BNL-223631-2022-TECH
C-A/AP/677

FETS-VFFA Cell using Contour-Based Magnets

S. Brooks

February 2020

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy
USDOE Office of Science (SC), Nuclear Physics (NP) (SC-26)

DISCLAIMER

FETS-VFFA Cell using Contour-Based Magnets

Stephen Brooks, Shinji Machida

February 28, 2020

1 Lattice Cell

A pure-conductor winding arrangement has been found that approximates the optics of the design given in [1] (Figure 1). The tunes are not the same but are reasonably flat and exist through the entire 3–12 MeV energy range, so could conceivably be fine-tuned using the vertical winding density function in the next section.

The lattice consists of two magnets: a reverse bending Bd magnet followed by a normal bending Bf magnet. These magnets are both sectors whose midpoints are displaced with respect to the midpoints of the sides of a 20-sided polygon with side length 1.25 m. The ring is said to have 10 cells (2 polygon sides per cell) that are 2.5 m long. The displacements are 17 cm inwards for Bd and 17 cm outwards for Bf.

Table 1: Lattice Parameters

Parameter	**Value**	**Units**
Particle	Protons	
Kinetic energy range	3–12	MeV
Cells	10	
Polygon circumference	25	m
Cell angle	-36	° (clockwise)
Polygon sides	20	
Polygon side length	1.25	m
-Bd/Bf	0.44	
k	1.6	m^{-1}
Equivalent tanh fringe extent	∼0.125	m
Machida decoupled tunes	0.137164 / 0.263641	cycles/cell

Table 2: Magnet Parameters

Parameter	**Bd**	**Bf**	**Units**
Length	0.5	0.5	m
Sector angle	16	-32	°
Transverse displacement	-0.17	0.17	m
Beam bend angle	28.28571429	-64.28571429	°
B_0 injection	-0.247309446	0.562066924	T
B 12 MeV extraction	-0.495801694	1.126822031	T
B magnet 'top' (12 MeV@$k = 1.2 m^{-1}$ tuning)	-0.625168527	1.420837562	T
Height of magnet 'top'	0.579613053		m

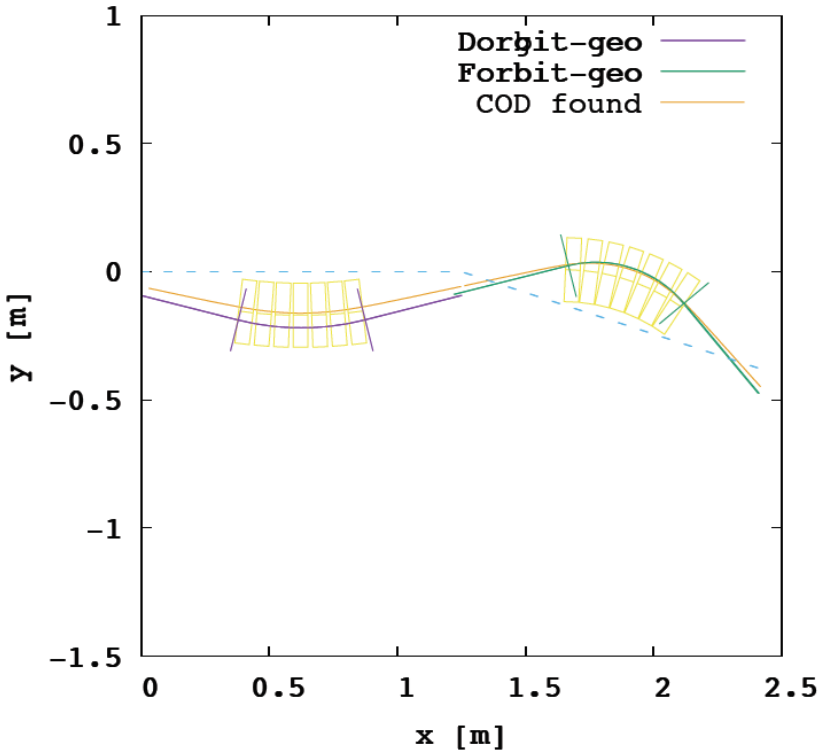

Figure 1: Periodic FODO cell from [1]. Physical magnet sectors are in yellow, actual beam path is in orange, two sides of 20-sided reference polygon are the blue dotted line. Green and purple arcs were a hard-edged starting point for optimisation, which had to be changed significantly.

