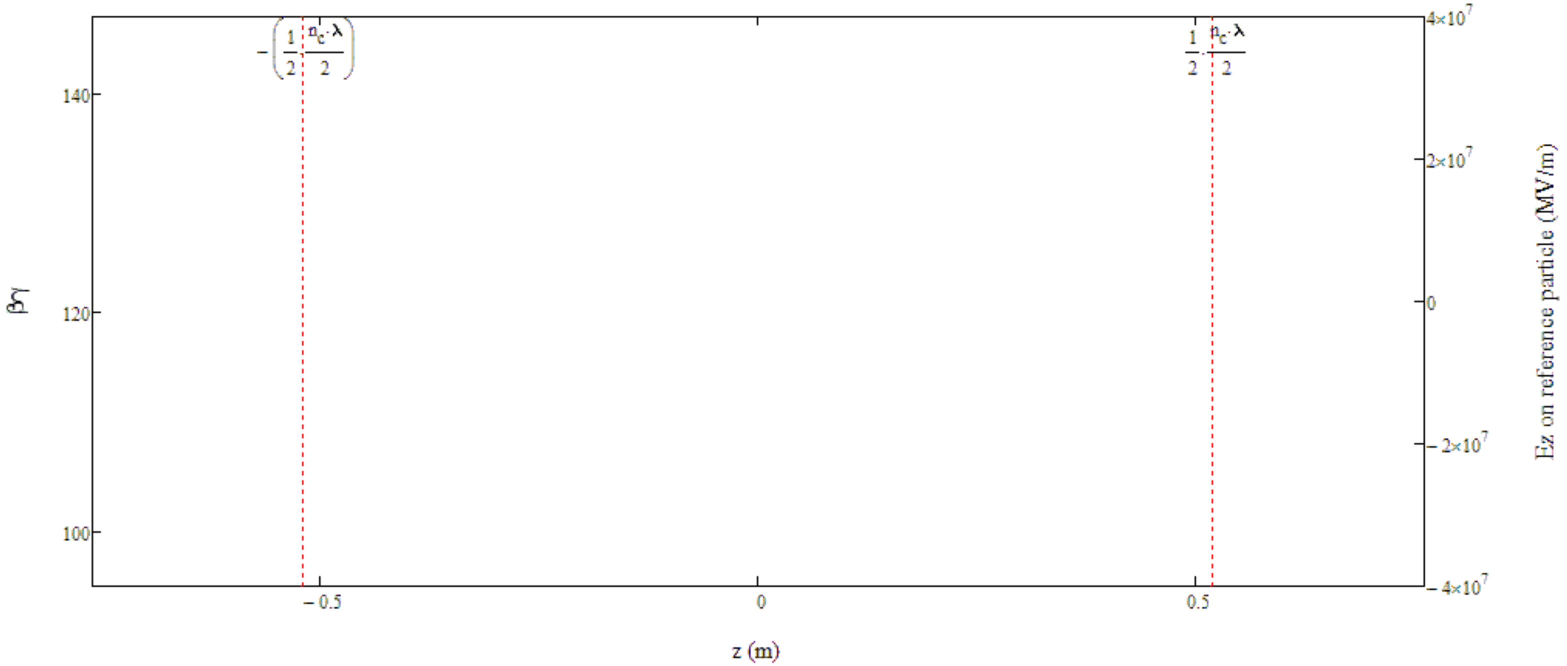
Energy Chirp – far off-crest

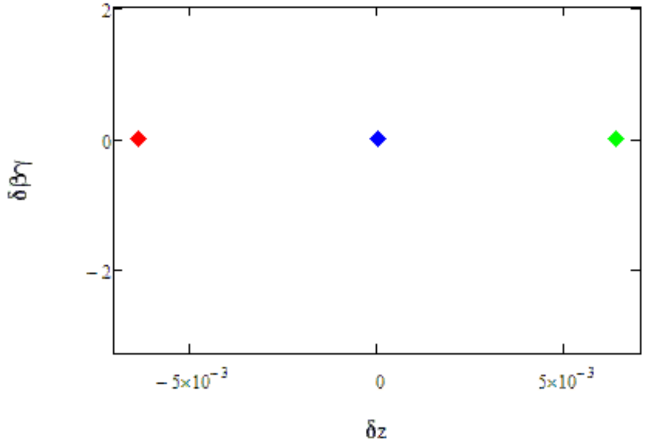


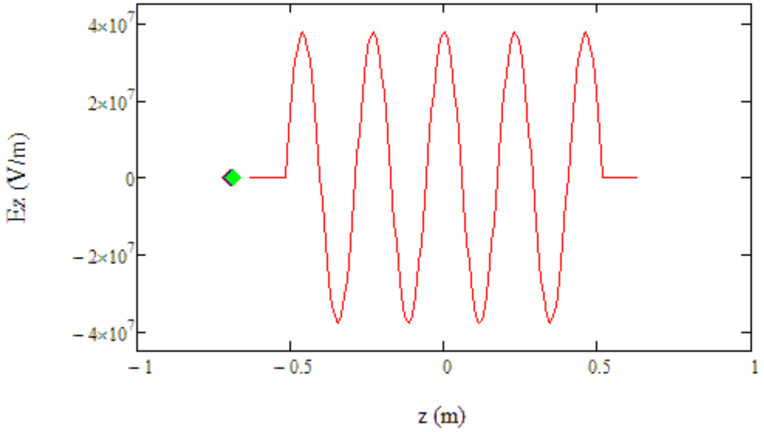




Energy Chirp – 20 degrees off-crest







How do we model the accelerator's effects on the beam?

- How do we simulate the process of transporting the beam from a source, through the accelerator, to our desired final state?

- What level of fidelity do we require?
  - What physics effects are included?
  - What's not there?
  - What do we not know (initially) we need?

How do we model the beam?

$$\nabla \cdot \vec{E} = \frac{\rho}{\varepsilon_o} \qquad \nabla \cdot \vec{B} = 0$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \qquad \nabla \times \vec{B} = \mu_o \left(\vec{J} + \varepsilon_o \frac{\partial \vec{E}}{\partial t} \right)$$

$$\frac{\partial \vec{p}}{\partial t} = q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

$$\frac{\partial \vec{r}}{\partial t} = \frac{c\vec{p}}{\sqrt{m^2c^2 + \left|\vec{p}\right|^2}}$$

And … we’re done, right?

Where do the fields come from?

$$\frac{\partial \vec{p}}{\partial t} = q\left(\vec{E} + \vec{v} \times \vec{B}\right)$$

Generally, E and B are, or can be,

- functions of both position and time;
- generated by sources
    - outside the beam (e.g. magnets, accelerating tanks);
    - generated by the beam itself (e.g. space charge); or
    - arise as a result of the structures and elements the beam traverses (e.g. wakefields, synchrotron radiation)

Approaches to Modeling

<u>Particle-in-Cell</u>

- Place a grid over the simulation space
- Find E, B on the grid points
  - external elements
  - fields from the beam
- Extrapolate and apply to the beam
- Integrate to advance the particle positions and momenta, fields

- Pros
  - somewhat intuitive
  - in principle, accurate to any desired order
  - does not rely on analytic description of the beam or elements of the accelerator

- Cons
  - tends to be rather slow
    - large number of grid points
    - small timesteps
  - hard to model an entire machine
  - practically, still needs analytic models for “external” fields
  - getting the physics right can be challenging

Approaches to Modeling

Matrix-Based

- Find single-particle solutions to the equations of motion through a element (e.g. drift space, bend, quad)
- Simplify equations to the desired order
- Define a matrix to solve the transport through that element
- Concatenate matrices for each element along the accelerator

- Describe the beam using a 1st-order matrix
- Then, multiply an individual particle vector or beam matrix to obtain the transport through the whole accelerator

- Pros
  - Based on analytic models of the accelerator
  - can be very fast
    - very amenable to optimization
    - can describe a multi-km accelerator with 36 numbers (to 1st order)
  - can handle “non-simple” beams by transporting particles, rather than the beam matrix

- Cons
  - Good for the order you select, no more
  - Multi-particle effects can be problematic to handle
  - If an element is not included in the library, it does not exist

Approaches to Modeling

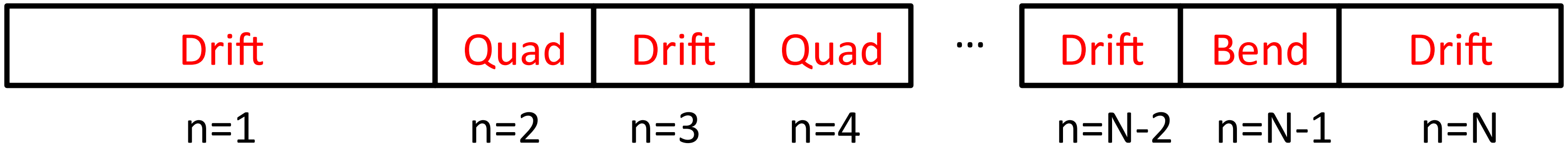
- Many codes are somewhere “in-between”
  - **elegant** can use matrix transport, but also has some numerically integrated elements
  - PARMELA, Astra, GPT are “particle pusher” codes: don’t include structure interactions directly, but
    - do incorporate Poisson solutions for working with space charge
    - do numerically integrate particle trajectories along the linac
    - may or may not include realistic “edge effects” on magnets, etc.

- There are many codes available!

- Critical: Know the limitations of the code you are using, whatever it is!

Matrix transport

- Paradigm
  - Accelerator model built from N discrete elements
  - Each element's effects on a particle (or the beam as a whole) can be described by a matrix
  - Effects of subsequent elements obtained by matrix multiplication



Drift Quad Drift Quad … Drift Bend Drift

n=1 n=2 n=3 n=4 n=N-2 n=N-1 n=N

Matrix Transport

1st-order matrix: the R-matrix

$$x_{n+1} = a_x x_n + b_x x'_n + c_x y_n + d_x y'_n + e_x t_n + f_x p_n$$

$$x'_{n+1} = a_{x'} x_n + b_{x'} x'_n + c_{x'} y_n + d_{x'} y'_n + e_{x'} t_n + f_{x'} p_n$$

$$\dots$$

$$\begin{pmatrix} x \\ x' \\ y \\ y' \\ t \\ p \end{pmatrix}_{n+1} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & R_{14} & R_{15} & R_{16} \\ R_{21} & R_{22} & R_{23} & R_{24} & R_{25} & R_{26} \\ R_{31} & R_{32} & R_{33} & R_{34} & R_{35} & R_{36} \\ R_{41} & R_{42} & R_{43} & R_{44} & R_{45} & R_{46} \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} & R_{56} \\ R_{61} & R_{62} & R_{63} & R_{64} & R_{65} & R_{66} \end{pmatrix} \cdot \begin{pmatrix} x \\ x' \\ y \\ y' \\ t \\ p \end{pmatrix}_{n}$$

Example: Drift Space

$$\begin{pmatrix} x \\ x' \\ y \\ y' \\ t \\ p \end{pmatrix}_{n+1} = \begin{pmatrix} 1 & L & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ x' \\ y \\ y' \\ t \\ p \end{pmatrix}_{n}$$

No change in momentum, angle, or arrival time (to 1st order)

$$x_{n+1}=x_n + L\, x'_n$$
$$y_{n+1}=y_n + L\, y'_n$$

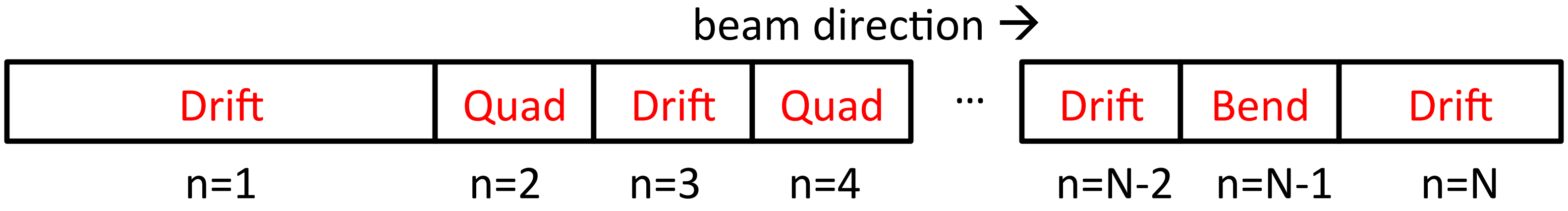
Example: Thin-Lens Quadrupole

$$\begin{pmatrix} x \\ x' \\ y \\ y' \\ t \\ p \end{pmatrix}_{n+1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{f} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ x' \\ y \\ y' \\ t \\ p \end{pmatrix}_{n}$$

f is the focal length of the lens
f depends on quad field gradient (T/m), length (m), and particle momentum

“thin” means particle transverse position change (x_{n+1}-x_n, y_{n+1}-y_n) is small.

