Terascale Accelerator School 2008

Solution to Exercises
Linear Accelerators

Exercise 1: Accelerator The given emittance ϵ of the particle source and its lateral extent define the angular divergence of the beam. At the origin both values are uncorrelated and $\sigma_{x'} = \epsilon/\sigma_x$.

a) The accelerator is constructed from individual FODO elements. An initial list is extended by nine identical FODO pieces.

```
FODO=list(F,O,D,O);
accelerator = list(F,O,D,O);
for k=1:9
  accelerator= lstcat( accelerator, FODO );
end
```

50 individual start positions are randomly chosen.

```
emittance=1.e-8;
sigmax=0.001;
sigmaxprime=emittance/sigmax;
N=50;  // define start positions
start=[];
for i=1:N
  x=[rand( N, 'normal')*sigmax; rand( N, 'normal')*sigmaxprime ];
  start = [start, x ];
end
```

The standard deviation *sigma* is calculated at each element of the accelerator. The particle position is evaluated from the transport matrix.

```
s=0.;
sx = 0.; sx2 = 0.;
for i=1:N
  x = start(:,i);
  sx = sx + x(1);
  sx2 = sx2 + x(1)*x(1);
end
sigma = sqrt( (sx2-sx*sx/N)/(N-1) )

for k=accelerator
  d=k*d;
  s=[s,d.s];
  sx = 0.; sx2 = 0.;
  for i=1:N
    x = d.matrix*start(:,i);
```

```
    sx = sx + x(1);
    sx2 = sx2 + x(1)*x(1);
  end
  sigma = [sigma, sqrt( (sx2-sx*sx/N)/(N-1) )];
end
```

The resulting *sigma* is shown in figure 1a) as a function of *s*.

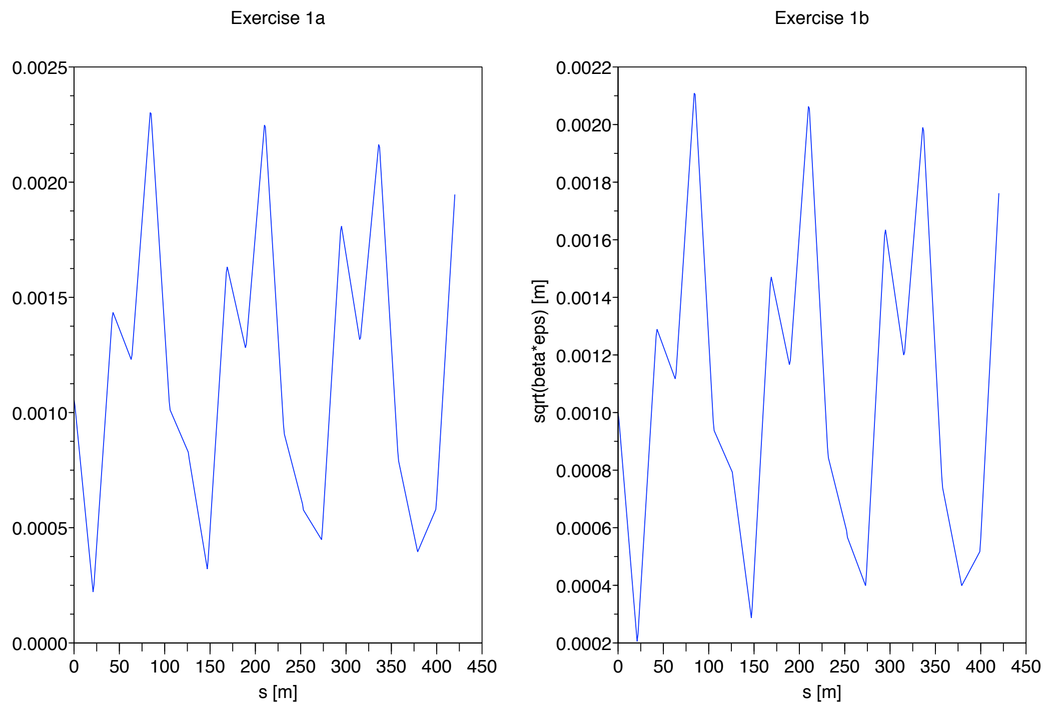

Abbildung 1: Comparison of the two approaches: a) standard deviation calculated from 50 particle trajectories; b) as extracted from the β-matrix.

