

# TAIGA 0.8.0

Felhasználói kézikönyv

plazma

user: taiga

pwd: Remszarvas

## Tartalomjegyzék

<b>1. TAIGA futtatása</b>	<b>3</b>
1.1. Fordítás . . . . .	3
1.2. Program indítása . . . . .	3
<b>2. Futtatás különböző input paraméterekkel</b>	<b>3</b>
2.1. Banánpálya kapcsoló . . . . .	3
2.2. Régi szemi-Runge–Kutta . . . . .	3
2.3. ABP számítás azzal a feltétellel, hogy minden ion adott $R$ -ből jön . .	4
2.4. ABP számítás az $n_e$ -ből és a $T_e$ -ből számolt ionpopulációval . . . . .	4
2.5. Hány részecske fusson? . . . . .	4
2.6. Detektorpozíció megadása $R$ -ben . . . . .	4
2.7. Nyalábparaméterek . . . . .	5
2.8. Futási paraméterek megadása . . . . .	5
<b>3. Adatbázisadatok beolvasása</b>	<b>6</b>
3.1. Mágneses tér legenerálása . . . . .	7
3.2. Mágneses tér spline-együtthatóinak legenerálása – ELM-ek nélkül	7
3.3. Mágneses tér spline-együtthatóinak legenerálása – ELM-et modellező áramfonalak esetén . . . . .	7
3.4. Ionpopuláció legenerálása . . . . .	7
<b>4. Kimeneti fájlok</b>	<b>8</b>
<b>5. Ionpályák kirajzolása</b>	<b>9</b>
5.1. Radiális–vertikális síkban a nyaláb koordinátái . . . . .	9
5.2. Háromdimenziós nyalábalak . . . . .	9
5.3. Detektorsík . . . . .	9

# 1. TAIGA futtatása

## 1.1. Fordítás

A programot először le kell fordítani végrehajtás előtt.

A fordítás a **projects/taiga\_#verziószám** könyvtárban található **compile** bash fájl indításával lehet elvégezni. Pl. így:

```
taiga@plazma:~/projects/taiga_0.8.0$ ./compile
```

## 1.2. Program indítása

A program automatikusan elindul fordítás után.

Ha ezt nem akarjuk, akkor a *compile*-ből ki kell szedni a **-run** kapcsolót.

Ha máskor is futtatni akarjuk, akkor a **./taiga.exe** fájlal tudjuk indítani.

```
taiga@plazma:~/projects/taiga_0.8.0$ ./taiga.exe
```

# 2. Futtatás különböző input paraméterekkel

Hogyha különböző paraméterekkel akarjuk futtatni a kódot, csak a **main.cu**-t kell módosítani.

Minden fontos paraméter a compiler kapcsolójába van építve, ezért csak az első néhány sorral kell foglalkozni!

```
taiga@plazma:~/projects/taiga_0.8.0$ vi main.cu
```

## 2.1. Banánpálya kapcsoló

Állítsuk a BANANA kapcsoló értékét 1-re. Máskülönben 0.

A részecskék vertikálisan felfelé indulnak. A nyaláb szélessége radiálisan **diameter** milliméter, közepe  $R = \mathbf{R\_midions}$  méter.

```
#define BANANA 1
#define R_midions 0.675
#define diameter 50
```

## 2.2. Régi szemi-Runge–Kutta

**Ezt ne használd!!!**

Ez hamarosan ki lesz véve. Ha 0-ra állítod, akkor az új kód fut, ha 1-re, akkor a régi.

```
#define RKOLD 0
```

## 2.3. ABP számítás azzal a feltétellel, hogy minden ion adott $R$ -ből jön

Állítsuk a RADIONS értékét 0-ra. A nyaláb szélessége vertikálisan **diameter** milliméter, közepe  $R = R\_midions$  méter.

```
#define BANANA 0
#define RADIONS 1
#define R_midions 0.69
#define diameter 5
```

## 2.4. ABP számítás az $n_e$ -ből és a $T_e$ -ből számolt ionpopulációval

Állítsuk a RADIONS értékét 1-re. A nyaláb szélessége vertikálisan **diameter** milliméter.

```
#define BANANA 0
#define RADIONS 1
#define diameter 5
```

Az ionpopulációt le kell gyártani, ha eltérő lövés esetén dolgozunk!