Concatenate elements



$$\vec{x}_n = \underbrace{R_n \cdot R_{n-1} \cdot \cdots \cdot R_1}_{} \cdot \vec{x}_0$$

The results of this multiplication, 36 numbers, allows us to very rapidly "translate" a bunch from the start to the end of the linac (to 1st order in the starting coordinates)

elegant: *Ele*ctron *G*eneration *an*d *T*racking

- Has its roots as a matrix code, but has been extended considerably over the years

- Define a beam (by one of several means)
- Define the accelerator

- Track beam particles through the accelerator;
- Calculate the accelerator's R-matrix;
- Vary parameters to obtain desired beam parameters at the end of the accelerator;
- etc.

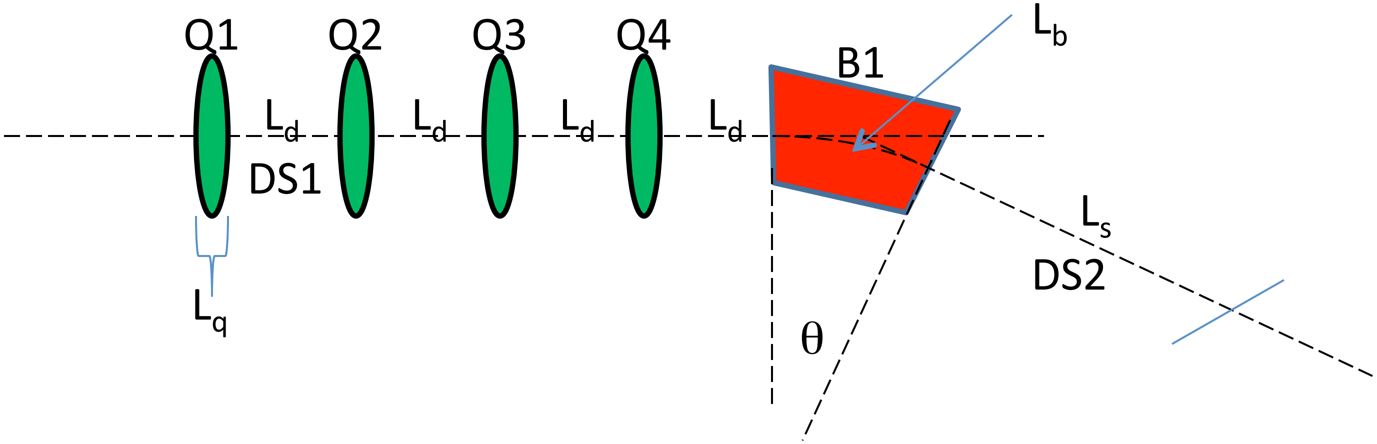
elegant

Particles represented as: $(x, x', y, y', t, p)_z$

Accelerator (aka lattice) represented as:
- Various individual elements (quads, drifts, etc.)
- Concatenated together to build up a beamline

Program execution controlled by a command file:
- defines the beam
- defines tasks to be performed (e.g. optimization)
- specifies the output to be generated
- based on a namelist formalism

Simple lattice file example



```
Q1:  quad, L=0.1, K1=2
Q2:  quad, L=0.1, K1=-4
Q3:  quad, L=0.1, K1=2
Q4:  quad, L=0.1, K1=0

DS1: drift, L=0.4
DS2: drift, L=1.0

B1: sbend, L=0.6, angle=“pi 6 /”

test: line=(DS1, Q1, DS1, Q2, DS1, Q3, DS1, Q4, DS1, B1, DS2)
```

Simple Command File Example

```
&run_setup
	lattice = M_01.lte
	default_order = 2
	use_beamline = test
	p_central = 200
	sigma = %s.sig
	centroid = %s.cen
	output = %s.out
	final = %s.fin
	magnets = %s.mag
	parameters = %s.param
	random_number_seed = 987654321
	print_statistics = 1
&end

&run_control
	n_steps = 1
&end

&bunched_beam
	bunch = %s.bun
	n_particles_per_bunch = 1000
	emit_nx = 1e-6
	emit_ny = 1e-6
	Po = 200
	sigma_dp = 0.
	beta_x = 200
	beta_y = 200
&end

&track &end
```

which lattice file to use

which beamline in the lattice file

initial momentum (βγ) the line is set up for

names of output files to generate
%s = (name of command file)
%s.sig → M_01.sig in this case

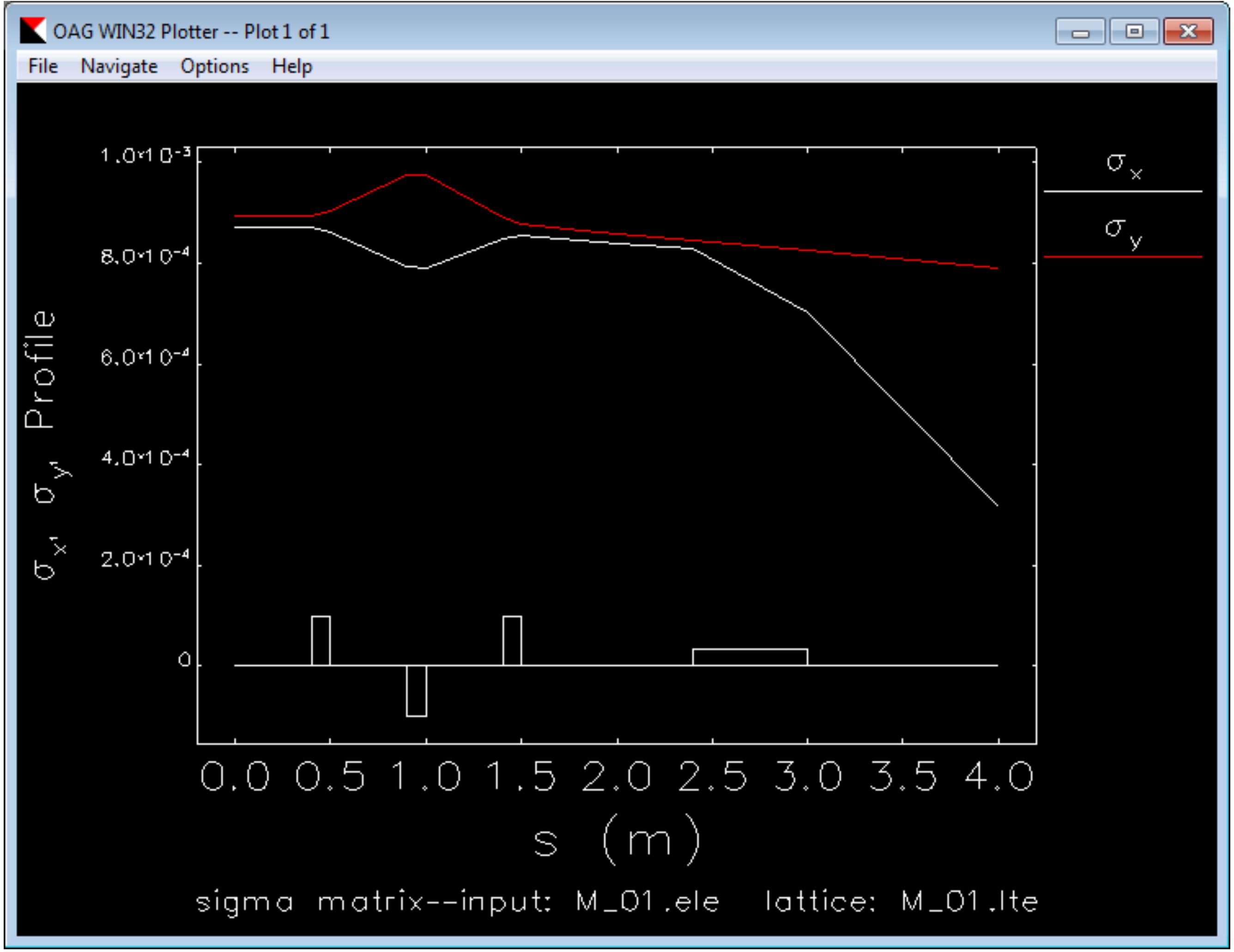
Describe the beam to be run through the line

Save the beam’s coordinates

Define (some of the) beam properties;
ones not specified go to their default values

Examine some of the output

```
sddsplot -col=s,"(Sx,Sy)" -legend -grap=line,vary M_01.sig
    -col=s,Profile M_01.mag -factor=ym=1e-4
```



Work with the example files

Directory: USPAS\2014-June\day_1\lattice_example

M_01.ele: elegant command file
M_01.lte: elegant lattice file

run_m_01.bat: batch file to

- run **elegant**
- generate a plot of rms beam size vs. distance along the line
- generate various phase-space plots at the start and end of the line
- report parameters at the end of the line

Try modifying

- quad strengths & lengths, bend angle, etc. in the lattice file
- starting beam parameters (esp. energy spread) in the command file

A couple of notes… Windows shell

- The Windows command shell uses the caret ^ to continue from one line to another

- example: the shell will treat this as one line:

```
sddsplot -col=t,p -grap=dot ^
          test.out
```

A couple of notes … lattice file

- **elegant** uses the ampersand & to continue lines in the lattice file
- Syntax is ***extremely*** important; in particular, continuation lines do not remove the need for punctuation (e.g. commas). Example:

```
Dq: quad, L=0.1, K1=2.9
```

Defining a quad on one line

```
Dq: quad, L=0.1, &
    K1=2.9
```

Defining a quad on two lines, correctly

```
Dq: quad, L=0.1 &
    K1=2.9
```

Defining a quad on two lines, ***incorrectly.***

Because the comma is missing, the **elegant** parser will not recognize that you're defining K1 and it will ***default to zero*** (or whatever the default value is)

Tuesday

Outline Tuesday

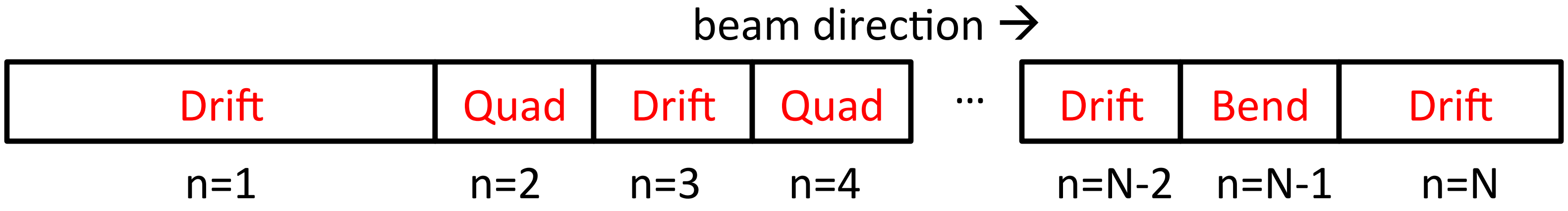
- Matrix transform
  - R-matrix review
  - Beam matrix
  - Twiss / Courant-Snyder parameters
  - Capabilities & limitations
  - Mixed-mode operation in **elegant**

- Acceleration
  - standing-wave vs. traveling-wave linacs
    - pros/cons
    - implications for modeling design
  - various models in **elegant**

- Emittance damping
  - normalized vs. unnormalized emittance

- Lab: exploring linac models

Concatenate elements



$$\vec{x}_n = \underbrace{R_n \cdot R_{n-1} \cdot \cdots \cdot R_1} \cdot \vec{x}_0$$

The results of this multiplication, 36 numbers, allows us to very rapidly “translate” a bunch from the start to the end of the linac (to 1st order in the starting coordinates)

