Science & Technology Facilities Council

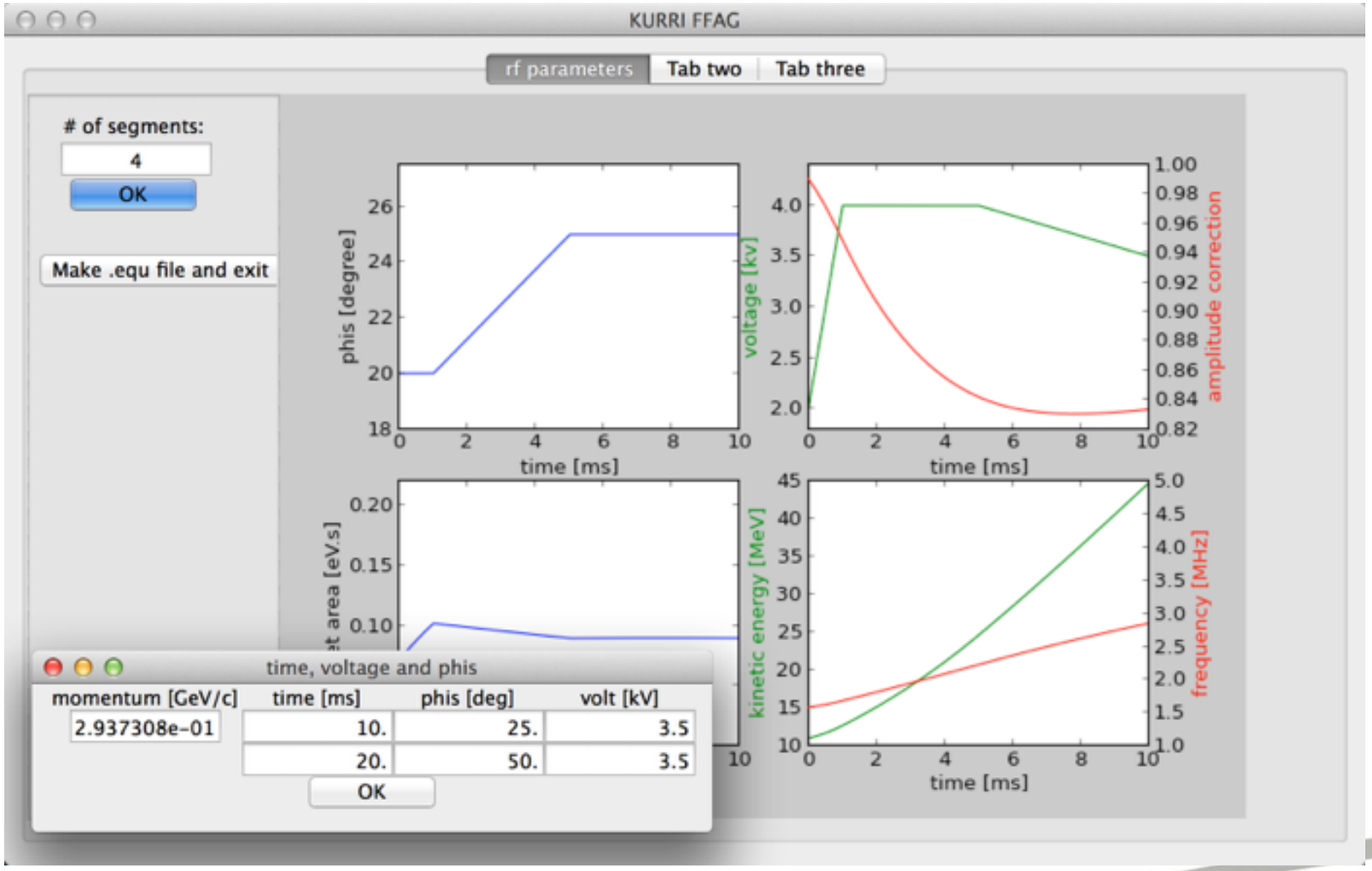
rf setting script

Shinji Machida

16 February 2015

Purpose

To find out the optimum rf parameters (voltage and frequency vs time), wrote a Python script with GUI.



