New coil calc for HDS (Peter's emails of 2020-10-22ff), AOSEM wire, larger coil, double length, ø2 mm x 6 mm magnet

In[•]:=
```
HDS0vals = Recurse[
    LOSvals~Join~{
        kB->1.38 10^-23,
        T->295,
        g->9.8,
        mu0->4 N[Pi] 10^-7,
        rhoeAl->2.65 10^-8,          (* Resistivity of Al *)
        rhoepianowire->30 rhoeAl,    (* Resistivity of P.W., GUESSTIMATE *)
        sigmaeAu->2.67,              (* Resistivity of Au paint, measured *)
        dipoleM->Bresid/mu0,         (* Magnetization *)
        massSOS->0.25,               (* Final design value *)
        massM->rhomM l N[Pi] a^2,
        Bresid->1.25,                (* Residual field *)
        Pi->N[Pi],
        rhomM->7400,                 (* Mass density of magnets *)
        ztube->0.0155,               (* Separation for magnet to push its own weight, measured
        zpickup->0.018-l,            (* Separation for magnet to pull its own weight, measured
        masspullapart->0.060,        (* Force (in kgf) to separate magnets *)
        (* preferred value of small magnet dipole moment *)
        mzpref->0.007945632135874319,
        (* magnet moment per unit volume for magnet stuff, assumed the same for big and small
        mz->mzpref/(lold*N[Pi]*aold^2),
        lold->0.003175,              (* length of small magnets *)
        aold->0.0009525,             (* radius of small magnets *)
        l->0.006,                (* length, Peter email 2020-10-22, as for AODLvals *)
        a->0.002/2,          (* radius, Peter email 2020-10-22, as for AODLvals *)
        (* coil length *)
        coillen->0.005, (* Peter email 2020-10-22 *)
        (* the ends of the coil relative to the centre of the coil *)
        coilz1->-coillen/2,
        coilz2->+coillen/2,
        (* the inner and outer radii of the coil *)
        coilrad1->0.010/2, (* Peter email 2020-10-22  *)
        coilrad2->coilrad1+coillen, (* make cross section square, Peter email 2020-10-22, 2020
        (* the number of turns *)
        coilturns->((coilrad2 - coilrad1)*(coilz2 - coilz1))/wireareaAO, (* derive from cross
        wireareaAO->6.47708x10^-8, (* 1/coilsigma/.AOvals - same wire as AOSEM *)
        coilsigma -> coilturns/((coilrad2 - coilrad1)*(coilz2 - coilz1)),
        wirearea -> 1/coilsigma
    }
]
```

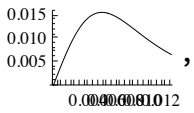
Out[•]= {w0 → 4.66841, rwire → 0.00155, awire → 7.54768×10^{-6}, dpend → 0.45, dyaw → 0.0333, massoptic → 10.7, loptic → 0.1, roptic → 0.125, Ioptic → 0.0507135, wpitch → 3.76991, wpend → 4.67469, wyaw → 3.14159, kpitch → 0.720753, kyaw → 0.500523, pitchlever → 0.0808, yawlever → 0.0808, kxpitch → 8.9202, kB → 1.38×10^{-23}, T → 295, g → 9.8, mu0 → 1.25664×10^{-6}, rhoeAl → 2.65×10^{-8}, rhoepianowire → 7.95×10^{-7}, sigmaeAu → 2.67, dipoleM → 994718., massSOS → 0.25, massM → 0.000139487, Bresid → 1.25, π → 3.14159, rhomM → 7400, ztube → 0.0155, zpickup → 0.012, masspullapart → 0.06, mzpref → 0.00794563, mz → 878021., lold → 0.003175, aold → 0.0009525, l → 0.006, a → 0.001, coillen → 0.005, coilz1 → -0.0025, coilz2 → 0.0025, coilrad1 → 0.005, coilrad2 → 0.01, coilturns → 385.976, wireareaAO → 6.47708×10^{-8}, coilsigma → 1.54391×10^{7}, wirearea → 6.47708×10^{-8}}

In[•]:=
```
vals[HDS0] := HDS0vals;
```

In[•]:=
```
optdata[HDS0] = If[
  useprecomputed,
  {-0.05125731583419942`, {z → 0.004554212103051588`}},
  Minimize[{-fz[HDS0, z], 0.004 < z < 0.008}, z]
 ]
```