Single-particle transform

$$\vec{x}_N = \mathbf{R_{0N}} \cdot \vec{x}_0$$

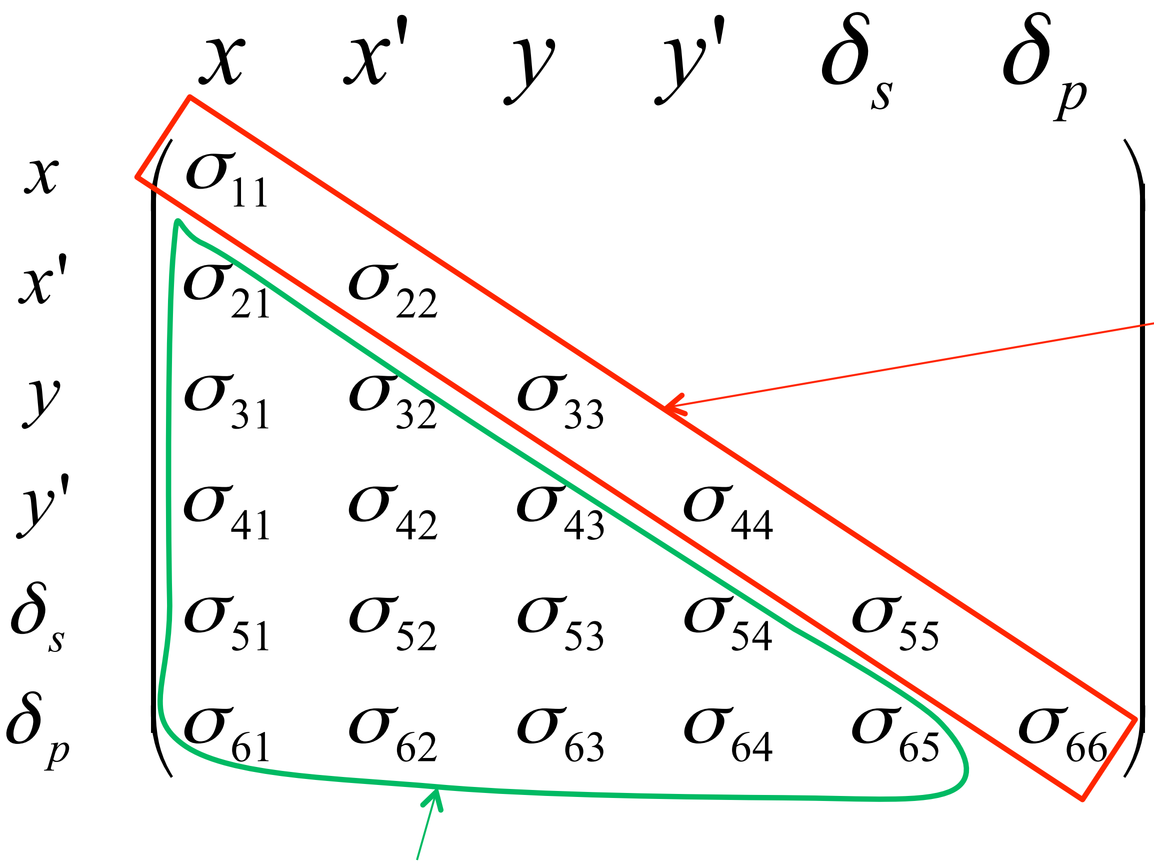
Transforms a *single particle* (to 1st order) from start to end.

We can do better: transfer the beam as a whole.

Beam Matrix Approach

- The beam exists in 6-d phase space.
- The phase space (in an RMS sense) can be described by
  - the RMS size along each of the 6 axes, and
  - correlations between pairs of axes

Beam Matrix / Sigma Matrix



$$\begin{array}{c|cccccc} & x & x' & y & y' & \delta_s & \delta_p \\ \hline x & \sigma_{11} & & & & & \\ x' & \sigma_{21} & \sigma_{22} & & & & \\ y & \sigma_{31} & \sigma_{32} & \sigma_{33} & & & \\ y' & \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} & & \\ \delta_s & \sigma_{51} & \sigma_{52} & \sigma_{53} & \sigma_{54} & \sigma_{55} & \\ \delta_p & \sigma_{61} & \sigma_{62} & \sigma_{63} & \sigma_{64} & \sigma_{65} & \sigma_{66} \end{array}$$

squares of the RMS sizes along each axis

Correlation terms between pairs of axes

Can be useful to look at these to identify areas of concern at the end of the linac (e.g. transverse-longitudinal correlations)

Why bother?

$$\Sigma_N = \mathbf{R_{0N}} \Sigma_0 \mathbf{R_{0N}}^t$$

So, not only can we transform a single particle (to 1st order) from start to end, we can translate a 1st-order description of the whole beam from start to end in a single step.

Detour: Twiss (or Courant-Snyder) Parameters

$$\sigma_x = \sqrt{\beta_x \varepsilon_x}$$

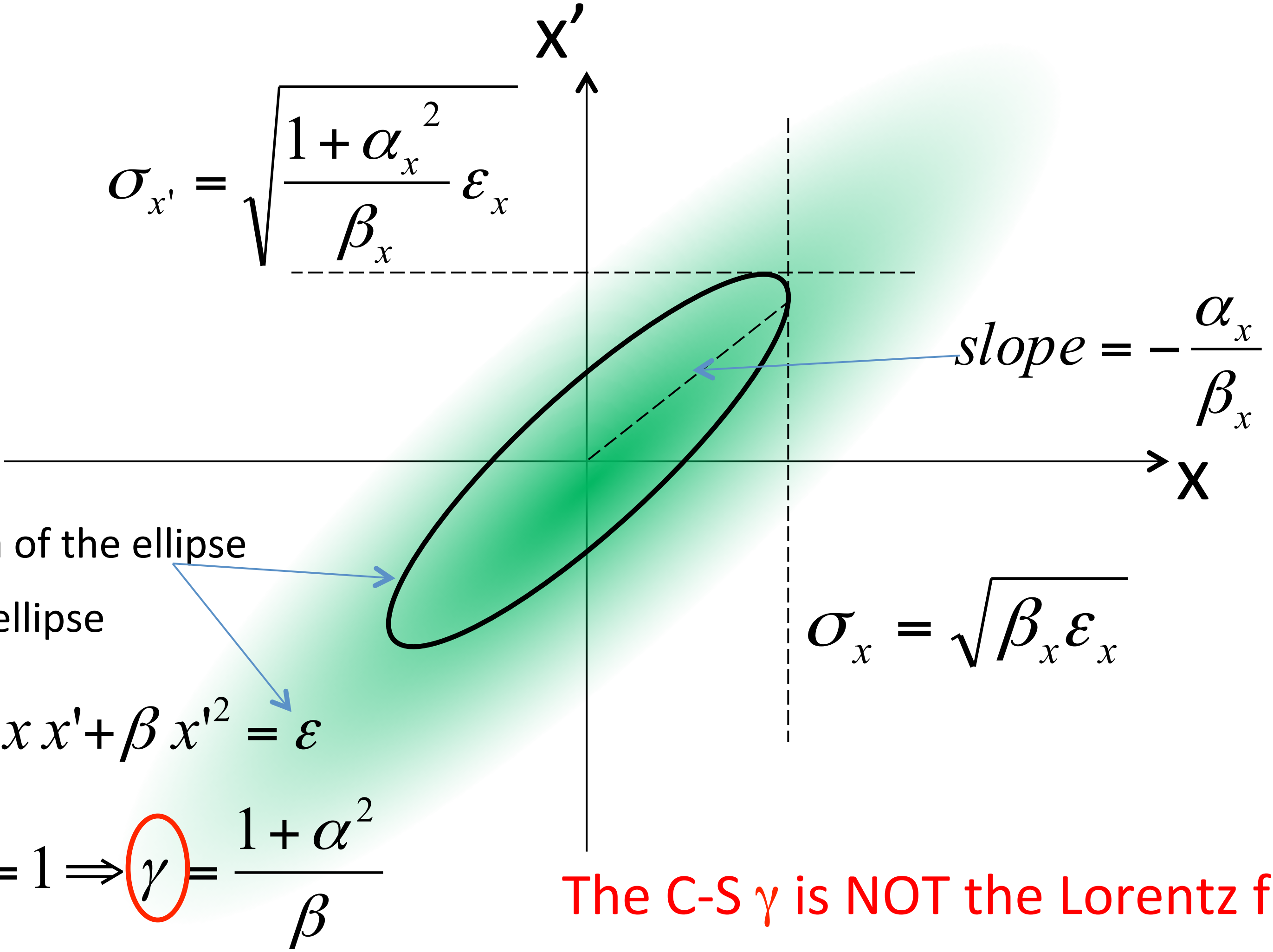
β_x = normalized beam size

$$\sigma_{x'} = \sqrt{\frac{1+\alpha_x{}^2}{\beta_x}\varepsilon_x}$$

α_x = normalized beam divergence

In a dispersion-free region (e.g. no correlation between energy and position or angle due to the accelerator)

Detour: Twiss (or Courant-Snyder) Parameters



$$\sigma_{x'} = \sqrt{\frac{1+\alpha_x{}^2}{\beta_x}\varepsilon_x}$$

$$slope = -\frac{\alpha_x}{\beta_x}$$

area of the ellipse

equation of an ellipse

$$\sigma_x = \sqrt{\beta_x \varepsilon_x}$$

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

$$\gamma\,\beta - \alpha^2 = 1 \Rightarrow \boxed{\gamma} = \frac{1+\alpha^2}{\beta}$$

The C-S γ is NOT the Lorentz factor!!

Mixed-mode operation

- Charge-independent: component ==> beam
  - matrix approach works fairly well
  - many (most?) **elegant** elements work in this fashion

- Charge-dependent: beam ==> component ==> beam
  - examples: wakefields, CSR
  - matrix approach doesn't work well at all
  - depending on the interaction, one can have **elegant**
    - add in charge-dependent effects "after" elements (e.g. wakefields)
    - integrate through elements (e.g. CSR)
  - generally not self-consistent
  - will *not* be included when calculating the R-matrix for the machine

- Charge-dependent: beam <==> beam
  - example: space charge
  - *very* limited support in **elegant** (e.g. longitudinal space charge)

Acceleration

Normal-conducting traveling-wave

- Example: SLAC 3-m S-band linac section
- Focusing effects minimal when beam || TW power flow
- Wakefields
  - strong short-range (small apertures)
  - relatively weak long-range (lossy structures)

Superconducting standing wave

- Example: 9-cell TESLA cavity
- Moderate focusing effects regardless of beam direction
- Wakefields
  - relatively weak short-range (large apertures)
  - long-range wakes (aka higher-order modes) typically require damping

Example 1: acceleration

Directory: USPAS\2014-June\day_2\acc_1

T_01.ele: elegant command file
T_01.lte: elegant lattice file

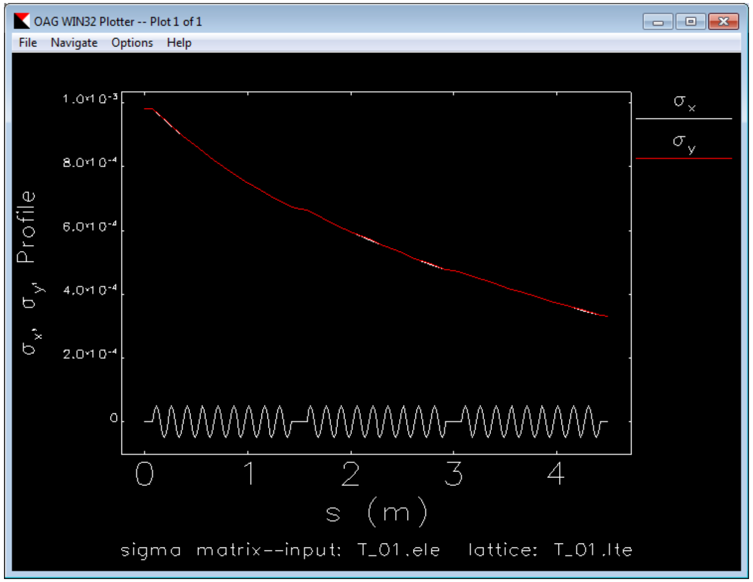
run_t_01.bat: batch file to

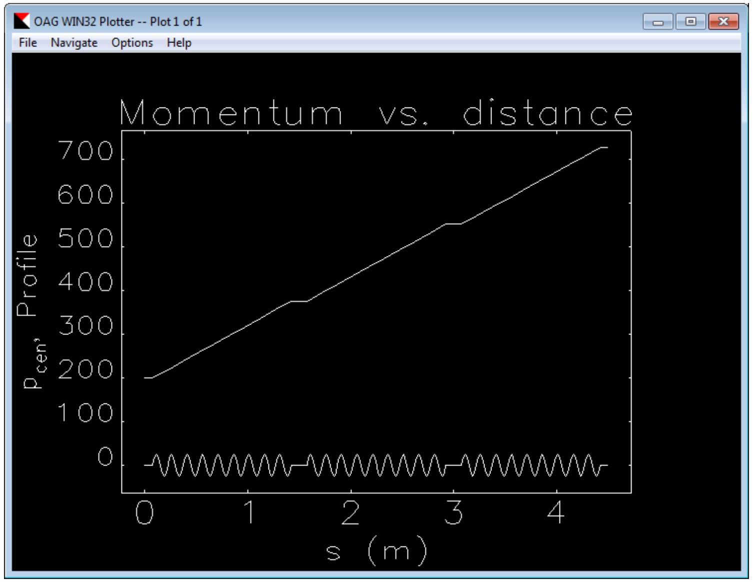
- run **elegant**
- generate a plot of rms beam size vs. distance along the line
- generate longitudinal phase-space plots at the start and end of the line
- plot mean momentum and momentum spread vs. distance along the line