Science & Technology Facilities Council

System requirements

I have developed on OS X, but it should be platform independent.

Python2.7 (version came with OS X).

wxPython3.0 (http://www.wxpython.org).

http://www.wxpython.org/download.php#osx

wxPython3.0-osx-cocoa-py2.7

Procedure

Divide one rf cycle into several regions.

e.g. capture, acceleration, storage, etc.

Specify start and end time of each region.

Specify voltage at start and end point. In between, it changes linearly.

Specify \phis at start and end point. In between, it changes linearly.

n.b. It is only used to specify energy gain per turn, namely \Delta E = eV\sin(\phis).

Beam sees different \phis with energy loss at foil.

Files

kurri_rf.py: main script.

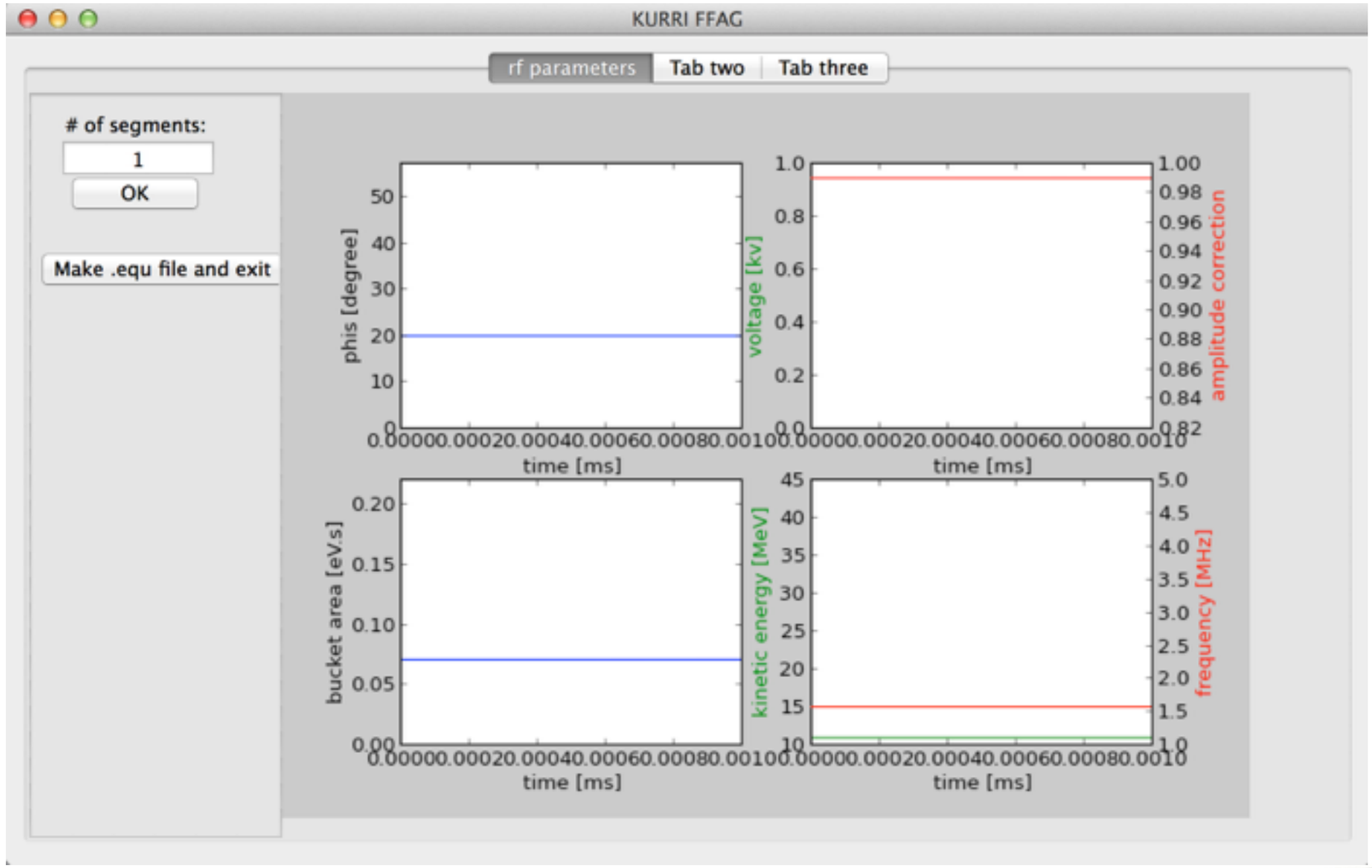
out_scode_r2_all: table from scode listing k and other parameters vs momentum.

tmp_all.equ: output file for AWG.

