ESSENTIALS OF LATTICE DESIGN CODES

Hywel Owen, University of Manchester/Cockcroft Institute

The Cockcroft Institute
of Accelerator Science and Technology

Lecture One – Linear Optics and Matrices

Recap – Numerical Methods

From equations to matrices

Periodic systems

Standard optical modules

Aperiodic systems

Real lattices

Lecture Two – Matching

Design and optimisation

Configuration space and objective functions

Hill-climbing methods

Multiple minima systems

Weights and constraints

Lecture Three - Randomness

Random numbers don't exist!

Why distributions are important to get right

How to do it

Monte-Carlo and other things

Lecture Four – Everything Else

A bestiary of codes

The structure of an accelerator code

'Gotchas' – things to watch out for

Various things that codes do

There is (probably) not going to be a one-to-one mapping of lectures into slots

Learning material for this course

Course Material

CI Courses (http://www.cockcroft.ac.uk/pages/education.htm)

Linear Dynamics (Wolski)

Accelerator Physics(Appleby)

Linear Optics and Lattice Design (Holzer)

CAS (http://www.cern.ch/CAS)

And many more….

(available from all good booksellers!)

Where to get codes

Accelerator Code Repository:

http://projects.astec.ac.uk/Plone/Codes

MAD – Methodical Accelerator Design

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

Elegant

http://www.aps.anl.gov/Accelerator_Systems_Division/Operations_Analysis/software.shtml

And many more….

The Cockcroft Institute
of Accelerator Science and Technology

LATTICE DESIGN CODES: LECTURE ONE: NUMERICS & MATRICES

Hywel Owen, University of Mancheseter/Cockcroft Institute

The Cockcroft Institute
of Accelerator Science and Technology

Recap of Numerical Methods

A numerical method is an algorithm that allows you to solve a problem numerically. Nowadays this is done on a computer

usually using some iterative procedure.

Examples:

- Solving systems of equations
- Finding roots of equations (esp. non-linear)
- Solving ordinary differential equations/partial differential equations (Project 2)
- Analysis of data (curve fitting, spectral analysis) (Project 1)
- Monte-Carlo simulation of physical systems

During numerical analysis, we must be aware that our predictions may be different from real system, e.g. to what accuracy can we believe them?

- Round-off error
- Range error
- Truncation error

The representation of numbers on computers

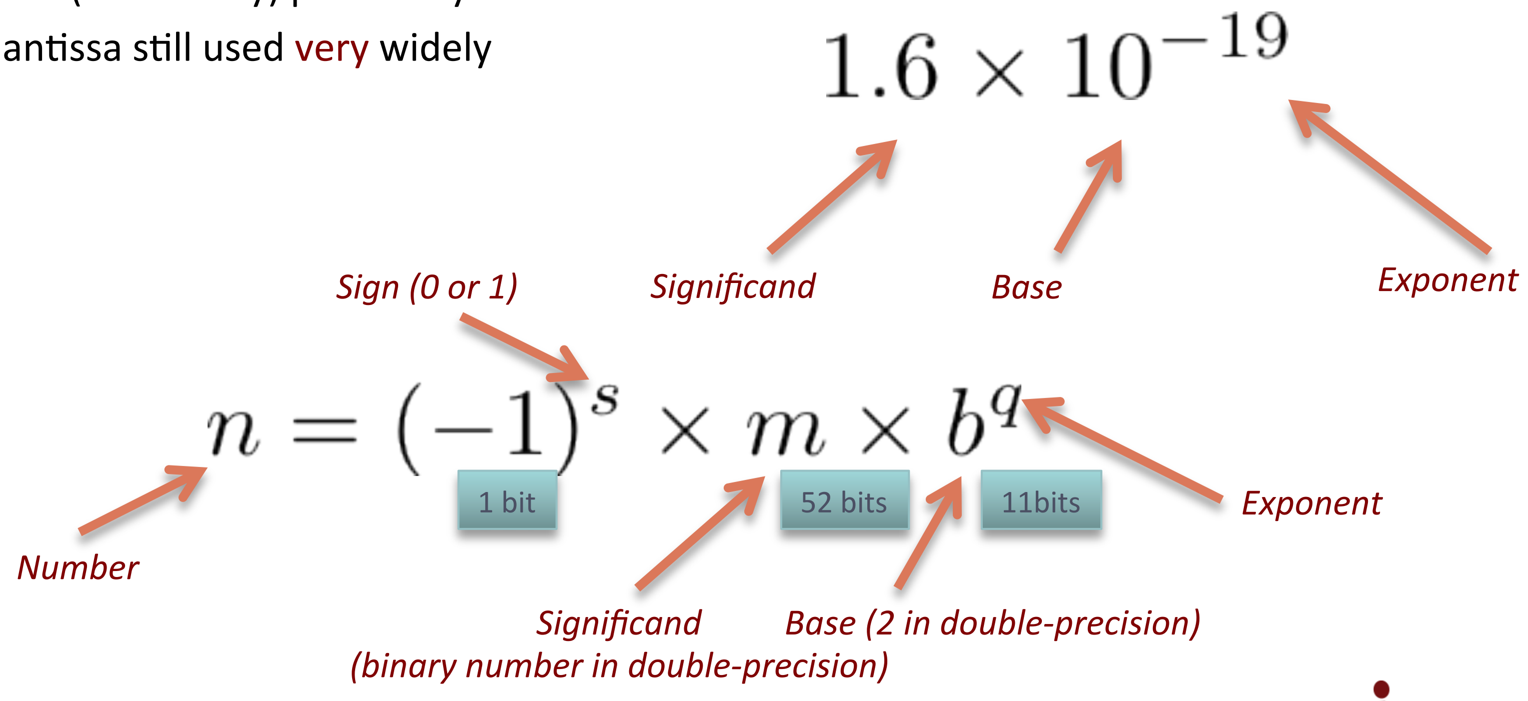
Normally, computers store individual floating-point numbers in either

- Single precision (4 bytes/32 bits)
- Double precision (8 bytes/64 bits)/binary64 – e.g. default in MATLAB
- If you define `a = 0.02`; then a will be in double-precision format

A floating-point number is represented by a significand and exponent

Significand (incorrectly) previously called mantissa

- mantissa still used very widely



$$1.6 \times 10^{-19}$$

Significand

Base

Exponent

Sign (0 or 1)

$$n = (-1)^s \times m \times b^q$$

1 bit

52 bits

11bits

Number

Exponent

Significand
(binary number in double-precision)

Base (2 in double-precision)

Accuracy of stored values

Accuracy is determined by the number of bits in the significand:

Double-precision accuracy is about 16 decimal places

$$\frac{1}{b^l} = \frac{1}{2^{52}} \simeq 1 \times 10^{-16}$$

Single precision accuracy is about 7 decimal places

$$\frac{1}{b^l} = \frac{1}{2^{23}} \simeq 1 \times 10^{-7}$$

Round-off errors can be very significant:

'Subtraction of similar numbers' problem

$$\frac{10^{-20}}{(\underbrace{3 + 10^{-20}}_{\textit{Cannot be represented with 16 sd}}) - 3)}$$

This immediately gives catastrophic round-off error

Try it! >>10^(-20)/((3+10^(-20))-3) gives Inf (inifinity) instead of 1

Matlab has variable eps that gives lower bound on accuracy

eps = 2.2204e-16

Round-off error

Example: Round-off error when calculating derivatives

$$f'(x) = \frac{f(x+h) - f(x)}{h} \qquad h \to 0$$

$$\Delta(h) = \left| \underbrace{f'(x)}_{\textit{Exact Derivative}} - \underbrace{\tfrac{f(x+h) - f(x)}{h}}_{\textit{Approximation}} \right|$$

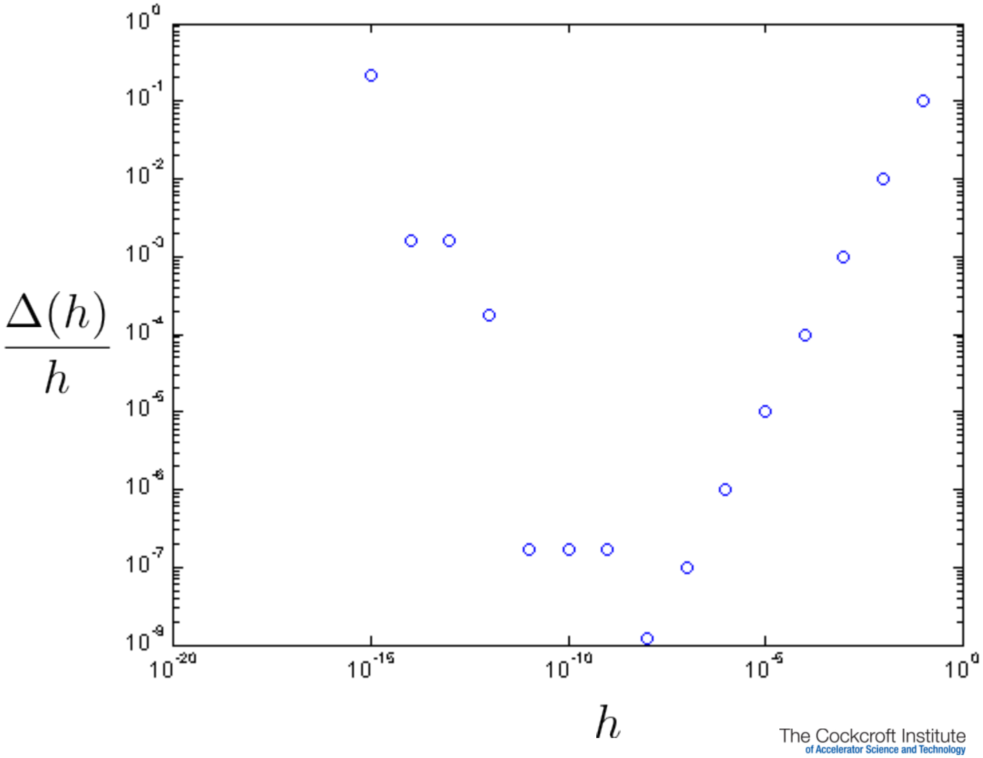
Example: $f(x) = x^2$

$$\Delta(h) = 2x - \frac{(x+h)^2 - x^2}{h}$$

An optimum value of h is often around

$$h \simeq 10^{-8}$$

But, it depends on the problem!



Range Error

Range errors occur when you use a number outside of the exponent range

Smallest representable number: $2^{2^{-(l_q-1)}}$

Largest representable number: $2^{-2^{(l_q-1)}}$

Single precision: $l_q = 8 \qquad 2^{\pm 2^{(l_q-1)}} \simeq 10^{\pm 38}$

Double precision: $l_q = 11 \qquad 2^{\pm 2^{(l_q-1)}} \simeq 10^{\pm 308}$

Exceeding the single precision range limit is not difficult in physics

$$a_0 = \frac{4\pi\epsilon_0\hbar^2}{m_e e^2} \boxed{\simeq 5.2 \times 10^{-11}} m$$

well within single precision limit

But, we would (without thinking) calculate the numerator and denominator separately in code:

$$4\pi\epsilon_0\hbar^2 \simeq 1.24 \times 10^{-78}$$

$$m_e e^2 \simeq 2.34 \times 10^{-68}$$

oops!

Solution: use unit scale appropriate to the problem!

Truncation Error

Suppose we need to find the first derivative $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

We have seen that h cannot be chosen too small because of round-off error

Therefore $f'(x) = \frac{f(x+h) - f(x)}{h} + error$

Can we find an expression for the error?

Consider the Taylor expansion $f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \ldots$

(equivalently) $f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\zeta)$ where $x \leqslant \zeta \leqslant x+h$

Therefore $f'(x) \simeq \frac{f(x+h) - f(x)}{h} - \frac{1}{2}hf''(x)$

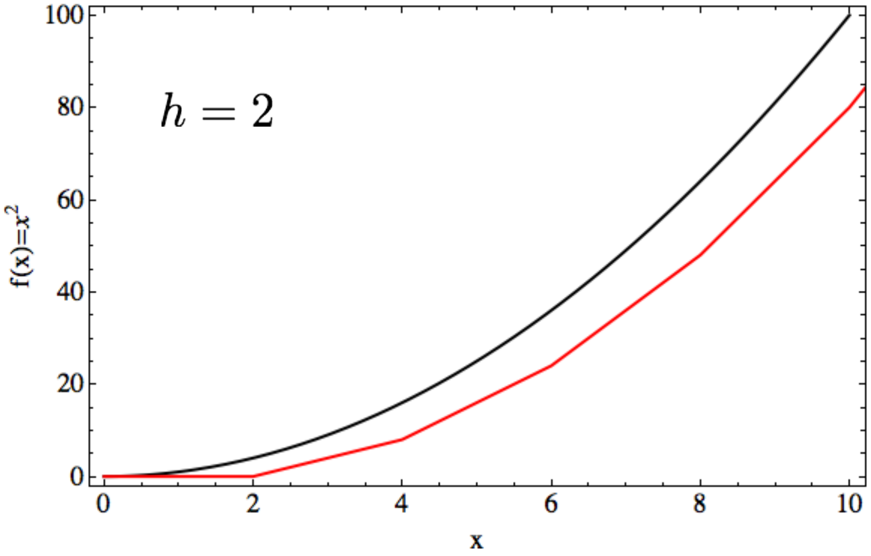
$\frac{1}{2}hf''(x) \sim O(h)$ is called the (local) truncation error

Numerical Integration

Now we are in a position to perform numerical integration

We will consider a universal equation of motion, and look at the simplest method we can imagine. This will turn out to be Euler's Method

Before we do that, let's picture what we are doing:



$$f(x) = x^2$$

$$f(x+h) = f(x) + hf'(x)$$

After n steps, we have effectively calculated the definite integral

$$\int_0^{nh} 2x dx \qquad f'(x) = 2x$$

Euler's Method

$$\frac{dv}{dt} = a(x,t) \qquad \frac{dx}{dt} = v(t)$$

(2 equations, not 1!)