2 Magnet Winding Function

The magnet windings are produced as contours of a 2D winding potential function defined on the vertical plane, transversely either side of the magnet midplane, following the technique in [2]. Consider first the magnet without curvature.

The $\tanh(z/f)$-shaped fringe field used by Machida falls off faster than most real magnet fringe fields. A good approximation to the rate of fall-off for $f = 0.125\,\text{m}$, at least in the middle of the fringe field, is obtained by returning the windings in a linear way across a $\pm 4\,\text{cm}$ longitudinal range either side of each nominal end of the magnet, meanwhile making the transverse positions of the two winding sheets $\pm 7\,\text{cm}$ from the midplane.

The winding potential can thus be written $\phi(y,z) = Y(y)Z(z)$ where $Z(z)$, the longitudinal function, is a linear ramp up from $z = -4$ to $4\,\text{cm}$, equal to 1 from $z = 4\,\text{cm}$ to $46\,\text{cm}$ and ramping back down to zero from $z = 46\,\text{cm}$ to $54\,\text{cm}$. The fringe field shape associated with this longitudinal winding shape can be approximated by Z' convolved with the function $\arctan(z/X)$ where $X = 7\,\text{cm}$ here is the transverse distance from the midplane to the winding sheets.

2.1 Ideal Vertical Potential

Since $Z(z)$ was made dimensionless, the vertical potential $Y(y)$ has units of Amps. The ideal vertical potential for an exponential VFFA assuming winding sheets close to the midplane is $Y(y) = \frac{1}{\mu_0} B_0 \frac{1}{k} e^{ky}$.

2.2 Vertical Potential: Joins and Ends

The ideal vertical potential has a number of problems: it goes to infinity at positive y and never fully reaches zero at negative y. In the magnet design I partly solved this problem by introducing a joining function (mathematically related to a 'bump function'):

$$\sigma(x;k,a) = \frac{1}{2}\tanh\left(k\tan\left(\left(x-\frac{1}{2}\right)\pi\right)+a\right)+\frac{1}{2} \quad \text{if } 0<x<1;\ 0 \text{ for } x\le 0;\ 1 \text{ for } x\ge 1.$$

This function is 0 at $x = 0$, 1 at $x = 1$ and joins between the two having an infinite number of continuous derivatives. The parameters k and a control the rate and asymmetry of the join.

With this function implemented as `sigy`, the full vertical potential is produced by the function `poty` in the code 2. This produces a symmetrical magnet with two possible acceleration areas (one inverted above the high-field end of the original magnet). The function `poty1` is the idealised potential in this case.

2.3 Curvature

Rather than compensating for curvature in any way, the windings produced from the contours were bent according the each magnet's radius of curvature. The centre point of the sector stays the same, while differences in transverse position become differences in radius. The magnet centre-line's length is also preserved and turned into arc length. The code 3 implements this transformation.

2.4 Vertical Potential: Polynomial Correction for Better Fields and Tunes

The above methods produce magnets that have stable optics through a lot of the required energy range, but the tunes are still not very constant (Figure 4). The largest excursions of tune happen at the top and bottom of the magnet where the effects of the non-zero winding separation allow the deviations from the ideal exponential law to be seen. Here, a simple correction with two polynomial terms was tried and resulted in improved tunes that were stable across the entire required energy range, although not completely constant yet (see Figure 5). It is thought that by using splines with more variables instead of the polynomials below, the tunes can be made even flatter.