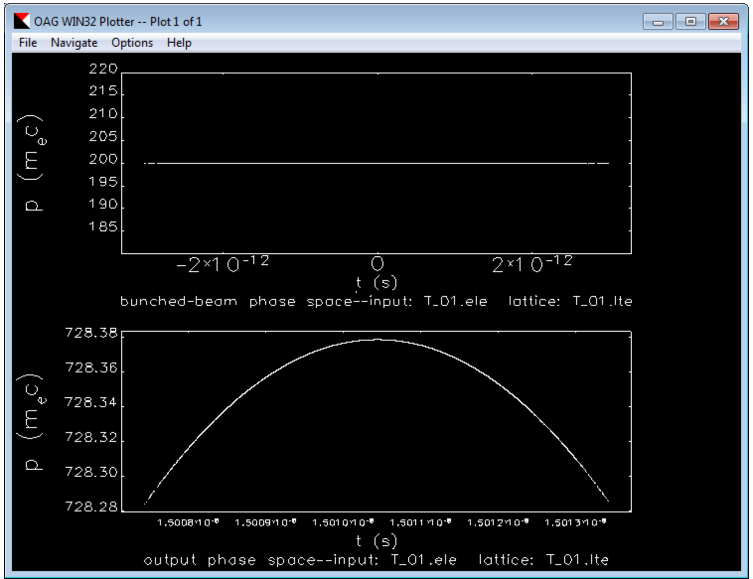
Try adjusting:

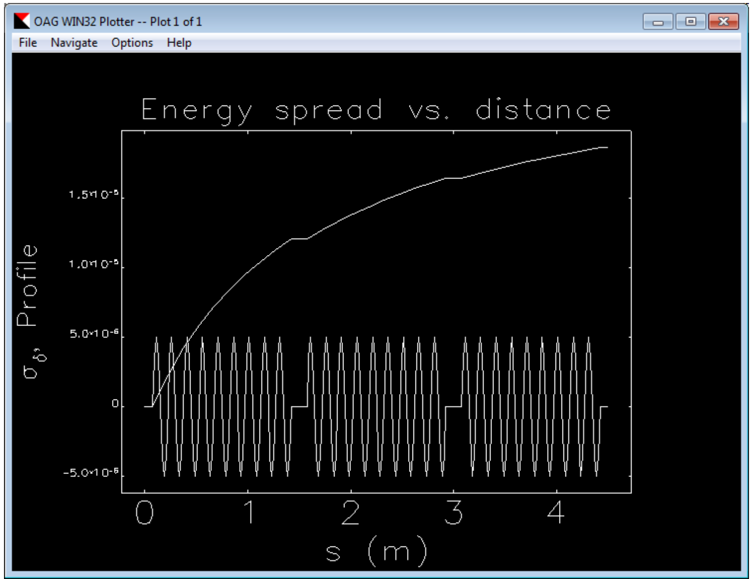
- the voltage and phase of the struct_cell element
- the beam starting aspect ratio via the initial beta functions
- the beam's starting bunch length
- whether the model uses end-of-cell focusing or not

Example 1: acceleration









Use SDDS Toolkit to Process Results

- The T_01.cen file provides the beam normalized *momentum* (βγ) as a function of distance
- We specify the linac tank voltage in MV
- Let’s calculate beam kinetic energy from βγ and replot
- The equation we want to use is:

$$KE = mc^2\left(\sqrt{(\beta\gamma)^2+1}-1\right)$$

sddsquery: what's in a file?

```
C:\AOT-HPE\demos\uspas\2014-June\day_2\acc_1>sddsquery T_01.cen

file T_01.cen is in SDDS protocol version 1
description: centroid output--input: T_01.ele  lattice: T_01.lte
contents: centroid output
data is little-endian binary

12 columns of data:
NAME            UNITS           SYMBOL          FORMAT          TYPE    FIELD  DESCRIPTION
                                                                        LENGTH
s               m               NULL            NULL            double  0       Distance
ElementName     NULL            NULL            %10s            string  0       Element name
ElementOccurence NULL            NULL            %6ld            long    0       Occurence of element
ElementType     NULL            NULL            %10s            string  0       Element-type name
Cx              m               <x>             NULL            double  0       x centroid
Cxp             NULL            <x'>            NULL            double  0       x' centroid
Cy              m               <y>             NULL            double  0       y centroid
Cyp             NULL            <y'>            NULL            double  0       y' centroid
Cs              m               <s>             NULL            double  0       mean distance traveled
Cdelta          NULL            <$gd$r>         NULL            double  0       delta centroid
Particles       NULL            NULL            NULL            long    0       Number of particles
pCentral        m$be$nc         p$bcen$n        NULL            double  0       Reference beta*gamma

1 parameters:
NAME                UNITS               SYMBOL              TYPE                DESCRIPTION
Step                NULL                NULL                long                Simulation step
```

sddsprocess: do math on a file

- We see that pCentral ($\beta\gamma$) is a column of data
- We want to make a new column, KE
- sddsprocess uses reverse Polish notation, so:

```
sddsprocess -define=col,KE,"pCentral 2 pow 1 + sqrt 1 - 0.511 *" T_01.cen
```

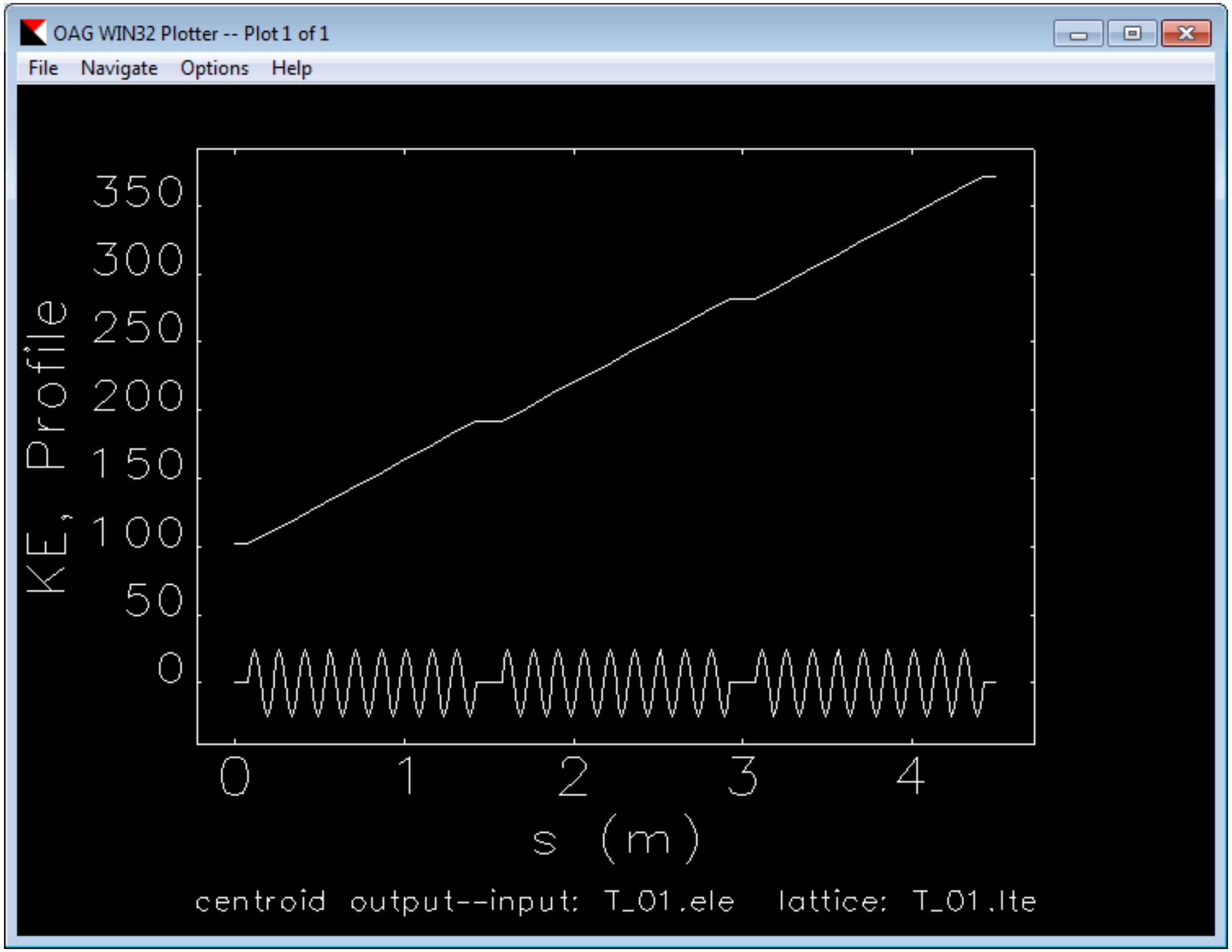
Defines a new column named KE, based on the equation from last slide, to convert $\beta\gamma$ to kinetic energy (for an electron). It does this on the file T_01.cen.

Result:

```
C:\acc_1>sddsprocess -define=col,KE,"pCentral 2 pow 1 + sqrt 1 - 0.511 *" T_01.cen
warning: existing file T_01.cen will be replaced (sddsprocess)

C:\acc_1>sddsquery T_01.cen

file T_01.cen is in SDDS protocol version 1
description: centroid output--input: T_01.ele  lattice: T_01.lte
contents: centroid output
data is little-endian binary

13 columns of data:
NAME            UNITS           SYMBOL          FORMAT          TYPE    FIELD  DESCRIPTION
                                                                        LENGTH
s               m               NULL            NULL            double  0       Distance
ElementName     NULL            NULL            %10s            string  0       Element name
ElementOccurence NULL            NULL            %6ld            long    0       Occurence of element
ElementType     NULL            NULL            %10s            string  0       Element-type name
Cx              m               <x>             NULL            double  0       x centroid
Cxp             NULL            <x'>            NULL            double  0       x' centroid
Cy              m               <y>             NULL            double  0       y centroid
Cyp             NULL            <y'>            NULL            double  0       y' centroid
Cs              m               <s>             NULL            double  0       mean distance traveled
Cdelta          NULL            <$gd$r>         NULL            double  0       delta centroid
Particles       NULL            NULL            NULL            long    0       Number of particles
pCentral        m$be$nc         p$bcen$n        NULL            double  0       Reference beta*gamma
KE              NULL            NULL            NULL            double  0       NULL

1 parameters:
NAME                UNITS               SYMBOL              TYPE                DESCRIPTION
Step                NULL                NULL                long                Simulation step
```

What should have happened?

```
sddsplot -col=s,KE t_01.cen -col=s,Profile T_01.mag -factor=ym=50
```



Did we get the energy gain we expect?