Using the definition of numerical derivatives, we can write

$$\frac{v(t+h) - v(t))}{h} + O(h) = a(x(t), v(t))$$

$$\frac{x(t+h) - x(t))}{h} + O(h) = v(t)$$

(forward derivatives)

Multiplying through,

$$v(t+h) = v(t) + h \cdot a(x(t), v(t)) + O(h^2)$$

$$x(t+h) = x(t) + h \cdot x(t) + O(h^2)$$

Re-writing in iterative notation,

$$v_{n+1} = v_n + h \cdot a_n$$
$$x_{n+1} = x_n + h \cdot v_n$$

That looks quite easy to calculate! Euler's Method.

Choose h and some initial conditions, and off you go!

Improved Euler's Method

An easy way to improve on Euler's method is to use one extra term in the Taylor expansion for the derivative:

$$v_{n+1} = v_n + h \cdot a_n$$

$$x_{n+1} = x_n + h \cdot v_n + \frac{h^2}{2} \cdot a_n \qquad error \sim O(h^3)$$

(cf.) $x(t+dt) = x(t) + x'(t)dt + x''(t)\frac{(dt)^2}{2} + \ldots$

Verlet's Method

$$\frac{d^2x}{dt^2} = a \qquad \frac{dx}{dt} = v$$

$$f'(t) = \frac{f(t+h) - f(t-h)}{2h} - \frac{1}{6}h^2 f'''(\xi)$$

$$f''(t) = \frac{f(t+h) + f(t-h) - 2f(t)}{h^2} - \frac{1}{12}h^2 f^{(4)}(\xi)$$

re-arranging the 2nd of these gives

$$x_{n+1} = 2x_n - x_{n-1} + h^2 \cdot a_n + O(h^4)$$

Note: quartic in h!

Wow! This should be much more accurate!
But notice that (n+1) term requires n and (n-1) terms – not self-starting.
Must use Euler method for first step.

The Cockcroft Institute
of Accelerator Science and Technology

Example: Spring-Mass with Damping

General SHM with damping

$$m\ddot{x} = -b\dot{x} - kx + F(t)$$

$$a(t) = -\frac{b}{m}v(t) - \frac{k}{m}x(t) + \frac{F(t)}{m}$$

$$a_n = -\frac{b}{m}v_n - \frac{k}{m}x_n + \frac{F(t)}{m}$$

Solvable if:

$$F(t) = 0$$

(and for certain functions)

Euler

$$a_n = -\frac{b}{m}v_n - \frac{k}{m}x_n$$

$$v_{n+1} = v_n + h \cdot a_n$$

$$x_{n+1} = x_n + h \cdot v_n$$

Imp. Euler

$$a_n = -\frac{b}{m}v_n - \frac{k}{m}x_n$$

$$v_{n+1} = v_n + h \cdot a_n$$

$$x_{n+1} = x_n + h \cdot v_n + \frac{1}{2}h^2 \cdot a_n$$

Verlet

$$x_{n+1} = A \cdot x_n + B \cdot x_{n-1}$$

$$A = 2\frac{(2m - kh^2)}{D} \quad B = \frac{(bh - 2m)}{D}$$

$$D = 2m + bh$$

The Cockcroft Institute
of Accelerator Science and Technology

Exact solution for F(t)=0:

(Refer to your notes from PHYS 10302)

$m\ddot{x} + b\dot{x} + kx = 0$ has solution $x(t) = Ae^{(i\omega - \frac{\gamma}{2})t}$ where $\gamma = \dfrac{b}{m}$ & $\omega_0 = \sqrt{\dfrac{k}{m}}$

$$\omega^2 = \omega_0^2 - \frac{\gamma^2}{4} = \frac{k}{m} - \frac{b^2}{4m^2}$$

Quality factor $Q = \dfrac{\omega_0}{\gamma} = \sqrt{\dfrac{km}{b^2}}$

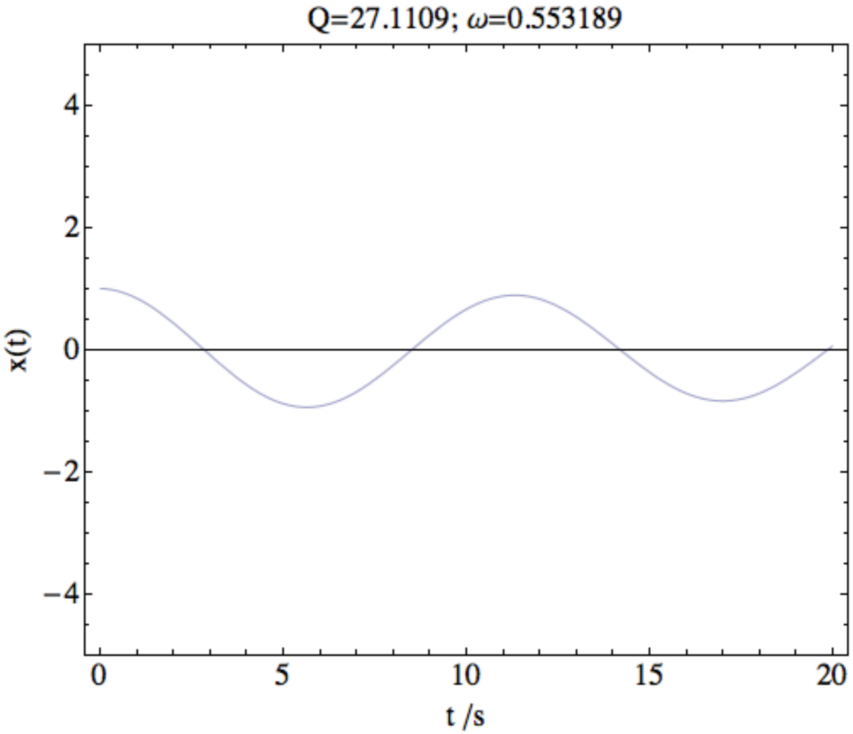
Critical damping is defined as $\omega = 0$, therefore $\omega_0 = \dfrac{\gamma^2}{4}$ and so $b = 2\sqrt{km}$

Substituting into equation for quality factor gives $Q = \dfrac{1}{2}$

$$b_{cr} = 2\sqrt{km} \qquad b = \frac{1}{2}b_{cr} \qquad Q = 1$$

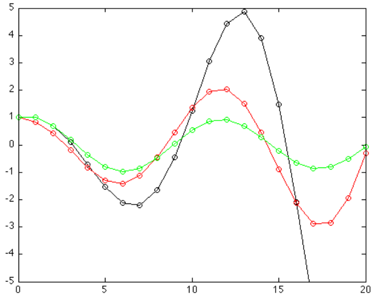
$$b = 2b_{cr} \qquad Q = \frac{1}{4}$$

Analytic versus numerical simulation

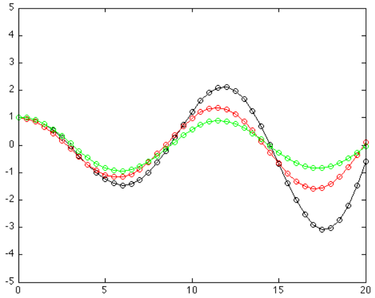


Analytical solution

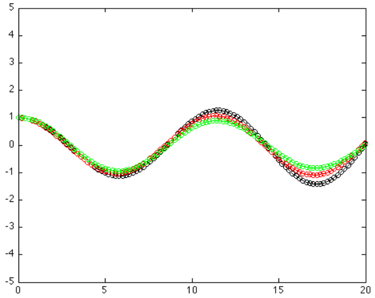
The important point here is to test your algorithm in the case where the solution is analytically known: then you know how good it is.



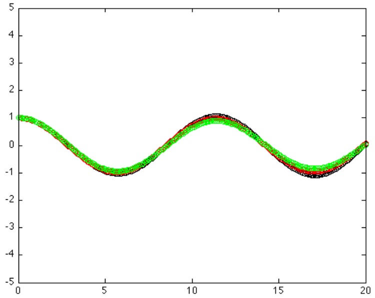
$h = 1s$



$h = 0.5s$

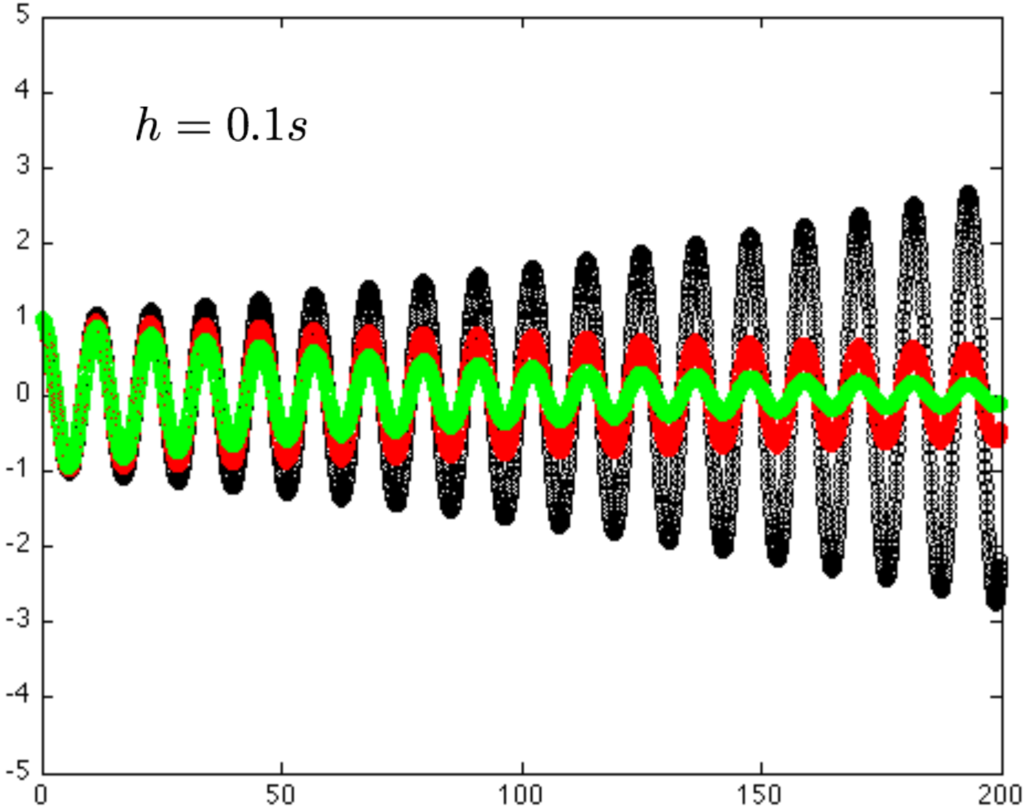


$h = 0.2s$



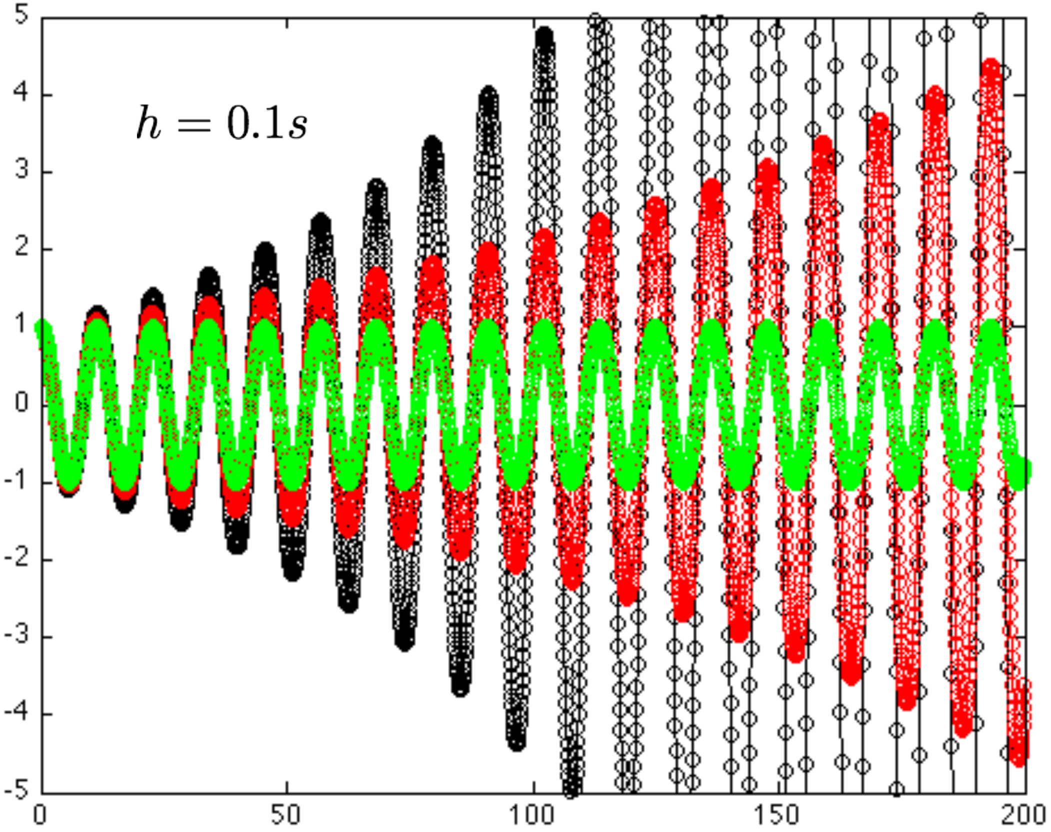
$h = 0.1s$

Integration over long times



Artificial addition of energy to system – numerical artefact.
Integrators are non-*symplectic*, one of them badly so!

