Zgoubi Workshop

Zgoubi User Experience

September 2014

Jimmy Garland and Sam Tygier

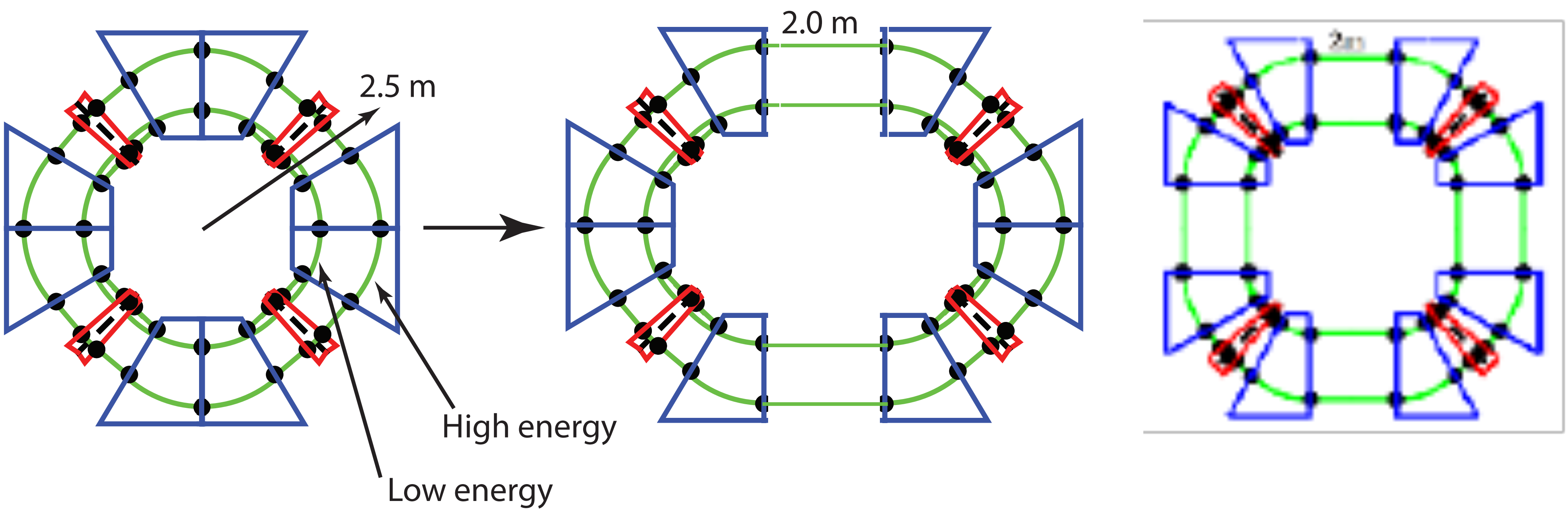
The University of Manchester

The Cockcroft Institute