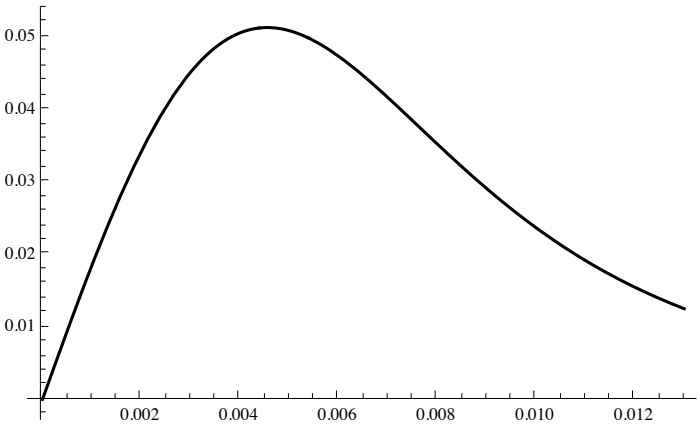
Out[•]= {-0.0512573, {z → 0.00455421}}

In[•]:=
```
plot[HDS0] = If[
  useprecomputed,
```




```
  Plot[fz[HDS0, p], {p, 0, 0.013}, PlotStyle → {Black}]
 ]
```

Out[•]=



In[•]:=
```
zmax[HDS0] = z /. optdata[HDS0] [[2]]
```

Out[•]= 0.00455421

In[•]:=
```
fmax[HDS0] = -optdata[HDS0] [[1]]
```

Out[•]= 0.0512573

In[•]:=
```
fzi[HDS0] = FunctionInterpolation[fz[HDS0, z], {z, 0, 0.013}]
```

Out[•]= InterpolatingFunction[Domain: {{0., 0.013}} Output: scalar]

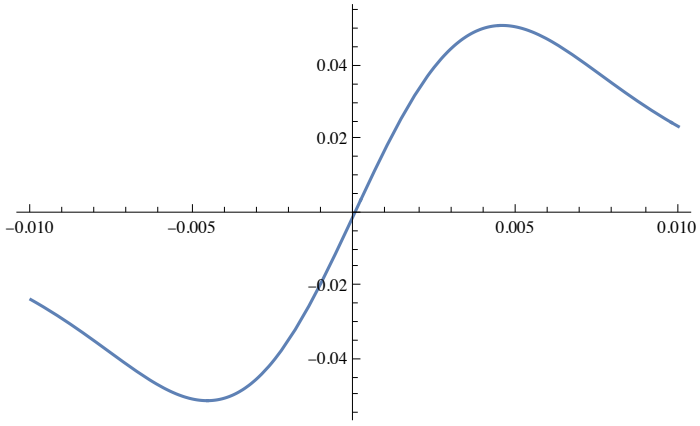
In[•]:=
```
coupling[HDS0] = -2 * Coefficient[Normal[Series[fzi[HDS0][z], {z, zmax[HDS0], 2}]], z^2]
```

Out[•]=
```
4533.87
```

Sweet spot relative to centre of coil

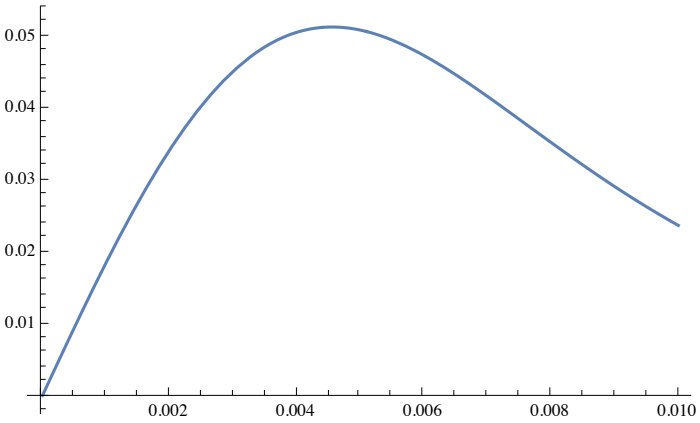
In[•]:=
```
Plot[fz[HDS0, p], {p, -0.010, 0.010}]
```

Out[•]=



In[•]:=
```
Plot[fz[HDS0, p], {p, 0.0, 0.01}]
```

Out[•]=



In[•]:=
```
FindMinimum[-fz[HDS0, p], {p, 0.005}]
```

Out[•]=
```
{-0.0512573, {p → 0.0045542}}
```