ETEAPOT EDM Benchmark-II: Updated Results: Transfer Matrices and Twiss Functions

J. D. Talman and R. M. Talman

November 11, 2019

Abstract

In this chapter ETEAPOT is used to obtain transfer matrices between arbitrary lattice positions, and, from them, Twiss functions as functions of longitudinal coordinate s. The same three benchmark lattices as in the November 6, 2019, "ETEAPOT EDM Benchmark I: Updated Results for Proton EDM Benchmark Lattices" chapter are analysed. The lattices have field indices $m = -1$, $m \approx 0$, and $m = 1$. Once-around transfer matrices were obtained for the Benchmark-I report, and used to determine beta functions at the origin as well as transverse tunes. But general transfer matrices had not yet been obtained. Also the discussion of aliasing ambiguities affecting the determination of integer tune values was incomplete. Results are compared to values obtained from a linearized MAPLE transfer matrix formalism. In all cases the design vertical tunes have been adjusted to Q_y=0.2.

1 Parameters of Benchmark Lattices

Table 1: Parameters of benchmark all-electric EDM lattices, and Twiss parameters calculated by MAPLE from linearized transfer matrices.

file name	variable name	unit	E_BM_M1.0.sxf	E_BM_Z.sxf	E_BM_P1.0.sxf
cells/arc	NCellPerArc		20	20	20
bend radius	r0	m	40.0	40.0	40.0
half drift length	Ldh	m	1.0	1.0	1.0
half bend per cell	Thetah	r	0.078539816	0.078539816	0.078539816
half bend length	Leh	m	3.141592	3.141592	3.141592
circumference	circum	m	331.327	331.327	331.327
inverse focal length	q	1/m	-0.002019	-0.00005960	0.0019075
field index	m		-1.0	1.0e-10	1.0
horizontal beta	betax	m	35.9237	36.1018	36.1910
vertical beta	betay	m	264.182	263.620	262.237
horizontal tune	Qx		1.4605	1.4578	1.4588
vertical tune	Qy		0.20024	0.20004	0.20047

2 Determination of Twiss Functions From Transfer Matrices

2.1 Analysis of the Once-Around Transfer Matrix at the Origin

In this section x and y and ct subscripts will be suppressed, and only transverse evolution is to be discussed. The most general transfer matrix is a six-by-six matrix $\mathbf{M}(s_i, s_j)$, which propagates a phase space vector $\mathbf{x}(s_i)$ at s_i, to its value $\mathbf{x}(s_j)$ at s_j;

$$\mathbf{x}(s_j) = \mathbf{M}(s_i, s_j)\,\mathbf{x}(s_i). \tag{1}$$

ETEAPOT starts by assigning coordinates at the origin, $s_0 = 0$, to the particles in a standard bunch (as described earlier), and then evolves the standard bunch and records the coordinates $\mathbf{x}(s_i)$ at all points s_i. From these results the transfer matrices $\mathbf{M}(0, s_i)$ can be calculated (also as described earlier). By definition

$$\mathbf{M}(0, 0) = \mathbf{I}, \tag{2}$$

where $\mathbf{I}$ is the 6×6 identity matrix.

To extract Twiss lattice functions one needs periodic, "once-around" transfer matrices, distinguished by overhead tildes, and defined by

$$\widetilde{\mathbf{M}}(s_i) \equiv \mathbf{M}(s_i, s_i + \mathcal{C}_0) = \mathbf{M}(0, s_i)\,\mathbf{M}(s_i, \mathcal{C}_0). \tag{3}$$

where $\mathcal{C}_0$ is the circumference of the design orbit; the final step has been taken because knowledge of $s_i + \mathcal{C}_0$ requires tracking for more than one complete turn, but we are assuming that tracking has been done only for exactly one complete turn. Propagation from $s = \mathcal{C}_0$ to $\mathcal{C}_0 + s_i$ is the same as propagation from $s = 0$ to s_i. The Twiss parameterization of (one partitioned diagonal 2×2 block of) such a once-around, symplectic transfer matrix is

$$\widetilde{\mathbf{M}}(s_i) = \begin{pmatrix} \cos\mu + \alpha \sin\mu & \beta \sin\mu \\ -\frac{1+\alpha^2}{\beta} \sin\mu & \cos\mu - \alpha \sin\mu \end{pmatrix}. \tag{4}$$