b) Using the same script the β-matrix at the origin is given by

```
function [y]=BetaMatrix( alpha, beta )
y = [ beta, -alpha; -alpha, (1+alpha*alpha)/beta ];
endfunction
```

This matrix can be "transported" through the structures.

```
beta0 = BetaMatrix( 0., sigmax*sigmax/emittance );
beta = list(beta0);
s=0.;
d=OptElem( 0., [1 0; 0 1]);
for k=accelerator
  d=k*d;
  s=[s,d.s];
  beta=lstcat( beta, d.matrix*beta0*d.matrix');
end
```

beta defines a list of β-matrices at the points of the optical elements. The current matrix is evaluated as $B = MB_0M^T$. The matrix M describes the transport from the source to the current point on the trajectory. The size of the β-function at a given point is the $B_{1,1}$-element of the matrix. The width is $\sigma = \sqrt{B_{1,1} * \epsilon}$.

```
z=[];
for k=beta
  z = [z, sqrt(k(1,1)*emittance)];
end
```

The result *sigma* as a function *s* is shown in figure 1b) and agrees well with the emprical result of figure 1a).

Exercise 2: Twiss Parameter Insert a *pause* and *resume*-statement into the previous code. Extract the Twiss-parameters from the elements of the β-matrix. The values are shown in figure 2.

$$\begin{aligned} \alpha &= -B_{1,2} \\ \beta &= B_{1,1} \\ \gamma &= B_{2,2} \end{aligned}$$

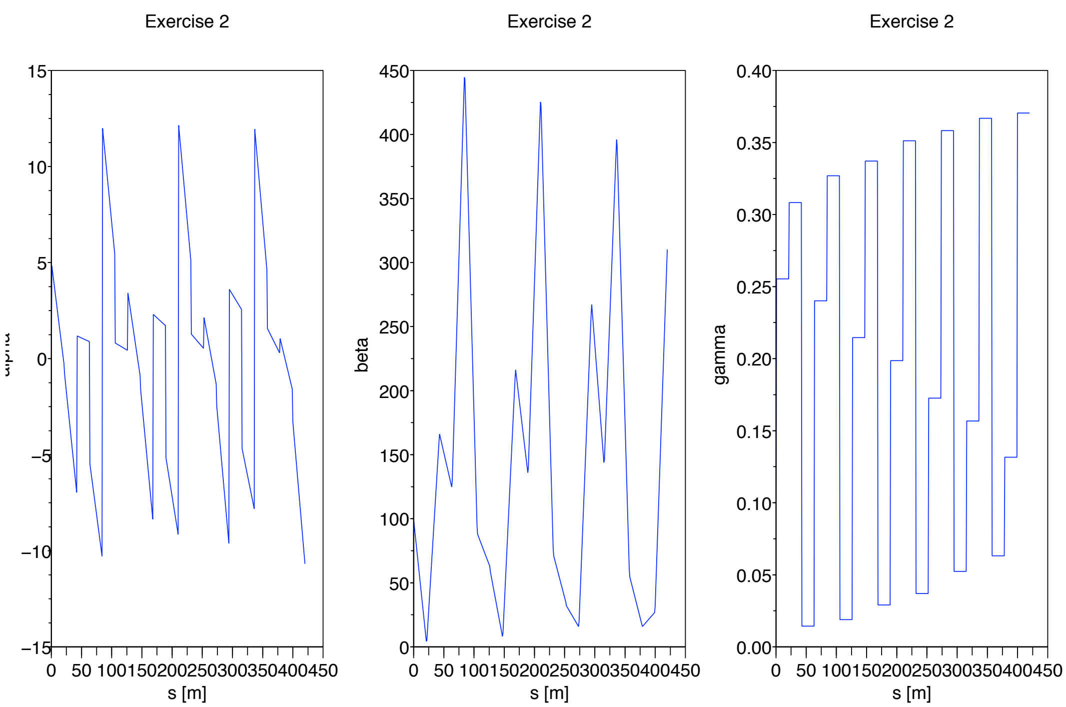

Abbildung 2: Twiss Parameter α, β und γ als Funktion des Weges durch die Struktur.

```
z=[];
```

```
for k=beta
  z = [z, [-k(1,2); k(1,1); k(2,2)]];
end
xbasc();
subplot(1,3,1);
xtitle( 'Exercise 2', 's [m]', 'alpha' );
plot(s, z(1,:) );
subplot(1,3,2);
xtitle( 'Exercise 2', 's [m]', 'beta' );
plot(s, z(2,:) );
subplot(1,3,3);
xtitle( 'Exercise 2', 's [m]', 'gamma' );
plot(s, z(3,:) );
```