Step 1

$ python kurri_rf.py

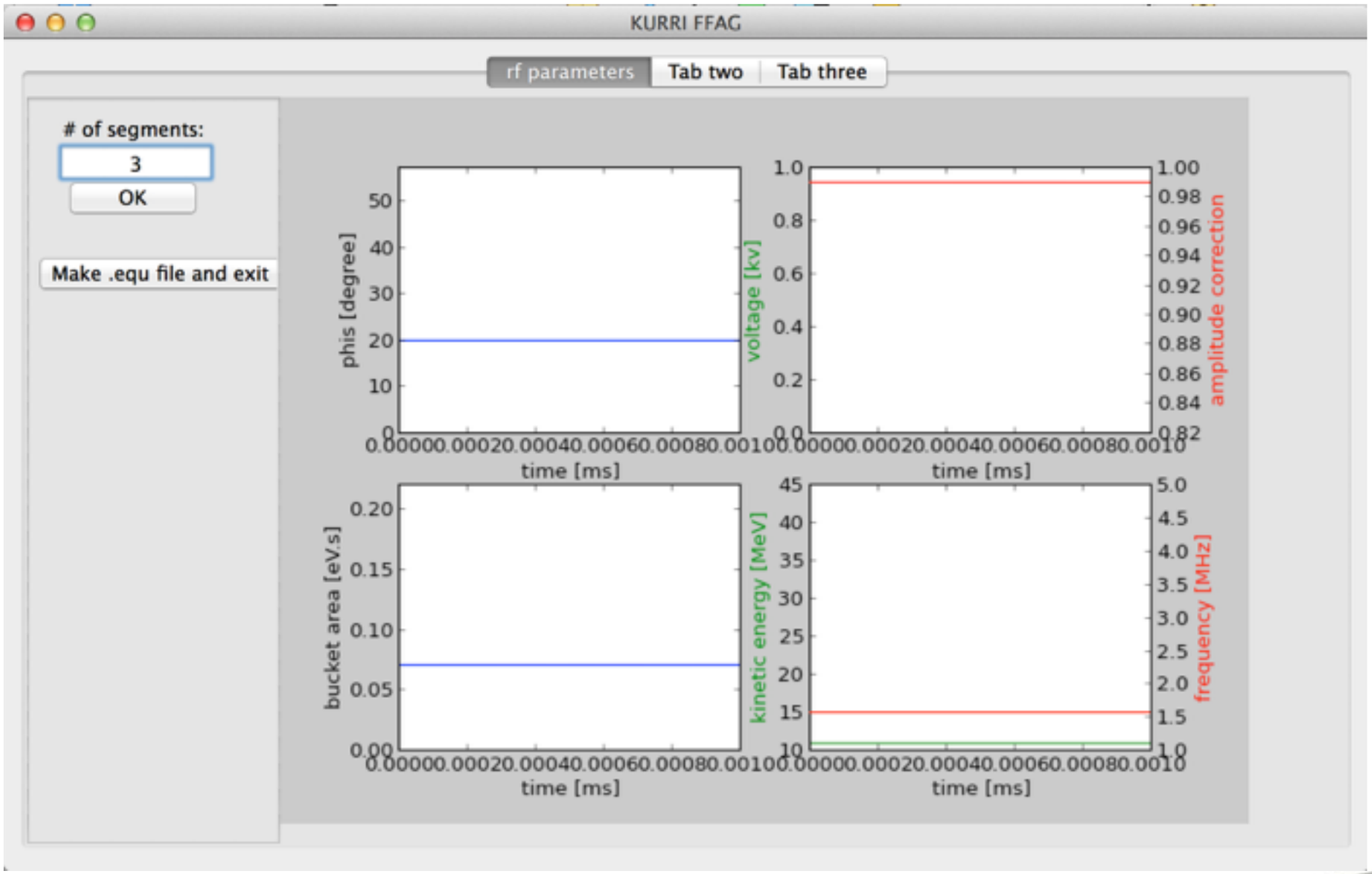
create a new window



Step 2

put number of regions in a cycle and push “OK”.