Extraction of the α_0 and β_0, the Twiss parameters at the origin, can start from

$$\cos\mu = \frac{1}{2}\left(\widetilde{\mathbf{M}}_{11}(0) + \widetilde{\mathbf{M}}_{22}(0)\right), \tag{5}$$

which fixes $\cos\mu$. Because of sign ambiguity, this determines only $|\sin\mu|$. One also has the relations

$$\beta_0 = \left|\frac{\widetilde{\mathbf{M}}_{12}(0)}{\sin\mu}\right|. \tag{6}$$

and

$$\alpha_0 = \frac{1}{2\sin\mu}\left(\widetilde{\mathbf{M}}_{11}(0) - \widetilde{\mathbf{M}}_{22}(0)\right). \tag{7}$$

With β being positive by convention, from the 1,2 element, $\mathrm{sign}(\sin\mu)$ can be seen to be the same as $\mathrm{sign}(\widetilde{\mathbf{M}}_{12})$. With $\cos\mu$ known, this fixes $\sin\mu$. Together, these relations fix $\sin\mu$, $\cos\mu$, α_0, and β_0.

Conventionally one also introduces a third Twiss parameter

$$\gamma_0 = \frac{1 + \alpha_0^2}{\beta_0}, \tag{8}$$

which can be obtained once β_0 and α_0 have been determined.

Because of the multiple-valued nature of inverse trig functions, these relations do not determine a unique value for μ. They do, however, determine the quadrant in phase space in which the angle μ resides. For $\mathrm{sign}(\sin\mu)>0$ the angle μ resides in the first or second quadrant, in which case the fractional tune is less than 1/2; otherwise the fractional tune is greater than 1/2. For $\mathrm{sign}(\cos\mu)>0$ the angle μ

resides in the first or fourth quadrant, in which case the fractional tune is below 1/4 or above 3/4. These considerations fix the fractional parts of the tunes.

An "aliasing" or "integer-tune" ambiguity remains, however, which cannot, even in principle, be obtained from the once-around matrix. Only if both transverse tunes are less than 1 (which is hardly ever the case) would the tunes be equal to the fractional tunes that have been determined. In general, to obtain the integer tunes, it is necessary to analyse the turn by turn data at sufficiently closely-space intermediate points in the lattice. (See examples in the previous chapter.)

2.2 Evolving the Twiss Functions Around the Ring

To find the Twiss parameters at an arbitrary location s_i in the ring requires the once-around transfer matrix $\widetilde{\mathbf{M}}(s_i)$. This can obtained most compactly by multiplying the equation

$$\mathbf{M}(0, s_j) = \mathbf{M}(s_i, s_j)\, \mathbf{M}(0, s_i) \tag{9}$$

on the right by $\mathbf{M}^{-1}(0, s_i)$ to produce

$$\mathbf{M}(s_i, s_j) = \mathbf{M}(0, s_j)\, \mathbf{M}^{-1}(0, s_i). \tag{10}$$

Substituting this with $s_j = \mathcal{C}_0$ into Eq. (**??**) produces

$$\widetilde{\mathbf{M}}(s_i) = \mathbf{M}(0, s_i)\, \widetilde{\mathbf{M}}(0)\, \mathbf{M}^{-1}(0, s_i). \tag{11}$$

Having obtained $\widetilde{\mathbf{M}}(s_i)$, the procedure described in the previous subsection can then be used to obtain $\alpha(s_i)$ and $\beta(s_i)$. But the integer tune ambiguity can, again, not be resolved. To resolve this ambiguity both x and y phases have to be tracked continuously through the lattice, requiring that they advance continuously and monotonically. (Later, at least in principle, the same ambiguity will have to be faced for longitudinal motion. But the integer longitudinal tune is almost always zero, so the problem is usually absent in the longitudinal case.)

ETEAPOT requires the lattice description to be in the form of an `.sxf` file. To be "legal" the granularity of such a file has to be fine enough that no phase can advance by more than a quarter integer through any element in the file. Before working out the α and β function evolution, ETEAPOT first works out the total phase advances from the origin to every node specified by the `.sxf` file (or, if some elements are sliced more finely, by every node after slicing).