The polynomial correction is obtained by replacing the ideal vertical potential by $Y(y) = \frac{1}{\mu_0}B_0\frac{1}{k}(e^{ky}+p(\frac{y}{H}-\frac{1}{2}))$, where p is a polynomial function and H is the full height of the magnet, so the argument $\frac{y}{H}-\frac{1}{2}$ ranges from $-\frac{1}{2}$ to $\frac{1}{2}$. The polynomial used in this example is

$$p(x) = 0.127x^5 + 0.464x^6,$$

with the relatively high powers of x restricting the changes to near the vertical extremities of the magnet. For the vertically-symmetrical double magnet generated by listing 2, this correction goes in the function `poty1` and gets used twice over in both good field regions.

This correction seems to reduce the RMS error between the midplane field generated by the windings and the ideal field, as well as levelling out the tunes.

3 Biot-Savart Calculation

The field maps for the tracking were typically generated with 5mm×5mm×2cm grid spacing where the longer grid spacing is in the beam direction (longitudinal). Assuming the windings have been approximated by a list of line segments carrying currents, the field values can be obtained by summing the Biot-Savart contributions of these segments as calculated in [3]. There is also some computer code for working out the field of the straight wire segment in listing 6.

It can be important to extend the field maps far enough that large discontinuities or loss of fringe field are avoided. I extended them by at least 0.625 m (5 tanh fringe field lengths) longitudinally from the magnet ends and 0.2 m transversely from the magnet to account for slanted entrance and exit orbits.

4 Winding Discretisation Algorithm

In order to go from the winding function $\phi(y, z)$ to a list of line segments needed for the calculation in the previous section, a discretisation algorithm is needed. The windings will be 'contour lines' of the winding function, each carrying a certain number of Amps. This is calculated assuming a 250 A/mm^2 current density and windings for the Bf magnet of 0.25 cm^2 (6250 Amps), with the weaker Bd magnet windings being 0.125 cm^2 (3125 Amps) to maintain vertical smoothness of the field on the midplane. The levels of these contours should be offset by half a winding spacing from zero so that there is not a winding at $\phi = 0$ Amps that may extend to infinity.

The spatial discretisation algorithm samples $\phi(y, z)$ on a square grid where 100 cells cover the vertical extent of the potential. Each square is split into two triangles. Within each triangle, ϕ may be approximated by a linear function that goes through the three potential values calculated for the corners of the triangle. The contours off this linear function are straight lines. One of the corner potential values will be in the 'middle' of the other two. The half of the triangle above and below this middle value may be treated separately, each half populated with zero or more straight contours between two particular sides of the triangle. The endpoints of these straight contours may be found by simple linear interpolation on each side.

The resulting winding stacks, with half the number of windings for clarity, are shown in Figures 7 and 8.

References

[1] *FFA Physics Internal - extra for Stephen* -, Shinji Machida, presentation file `machida20200130.pdf`

[2] *Magnetic Field of a Winding Sheet defined by a Contour Function*, Stephen Brooks, available from `http://stephenbrooks.org/ap/report/2019-7/contourwinding.pdf`

[3] *Magnet Field of a Finite Wire*, Stephen Brooks, available from `http://stephenbrooks.org/ap/report/2011-7/finitewire.pdf`

Figure 2: Code for vertical potential of more realistic magnet

```
REAL yfalloff=0.5,ymiddle=yfalloff;

REAL sigy(const REAL x,const REAL k=2.0,const REAL asym=0)
{ // Goes from (0,0) to (1,1) smoothly, f'(0)=f'(1)=0
    if (x<=0) return 0; if (x>=1) return 1;
    //return x*x*(3-x-x);
    return tanh(k*tan((x-0.5)*M_PI)+asym)*0.5+0.5;
}

REAL poty1(const REAL y)
{ // Ideal field only
    return Bmax[magnum]/mu0*exp(k*y)/k;
}

REAL poty2(const REAL y)
{ // Ideal field symmetrically above middle
    return poty1(ymiddle-y);
}

REAL mid_asym=-0.125,mid_k=1.6,end_asym=0,end_k=1.6;

REAL poty(const REAL y)
{ // Try two-sided (symmetrical) one
    if (y<=0)
    {
        return poty1(y)*sigy(1+(y-(-H))/yfalloff,end_k,end_asym);
    }
    else if (y<ymiddle)
    {
        return poty1(y)*sigy(1-y/ymiddle,mid_k,mid_asym)+poty2(y)*
            sigy(y/ymiddle,mid_k,mid_asym);
    }
    else
    {
        return poty2(y)*sigy(1+((ymiddle+H)-y)/yfalloff,end_k,end_asym);
    }
}
```