- We started with:
  - a peak voltage per cell of 10 MV;
  - 9 cells per structure; and
  - 3 structures; so the beam should have gained
  - (10 MV)(3)(9) = **270 MeV**

- Let’s check that using sddsprintout

sddsprintout

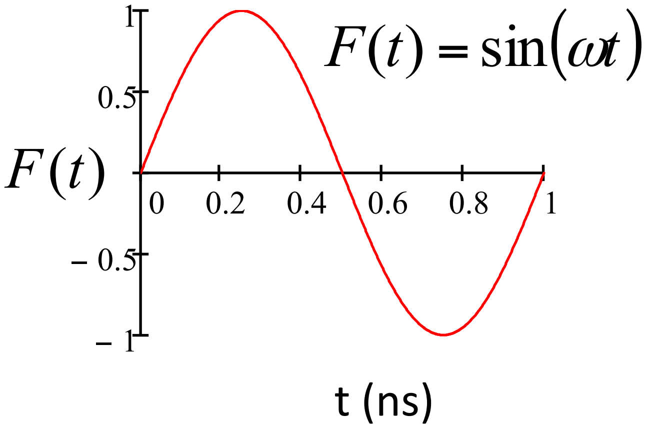
```
C:\day_2\acc_1>sddsprintout -col=s -col=ElementType -col=pCentral -col=KE T_01.cen
Printout for SDDS file T_01.cen

      s         ElementType      pCentral            KE
      m                          m$be$nc
-----------------------------------------------------------------
 0.000000e+000         MARK   2.000000e+002  1.016903e+002
 7.500000e-002         DRIF   2.000000e+002  1.016903e+002
 2.250000e-001         RFCA   2.195694e+002  1.116901e+002
 3.750000e-001         RFCA   2.391387e+002  1.216899e+002
 5.250000e-001         RFCA   2.587080e+002  1.316898e+002
 6.750000e-001         RFCA   2.782772e+002  1.416896e+002
 8.250000e-001         RFCA   2.978465e+002  1.516894e+002
 9.750000e-001         RFCA   3.174157e+002  1.616892e+002
 1.125000e+000         RFCA   3.369850e+002  1.716891e+002
 1.275000e+000         RFCA   3.565542e+002  1.816889e+002
 1.425000e+000         RFCA   3.761234e+002  1.916887e+002

 (Some lines deleted for readability on the screen)

 3.000000e+000         DRIF   5.522460e+002  2.816872e+002
 3.075000e+000         DRIF   5.522460e+002  2.816872e+002
 3.225000e+000         RFCA   5.718152e+002  2.916870e+002
 3.375000e+000         RFCA   5.913843e+002  3.016868e+002
 3.525000e+000         RFCA   6.109535e+002  3.116867e+002
 3.675000e+000         RFCA   6.305227e+002  3.216865e+002
 3.825000e+000         RFCA   6.500918e+002  3.316863e+002
 3.975000e+000         RFCA   6.696610e+002  3.416861e+002
 4.125000e+000         RFCA   6.892301e+002  3.516860e+002
 4.275000e+000         RFCA   7.087993e+002  3.616858e+002
 4.425000e+000         RFCA   7.283685e+002  3.716856e+002
 4.500000e+000         DRIF   7.283685e+002  3.716856e+002
```

elegant automatically adds a “marker” element at the start of the selected beamline, so we can see the beam properties at the start of the first element

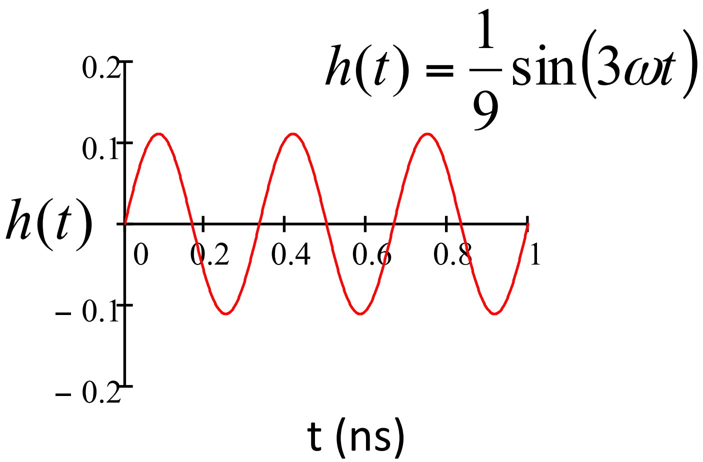
ΔKE = 269.9953, not 270

So .. why?

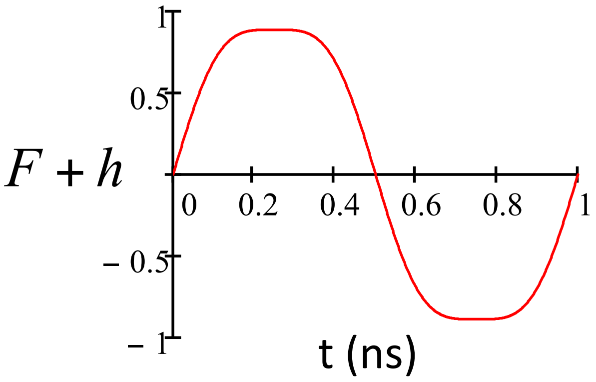
Adding Multiple Frequencies

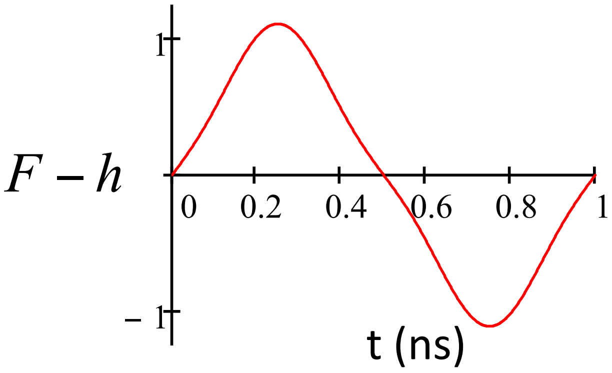


$$F(t) = \sin(\omega t)$$



$$h(t) = \frac{1}{9}\sin(3\omega t)$$





Example 2: harmonics

Directory: USPAS\2014-June\day_2\acc_2

T_02.ele: elegant command file
T_02.lte: elegant lattice file

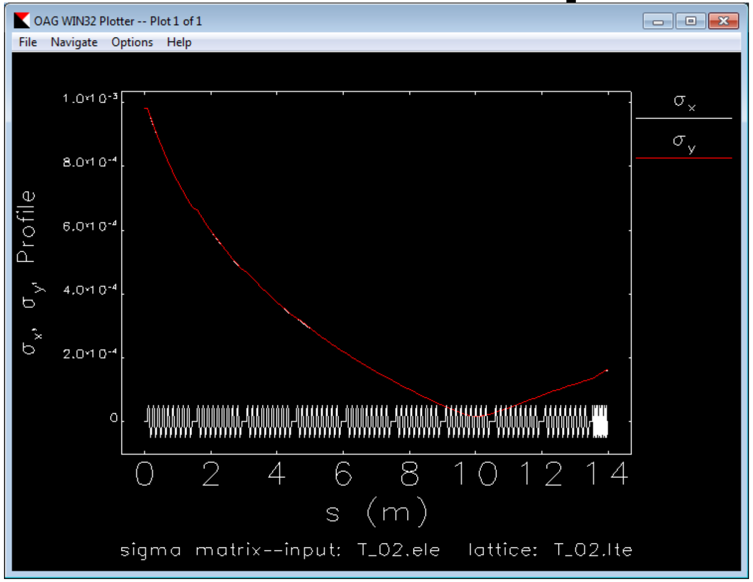
run_t_02.bat: batch file to

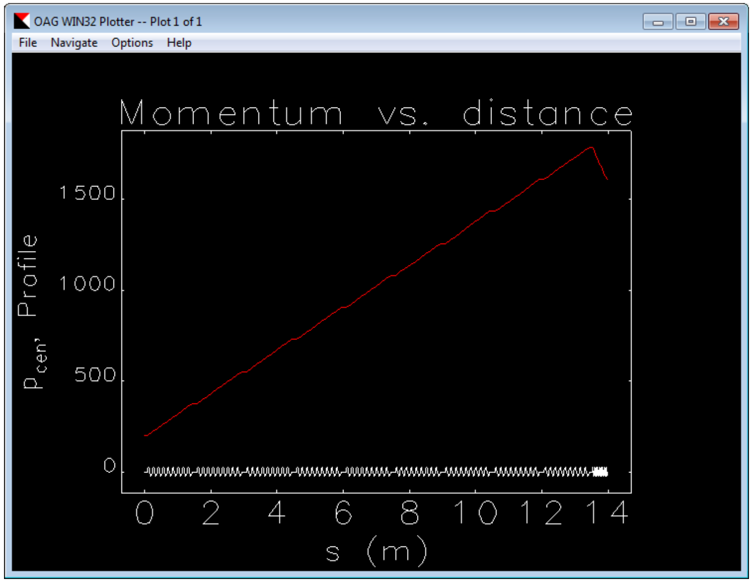
- run **elegant**
- generate a plot of rms beam size vs. distance along the line
- generate longitudinal phase-space plots at the start and end of the line
- plot mean momentum and momentum spread vs. distance along the line

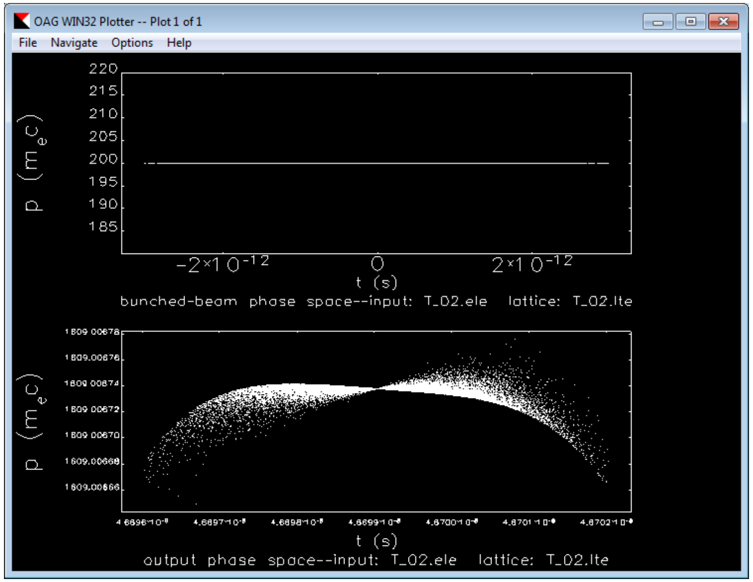
Try varying:

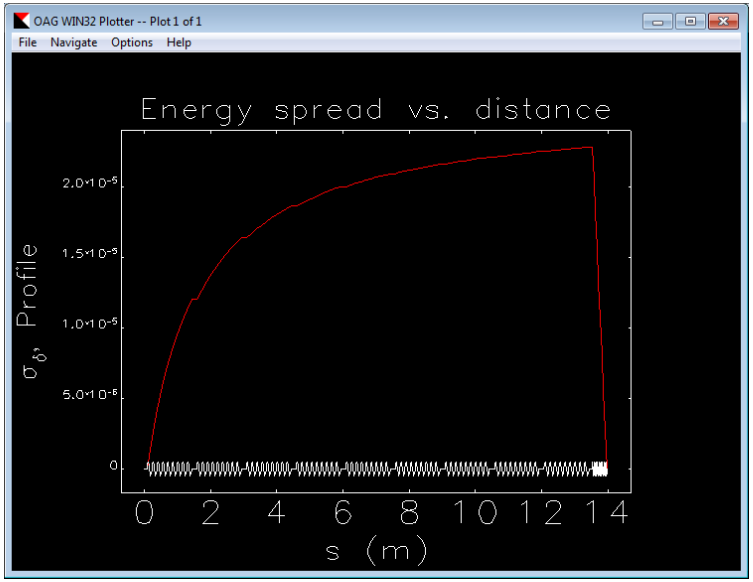
- the voltage and phase of the F_cell and h_cell elements (or how many of each there are)
- the harmonic number of the h_cell element. How well do the 4th and 2nd harmonics work?
- the beam's starting bunch length
- the order of the F_cell and h_cell structures
- whether the models for F_ and h_cells use end-of-cell focusing or not

