# Magnetic Mirror Effect in Magnetron Plasma: Modeling of Plasma Parameters

## 1 Control on Particle Update strategies - NOT a control knob

Not really a control knob. Concerns precision of solution / update.

Update strategies used in **particle.ipynb**

Describe the Boris algorithm update strategy.

```python
class Particle:
    #... other things

    def Boris_update(self, afield, argsE, argsB):
        # Define q_prime
        q_prime = (self.charge / self.mass) * (dt / 2)

        # Get E and B fields from the afield argument by passing in the current position of the particle
        argsE = V, center
        E = afield.get_E_field(self.r, V, center)
        argsB = n, I, R, B_hat, mu_0
        B = afield.get_B_field(self.r, n, I, R, B_hat, mu_0)

        #Boris velocity update
        v_minus = self.v + q_prime * E
        v_plus = v_minus + q_prime * 2 * np.cross(v_minus, B)
        v_new = v_plus + q_prime * E

        self.v = v_new


        #could have also done:
        #self.v += (2 * q_prime) * (E + np.cross( (self.v + q_prime * E), B))

        #update position
        self.r += v_new * dt

    #... some other things
```

# 2 Control of Electric and Magnetic fields - Control Knobs here

Electric and Magnetic field configurations described in **field.ipynb**

```
class Field:
    #... something

    def uniform_E_field(self, E):
        return E

    def radial_E_field(self, r, V, center = [0,0,0]):
        #Get the distance vector of the particle from the electrode
        dr = [(r[0] - center[0]), (r[1] - center[1]), 0]
        #Get the electric field
        E = V * dr
        return E

    def uniform_B_field(self, B):
        return B

    def helmholtz_coil_B_field(self, n, I, R, B_hat, mu_0):
        return ( (4/5)**1.5 * ( (mu_0 * n * I) / (R) ) * B_hat)

    def two_helmholtz_B_field(self, n1, I1, R1, B1_hat, n2, I2, R2, B2_hat
        , mu_0):
        B1 = helmholtz(self, n1, I1, R1, mu_0, B1_hat)
        B2 = helmholtz(self, n2, I2, R2, mu_0, B2_hat)

        #Calculate the resultant of two magnetic fields
        B_hat = B1_hat + B2_hat
        return B_hat

    #...something else
```

Describe the Helmholtz coil magnetic field and electrode potential electric field configurations used.

Different Electric field configurations could be used. Simple example: changing the electrode voltages.

Different Magnetic field configurations could be used. Simple example: using many Helmholtz coils (number controllable), at different angles (angle controllable).

# 3 Controlling particle initialization - Control knobs here

Sampling particles with different initial velocities, and positions for example using different density functions f. For example: based on parameters like plasma Temperature.

Different particle sampling and initialization strategies used in **sampling.ipynb**

```
class Sampler:
    #... something

    def sample_uniform_position(self):
        pass

    def sample_uniform_veclocity(self):
        pass

    def sample_velocity_uniformKE(self):
        pass

    def sample_Maxwellian_velocity(self):
        pass

    def sample_parabolic_velocity(self):
        pass

    #... something else
```

Also track how the velocity distribution changes with time.

# References


[1] Qin, H., Zhang, S., Xiao, J., & Tang, W. M. (April, 2013). *Why is Boris algorithm so good?*. Princeton Plasma Physics Laboratory, PPPL-4872.