Figure 3: Code to bend coordinates in a magnet

```
V3 bendfunc(const V3 p)
{ // Bends about Y axis (Z is forward, X is radial, centre is (R,*,0))
    REAL R=L/angle[magnum];
    if (fabs(p.z)<=0.5*L)
    {
        REAL th=p.z/R,
            r=R-p.x;
        return V3_new(R-r*cos(th),p.y,r*sin(th));
    }
    else
    {
        REAL ca=cos(0.5*angle[magnum]),sa=sin(0.5*angle[magnum]),
            zo=R*sa,xo=R*(1.0-ca),zd=fabs(p.z)-0.5*L;
        if (p.z>0) return V3_new(xo+p.x*ca+zd*sa,p.y,zo+zd*ca-p.x*sa);
        else return V3_new(xo+p.x*ca+zd*sa,p.y,-zo-zd*ca+p.x*sa);
    }
}
```

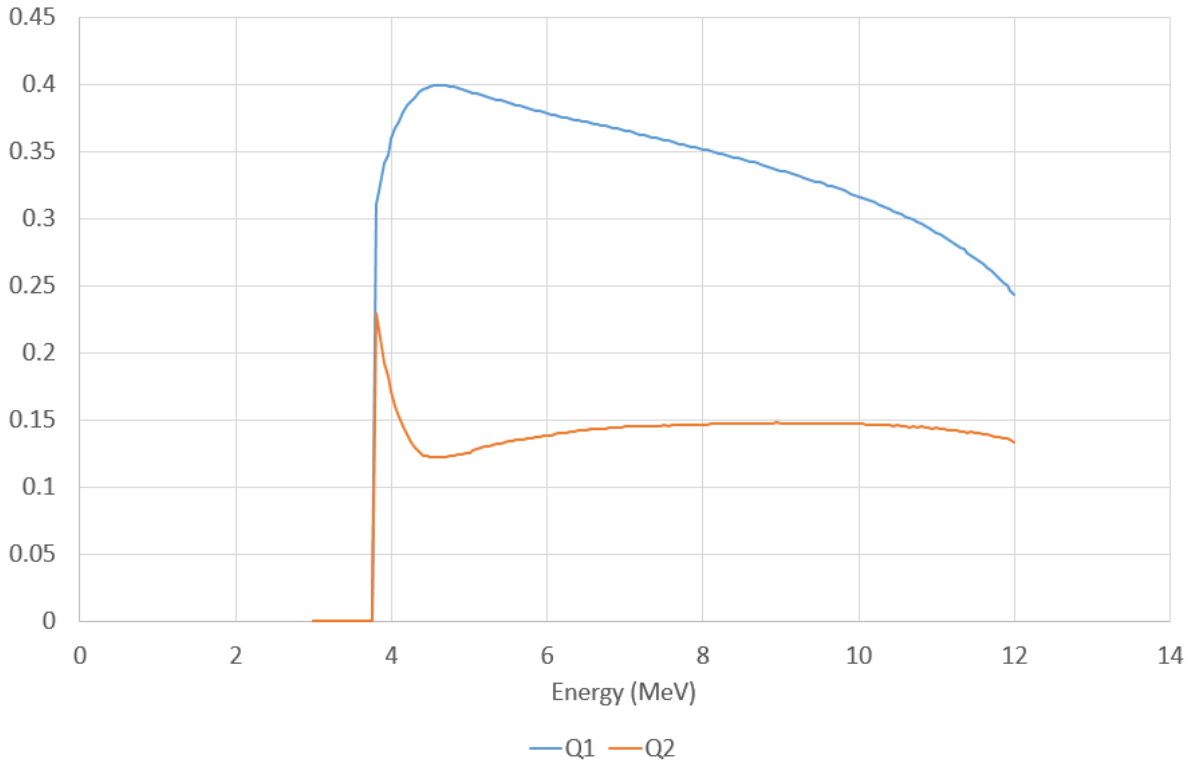

Figure 4: Tune variation before polynomial correction of vertical winding density.

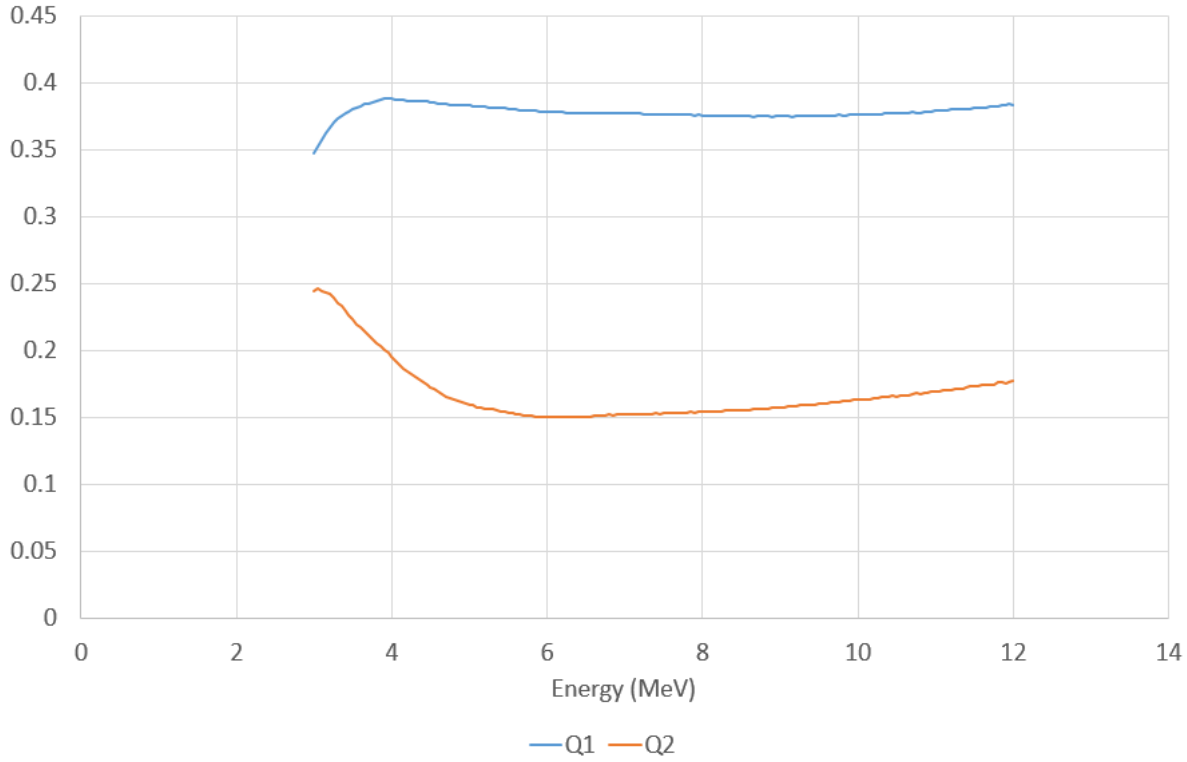

Figure 5: Tune variation after polynomial correction of vertical winding density.

