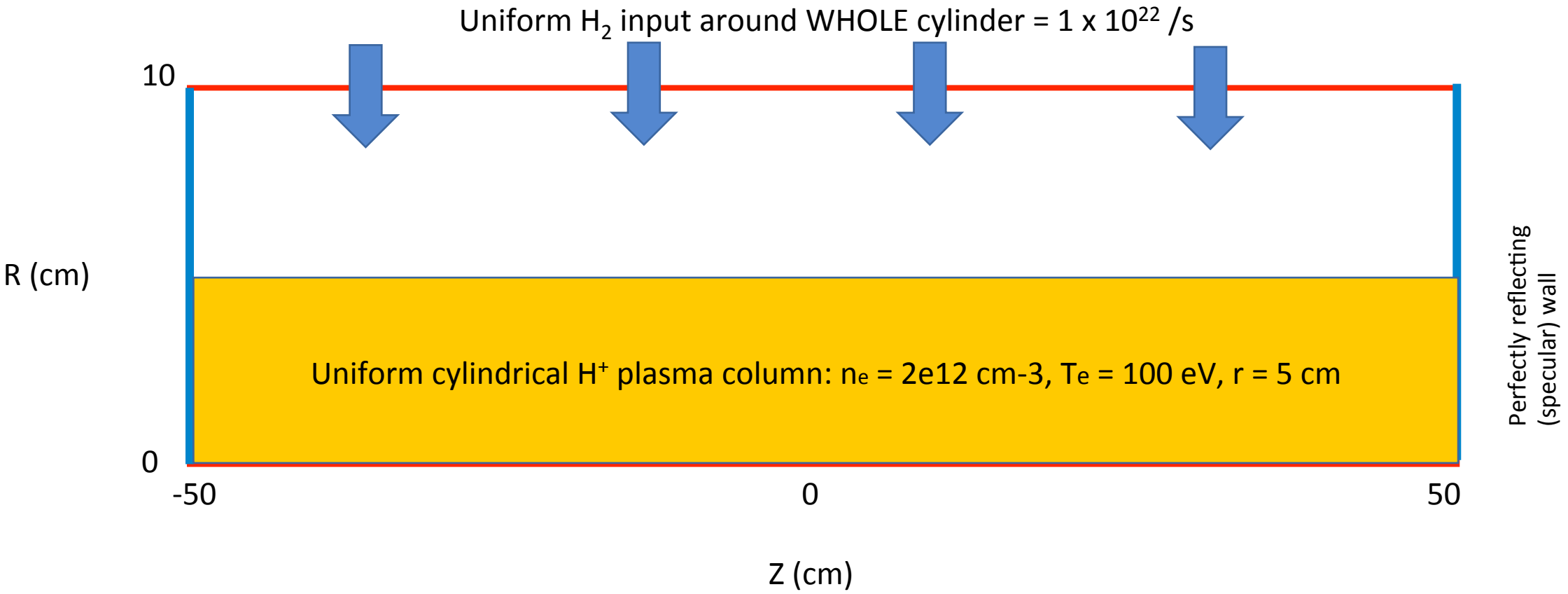
Degas Simulation Updates

C Biava

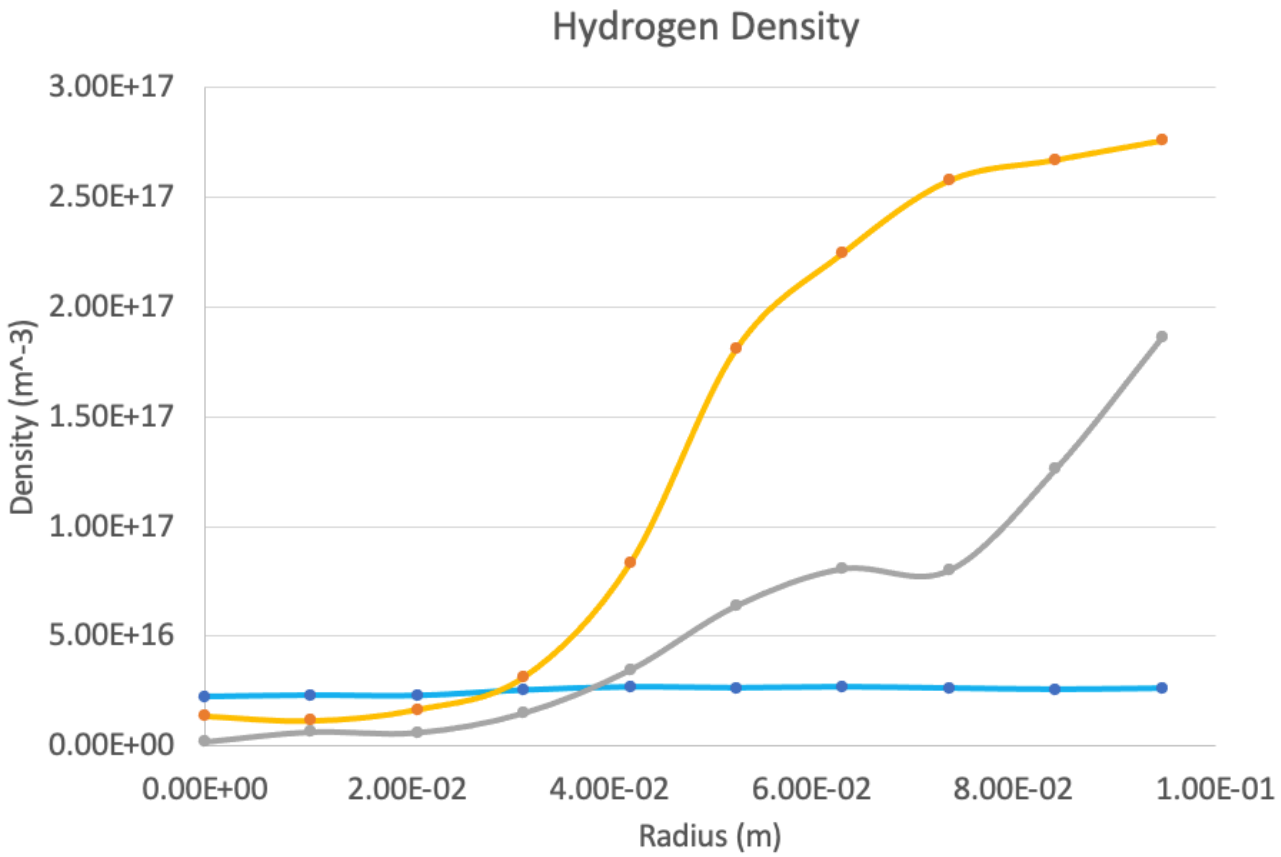
Hydrogen Setup



Hydrogen Setup

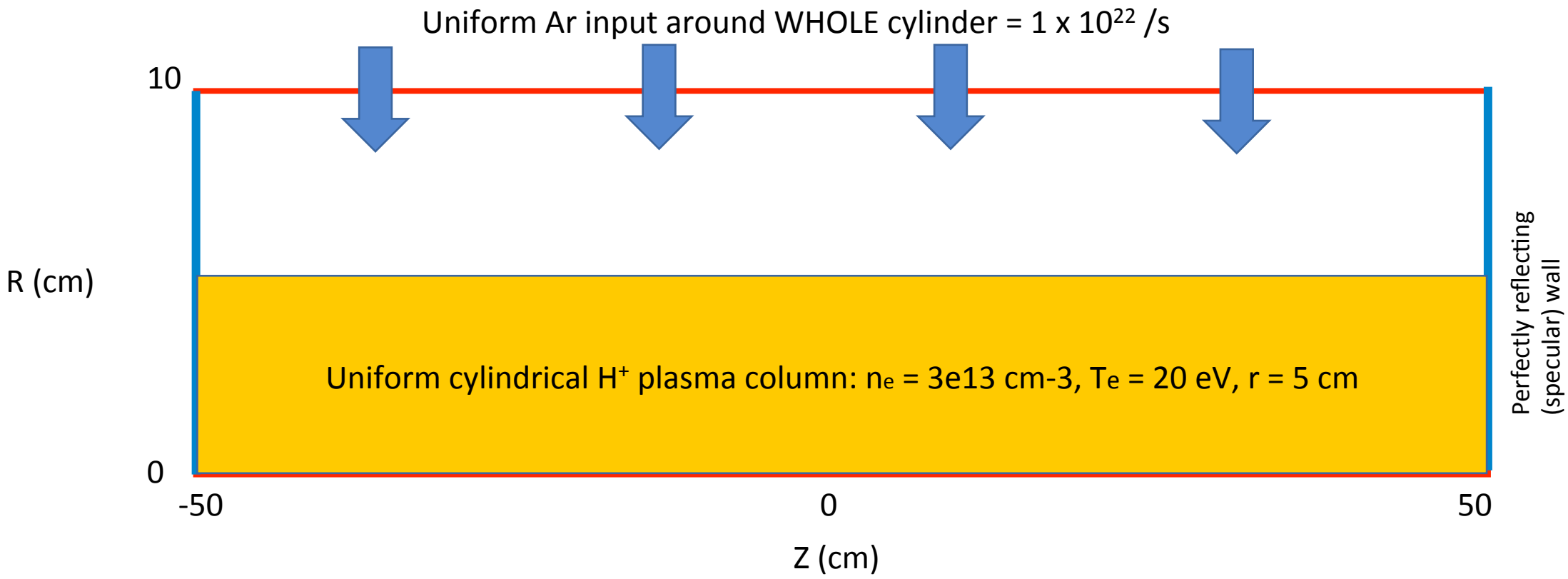
```
# source strength in molecules per m^2 per s
source = 1.0e22

# total radius in meters
R_tot=0.1

# radial resolution
NR=10

# constant ratio of Ti/Te
TiTe_ratio=1.0

# Number of flight samples to use for simulation
Nflights=1000

# These functions must be defined as functions taking a single argument (the radius in
# meters). You can use other (previously defined) parameters or tables in these functions if
 you wish. Units returned ought to be density in m^-3 and temperature in eV.
def ne_func(r):
    n0 = 2.0e18
    if r <= 0.05:
        return n0
    elif r > 0.05:
        return n0*(0.1-r)

def Te_func(r):
    T0 = 100.0
    if r <= 0.05:
        return T0
    elif r > 0.05:
        return T0*np.cos(r)
```