## 2.5. Hány részecske fusson?

A részecskék számát nem lehet direktben megadni, az a blokkok számának és a blokkonkénti szálak számának szorzata lesz.

A blokk az az egység, ami egy maghoz rendelhető. Azaz érdemes a blokkok számát magasan tartani!

**Jó tudni:** Az Nvidia Geforce GTS450-es kártyán 192 mag van.

### 2.5.1. Blokkok száma

```
#define n_blocks 192
```

### 2.5.2. Szálak száma

```
#define block_size 1
```

## 2.6. Detektorpozíció megadása $R$ -ben

A detektorpozíció megadása a **main.cu** 128. sorában kezdődő részben lehetséges (detector position).

Itt megadhatjuk az a  $R$  radiális koordinátát, melyen lévő vertikális–toroidális síkra kiinterpolálja a kilépő részecskéket.

```
l_ri = 0.7089;
```

A banánpályák esetén ennek nincs jelentősége. Azonban azért, hogy a kód ne haljon le, meg kell adni egy olyan értéket, ami kellően nagy (pl.  $R = 1$ )

## 2.7. Nyalábparaméterek

Külön meg kell adni a nyalábparamétereket ABP-számolásra és banánpályaszámolásra.

### 2.7.1. Nyalábenergia

A nyalábenergiát keV-ben kell megadni.

```
#define energy 80
```

### 2.7.2. Nyalábrészecskék tömege

A nyaláb részecskéinek tömegét AMU-ban kell megadni, azaz a  $^{12}\text{C}$  tömegének 1/12-ében.

```
#define mass 7.016004558
```

### 2.7.3. Nyalábátmérő

A nyalábátmérőt mm-ben kell megadni.

```
#define diameter 5
```

## 2.8. Futási paraméterek megadása

Külön meg kell adni a futási paramétereket ABP-számolásra és banánpályaszámolásra is.

### 2.8.1. Időlépés

Időlépés másodpercben.

```
#define dt 1e-9
```

### 2.8.2. Léptetések száma

Léptetések száma egy GPU futás alatt.

```
#define Nstep 1000
```

### 2.8.3. Futtatások száma

Ha ABP-módban vagyunk ez mindenképpen legyen 1.

Banánpálya esetén érdemes megadni, hogy hányszor futtassuk egymás után a kódot. A kód kimenete a következő GPU-ciklus bemenete lesz. A kimenet kiíródik a results mappába.

```
#define Nloop 1
```