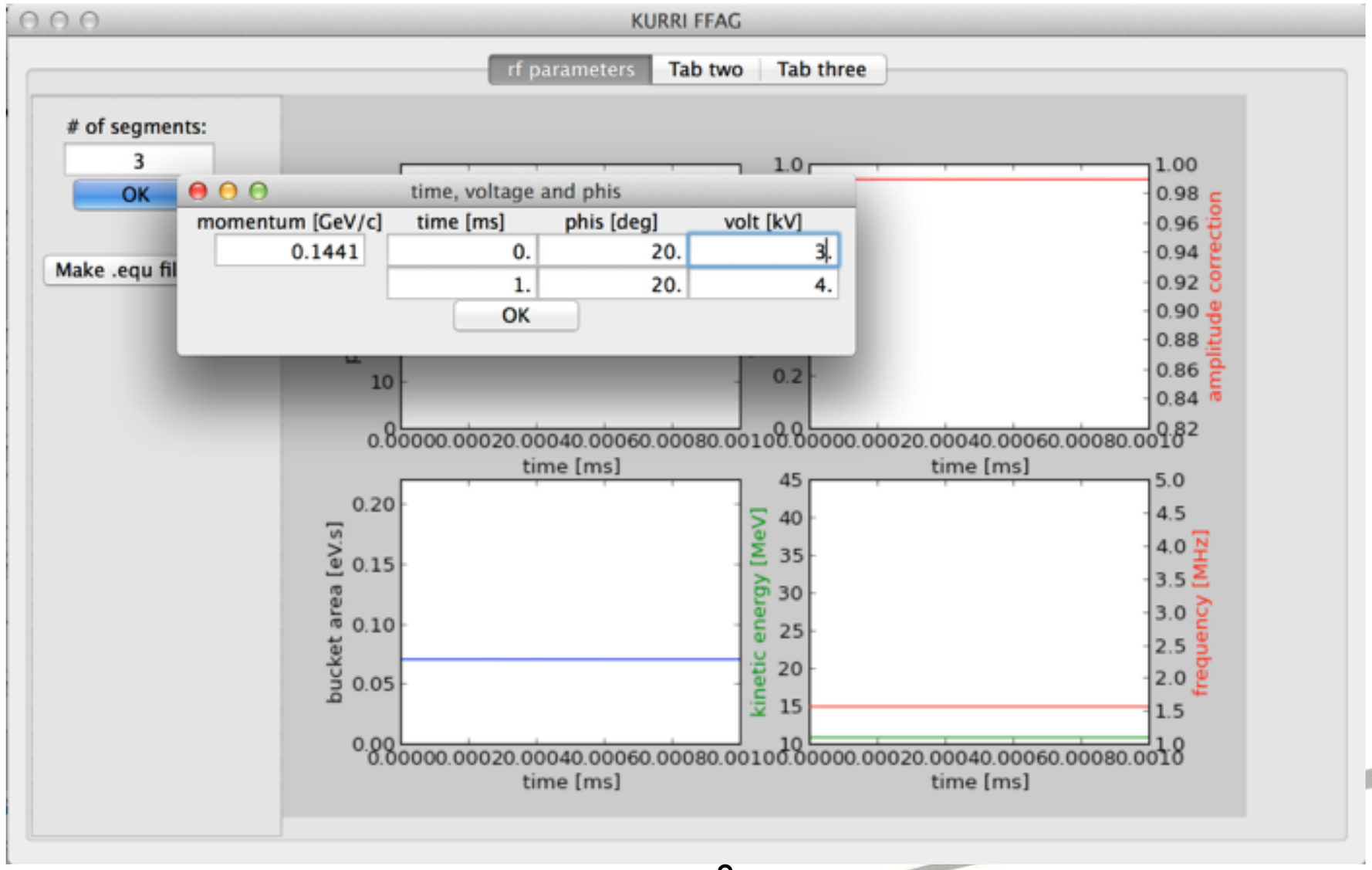
in this example, “3” to specify capture, acc and storage.