Exercise 3: Position Accuracy The individual optical elements can be positioned to 300 μm in the lateral direction. The individual coordinates are hence properties of the individual quadrupoles and have to be incorporated in the definition. The initial assignment for the offsets is taken as 0.

```
// typed list definition of an optical element
function [y]=OptElem( s, mat )
  offset = [0.; 0.];
  y=tlist(['ele';'s';'matrix';'offset'],s, mat, offset)
endfunction
```

The transport of a particle through the magnet structure is calculated by an *overloaded* multiplication. The multiplication has to include the respective transformations into the coordinate system of the quadrupole.

```
// overloaded transport of a vector through an optical element
function [z]=%ele_m_s(a,x)
  z = a.matrix*(x-a.offset)+a.offset
endfunction
```

The positions of the individual quadrupoles have to be randomly chosen. The coordinates of focussing and defocussing magnets are calculated using Gaussian distributions.

```
sigma_survey = 0.0003;  // position accuracy

//  define generic components
ds=20;qs=1.; k=0.05;
O=OptElem( ds, Drift(ds) );
F=OptElem( qs, Quadrupole(-k, qs ) );
D=OptElem( qs, Quadrupole(k, qs) );
d=OptElem( 0., [1 0; 0 1] );

F.offset = [rand( sigma_survey, 'normal' ) * sigma_survey; 0.];
```

```
D.offset = [rand( sigma_survey, 'normal' ) * sigma_survey; 0.];
FODO = list( F, O, D, O );
accelerator = FODO;
N = 200 - 1;
for k=1:N
  F.offset = [rand( sigma_survey, 'normal' ) * sigma_survey; 0.];
  D.offset = [rand( sigma_survey, 'normal' ) * sigma_survey; 0.];
  FODO = list( F, O, D, O );
  accelerator= lstcat( accelerator, FODO );
end
```

The particle starts at $(0, 0)$ and is sequentially transported through the structures. The results is shown in figure 3.

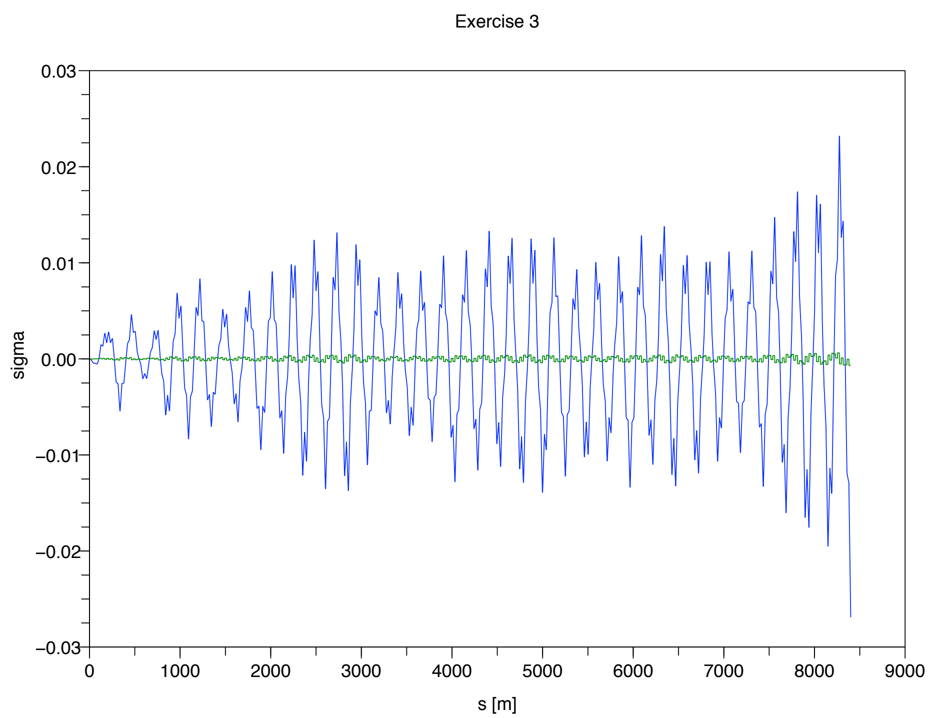

Abbildung 3: Transport through 200 FODO elements and a lateral position accuracy of 300 μm.