H Results



izone	radius (m)	H Density	rel. err.	H2 Density	rel. err.	H2+ Density	rel. err.
0	0.00E+00	2.26E+16	7.03E-02	1.38E+16	3.30E-01	2.12E+15	4.81E-01
1	1.05E-02	2.32E+16	4.04E-02	1.16E+16	1.29E-01	6.66E+15	2.12E-01
2	2.11E-02	2.30E+16	2.97E-02	1.67E+16	7.86E-02	6.34E+15	1.26E-01
3	3.16E-02	2.55E+16	2.59E-02	3.15E+16	4.92E-02	1.52E+16	7.94E-02
4	4.21E-02	2.68E+16	2.24E-02	8.32E+16	2.58E-02	3.49E+16	3.90E-02
5	5.26E-02	2.65E+16	2.54E-02	1.81E+17	2.75E-02	6.42E+16	1.24E-01
6	6.32E-02	2.69E+16	2.58E-02	2.25E+17	2.99E-02	8.10E+16	1.38E-01
7	7.37E-02	2.63E+16	2.71E-02	2.58E+17	2.94E-02	8.03E+16	1.25E-01
8	8.42E-02	2.58E+16	2.67E-02	2.67E+17	2.73E-02	1.26E+17	1.41E-01
9	9.47E-02	2.62E+16	2.69E-02	2.76E+17	2.79E-02	1.86E+17	2.21E-01