Getting it wrong with no damping



Question: How would you check the green one was really constant amplitude?

Energy Errors in Euler's Method

Ignoring for simplicity an initial velocity, we may write an expression for the damped SHM motion

$$x(t) = Ae^{-bt/2m}\cos(\sqrt{(\frac{k}{m}-\frac{b^2}{4m^2})t})$$

The energy is just

$$E(t) = \frac{1}{2}kx^2(t) + \frac{1}{2}mv^2(t)$$

Substituting and expanding out, we get (eventually)

$$E(t) = \frac{A^2e^{-bt/m}}{8m}(4km + b^2\cos(\frac{\sqrt{4km-b^2}t}{m}) + b\sqrt{4km-b^2}\sin(\frac{\sqrt{4km-b^2}t}{m}))$$

The Cockcroft Institute
of Accelerator Science and Technology

Euler’s Method With No Damping

Start by writing out expression for Euler step

$$\begin{aligned} x_{n+1} &= x_n + hv_n \\ v_{n+1} &= v_n - \frac{kh}{m}x_n \end{aligned}$$

Then substitute into expression for energy

$$E_{n+1} = \frac{1}{2}kx_n^2 + \frac{1}{2}mv_n^2 + \frac{1}{2m}h^2k^2x_n^2 + \frac{1}{2}h^2kv_n^2$$

and collect terms to give

$$E_{n+1} = E_n(1 + \frac{k}{m}h^2)$$

In other words, our simulation predicts a steadily increasing energy!

The Euler-Cromer Method

Let’s make a slight change to Euler’s method

$$\begin{aligned} x_{n+1} &= x_n + h v_{n+1} \\ v_{n+1} &= v_n - \frac{kh}{m} x_n \end{aligned}$$

Expanding out the expression for energy again, we obtain (eventually)

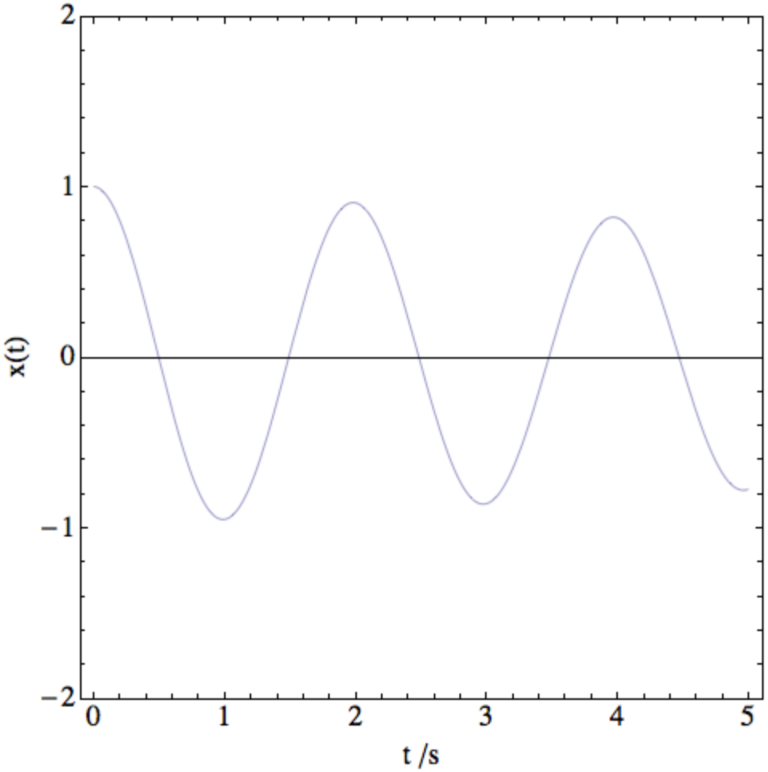
$$E_{n+1} = E_n - \frac{1}{2} h^2 \left(\frac{k^2 x_n^2}{m} - k v_n^2\right) - h^3 \frac{k^2 x_n v_n}{m} + h^4 \frac{k^3 x_n^2}{2m^2}$$

The second term averages out over one oscillation. The result is that the overall energy is conserved.

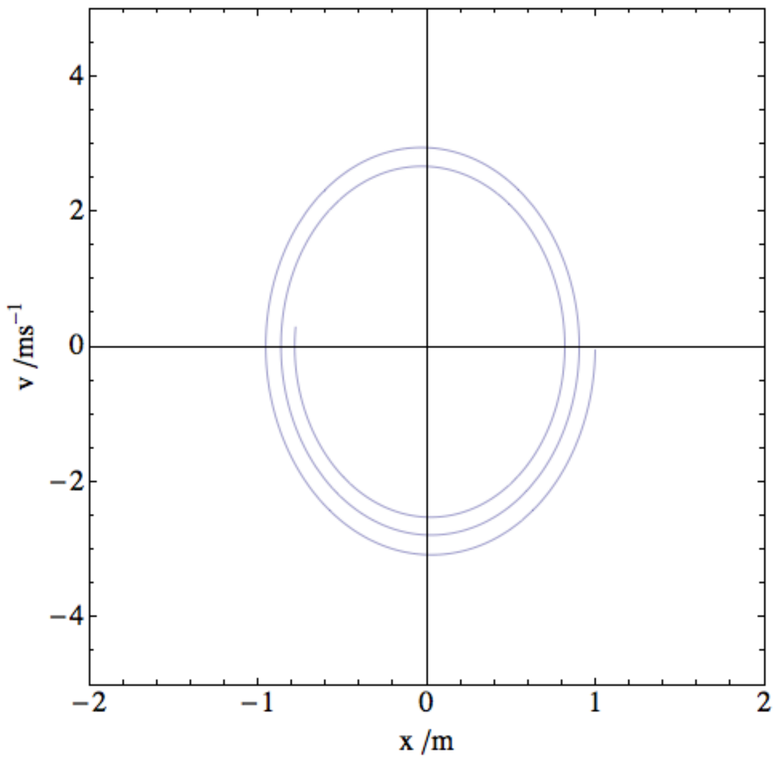
But… there are oscillations about this average, compared to the true value of the energy. Better than Euler though!

Euler-Cromer is a Symplectic Integrator, i.e. it is energy-preserving

A=1,k=10,b=0.1,m=1



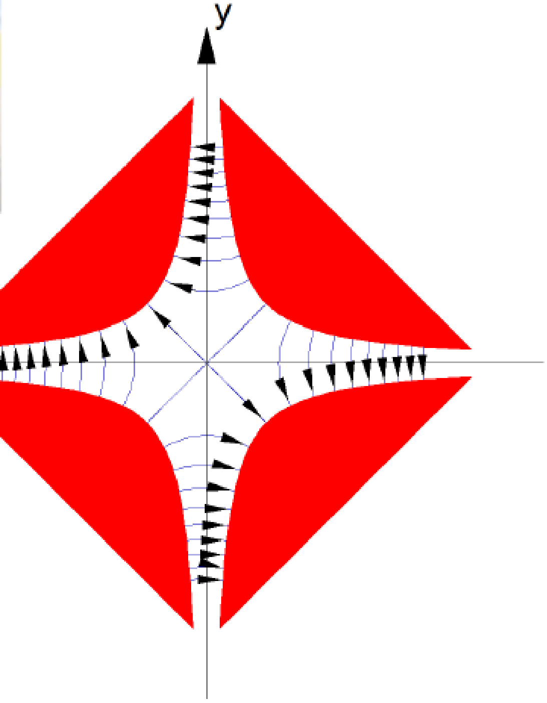
Time Plot



Phase Space Plot

The Cockcroft Institute
of Accelerator Science and Technology

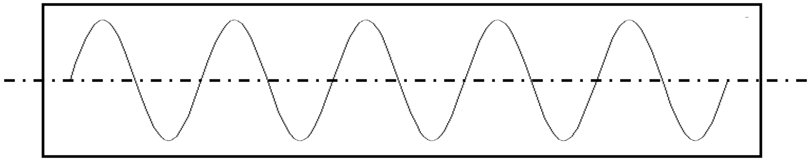
SHM in Linear Optical Systems

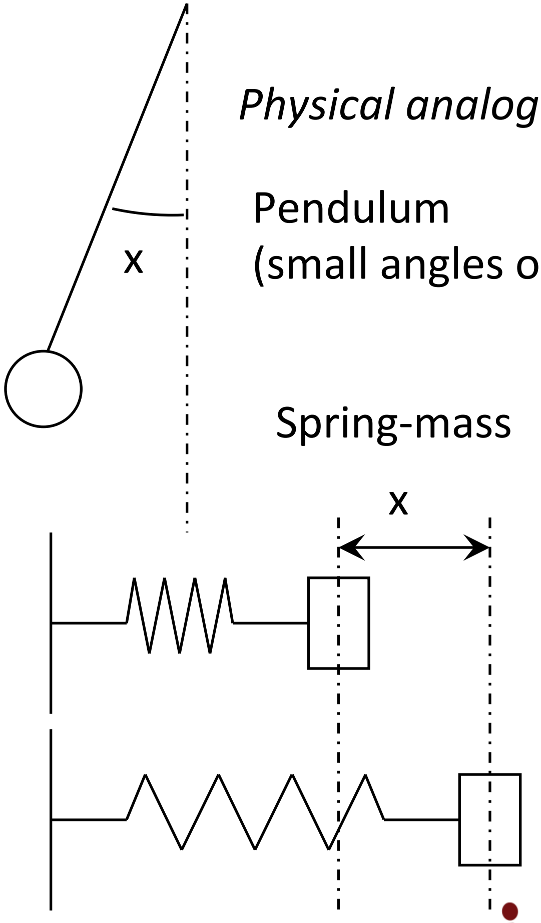


Normal quadrupole

$$B_x = b_2\frac{y}{r_0}, \qquad B_y = b_2\frac{x}{r_0}.$$

SHM through a long quadrupole





Physical analogies

Pendulum
(small angles only!)

Spring-mass

Weak Focusing

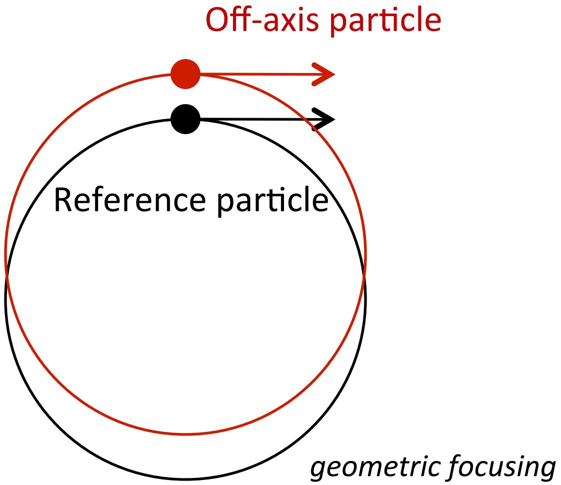
Consider a constant energy particle in a fixed dipole field (into the page)

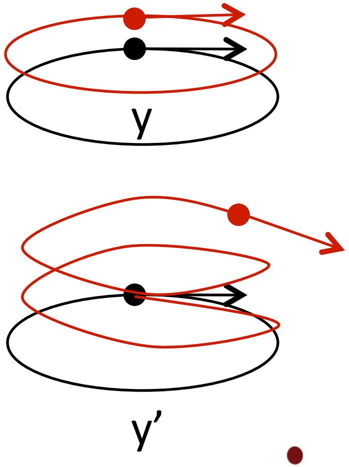
Reference particle executes cyclotron motion

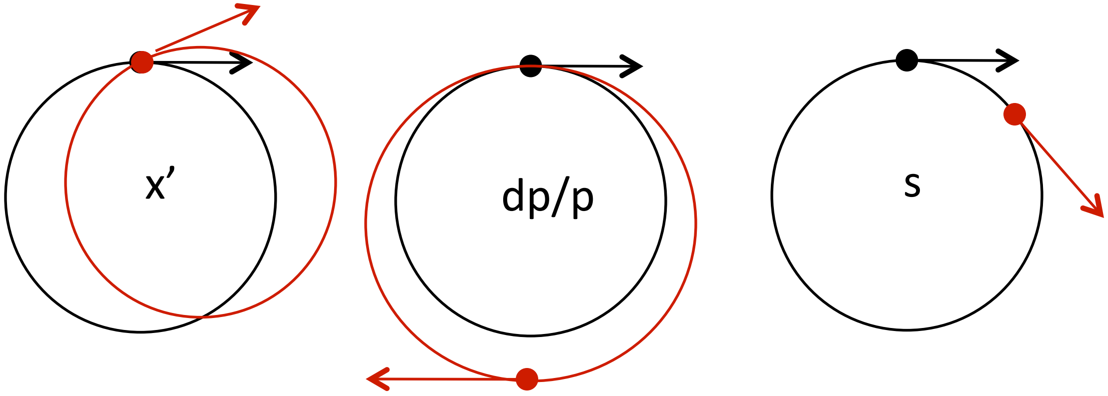
Other particles of the same energy execute cyclotron motion of same radius

Compared to reference trajectory, this looks like an oscillation

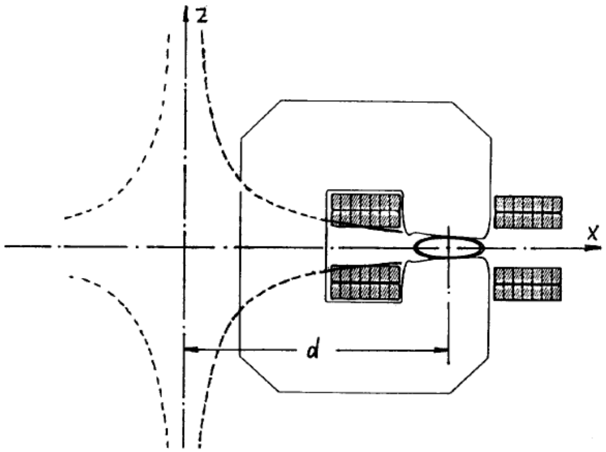
Number of oscillations per turn (the ‘tune’) is 1.