There is an alternative way of finding the betatron phase advances. It starts with a Twiss parameterization of $\mathbf{M}(0, s)$ from the origin to an arbitrary position s in the lattice;

$$\mathbf{M}(0, s) = \begin{pmatrix} \sqrt{\frac{\beta(s)}{\beta_0}}\Big(\cos\psi(s) + \alpha_0 \sin\psi(s)\Big) & \sqrt{\beta_0\beta(s)}\, \sin\psi(s) \\ \cdot & \sqrt{\frac{\beta_0}{\beta(s)}}\Big(\cos\psi(s) - \alpha(s) \sin\psi(s)\Big) \end{pmatrix}. \tag{12}$$

The 2,1 element is quite complicated; it is not shown here since it will not be needed for the following analysis. Dividing the 1,1 element by the 1,2 element produces

$$\frac{\mathbf{M}_{1,1}(0, s)}{\mathbf{M}_{1,2}(0, s)} = \frac{\sqrt{\frac{\beta(s)}{\beta_0}}\Big(\cos\psi(s) + \alpha_0 \sin\psi(s)\Big)}{\sqrt{\beta_0\beta(s)}\, \sin\psi(s)} = \frac{\cot\psi(s) + \alpha_0}{\beta_0}. \tag{13}$$

Rearranging this equation produces

$$\psi(s) = \tan^{-1} \frac{\mathbf{M}_{1,2}(0, s)}{\beta_0 \mathbf{M}_{1,1}(0, s) - \alpha_0 \mathbf{M}_{1,2}(0, s)}. \tag{14}$$

Like all inverse trigonometric formulas, this equation has multiple solutions. But, with the `.sxf` granularity being required to be fine enough, one can (in principle) sequentially obtain unique phases. With $\psi(s)$ starting from $\psi(0) = 0$, as s increases from s_i to s_{i+1} there is a unique solution of Eq. (**??**),

$\psi(s_i) \geq \psi(s_{i-1})$ such that the function $\psi(s)$ increases monotonically, as required. Because the interval from $s_{i=1}$ to s_i is non-zero, ψ will, superficially, advance discontinuously; the correct solution is the least discontinuous. The same calculation has to be done for both the x and y betatron sectors. Unfortunately, even though, theoretically, the phase advances monotonically, numerical errors can cause Eq. (**??**) to give local phase decrease. This can cause Eq. (**??**) to give occasionally erratic results.

2.3 Lattice E_BM_P1.0, $m = 1$, "spherical" bending results

Figures **??** through **??** contain evaluations for the "benchmark" lattice `E_BM_P1.0_sl4.sxf`. which has field index $m = 1$; this is the "Kepler" or "spherical" case with $E_r \sim 1/r^2$. Agreement between EPEAPOT and MAPLE is excellent.

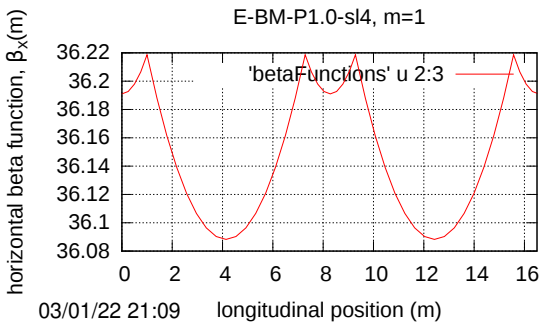

Figure 1: β_y calculated by ETEAPOT for lattice "E_BM_P1.0_sl4".

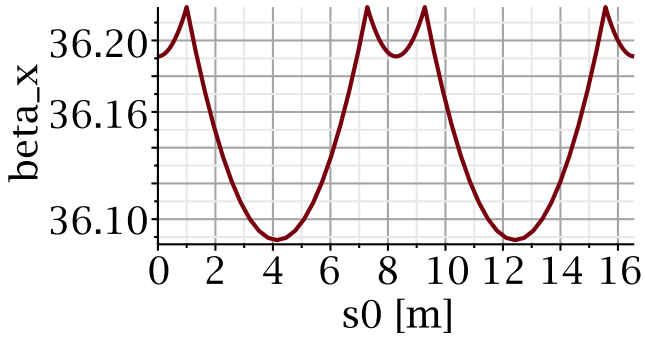

Figure 2: β_y calculated by MAPLE linearized theory for lattice "E_BM_P1.0_sl4".

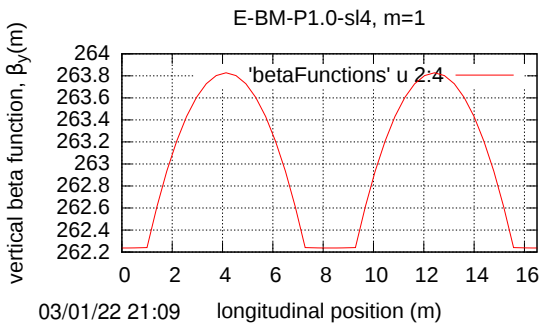

Figure 3: β_y calculated by ETEAPOT for lattice "E_BM_P1.0_sl4".