Carol's 4 Cell Compact 200 – 1000 MeV Proton FFAG

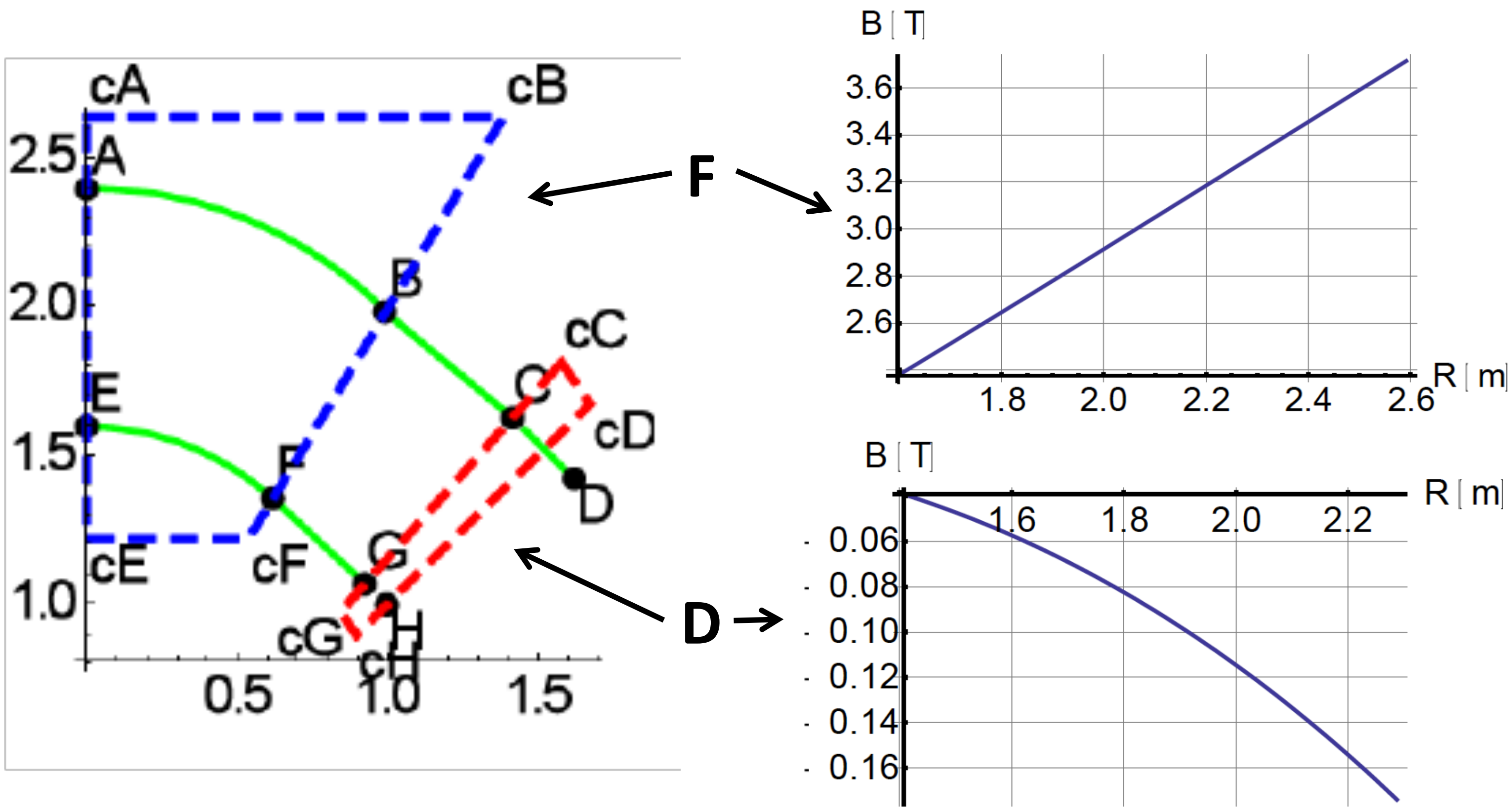
- We implemented Carols compact FFAG in PyZgoubi.
- Cell definition presented some challenges.