Figure 6: Code to calculate the field contribution of a finite wire segment, following [3]

```
V3 Bcurrentwire(const V3 a,const V3 b,const V3 x,const REAL I)
{ // Straight wire from a to b carrying I Amps, field evaluated at x
    V3 u=b-a,v=x-a;
    REAL q=V3_normsq(u),r=-2.0*(u*v),s=V3_normsq(v); // u*v is dot product
    return ((mu0/M_TWOPI)*I/(4.0*q*s-r*r)*
        ((q+q+r)/sqrt(q+r+s)-r/sqrt(s)))*(u^v); // u^v is cross product
}
```

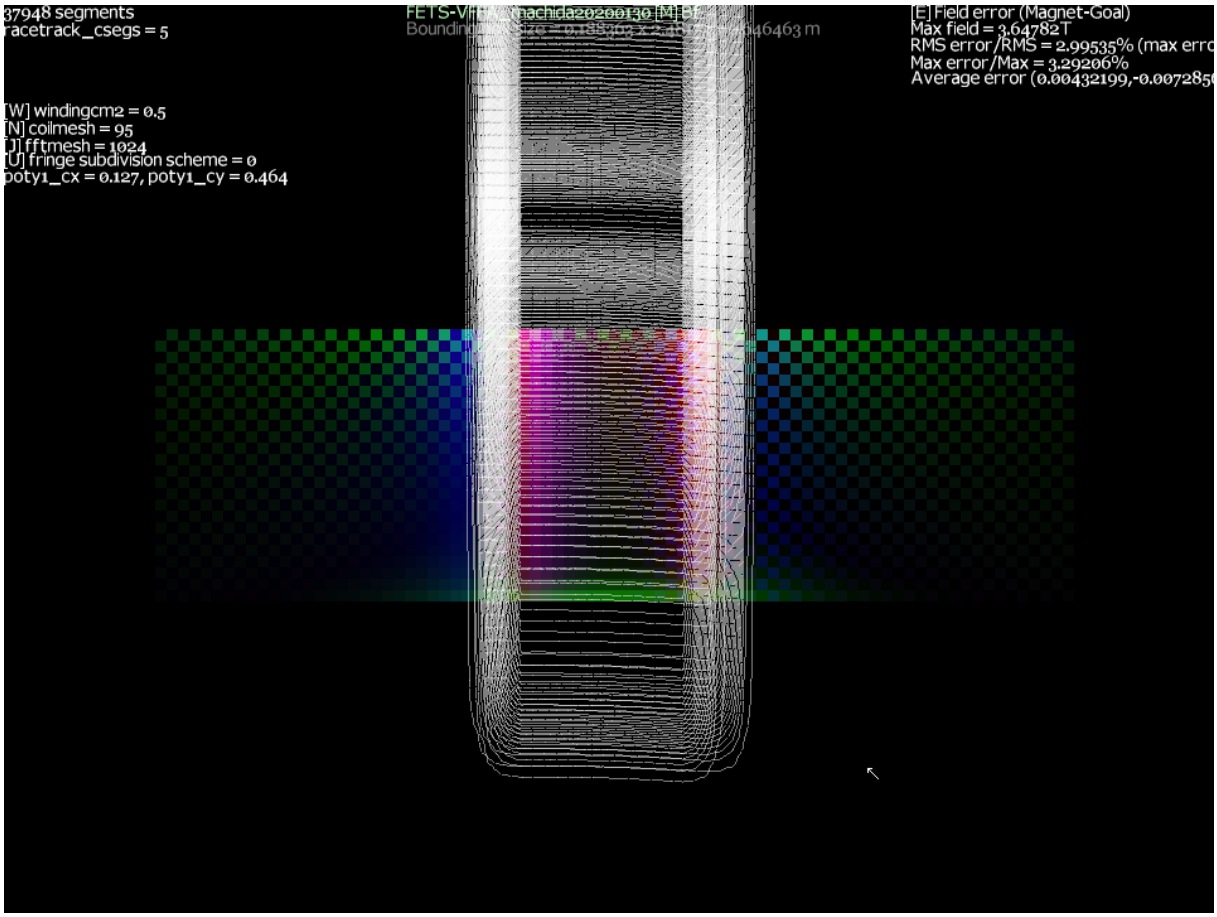

Figure 7: Discretised windings of the Bf magnet (NB: double the winding Amps spacing specified in the text)