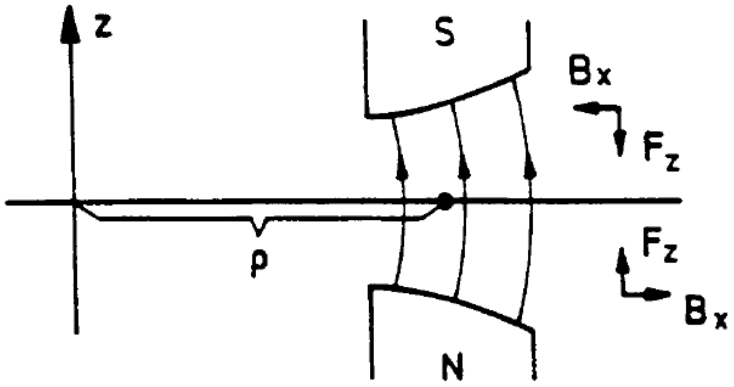




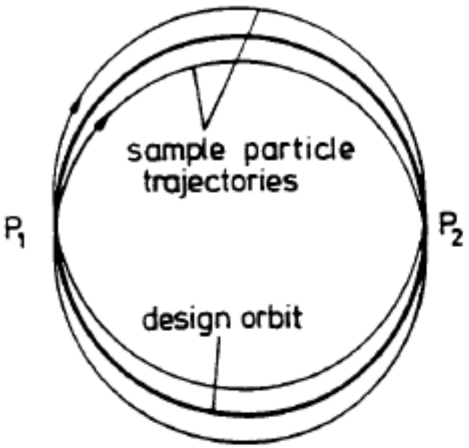


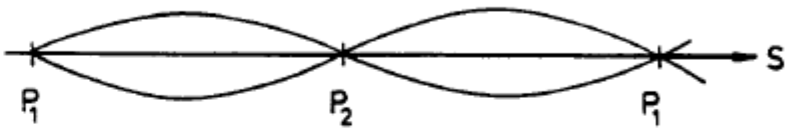
Stability in a combined function magnet





$$n = -\frac{\rho}{B_0}\left(\frac{\partial B_z}{\partial r}\right)_{r=\rho}$$

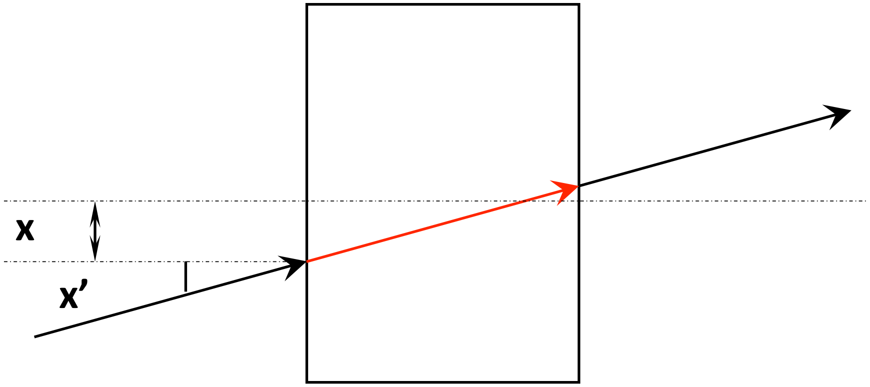


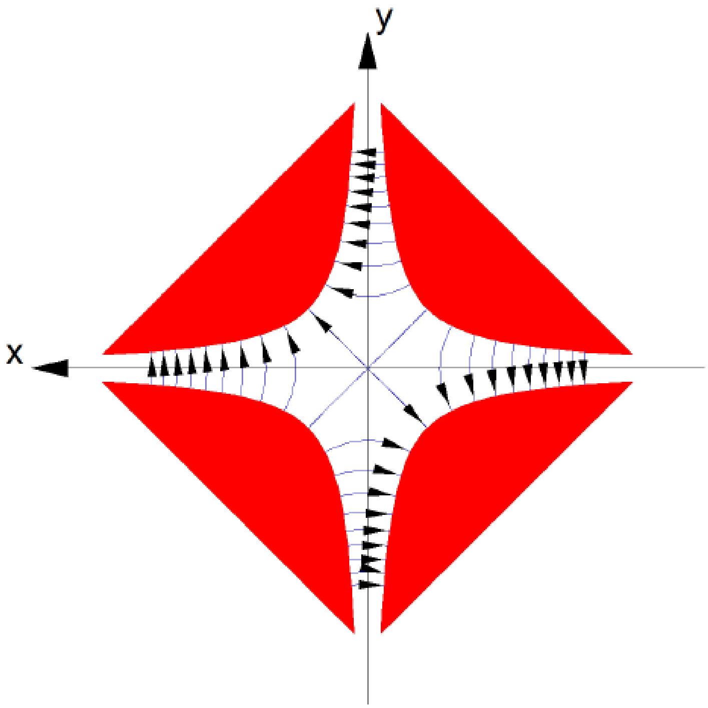


The 6 coordinates and the linear drift space

$$\vec{x} = \begin{pmatrix} x \\ p_x \\ y \\ p_y \\ z \\ \delta \end{pmatrix} \qquad R = \begin{pmatrix} 1 & L & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$x_1 = R.x_0$$





Normal quadrupole

$$B_x = b_2 \frac{y}{r_0}, \qquad B_y = b_2 \frac{x}{r_0}.$$

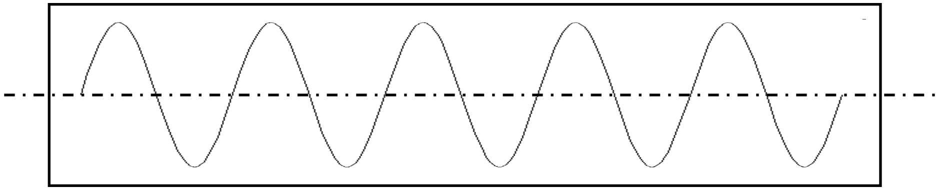
$$k_1 = \frac{q}{P_0} \frac{b_2}{r_0}$$

Normalised field gradient
(this is the number used in nearly all codes)
Doesn't depend on energy

$$R = \begin{pmatrix} \cos \omega L & \frac{\sin \omega L}{\omega} & 0 & 0 & 0 & 0 \\ -\omega \sin \omega L & \cos \omega L & 0 & 0 & 0 & 0 \\ 0 & 0 & \cosh \omega L & \frac{\sinh \omega L}{\omega} & 0 & 0 \\ 0 & 0 & \omega \sinh \omega L & \cosh \omega L & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2 \gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\omega = \sqrt{k_1}$$

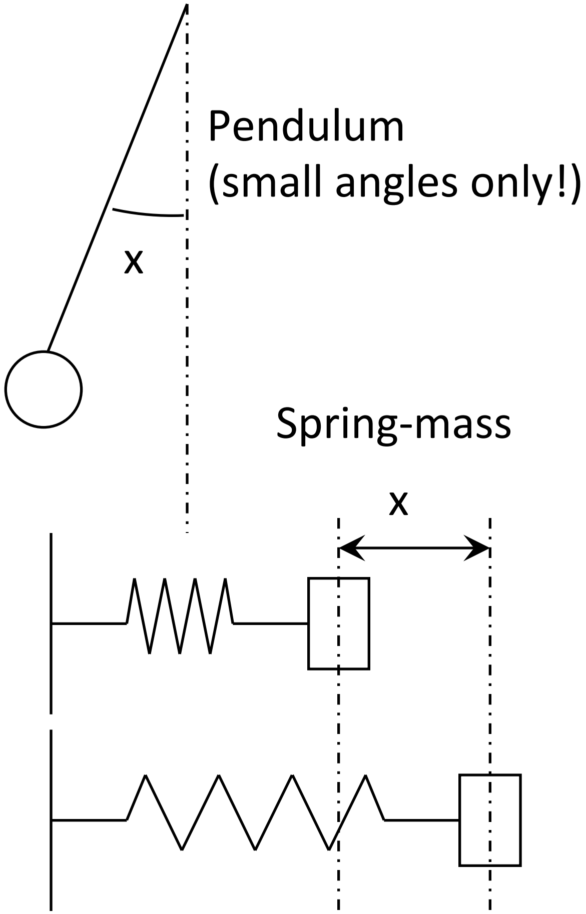
$$R = \begin{pmatrix} \cos\omega L & \frac{\sin\omega L}{\omega} & 0 & 0 & 0 & 0 \\ -\omega\sin\omega L & \cos\omega L & 0 & 0 & 0 & 0 \\ 0 & 0 & \cosh\omega L & \frac{\sinh\omega L}{\omega} & 0 & 0 \\ 0 & 0 & \omega\sinh\omega L & \cosh\omega L & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



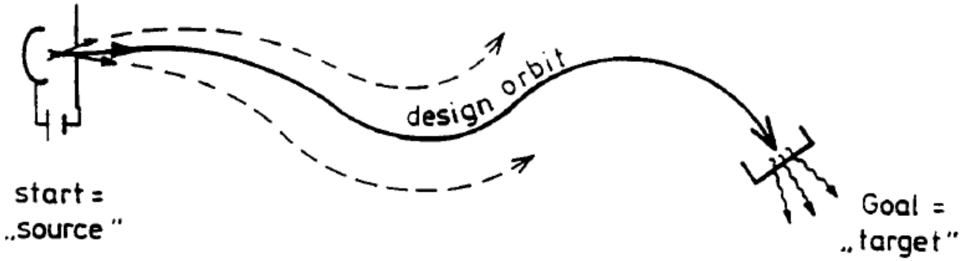
A particle executes simple harmonic motion in an infinitely long quadrupole

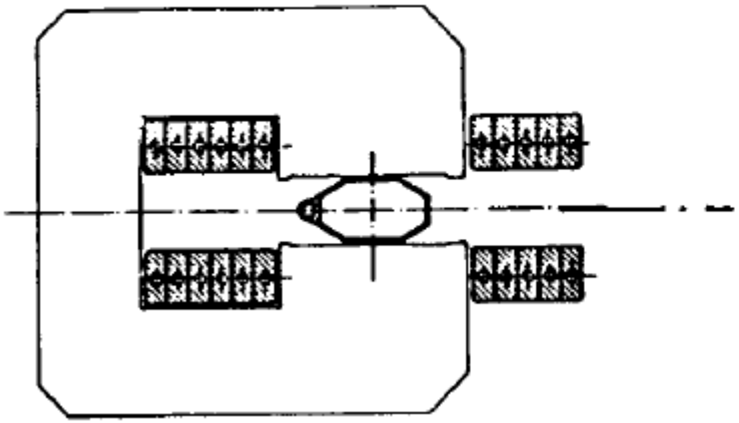
Transversely, an accelerator is analogous to a pendulum where gravity changes as a function of time

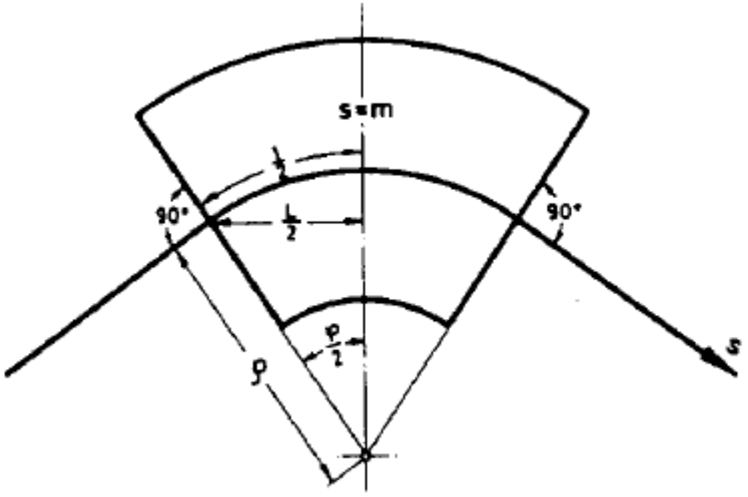
Physical analogies



The Sector Bend – assumption of a curved coordinate system



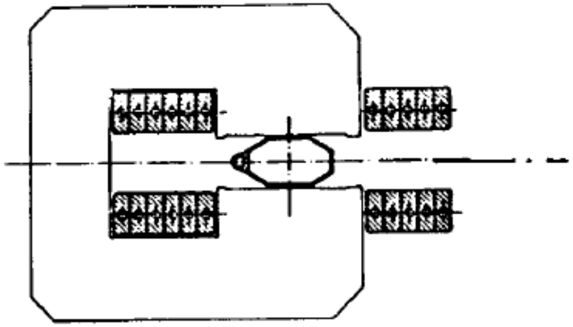


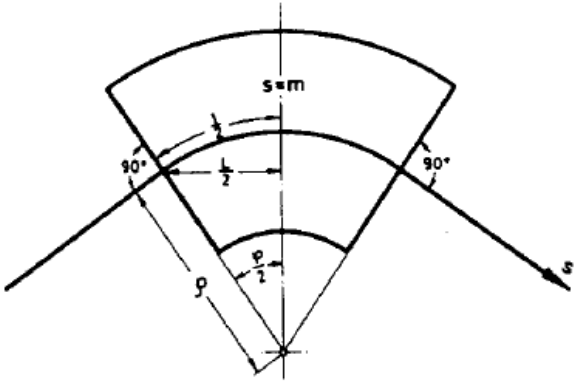


The Linear Wedge Dipole

$$R = \begin{pmatrix} \cos\omega L & \frac{\sin\omega L}{\omega} & 0 & 0 & 0 & \frac{1-\cos\omega L}{\omega\beta_0} \\ -\omega\sin\omega L & \cos\omega L & 0 & 0 & 0 & \frac{\sin\omega L}{\beta_0} \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{\sin\omega L}{\beta_0} & -\frac{1-\cos\omega L}{\omega\beta_0} & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} - \frac{\omega L-\sin\omega L}{\omega\beta_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\omega = k_0 = \frac{q}{P_0} B_0$$