200 - 1000 MeV Proton FFAG

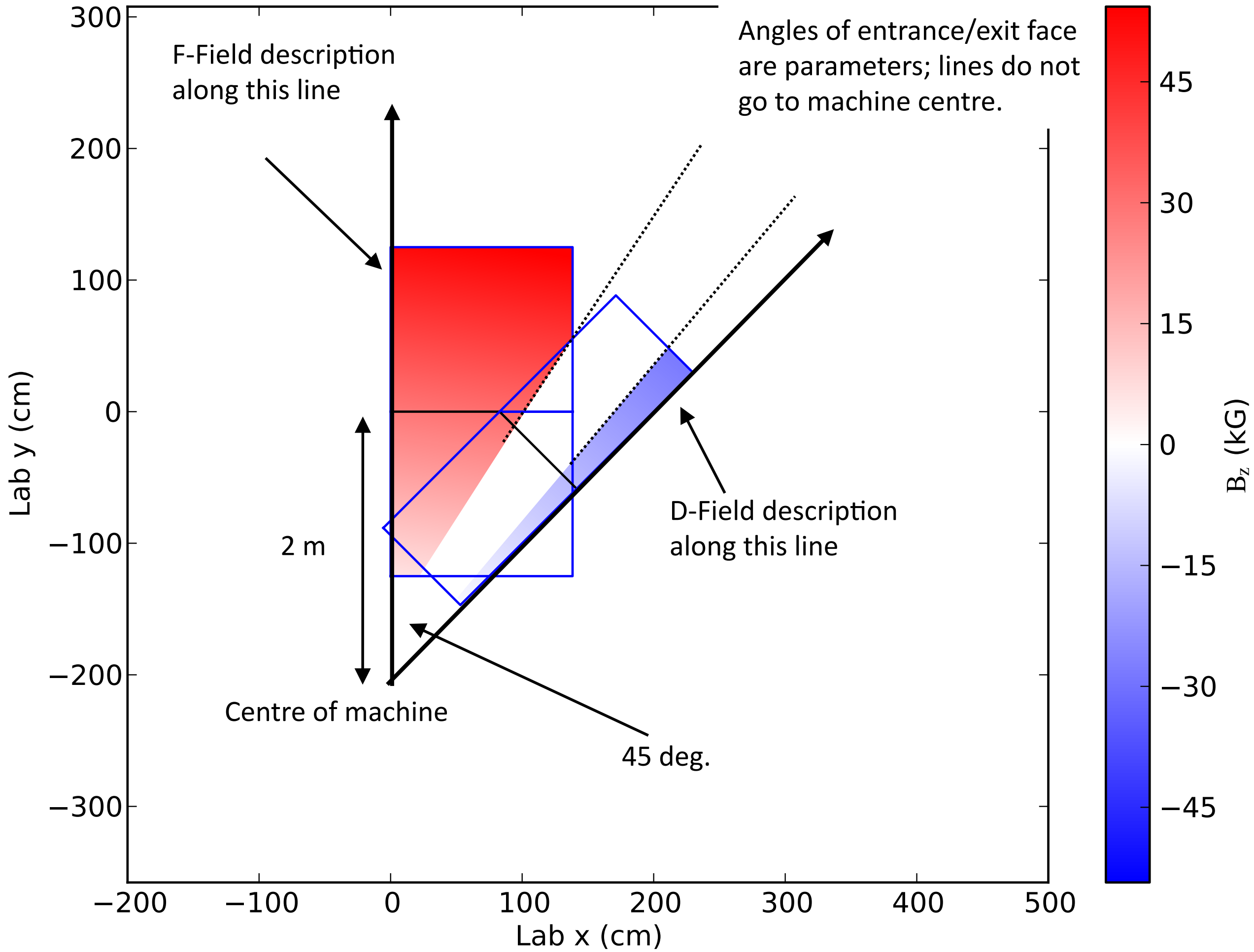


Carol’s 4 Cell Compact FFAG 200 – 1000 MeV

- Carol provided the specifications of the cell and magnetic field parameters.
- Magnetic field is non-linear, non-scaling, therefore DIPOLES module used in Zgoubi/PyZgoubi.
- As the magnets are not rectangular or sector type and wedge angles are utilised in Carol’s design, we must think a little more about how to implement the magnets.

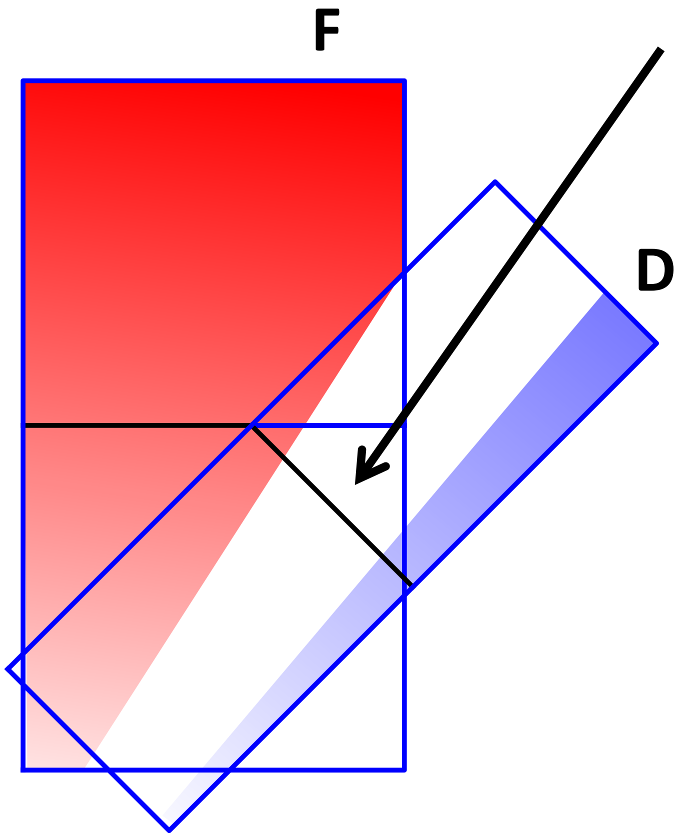


Magnets in Zgoubi



- Two DIPOLES units; one for the F and one for the D.
- The radius in DIPOLES is set to a very large value to make it go essentially to infinity as this is NOT a sector magnet.
- Entrance and exit field is defined *inside* the DIPOLES element (as in PAMELA Zgoubi model).
- Mid-plane of DIPOLES is 2m from centre of machine.
- Field description radial from centre of machine along the black solid lines.
- Extent of DIPOLES element incorporates all the intermediate drift with no field.