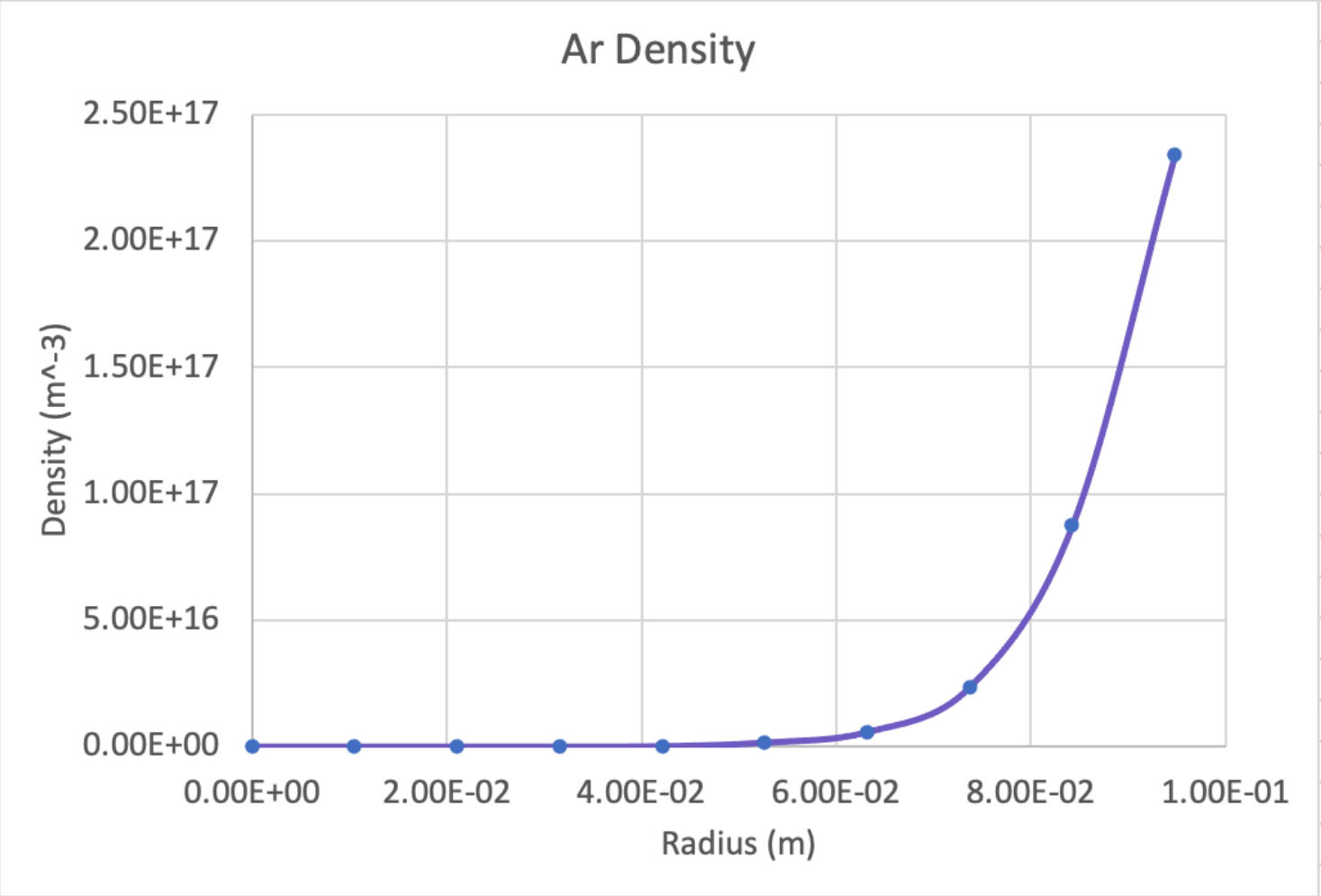
Argon Setup


Uniform Ar input around WHOLE cylinder = 1 x 10^22 /s
10
0
R (cm)
Uniform cylindrical H+ plasma column: ne = 3e13 cm-3, Te = 20 eV, r = 5 cm
Perfectly reflecting (specular) wall
-50
0
50
Z (cm)

Argon Setup

```
# source strength in particles per m^2 per s
source = 1.0e22

# total radius in meters
R_tot=0.1

# radial resolution
NR=10

# constant ratio of Ti/Te
TiTe_ratio=1.0

# Number of flight samples to use for simulation
Nflights=10000

# These functions must be defined as functions taking a single argument (the radius in
# meters). You can use other (previously defined) parameters or tables in these functions if
 you wish. Units returned ought to be density in m^-3 and temperature in eV.
def ne_func(r):
    n0 = 3.0e19
    if r <= 0.05:
        return n0
    elif r > 0.05:
        return n0*(0.1-r)

def Te_func(r):
    T0=20.0
    if r <= 0.05:
        return T0
    elif r > 0.05:
        return T0*np.cos(r)
```

Ar Results



izone	radius (m)	Ar Density	rel. err.
0	0.00E+00	4.41E+11	4.46E-01
1	1.05E-02	5.71E+11	2.18E-01
2	2.11E-02	1.91E+12	1.29E-01
3	3.16E-02	9.62E+12	7.32E-02
4	4.21E-02	1.43E+14	3.50E-02
5	5.26E-02	1.59E+15	1.98E-02
6	6.32E-02	5.76E+15	1.24E-02
7	7.37E-02	2.36E+16	7.43E-03
8	8.42E-02	8.74E+16	3.74E-03
9	9.47E-02	2.34E+17	5.73E-03