Example 2: harmonics









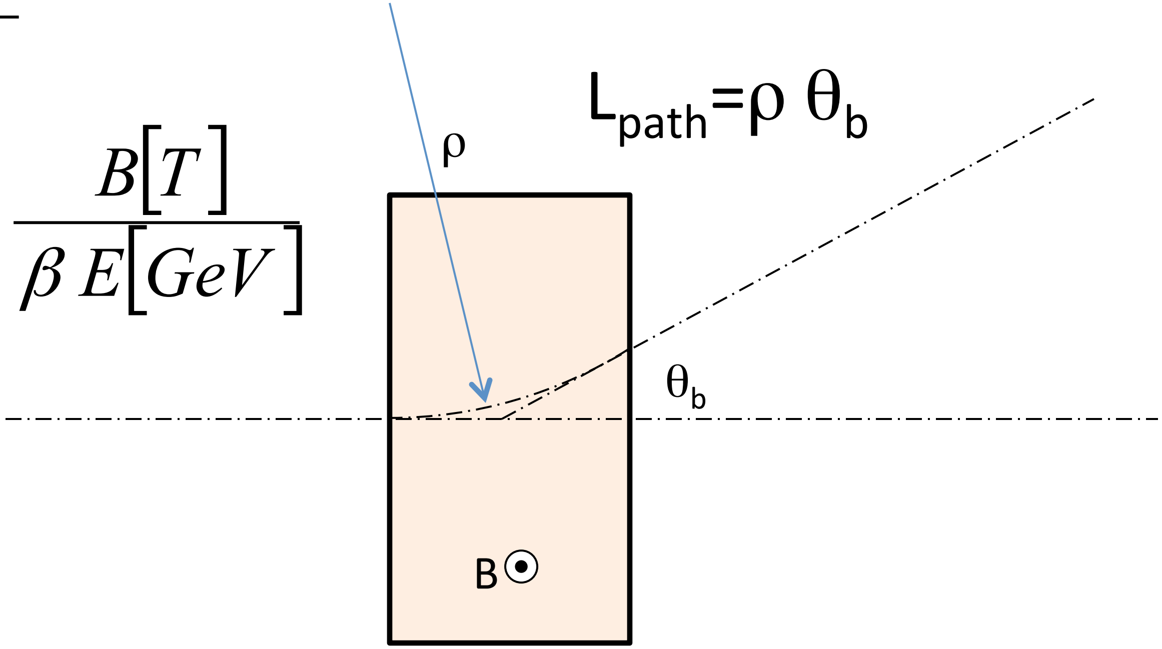
Wednesday

Bend Magnet Basics

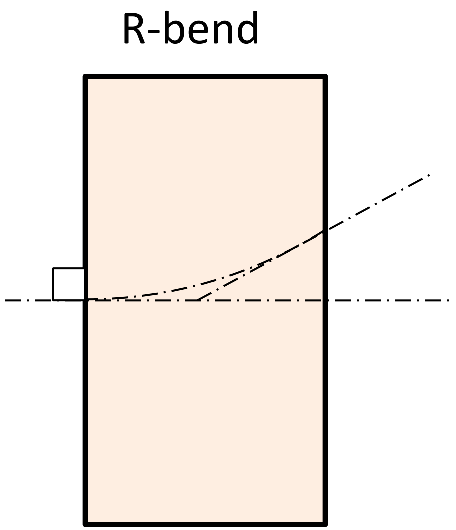
Beam through a bend – bend angle proportional to beam energy

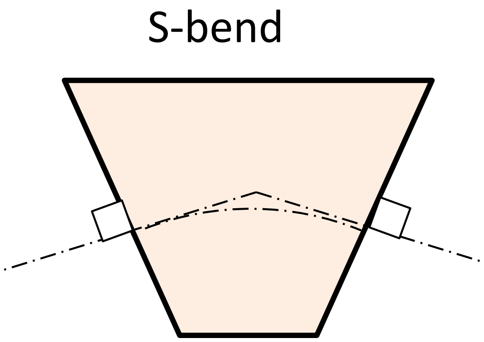
$$B\rho = \frac{\beta\gamma m c}{q}$$

$$\frac{1}{\rho}[m] = 0.2998\,\frac{B[T]}{\beta E[GeV]}$$



Bend Magnet Basics





Bend magnet with rectangular poles

- Entrance and exit pole faces are parallel
- Typically used for bunch compressors
- Some focusing in the non-bend plane
- Horizontally offset particles have the same path length through the magnet

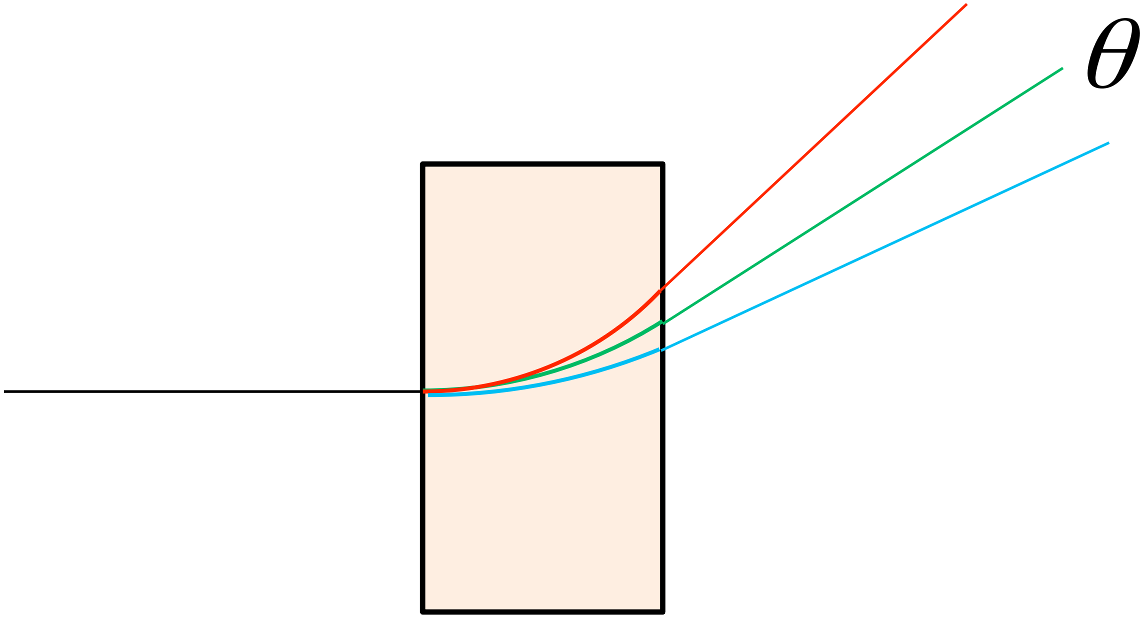
Bend magnet with sector poles

- Nominal beam path perpendicular to entrance and exit pole faces
- Typically used for spectrometers
- Focuses the beam in the bend plane

Generally, bend magnets can have arbitrary entrance and exit angles, curved pole faces, etc.

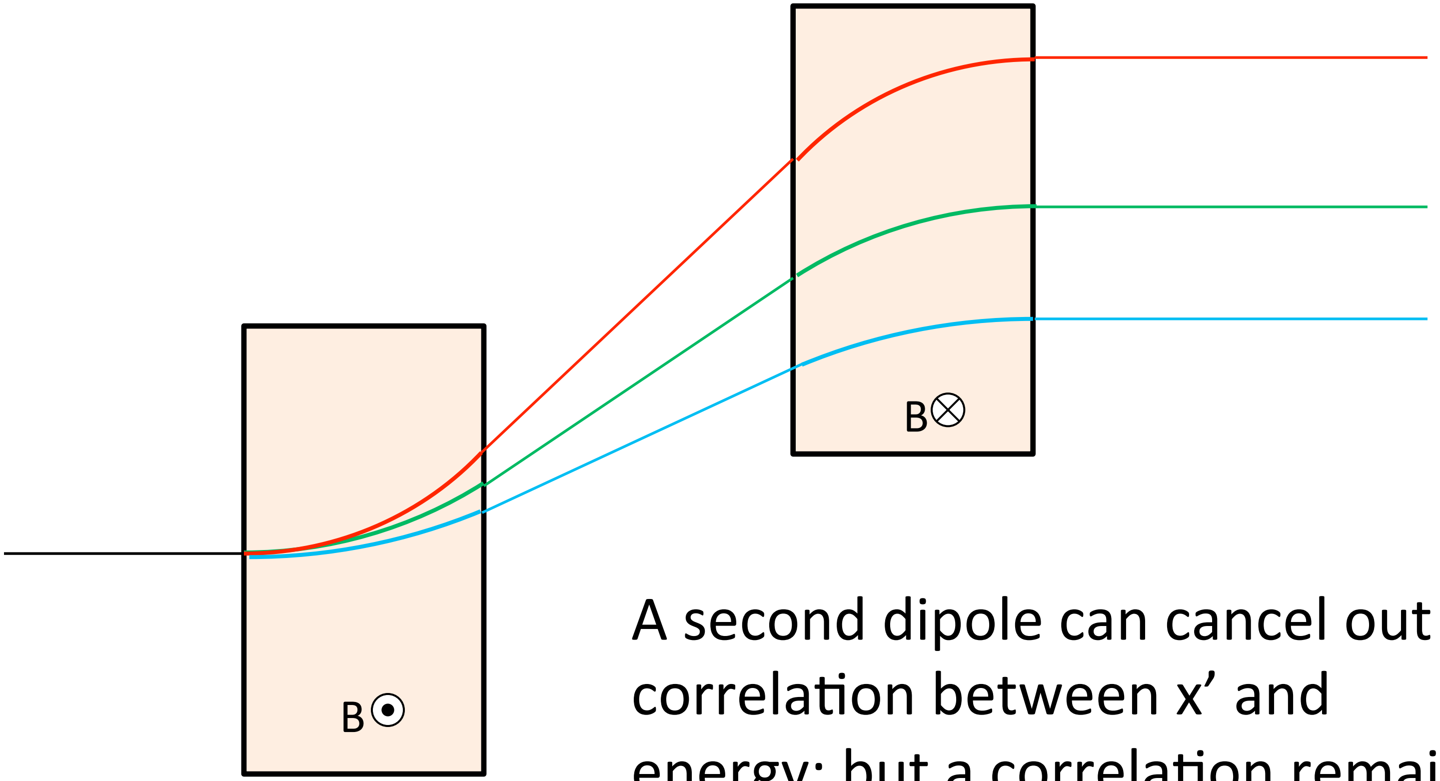
Dispersion (1)

$$\frac{d\theta}{\theta} = \frac{dp}{p}$$



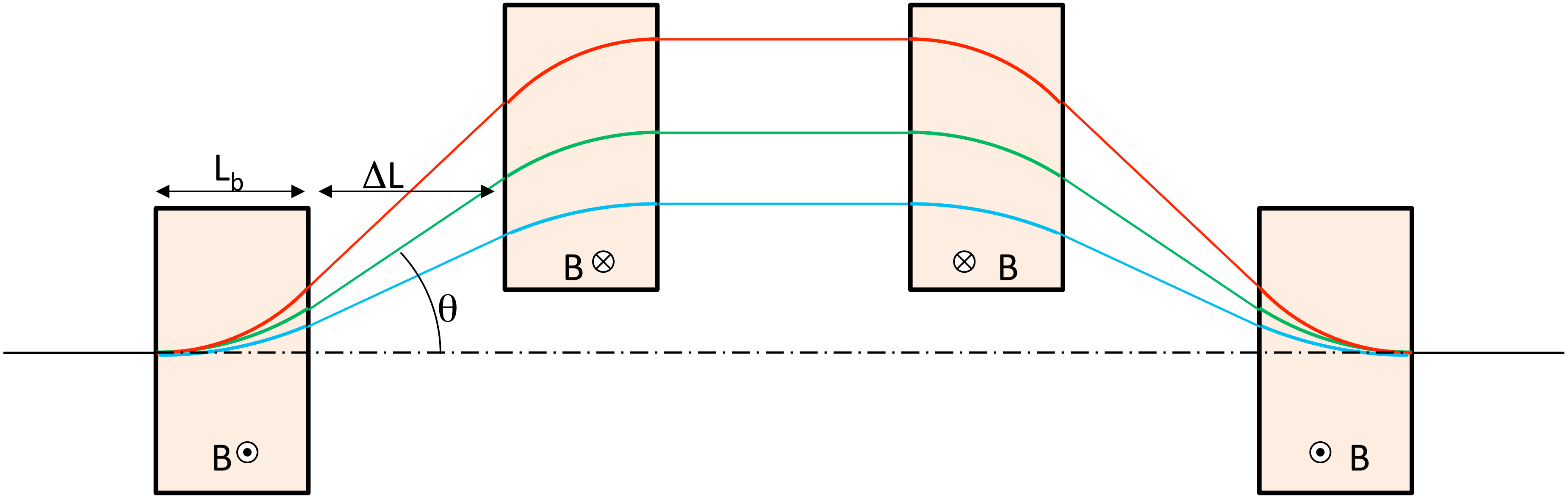
A single bending magnet can introduce a correlation between a particle's energy and x'. The symbol for this is η'

Dispersion (2)



A second dipole can cancel out the correlation between x’ and energy; but a correlation remains between x and energy: η

Bunch Compressor



In this arrangement, my final dispersion η and η' equal zero (to 1st order).