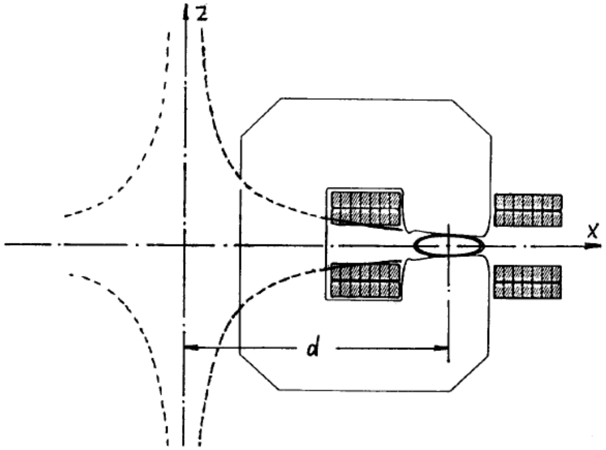
The Linear Wedge Dipole – now with focusing!

$$R = \begin{pmatrix} \cos\omega_x L & \frac{\sin\omega_x L}{\omega_x} & 0 & 0 & 0 & \frac{k_0}{\beta_0}\frac{(1-\cos\omega_x L)}{\omega_x^2} \\ -\omega_x \sin\omega_x L & \cos\omega_x L & 0 & 0 & 0 & \frac{k_0}{\beta_0}\frac{\sin\omega_x L}{\omega_x} \\ 0 & 0 & \cosh\omega_y L & \frac{\sinh\omega_y L}{\omega_y} & 0 & 0 \\ 0 & 0 & \omega_y \sinh\omega_y L & \cosh\omega_y L & 0 & 0 \\ -\frac{k_0}{\beta_0}\frac{\sin\omega_x L}{\omega_x} & -\frac{k_0}{\beta_0}\frac{(1-\cos\omega_x L)}{\omega_x^2} & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} - \frac{k_0^2}{\beta_0^2}\frac{(\omega_x L - \sin\omega_x L)}{\omega_x^3} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\omega_x = \sqrt{k_0^2 + k_1}, \qquad \omega_y = \sqrt{k_1}$$

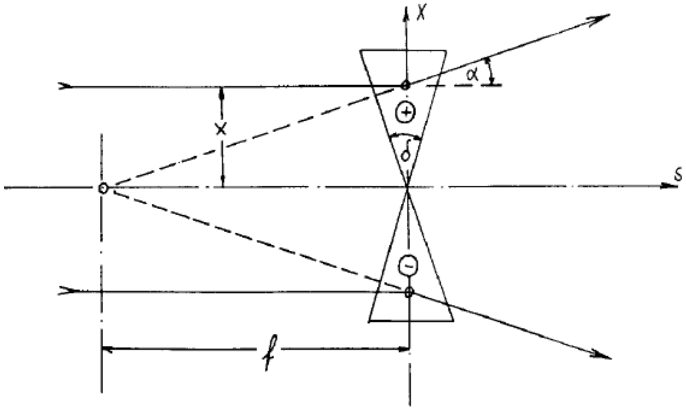
$$B_x = b_2 \frac{y}{r_0}, \qquad B_y = b_1 + b_2 \frac{x}{r_0}$$

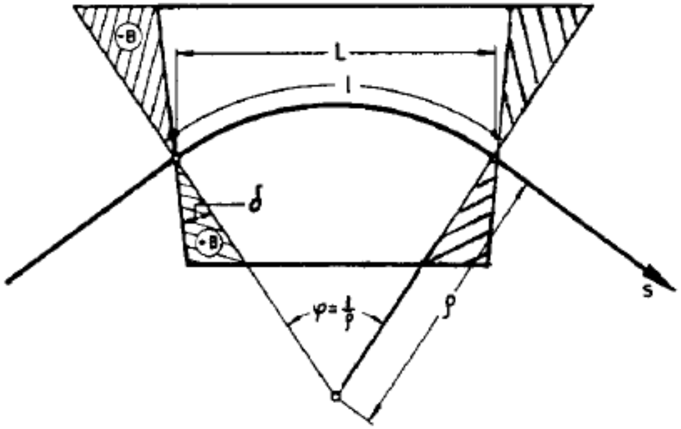
$$k_0 = \frac{q}{P_0} b_1, \qquad k_1 = \frac{q}{P_0}\frac{b_2}{r_0}$$



$$x_1 = R.x_0$$

Edge focusing

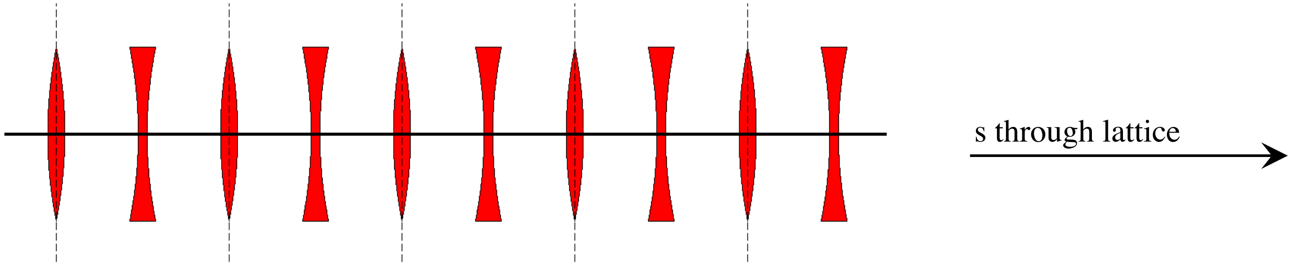




$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -K_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & K_1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$K_1 = -\frac{q}{P_0} B_0 \tan \psi$$

The FODO Lattice – The Simplest Strong-Focusing Lattice

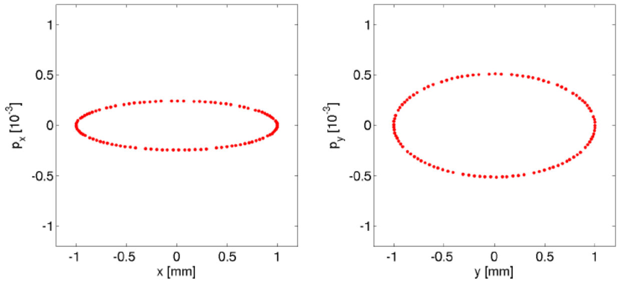


$L \to 0 \qquad k_1 L \to \frac{1}{f}$ (thin-lens approximation)

$$R_Q(f) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1/f & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/f & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad R_D(L) = \begin{pmatrix} 1 & L & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & L & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R = R_Q\left(2f_0\right) \cdot R_D(L) \cdot R_Q\left(-f_0\right) \cdot R_D(L) \cdot R_Q\left(2f_0\right)$$

$$R = \begin{pmatrix} 1 - \frac{L^2}{2f_0^2} & \frac{L}{f_0}(L + 2f_0) & 0 & 0 & 0 & 0 \\ \frac{L}{4f_0^3}(L - 2f_0) & 1 - \frac{L^2}{2f_0^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 - \frac{L^2}{2f_0^2} & -\frac{L}{f_0}(L - 2f_0) & 0 & 0 \\ 0 & 0 & -\frac{L}{4f_0^3}(L + 2f_0) & 1 - \frac{L^2}{2f_0^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{2L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



(hint: this is a Poincare section)

From rays to Twiss values

$$R = \begin{pmatrix} 1-\frac{L^2}{2f_0^2} & \frac{L}{f_0}(L+2f_0) & 0 & 0 & 0 & 0 \\ \frac{L}{4f_0^3}(L-2f_0) & 1-\frac{L^2}{2f_0^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-\frac{L^2}{2f_0^2} & -\frac{L}{f_0}(L-2f_0) & 0 & 0 \\ 0 & 0 & -\frac{L}{4f_0^3}(L+2f_0) & 1-\frac{L^2}{2f_0^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{2L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

It can be shown that: $R^\mathsf{T} \cdot S \cdot R = S$

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

w.l.o.g. we can write: $R_2 = I_2 \cos\mu_x + S_2 \cdot A_2 \sin\mu_x$

with: $A_2 = \begin{pmatrix} \gamma_x & \alpha_x \\ \alpha_x & \beta_x \end{pmatrix}$

which gives: $\beta_x\gamma_x - \alpha_x^2 = 1$ and $R_2^\mathsf{T} \cdot A_2 \cdot R_2 = A_2$

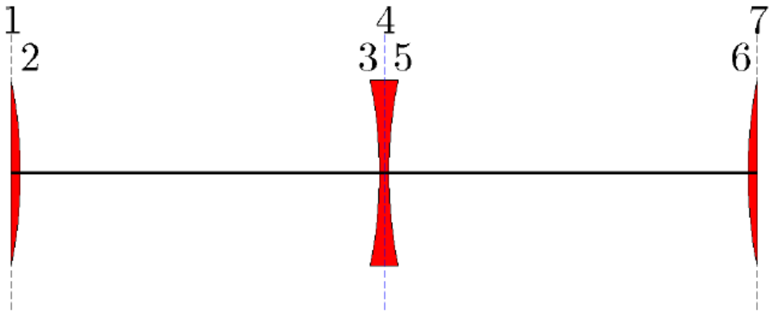
So what?

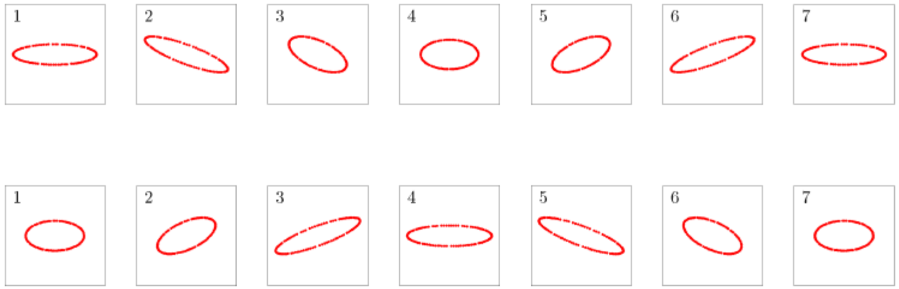
Well, this means that the values of alpha, beta, gamma are constants of the accelerator lattice, and not of any particular particle

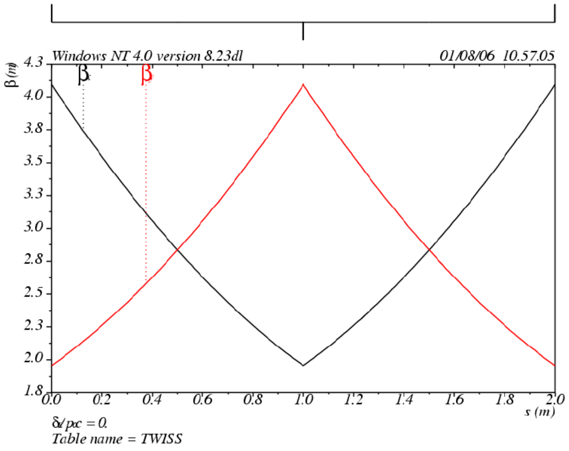
Beta in particular describes the envelope of the particles

Alpha, beta, gamma are the Twiss functions (functions of ‘s’).