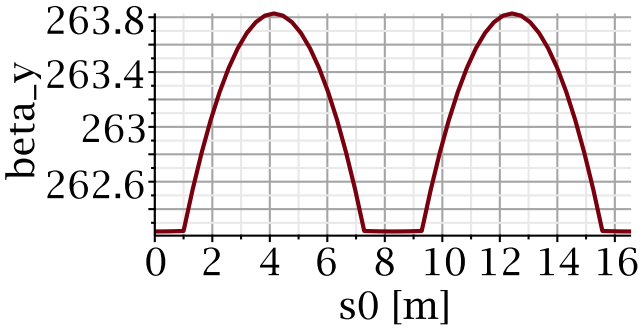

Figure 4: β_y calculated by MAPLE linearized theory for lattice "E_BM_P1.0_sl4".

2.4 Lattice E_BM_Z, $m = 0$, "cylindrical" bending results

Figures **??** and **??** compare β_x evaluations for the "benchmark" lattice `E_BM_Z_sl4.sxf` which has field index $m = 0$. This is the lattice closest to "cylindrical" with lumped quadrupoles just barely strong enough to move the lattice away from the boundary between vertical stability and vertical instability. The figure on the left is evaluated by ETEAPOT, the one on the right by the linearized (Wollnik) formulas re-derived in an appendix [**?**]. There is agreement to excellent accuracy on local beta function dependence $\beta_x(s)$. Similarly good agreement for vertical beta function $\beta_y(s)$ is exhibited in Figures **??** and **??**;

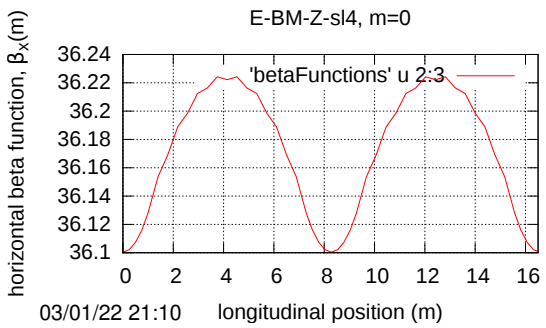

Figure 5: $\beta_x(s)$ calculated by ETEAPOT for lattice "E_BM_Z_sl4".

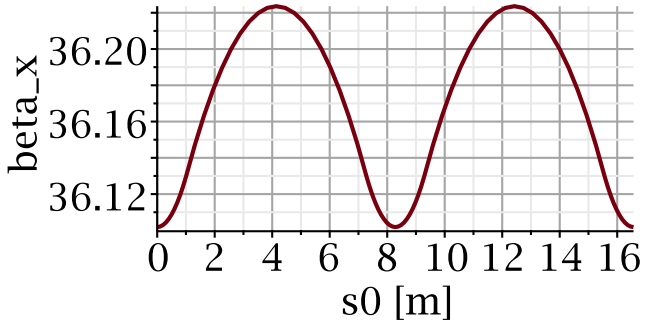

Figure 6: $\beta_x(s)$ calculated by MAPLE linearized theory for lattice "E_BM_Z_sl4".

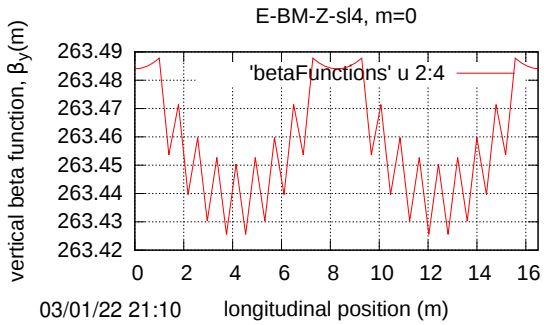

Figure 7: $\beta_y(s)$ calculated by ETEAPOT for lattice "E_BM_Z_sl4".

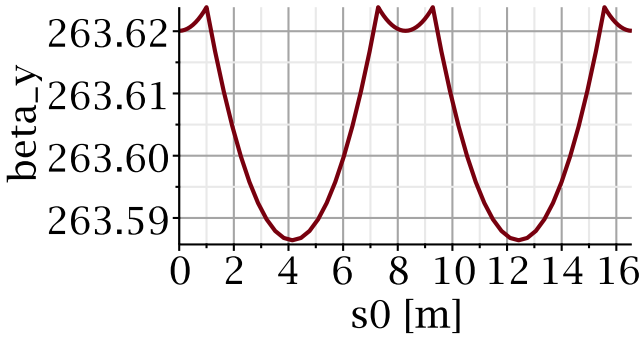

Figure 8: $\beta_y(s)$ calculated by MAPLE linearized theory for lattice "E_BM_Z_sl4".