Magnets in Zgoubi



Because of the definition of the field along the centre of the magnet from the centre of the machine, cannot put more than one magnet in any one DIPOLES element.

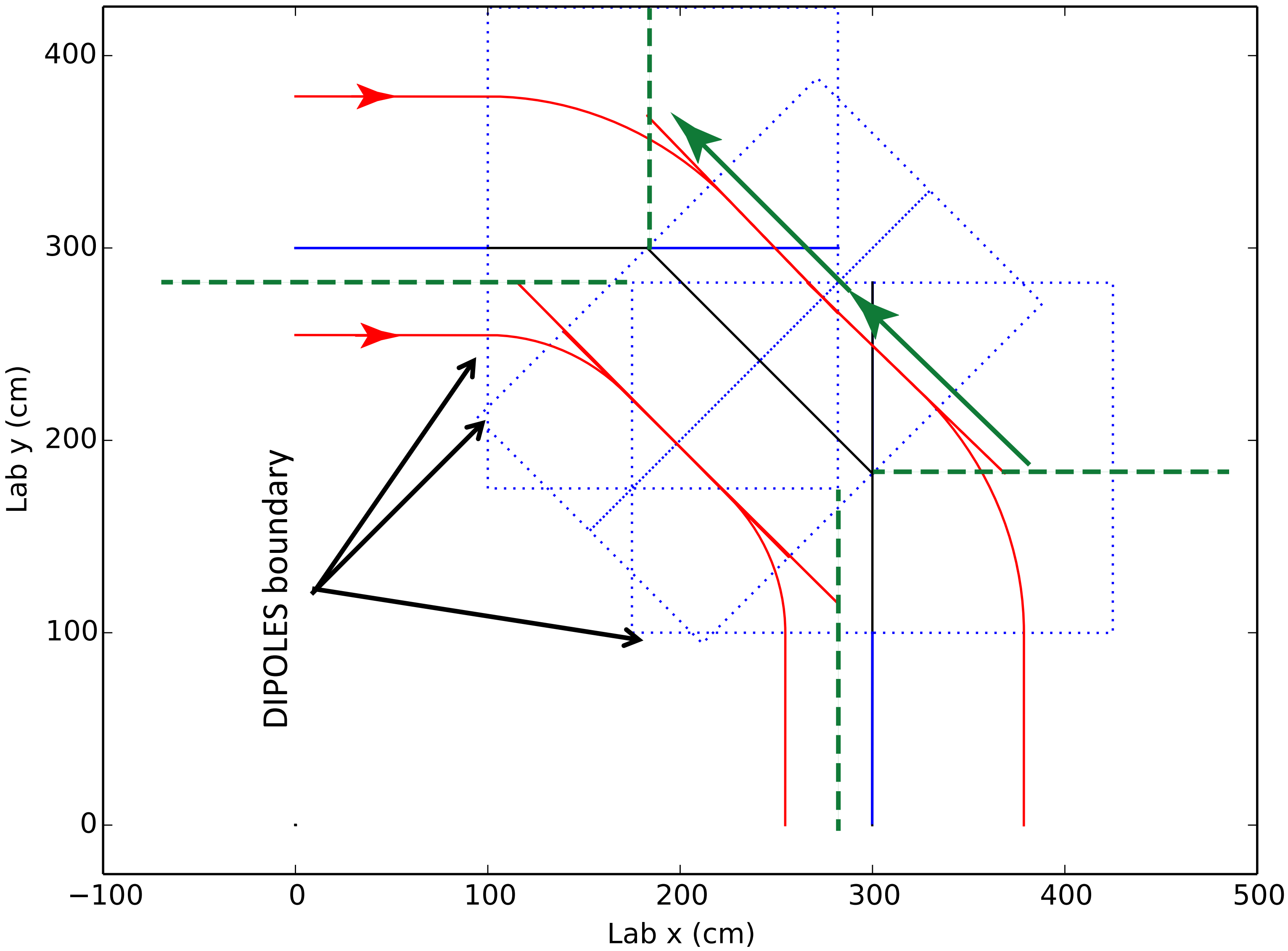
Cannot add a DRIFT here because of the size of the F magnet DIPOLES element.

Instead the D magnet DIPOLES element must be long enough to incorporate the intermediate drift space.

Zgoubi must track to the end of an element before starting another element.

But, we need to add fake drifts to make sure the beam starts in the right place in the next element.