Propagation of the Twiss Values







$$A_2^{-1}(s_1) = R_2(s_0,s_1)\cdot A_2^{-1}(s_0)\cdot R_2^{\mathrm{T}}(s_0,s_1)$$

where: $$A_2^{-1}(s) = \begin{pmatrix} \beta_x(s) & -\alpha_x(s) \\ -\alpha_x(s) & \gamma_x(s) \end{pmatrix}$$

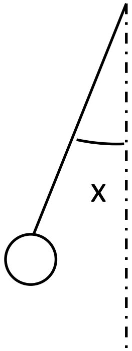
$$\begin{aligned} x &= \sqrt{2\beta_x J_x}\cos\phi_x \\ p_x &= -\sqrt{\frac{2J_x}{\beta_x}}(\sin\phi_x + \alpha_x\cos\phi_x) \end{aligned}$$

$$J_x = \frac{1}{2}\left(\gamma_x x^2 + 2\alpha_x x p_x + \beta_x p_x^2\right)$$

is a conserved quantity

$$\Delta\phi_x = \int_{s_1}^{s_2}\frac{1}{\beta_x}ds$$

$$\nu_x = \frac{\Delta\phi_x}{2\pi} = \frac{1}{2\pi}\int_0^{C_0}\frac{1}{\beta_x}ds$$



The Periodic Solution

$$R_2 = \begin{pmatrix} \cos\mu_x + \alpha_x \sin\mu_x & \beta_x \sin\mu_x \\ -\gamma_x \sin\mu_x & \cos\mu_x - \alpha_x \sin\mu_x \end{pmatrix}$$

This is always the form of the 1-turn matrix

$$R_2 = \begin{pmatrix} 1 - \frac{L^2}{2f_0^2} & \frac{L}{f_0}(L+2f_0) \\ \frac{L}{4f_0^3}(L-2f_0) & 1 - \frac{L^2}{2f_0^2} \end{pmatrix}$$

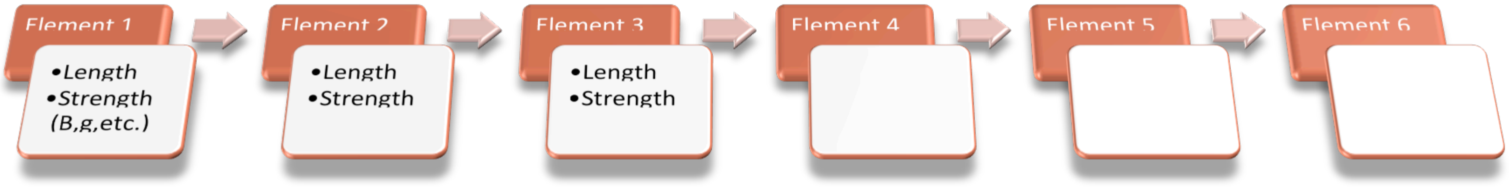
This is the particular form for our FODO example

For a real-valued phase advance (and therefore real-valued Twiss functions), the Trace of the transfer matrix must be less than 2.

- Don’t forget the other plane of motion – you can be stable in one plane and not in the other…
- In other words, you have to get the focusing strengths right.

So how do we do it in a code?

Accelerator codes simply assume a piecewise-continuous representation of the accelerator structure.



Element 1
- Length
- Strength (B,g,etc.)

Element 2
- Length
- Strength

Element 3
- Length
- Strength

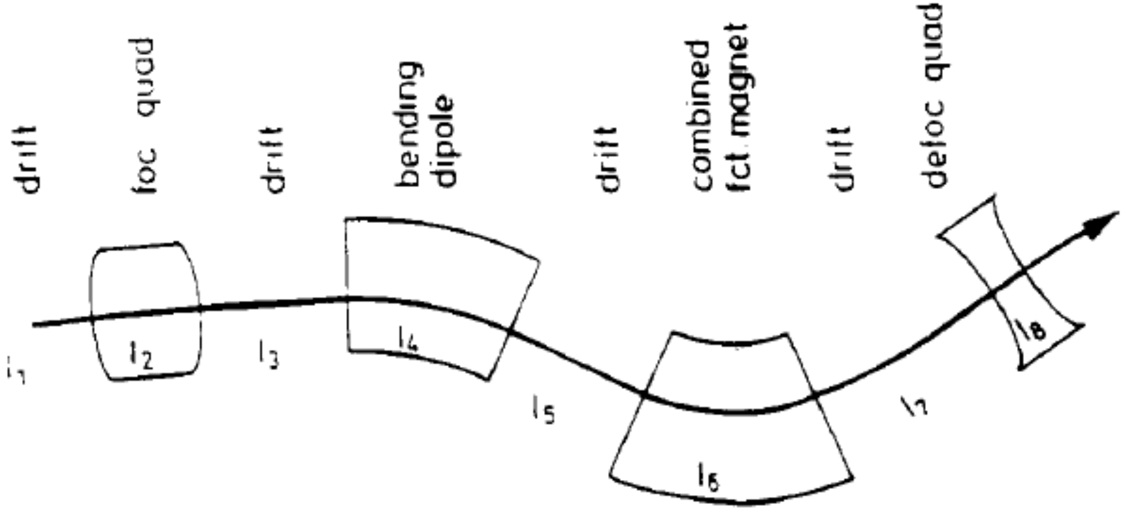
Element 4

Element 5

Element 6

$$R_T = R_n.R_{n-1}...R_3.R_2.R_1$$

but because of edge focusing the number of matrices is **not** the same as the number of elements.



Calculation of the Twiss values – periodic solution procedure

Linear optics codes work using the following procedure:

1. Parse lattice structure into R matrices
2. Calculate one-turn matrix: $R_T = R_n.R_{n-1}...R_3.R_2.R_1$
3. Determine stability in each plane using $Tr(R_2) < 2$

$$R_2 = \begin{pmatrix} \cos \mu_x + \alpha_x \sin \mu_x & \beta_x \sin \mu_x \\ -\gamma_x \sin \mu_x & \cos \mu_x - \alpha_x \sin \mu_x \end{pmatrix}$$

4. Calculate the periodic phase advance: $\mu = arccos[(R_{11} + R_{22})/2]$

5. Calculate the initial Twiss values: $\beta_0 = \dfrac{R_{12}}{sin(\mu)}$ $\alpha_0 = \dfrac{R_{11} - R_{12}}{2sin(\mu)}$ $\gamma_0 = \dfrac{1+\alpha_0^2}{\beta_0}$

   (Exercise: what happens when β_0 comes out negative?)

6. Propagate the Twiss values.

Propagating Twiss and phase values

This is easy…

$$T = A_2^{-1} = \begin{pmatrix} \beta & -\alpha \\ -\alpha & \gamma \end{pmatrix}$$

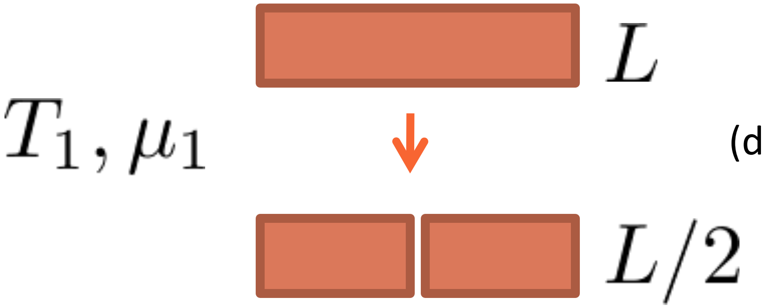
$$T_2 = R_{21}.T_1.R_{21}^T$$

and phases are propagated as

$$\mu_2 = arctan\left(\frac{R_{12}}{\beta_1 R_{11} - \alpha_1 R_{12}}\right)$$

the important thing to note here is that elements can be split…

(but the edge focusing is not split over dipoles)



(don’t have to split for the one-turn matrix)

(hint: you can put other calculations in when you do this)

Continuous versus discrete calculations

Most accelerator physics courses talk about **continuous** functions.

Of course, virtually all codes calculate **discrete** values using the linear matrix formalism.

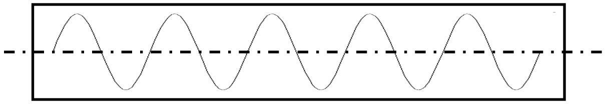
Depending on the code and/or the particular lattice, you can have problems. Most of these problems arise because of the oscillatory form of the trajectory solutions.

Example: Betatron tune through a long quadrupole

Continuously, we have $\nu(s) = \int_0^s \frac{ds}{\beta(s)}$

but in a discrete quadrupole the matrix calculation loses the integer part of the phase advance.

$$R = \begin{pmatrix} \cos\omega L & \frac{\sin\omega L}{\omega} & 0 & 0 & 0 & 0 \\ -\omega\sin\omega L & \cos\omega L & 0 & 0 & 0 & 0 \\ 0 & 0 & \cosh\omega L & \frac{\sinh\omega L}{\omega} & 0 & 0 \\ 0 & 0 & \omega\sinh\omega L & \cosh\omega L & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{L}{\beta_0^2\gamma_0^2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



Few codes check for this!

Dispersion

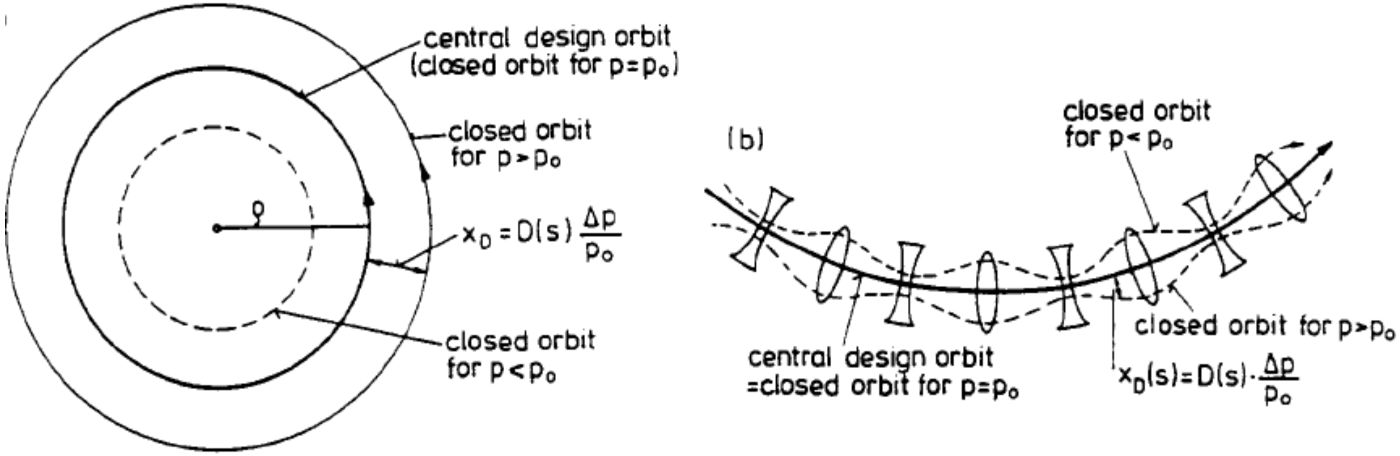
Dispersion is actually much easier than it sounds.

The dispersion function is that ray which has unit momentum deviation, i.e. $\Delta p/p = 1$

In other words, at any location through a lattice it can be described as a vector of the form:

$$\vec{D} = \begin{pmatrix} \eta_x \\ \eta_x' \\ \eta_y \\ \eta_y' \\ 0 \\ 1 \end{pmatrix}$$

Assuming a linear system, all other energies just scale from this value.



Propagating dispersion

Propagating dispersion is even easier…

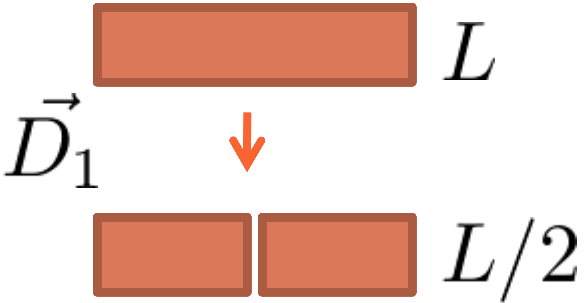
$$\vec{D}_2 = R_{21}.\vec{D}_1$$

(remember, the dispersion is just another ray/particle)

For a periodic system with one-turn 6x6 matrix R, it can be shown (!) that the periodic solution is:

$$\vec{D}_0 = \begin{pmatrix} \frac{(1-R_{22})R_{16}+R_{12}R_{26}}{(1-R_{11})(1-R_{22})-R_{21}R_{12}} \\ \frac{(1-R_{11})R_{26}+R_{21}R_{16}}{(1-R_{11})(1-R_{22})-R_{21}R_{12}} \\ \frac{(1-R_{44})R_{36}+R_{34}R_{46}}{(1-R_{33})(1-R_{44})-R_{43}R_{34}} \\ \frac{(1-R_{33})R_{46}+R_{43}R_{36}}{(1-R_{33})(1-R_{44})-R_{43}R_{34}} \\ 0 \\ 1 \end{pmatrix}$$

again, we can split elements when propagating…



Optical modules

There are many possible configurations of dipoles and quadrupoles that can give stable motion