2.5 Lattice E_BM_M1.0, $m = -1$, bending results

Figures **??** through **??** contain evaluations for the "benchmark" lattice `E_BM_M1.0_sl4.sxf`. which has field index $m = -1$; in this case the radial electric field E_r is independent of radial displacement $x \approx r - r_0$. Agreement is again excellent.

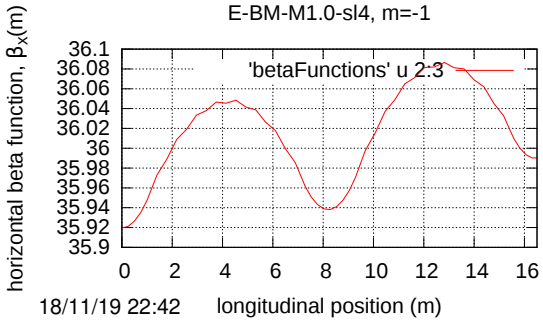

Figure 9: β_x calculated by ETEAPOT for lattice "E_BM_M1.0_sl4".

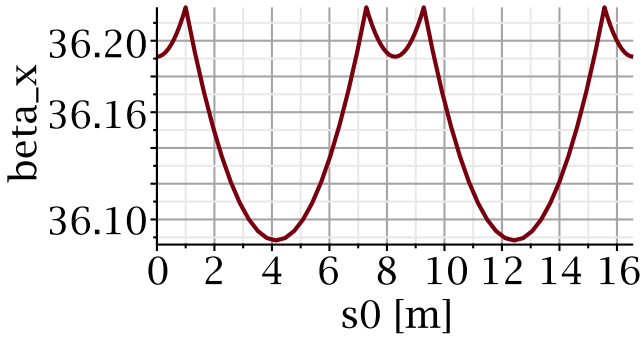

Figure 10: β_x calculated by MAPLE linearized theory for lattice "E_BM_M1.0_sl4".

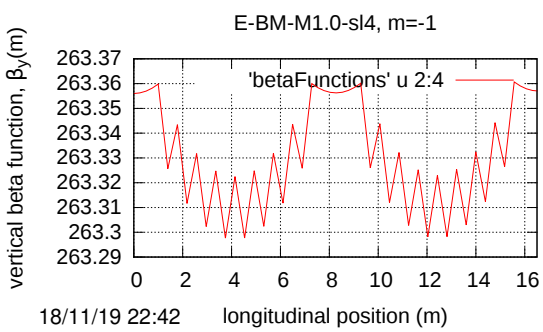

Figure 11: β_y calculated by ETEAPOT for lattice "E_BM_M1.0_sl4".

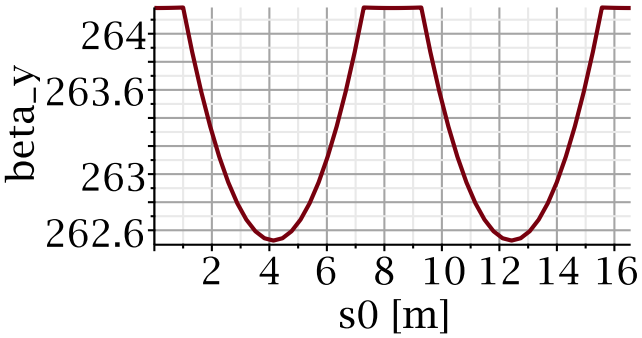

Figure 12: β_y calculated by linearized theory for lattice "E_BM_M1.0_sl4".

Figure **??** compares different ETEAPOT results for lattices `E_BM_Z_sl4.sxf` and `E_BM_Z.sxf`. These are actually the identical lattice treated two different ways. The "_sl4" suffix in the first file name indicates that the bend elements have been split into four slices in the (external) `sxf` file. In the other case the bends split into the same four slices, but ETEAPOT performs the slicing (internally) called for by the `splitForBends=2` parameter. The apparent disagreement is caused by a bug in the plotting procedure which joins points by straight lines. NOTE to JDT: either we fix the bug or suppress the plot? The actual beta function evaluations agree perfectly, at least to the accuracy of the plot.

The following tables give parameter values for the three benchmark lattices.