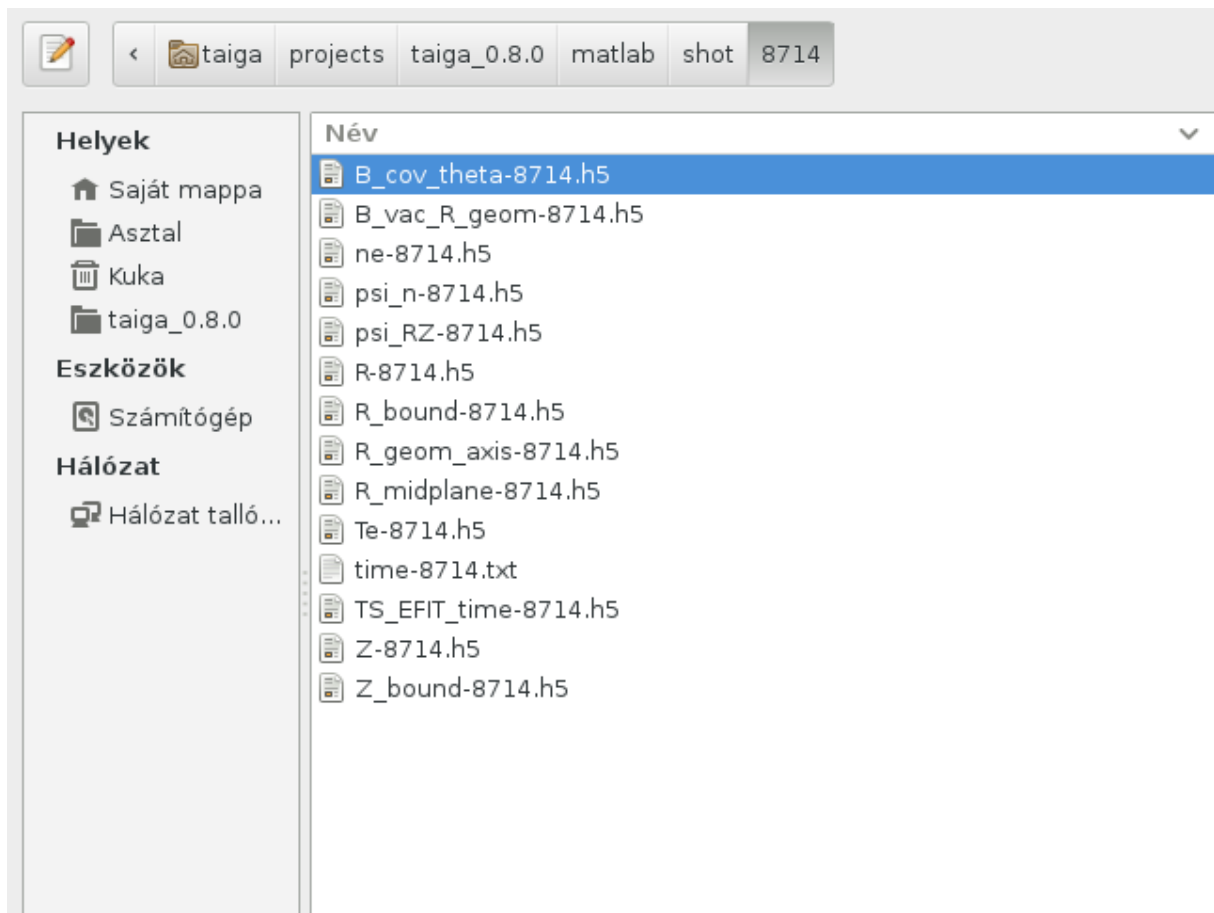
### 3. Adatbázisadatok beolvasása

A **projects/taiga\_0.8.0/matlab** mappából kell hívni a fájlokat. A bemeneti fájlokat a **projects/taiga\_0.8.0/matlab/shot/#shot\_number** mappába kell másolni.

A fájlok az EFIT és a THOMSON adatsorokból jönnek, amit az adatbázisból kézzel kell kimenteni.

**Figyelem! Az EFIT time mezője nem exportálható hdf5-be, ezért azt ASCII-ként kell lementeni, **.txt** kiterjesztéssel!**

A többi adat hdf5-ben van kimentve **.h5** kiterjesztéssel.



1. ábra. Fájlok a lövés adatait tartalmazó mappában

### 3.1. Mágneses tér legenerálása

A mágneses teret a **reader.m** segítségével lehet lefuttatni.

```
taiga@plazma:~/projects/taiga_0.8.0/matlab$ matlab -nojvm -r reader
```

Legenerálja a mágneses tér rácsát a fluxus segítségével. A rácsadatokat a **projects/field/default** mappába rakja.

### 3.2. Mágneses tér spline-együtthatóinak legenerálása – ELM-ek nélkül

A spline együtthatókat a **readerSpl.m** segítségével lehet kinyerni.

```
taiga@plazma:~/projects/taiga_0.8.0/matlab$ matlab -nojvm -r readerSpl
```

A spline együtthatókat a **projects/field/cuda/field** mappába rakja.

### 3.3. Mágneses tér spline-együtthatóinak legenerálása – ELM-et modellező áramfonalak esetén

A spline együtthatókat a **readerSpl2.m** segítségével lehet kinyerni.

```
matlab -nojvm -r readerSpl2(1000)
```

A spline együtthatókat a **projects/field/cuda/ipol** mappába rakja.

### 3.4. Ionpopuláció legenerálása

Az ionpopulációt a **readerDens.m** segítségével lehet lefuttatni.

```
matlab -nojvm -r readerDens(100000)
```

Legenerálja a nyalábmenti ionpopulációt a hőmérséklet és a sűrűség segítségével. A zárójelben lévő szám a részecskék számát adja meg.

Alapértelmezett érték: 100 000 részecske.