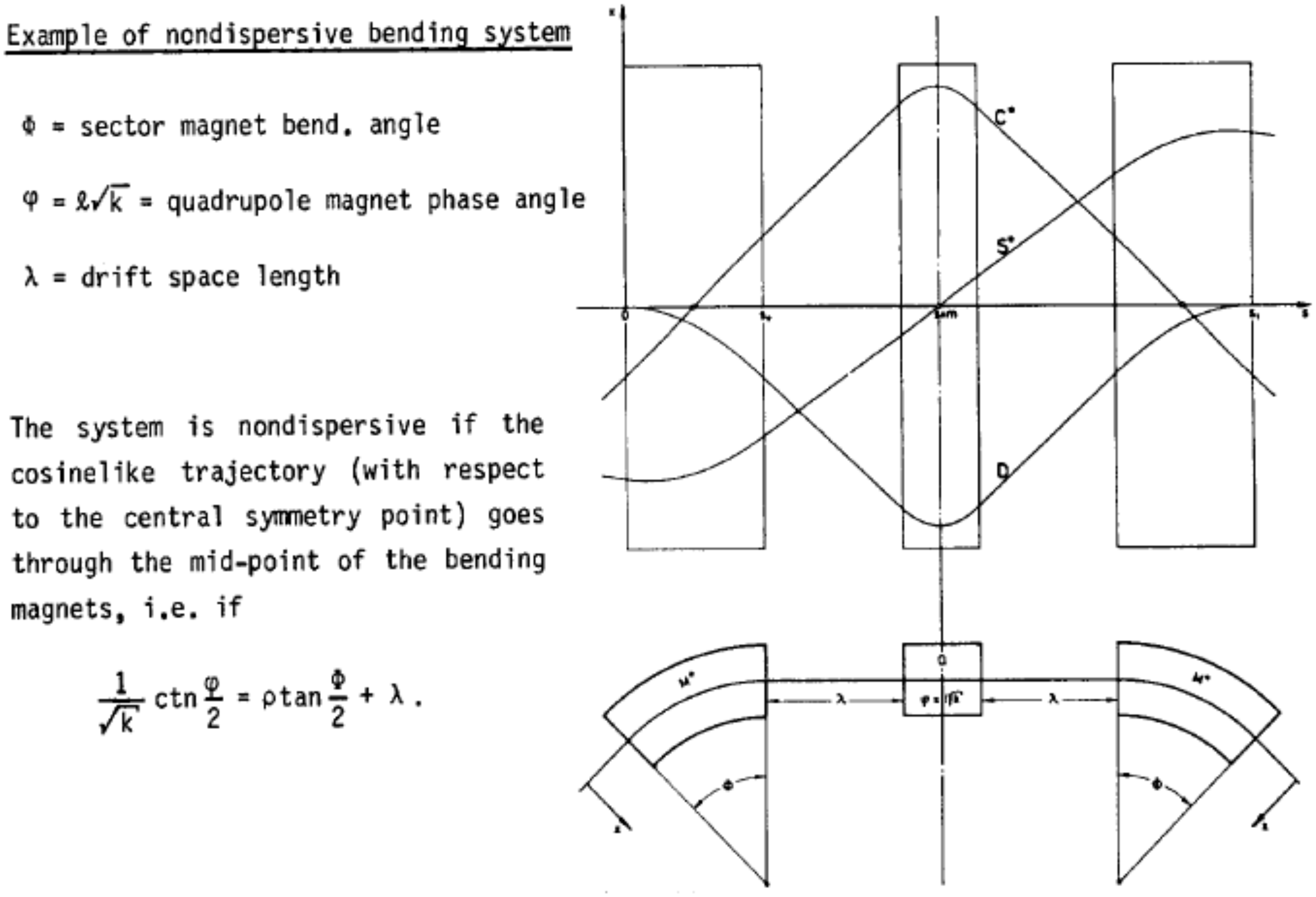
- In particular, we can talk about *dispersion-free* lattices, which are important in many applications
  - Colliders, SR sources
- Chasman-green, double-bend achromat, triple-bend achromat…

Nearly always, someone has worked out the rules for a typical optical module that does a particular job.

- It then needs adapting to your particular problem, e.g. taking account of space, beam energy, technological limitations/choices etc.

In the following slides we will see a few examples of such modules.

A non-dispersive bending system – the achromat



Example of nondispersive bending system

Φ = sector magnet bend. angle

$\varphi = \ell\sqrt{k}$ = quadrupole magnet phase angle

λ = drift space length

The system is nondispersive if the cosinelike trajectory (with respect to the central symmetry point) goes through the mid-point of the bending magnets, i.e. if

$$\frac{1}{\sqrt{k}}\,\mathrm{ctn}\,\frac{\varphi}{2} = \rho\tan\frac{\Phi}{2} + \lambda\,.$$

Fig. 14: Nondispersive deflecting system.

(from K. Steffen's excellent CAS lectures)

A non-dispersive translating system – the dog-leg

Example of nondispersive translating system

Φ = sector magnet bend. angle

$\varphi = \ell\sqrt{k}$ = quadrupole magnet phase angle

d, λ = drift space lengths.

The system is nondispersive if the sinelike trajectory (with respect to the central symmetry point) goes through the mid-point of the bending magnets, i.e. if

$$\rho \tan\frac{\Phi}{2} + \lambda = \frac{1}{\sqrt{k}} \; \frac{d\sqrt{k}\cos\varphi + 2\sin\varphi}{d\sqrt{k}\sin\varphi - 2\cos\varphi} .$$

Focusing also in the other plane may be obtained by adding a third quadrupole of opposite polarity at the symmetry point.

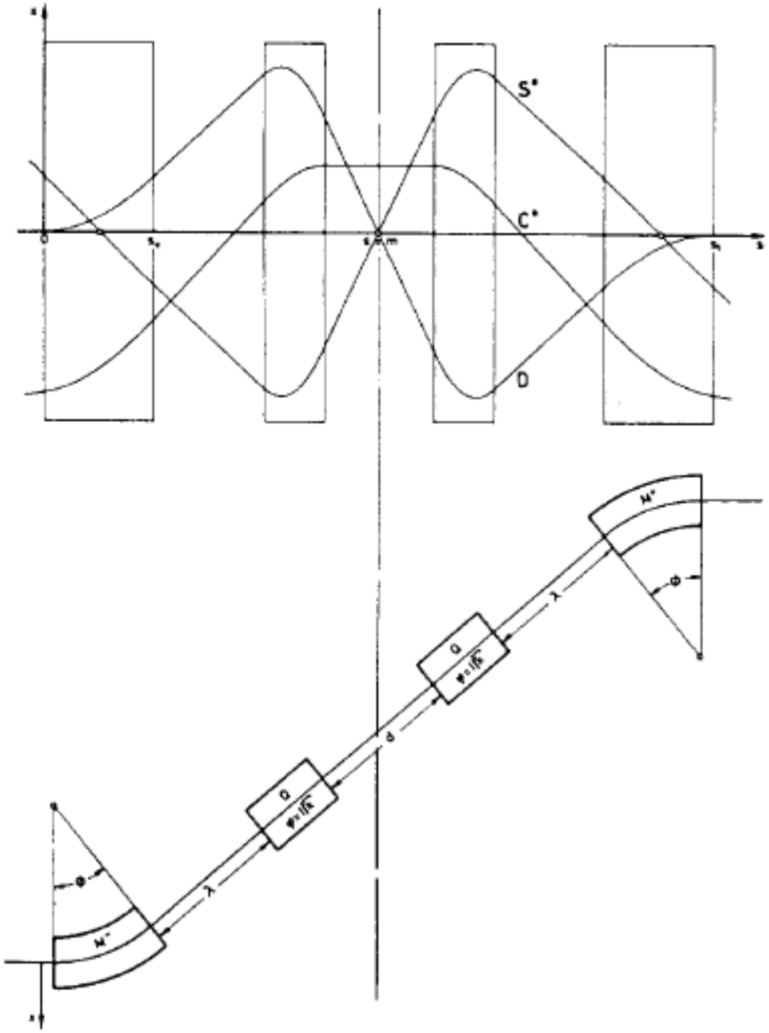
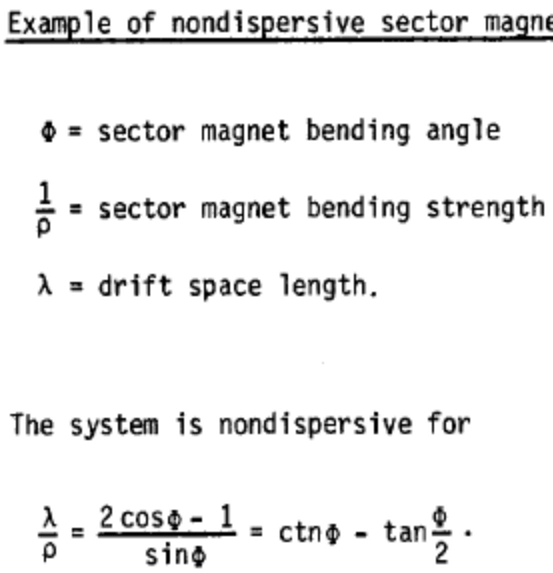


Fig. 15: Nondispersive translating system.

A non-dispersive bending system using only dipoles



Example of nondispersive sector magnet system

Φ = sector magnet bending angle

$\frac{1}{\rho}$ = sector magnet bending strength

λ = drift space length.

The system is nondispersive for

$$\frac{\lambda}{\rho} = \frac{2\cos\Phi - 1}{\sin\Phi} = \mathrm{ctn}\,\Phi - \tan\frac{\Phi}{2}\,.$$

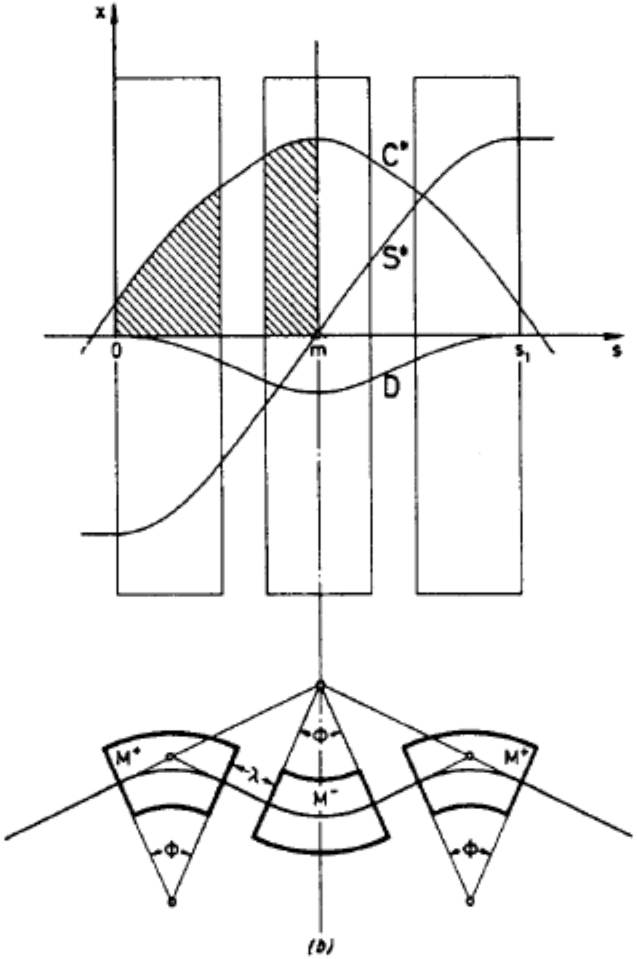


Fig. 17: Nondispersive sector magnet system.

A non-dispersive 4-magnet system – the chicane

Example of nondispersive rectangular magnet system

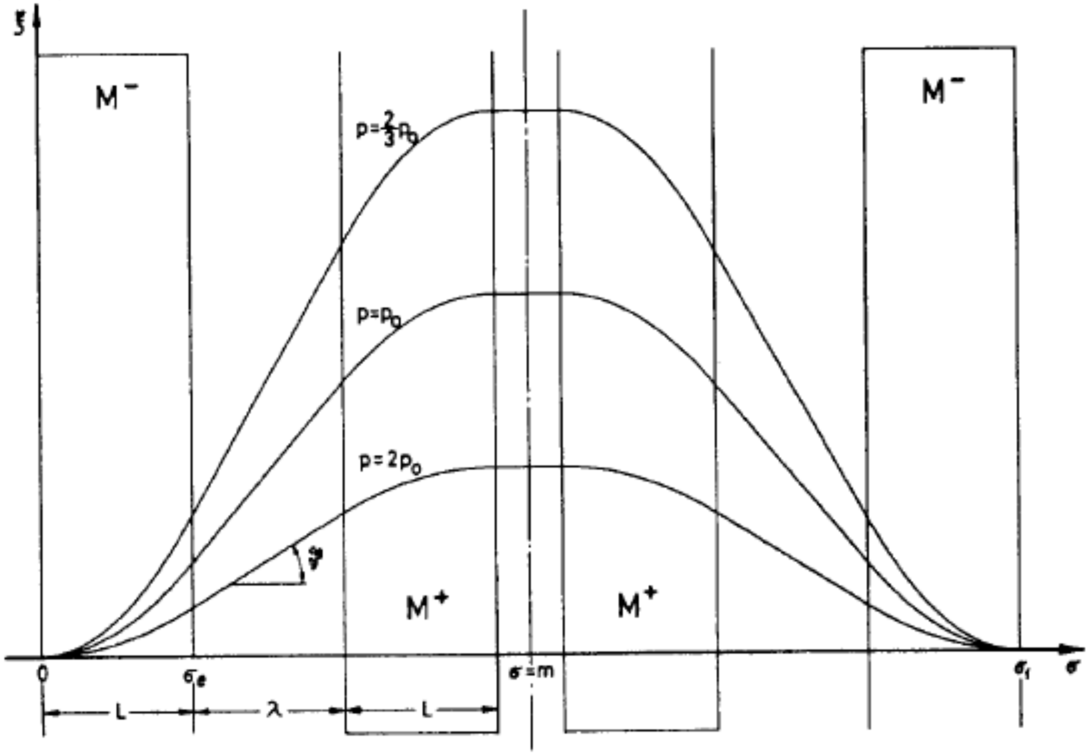


Fig. 18: Nondispersive momentum selecting system for large momentum spread.

Q: Why would you want one of these?