Step 3-1

another small window appears

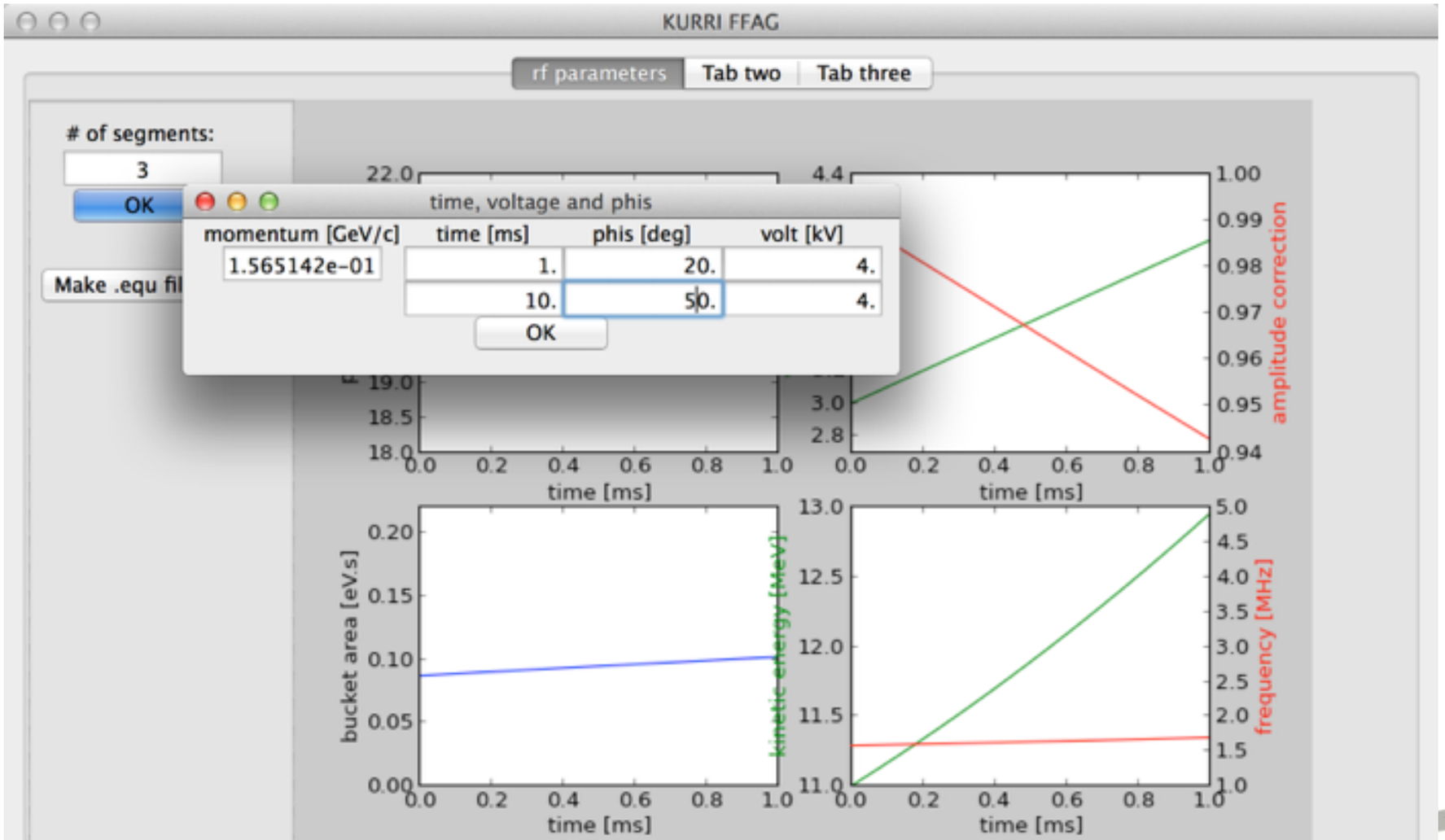
For the capture region (0-1 ms), phis is 20 deg., voltage increases from 3 to 4 kV. Push “OK”.