Fake Drifts



½ F mag
Fake drift
½ D mag

½ D mag
Fake drift
½ F mag

Fake Drifts

```
m1 = DIPOLES("F", IL=2, N=1,AT=degrees(AT),RM=RM/cm,
             KIRD=2.01, RESOL=4,
             XPAS=XPAS/cm,
             RE=RE/cm, RS=RS/cm, KPOS=2)
m1.add(ACN=degrees(ACN), B_0=Fcoefs[0], IND=5,
       BCOEF1 = Fcoefs[1], BCOEF2 = Fcoefs[2], BCOEF3 = Fcoefs[3], BCOEF4 = Fcoefs[4], BCOEF5 = Fcoefs[5],
       OMEGA_E = degrees(OMEGA_E), R1_E=big_len/cm, R2_E=big_len/cm, U1_E=-big_len/cm, U2_E=big_len/cm,
       #G0_E = g0*cm_, KAPPA_E = kappa,
       G0_E = 1e-20*cm_, KAPPA_E = kappa,
       NCE = nc, CE_0 = c0, CE_1 = c1, CE_2 = c2, CE_3 = c3,
       #NCE = nc, CE_0 = 0, CE_1 = 0, CE_2 = 0, CE_3 = 0,
       OMEGA_S = degrees(OMEGA_S), R1_S=big_len/cm, R2_S=big_len/cm, U1_S=-big_len/cm, U2_S=big_len/cm,
       THETA_S = degrees(Af),
       G0_S=g0*cm_, KAPPA_S=kappa,
       NCS = nc, CS_0 = c0, CS_1 = c1, CS_2 = c2, CS_3 = c3,
       )

cell.add(m1)

#inter - drift length
#LongD_1 = -50
#LongD_2 = Drift_L/cm


cell.add(DRIFT('d0', XL=1e-9))
#cell.add(CHANGREF(ALE=-60))
cell.add(DRIFT('negD', XL=-Fake_Drift_L/cm))
#cell.add(DRIFT('d0', XL=1e-9))
cell.add(CHANGREF(ALE=-45))
cell.add(DRIFT('d0', XL=1e-9))
```

Fake drift

```
    Ld = NEW_D_MAG_EDGE#r_D_field_edge# 0.144 *m
    Ld2 = r_D_field_edge
    Ad =  (pi/4) - D_in_ang#3.84181467/(360/(2*numpy.pi)) #pi/10 #0.718345848#
    #print Ad*(360/(2*pi))
    #exit()

    RM = RE = RS = fake_rad
    angle_mag = Mag_ref_cent*asin(Ld2/2/fake_rad)
    angle_dipoles = Mag_ref_cent*asin(Ld/2/fake_rad)
    AT = angle_dipoles
    ACN = angle_dipoles - angle_mag / 2
    OMEGA_E = angle_mag/2
    OMEGA_S = -angle_mag/2
    #bcoefs = numpy.zeros([6])
    #bcoefs[0] = -10
    #bcoefs[1] = -0.1


    m1 = DIPOLES("D", IL=2, N=1,AT=degrees(AT),RM=RM/cm,
                 KIRD=2.01, RESOL=4,
                 XPAS=XPAS/cm,
                 RE=RE/cm, RS=RS/cm, KPOS=2)
    m1.add(ACN=degrees(ACN), B_0=Dcoefs[0], IND=5,
           BCOEF1 = Dcoefs[1], BCOEF2 = Dcoefs[2], BCOEF3 = Dcoefs[3], BCOEF4 = Dcoefs[4], BCOEF5 = Dcoefs[5],
           OMEGA_E = degrees(OMEGA_E), R1_E=big_len/cm, R2_E=big_len/cm, U1_E=-big_len/cm, U2_E=big_len/cm,
           THETA_E = degrees(-Ad),
           G0_E = g0*cm_, KAPPA_E = kappa,
           NCE = nc, CE_0 = c0, CE_1 = c1, CE_2 = c2, CE_3 = c3,
           OMEGA_S = degrees(OMEGA_S), R1_S=big_len/cm, R2_S=big_len/cm, U1_S=-big_len/cm, U2_S=big_len/cm,
           #G0_S=g0*cm_, KAPPA_S=kappa,
           G0_S=1e-20*cm_, KAPPA_S=kappa,
           NCS = nc, CS_0 = c0, CS_1 = c1, CS_2 = c2, CS_3 = c3,
           #NCS = nc, CS_0 = 0, CS_1 = 0, CS_2 = 0, CS_3 = 0,
           )

    cell.add(m1)
    cell.add(DRIFT('d0', XL=1e-9))

    return cell
```