However, different energy particles take different path lengths through the *4-dipole chicane*. The first-order dependence is the R_{56} R-matrix element.

$$R_{56} \approx -2\theta^2 \left(\Delta L + \tfrac{2}{3} L_b \right)$$

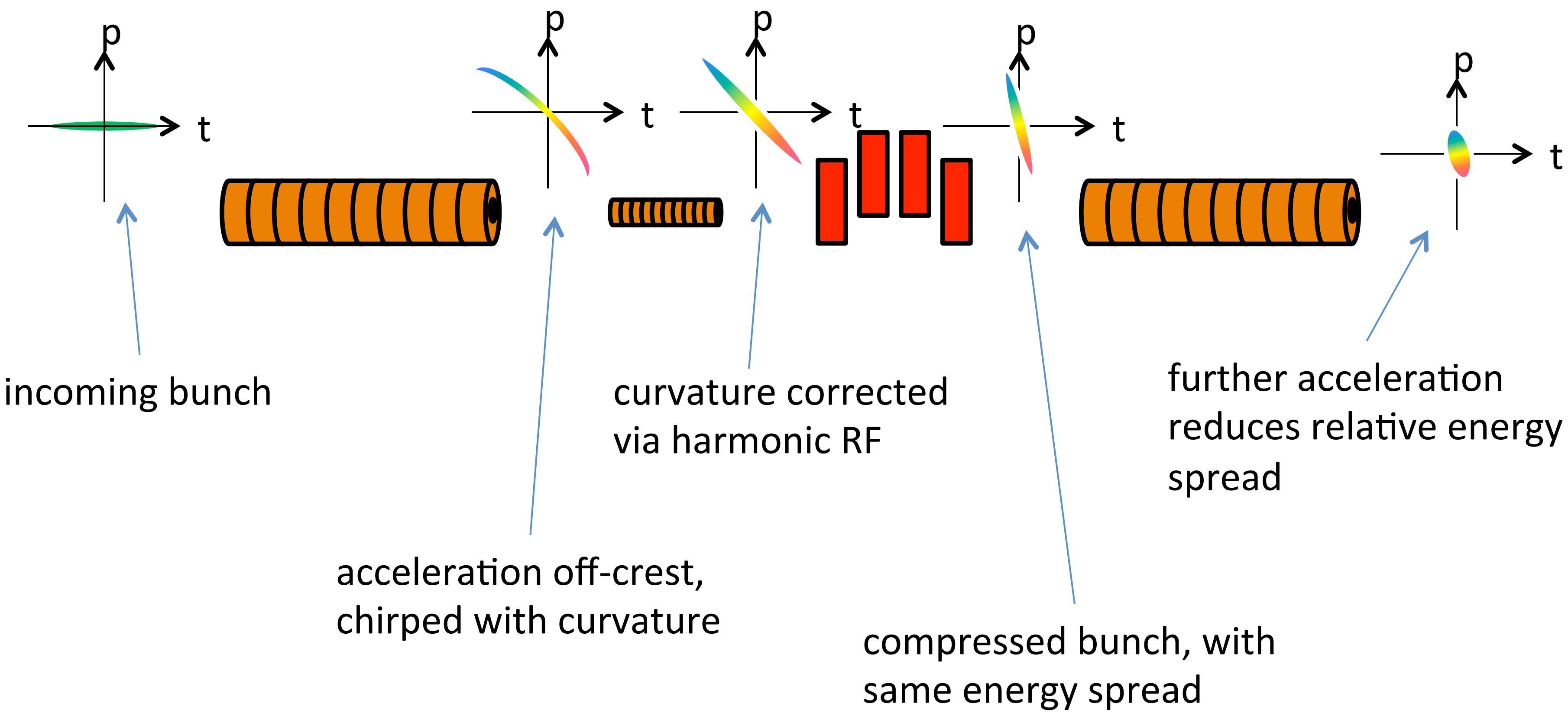
How to Change Bunch Lengths

- We need to adjust the length of the bunch
  - For a relativistic beam ($\gamma \geq 10$ or so) velocity difference too small
  - So, use energy-time correlation + *local* dispersion

- Impart the energy-time correlation with the linac, e.g. run off-crest
  - note: this means *increasing* the energy spread on the beam!
  - can use harmonics to “linearize” the phase space for cleaner compression

- Deal with the energy spread after compression
  - go to such a high energy, the relative energy spread is low enough
  - don’t go to full compression and use linac to pull off energy spread
  - use wakefields to help “flatten” the bunch

Schematic



incoming bunch

acceleration off-crest, chirped with curvature

curvature corrected via harmonic RF

compressed bunch, with same energy spread

further acceleration reduces relative energy spread

Example 1: bunch compressor

Directory: USPAS\2014-June\day_3\comp_1

W_01.ele: elegant command file
W_02.lte: elegant lattice file

run_W_01.bat: batch file to

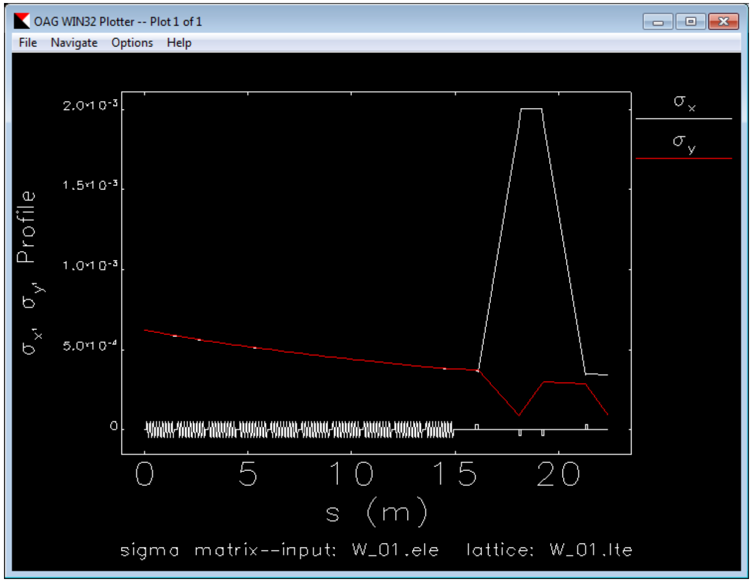
- run **elegant**
- perform some post-processing chores, e.g. data conversion and histogramming
- generate various plots
- print out some relevant data (final bunch length, emittance, R56, compression ratio)

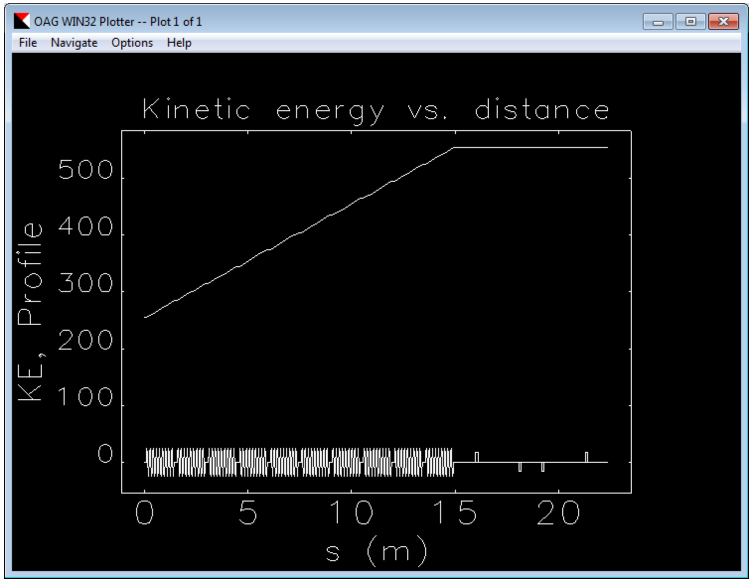
Note: Check the .lte file comments before modifying it.

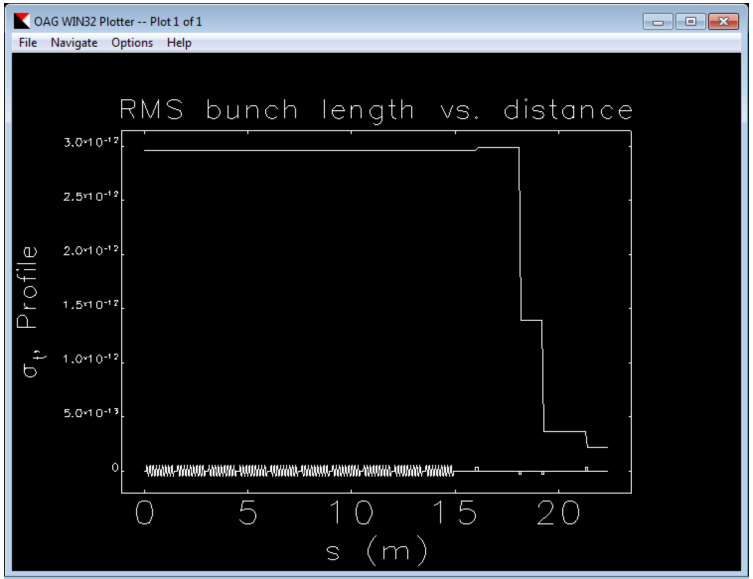
Try varying:

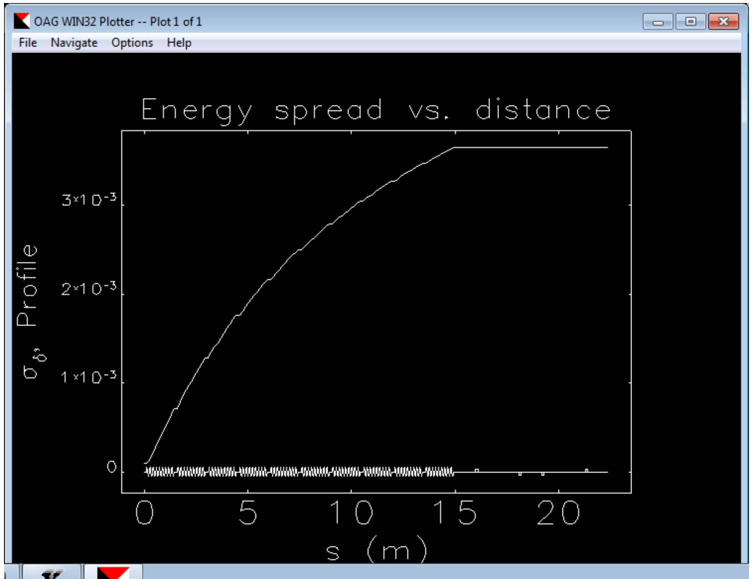
- the bunch compressor dipole bend angle
- the phase and gradient of the linac (.lte file header)
- the beam's starting bunch length and energy spread