Az ionpopulációt a **projects/taiga\_0.8.0/dataio/data/rad.dat** fájlba rakja.

Minden egyes valós populáció előtt le kell generálni a rad.dat fájlt, azért, hogy garantáljuk, hogy ugyanannyi részecskét generáljuk, mint a CUDA számol!

## 4. Kimeneti fájlok

A program a futása során hat darab ASCII file-t generál. Ezeket a **results/#timestamp** mappába rakja a program.

Az időbélyeg formátuma:

10Dec2014\_133635

Ennek jelentése: a futtatás 2014. december 10-én 13:36:35-kor kezdődött.

A három fájl, ami a detektálás helyét megmondja:

### **rad.dat**

A detektáláskor a részecskék radiális ( $R$ ) koordinátái

### **z.dat**

A detektáláskor a részecskék – a középsík feletti – vertikális ( $z$ ) koordinátái

### **tor.dat**

A detektáláskor a részecskék toroidális koordinátái. A toroidális irány – a tokamak felülnézetéből – negatív körbejárású (mint az óramutató).

Ezek egy sorból álló, tabulátorral elválasztott, double-ként tárolt számok. Az oszlopok száma a részecskeszám.

A másik három fájl a pályaadatokat tartalmazza.

### **t\_rad.dat**

Az ionpályák radiális ( $R$ ) koordinátái

### **t\_z.dat**

Az ionpályák vertikális ( $z$ ) koordinátái

### **t\_tor.dat**

Az ionpályák toroidális koordinátája. A toroidális irány – a tokamak felülnézetéből – negatív körbejárású (mint az óramutató).

Hasonlóan a korábbiakhoz, oszlopokként, tabulátorral elválasztva találjuk az egyes részecskéket. Sortöréssel pedig az egyes kimentési időközkhöz tartozó adatokat (lásd a 2.8.2 alfejezetben).

Hogyha a kimentési időköz nagyobb, vagy egyenlő, mint a lépésszám, akkor csak a végén írjuk ki a kártyáról a koordinátákat. Azaz a **t\_...** fájlokban csak a kezdeti és a detektált koordinátákat tároljuk el!

## 5. Ionpályák kirajzolása

Az ionpályákat a **projects/plotter** mappából indítva lehet futtatni.

Minden egyes futtatás után a kimenet pályáit át kell másolni a **traj** almappába!

Hol találom meg a futtatásból kijövő ionpályákat? Ugorj a 4. fejezethez!

### 5.1. Radiális–vertikális síkban a nyaláb koordinátái

```
taiga@plazma:~/projects/plotter$ matlab -nojvm -r taigaPlot
```

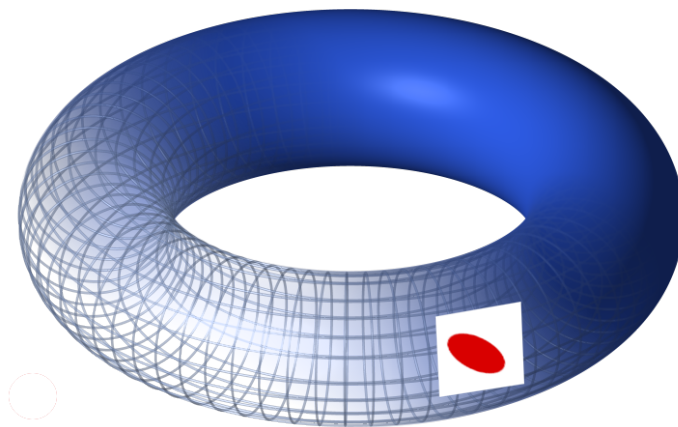
A piros pálya jelenti a toroidális geometriára ( $R$  és  $z$ ) visszatranszformált, míg a kék a  $\phi = 0$ -hoz illesztett descartes-i, amiben a program számol.

### 5.2. Háromdimenziós nyalábalak

```
taiga@plazma:~/projects/plotter$ matlab -nojvm -r plot3dtaiga
```

### 5.3. Detektorsík

```
taiga@plazma:~/projects/plotter$ matlab -nojvm -r detPlot
```



2. ábra. Detektorsík a **detPlot.m**-ből