Exercise: derive the analytic value in such a system for R_{56}

An isochronous, achromatic bending system

Example of symmetric isochronous deflecting system

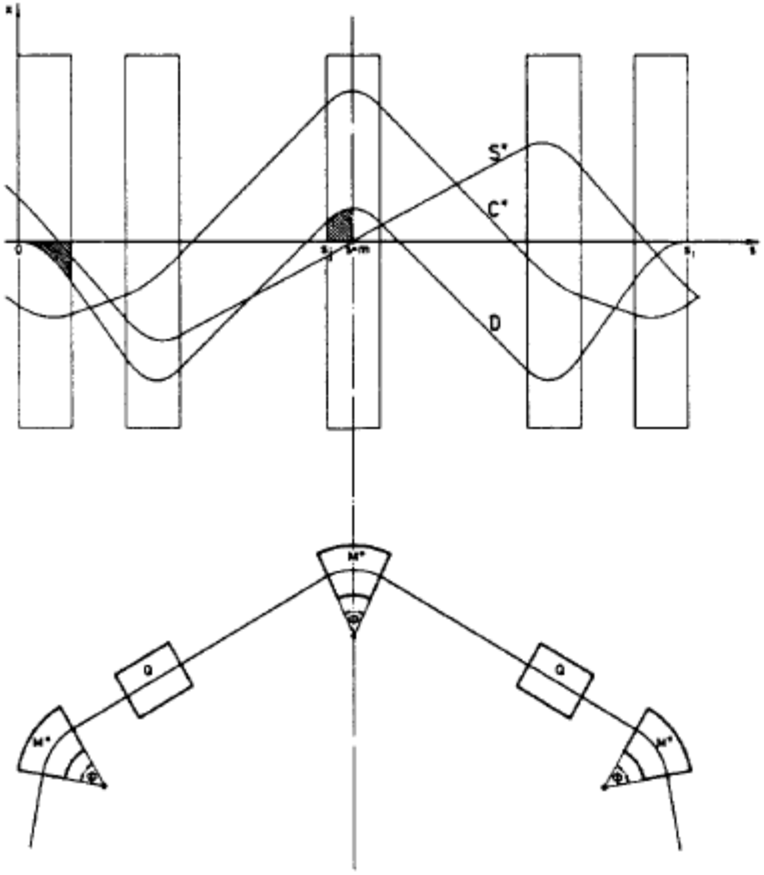


Fig. 19: Symmetric isochronous deflecting system.

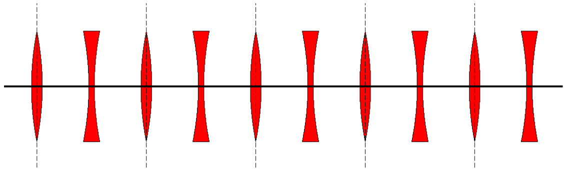
Periodic and aperiodic systems

There is no basic difference between a **periodic** and an **aperiodic** system.

The propagation of individual rays is identical, $\vec{x_2} = R.\vec{x_1}$

In an aperiodic system, initial Twiss and dispersion values must be supplied. The values obtained when propagated will then differ depending upon the initial values.

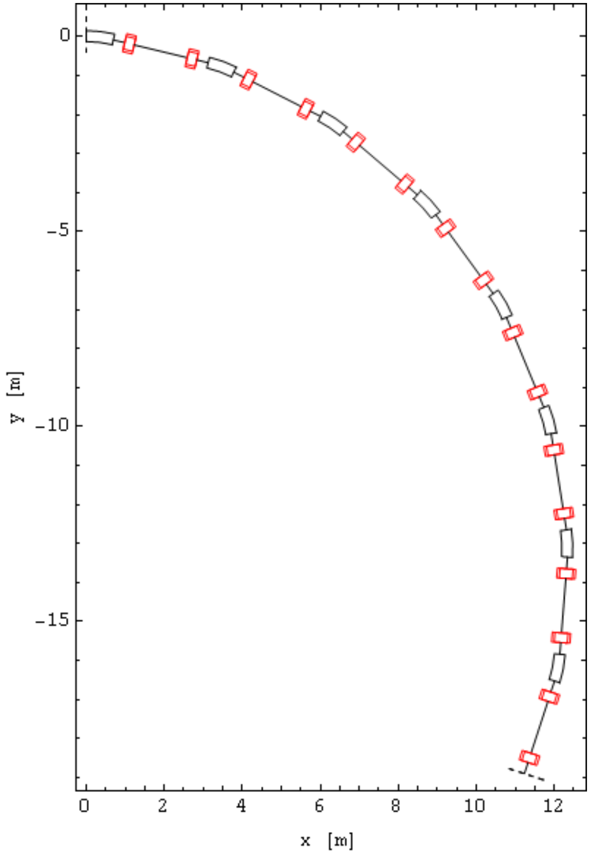
Example: a FODO channel is set up which has a particular periodic solution for the Twiss values. We propagate an initial (different) Twiss matrix through the system – the propagated values are **different**. This is called **mismatch**.

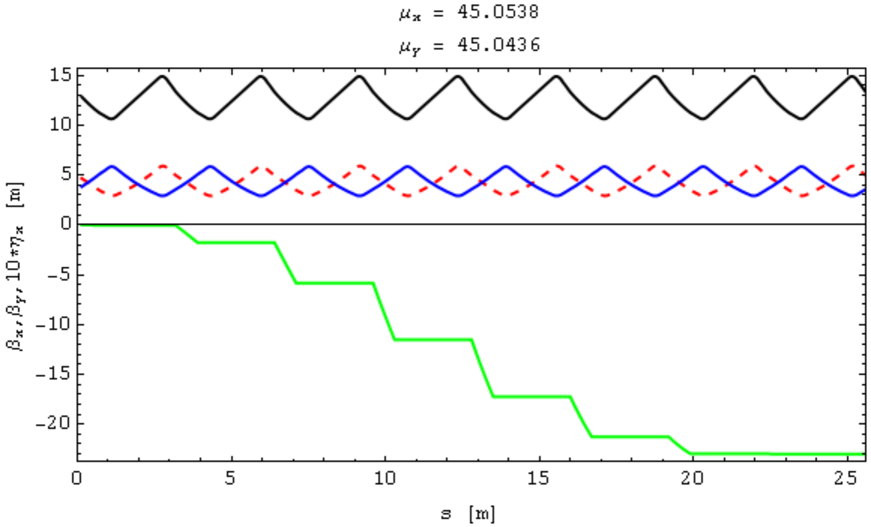


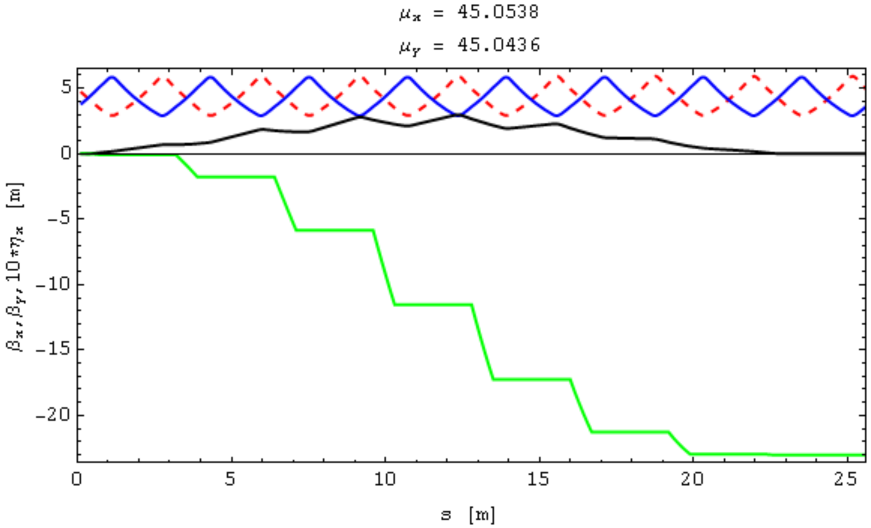
Exercise: Try it out for yourself!

The same thing is true of dispersion.

Deliberate dispersion mismatch – the dispersion suppressor







A FODO lattice with 45-degree phase advance

The Cockcroft Institute
of Accelerator Science and Technology

Real lattices and lattice design

Remember that when we design lattices, that eventually it will get built (hopefully!)

Reality uses up more space than ideal elements….

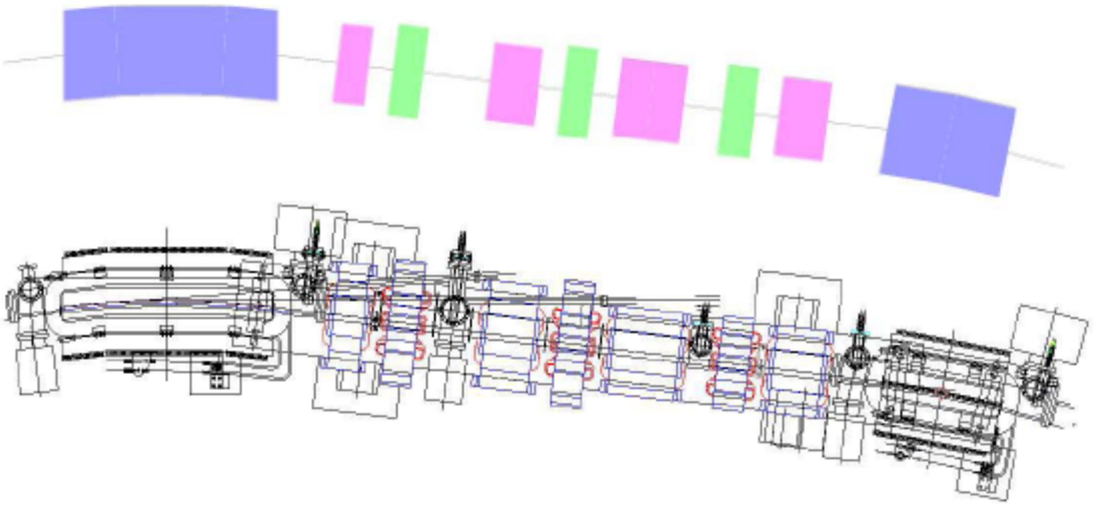
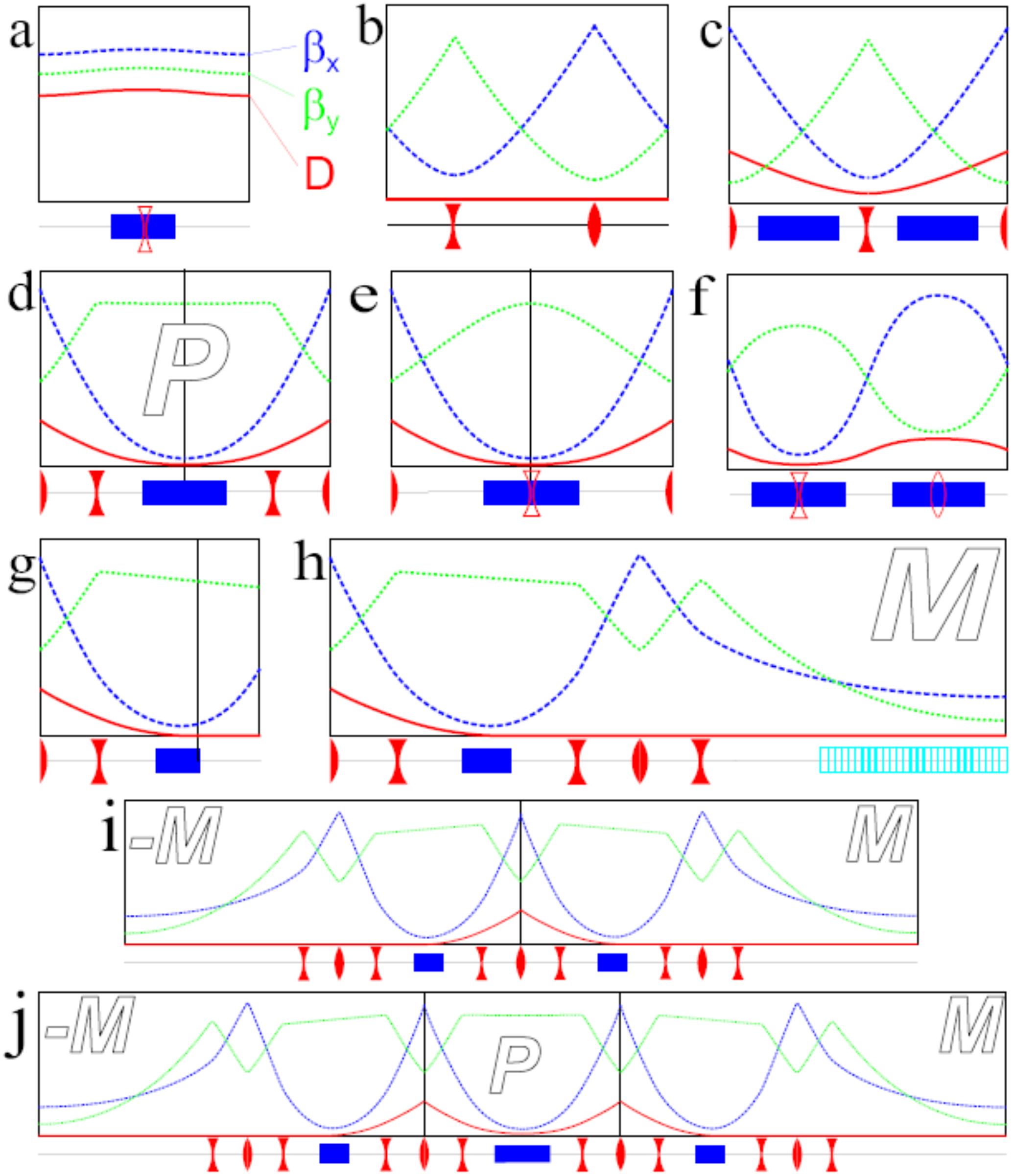


Fig. 1: Lattice sections as seen by the lattice designer (top) and the design engineer (bottom). Note how the space between ideal magnets is consumed by coils, beam-position monitors, absorbers, pumps, etc.

From simple to complex lattices



a: Weak focusing

b: FODO channel

c: FODO cell

d: Low-emittance cell

e: CF low-emittance cell

f: Low-emittance FODO

g: Dispersion match

h: Periodic dispersion match

i: Double-bend achromat

j: Triple-bend achromat

(see A.Streun's excellent course on low-emittance lattice design)

In the next lecture

In the next lecture, we will talk about taking an optical module, and adapting (i.e. improving) it for a particular purpose. This is often one of the principal jobs people use codes for.

The general method is called **matching**.