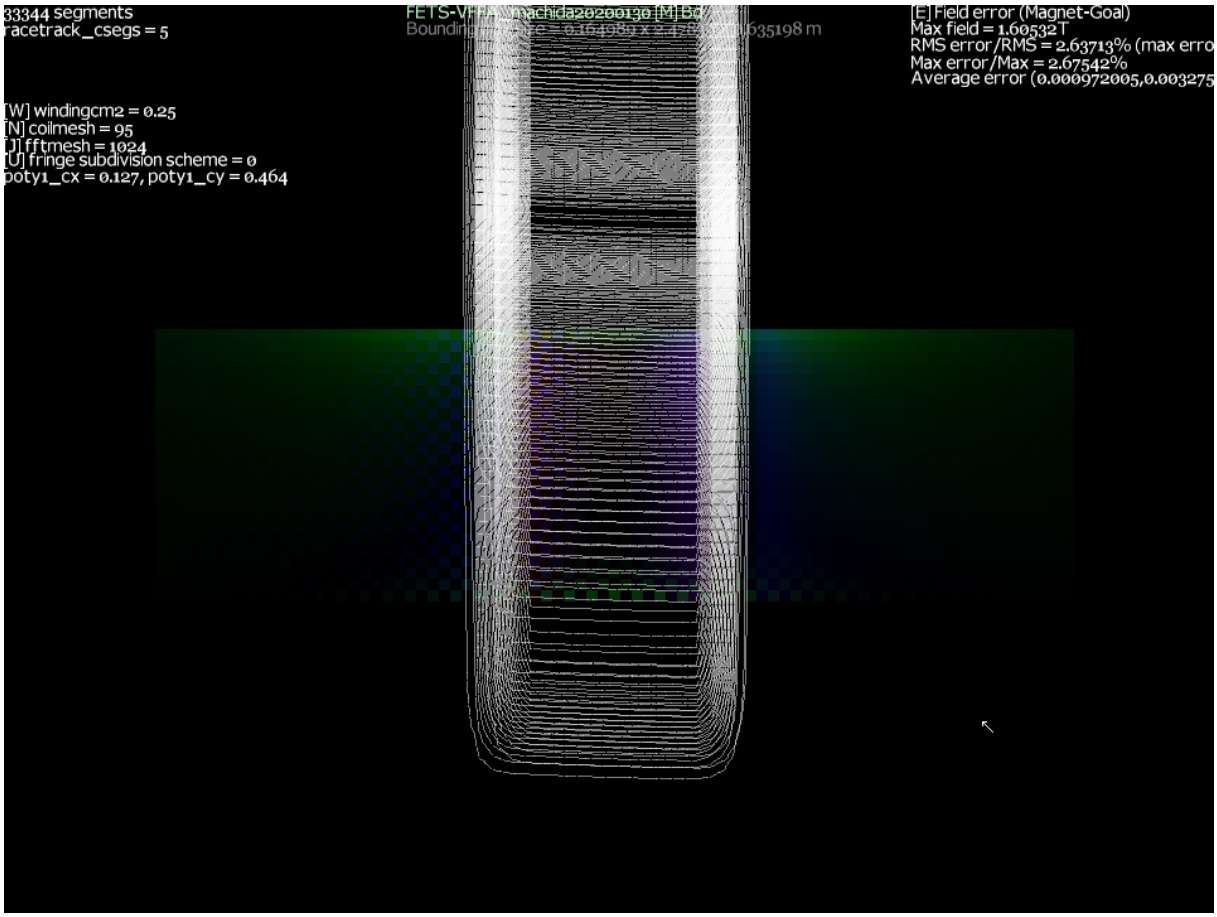

Figure 8: Discretised windings of the Bd magnet (NB: double the winding Amps spacing specified in the text)