Example 1: bunch compressor



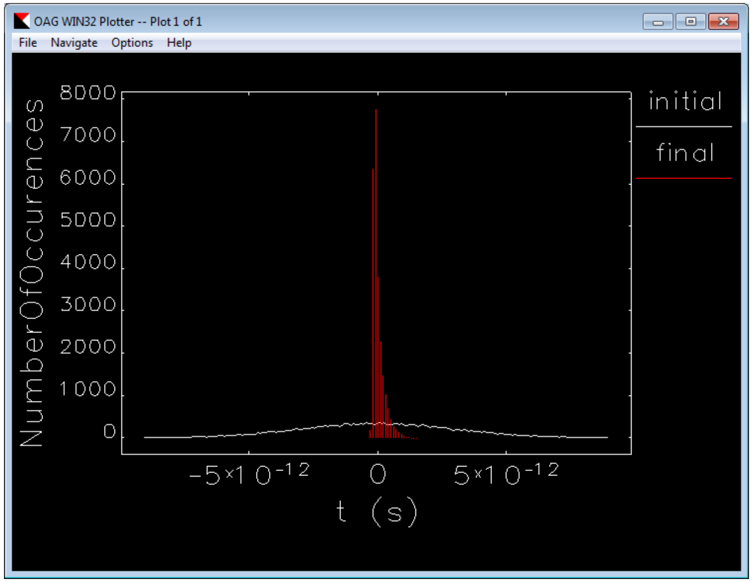




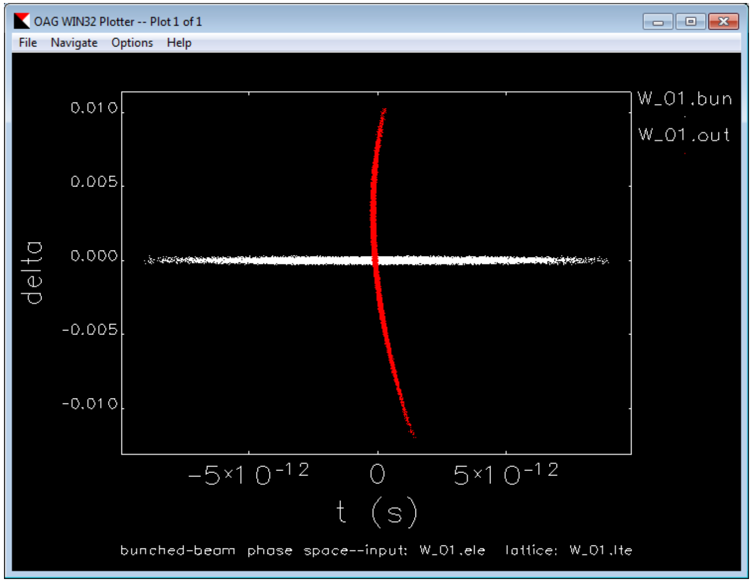


Example 1: bunch compressor

current histograms



longitudinal phase space



white = start of linac red = after bunch compressor

<u>parameters of interest at end of line</u>

σ_t	2.2 ps
σ_δ	0.36%
$\varepsilon_{n,x}$	1.06 μm
$\varepsilon_{n,y}$	1.23 μm
comp. ratio	13:1

Example 2: with 3rd harmonic

Directory: USPAS\2014-June\day_3\comp_2

W_01.ele: elegant command file
W_02.lte: elegant lattice file

run_W_02.bat: batch file to

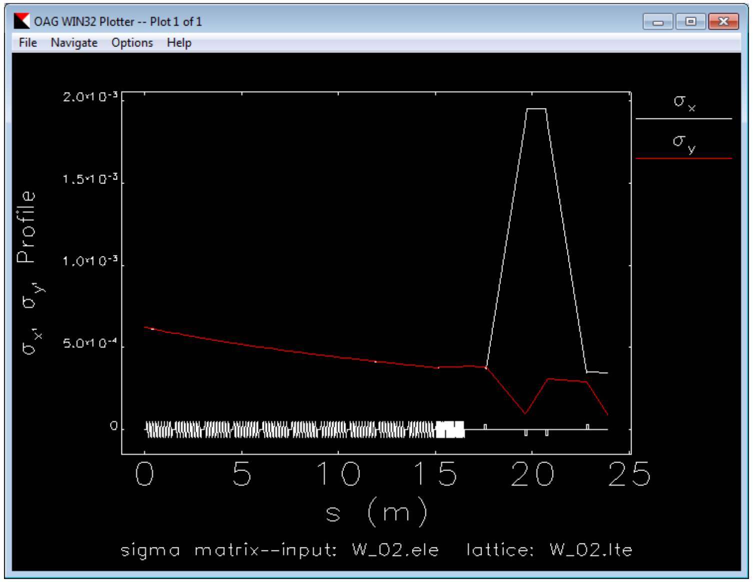
- run **elegant**
- perform some post-processing chores, e.g. data conversion and histogramming
- generate various plots
- print out some relevant data (final bunch length, emittance, R56, compression ratio)

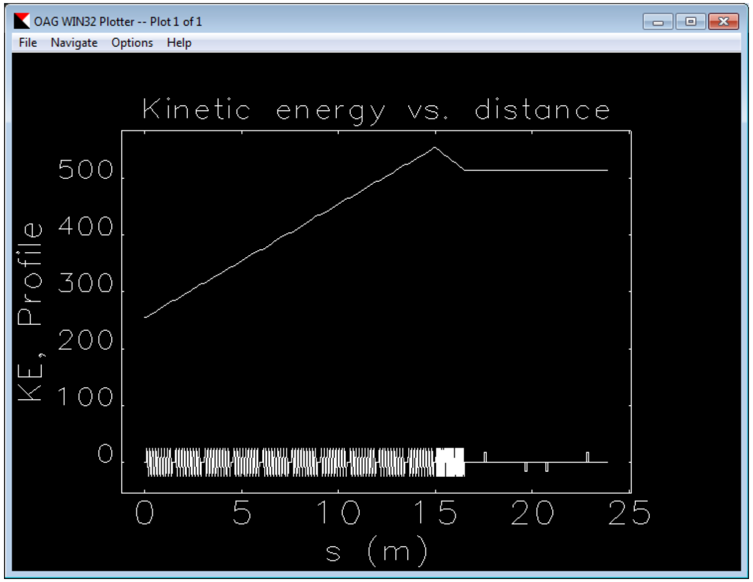
Note: Check the .lte file comments before modifying it.

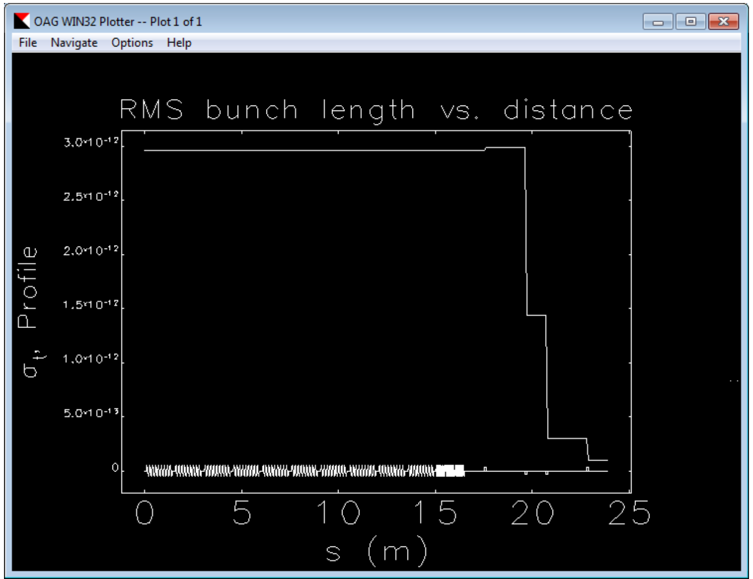
Try varying:

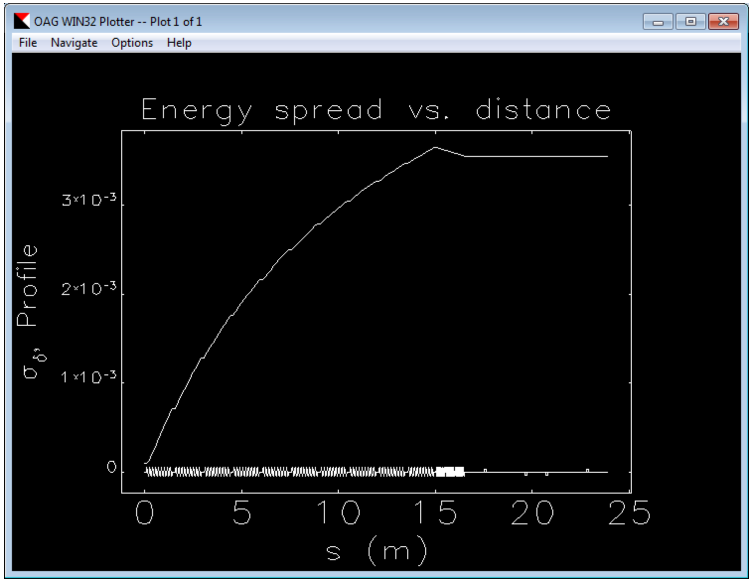
- the bunch compressor dipole bend angle
- the phase and gradient of the linac (.lte file header) fundamental and 3rd harmonics
- the beam's starting bunch length and energy spread

Example 2: with 3rd harmonic



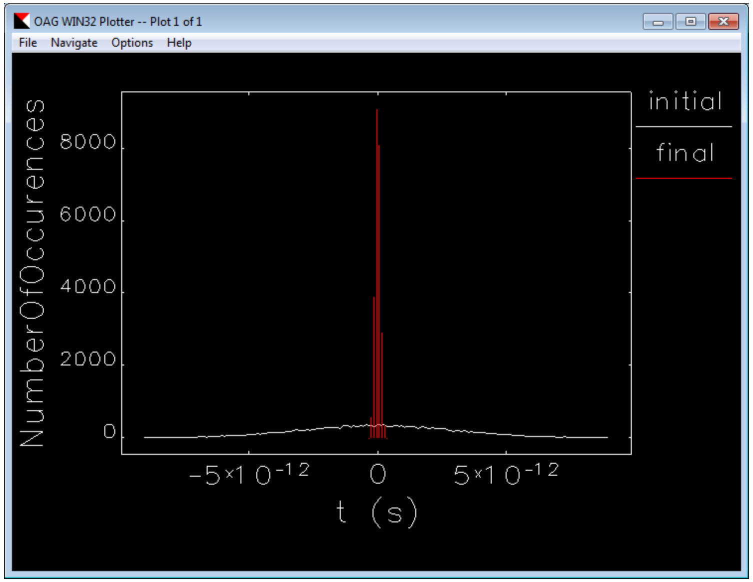




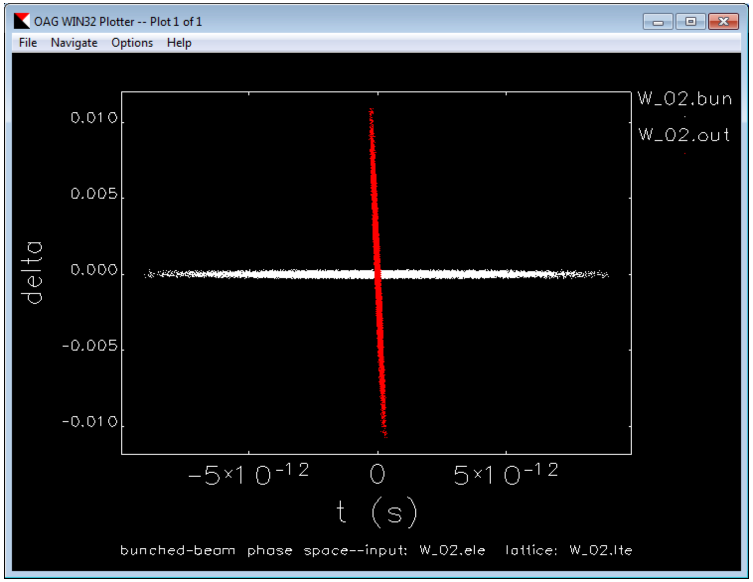


Example 2: with 3rd harmonic

current histograms



longitudinal phase space



white = start of linac red = after bunch compressor

<u>parameters of interest at end of line</u>

σ_t	0.1 ps
σ_δ	0.35%
$\varepsilon_{n,x}$	1.04 μm
$\varepsilon_{n,y}$	1.20 μm
comp. ratio	30:1

Final Project: Design a linac

- Not based on a specific proposed X-FEL

- You will not be including charge-dependent effects
  - wakefields
  - CSR, LCS, etc.

- Therefore, this represents the equivalent to an initial design study, rather than a final design.