References

[1] J.D. Talman and R.M. Talman, *UAL/ETEAPOT Results (Augmented) for Proton EDM Benchmark Lattices,* BNL internal report, April 29, 2012

[2] N. Malitsky, J. Talman, and R. Talman, *Appendix UALcode: Development of the UAL/ETEAPOT Code for the Proton EDM Experiment,* UAL/ETEAPOT documentation (frequently revised), August, 2012

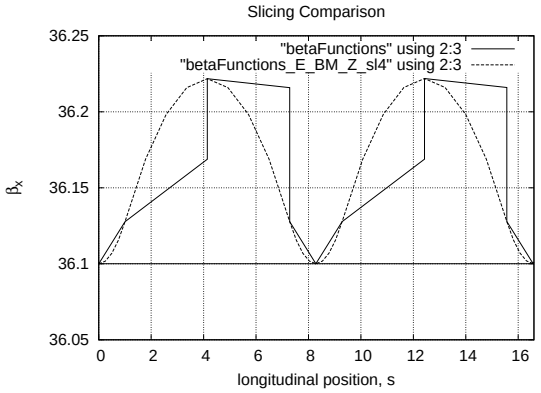

Figure 13: Comparison of external slicing (via the `sxf` file) and UAL-internal slicing, via the `splitForBends=2` parameter. Only selected points (connected by straight lines) are plotted in the internal slicing case. This accounts for the different local appearance of the plots.

Table 2: E_TEAPOT comparisons for `E_BM_P1.0.sxf`.

file name	variable name	unit	linearized	2 slices/bend	
cells/arc	`NCellPerArc`		20		
bend radius	`r0`	m	40.0		
half drift length	`Ldh`	m	1.0		
half bend per cell	`Thetah`	r	0.078539816		
half bend length	`Leh`	m	3.141592		
circumference	`circum`	m	331.327		
inverse focal length	`q`	1/m	0.0019075		
field index	`m`		1.0		
horizontal beta	`betax`	m	36.1910	36.1910	
vertical beta	`betay`	m	262.237	262.2370	
horizontal tune	`Qx`		1.4588	1.4588	
vertical tune	`Qy`		0.20047	0.2005	

[3] Storage Ring EDM Collaboration, *A Proposal to Measure the Proton Electric Dipole Moment with* 10^{-29} *e-cm Sensitivity,* especially Appendix 1. October, 2011

[4] C. Møller, *The Theory of Relativity,* Clarendon Press, Oxford, 1952,

[5] G. Muñoz and I. Pavic, *A Hamilton-like vector for the special-relativistic Coulomb problem,* Eur. J. Phys. **27**, 1007-1018, 2006

[6] R. Talman, *Geometric Mechanics,* John Wiley and Sons, 2000

[7] J. Aguirregabiria et al., Archiv:physics/0407049v1 [physics.ed-ph] 2004,

[8] U. Torkelsson, Eur. J. Phys., **19**, 459, 1998,

[9] T. Boyer, Am. J. Phys. **72** (8) 992, 2004

Table 3: E_TEAPOTComparisons for `E_BM_Z.sxf`.

file name	variable name	unit	linearized	2 slices/bend	
cells/arc	`NCellPerArc`		20		
bend radius	`r0`	m	40.0		
half drift length	`Ldh`	m	1.0		
half bend per cell	`Thetah`	r	0.078539816		
half bend length	`Leh`	m	3.141592		
circumference	`circum`	m	331.327		
inverse focal length	`q`	1/m	-0.00005960		
field index	`m`		1.0e-10		
horizontal beta	`betax`	m	36.1018	36.0795	
vertical beta	`betay`	m	263.620	261.4688	
horizontal tune	`Qx`		1.4578	1.4581	
vertical tune	`Qy`		0.20004	0.2018	

Table 4: E_TEAPOT comparisons for `E_BM_M1.0.sxf`

file name	variable name	unit	linearized	2 slices/bend	
cells/arc	`NCellPerArc`		20		
bend radius	`r0`	m	40.0		
half drift length	`Ldh`	m	1.0		
half bend per cell	`Thetah`	r	0.078539816		
half bend length	`Leh`	m	3.141592		
circumference	`circum`	m	331.327		
inverse focal length	`q`	1/m	-0.002019		
field index	`m`		-1.0		
horizontal beta	`betax`	m	35.9237	35.8566	
vertical beta	`betay`	m	264.182	251.8522	
horizontal tune	`Qx`		1.4605	1.4620	
vertical tune	`Qy`		0.20024	0.2102	