Step 3-2

another small window appears

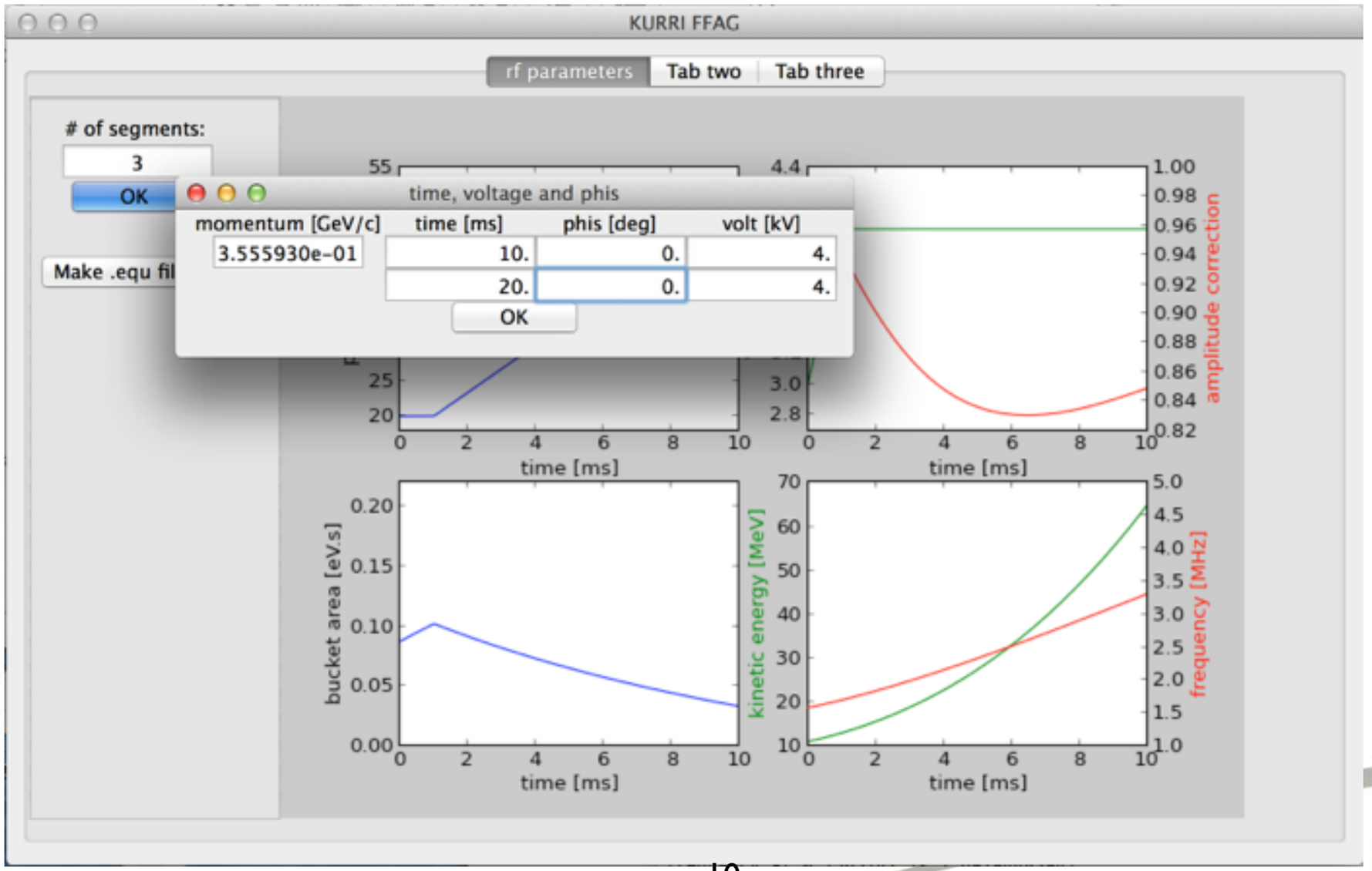
For the acc region (1-10 ms), phis is 20 to 50 deg., voltage is constant with 4 kV. Push “OK”.



Step 3-3

another small window appears

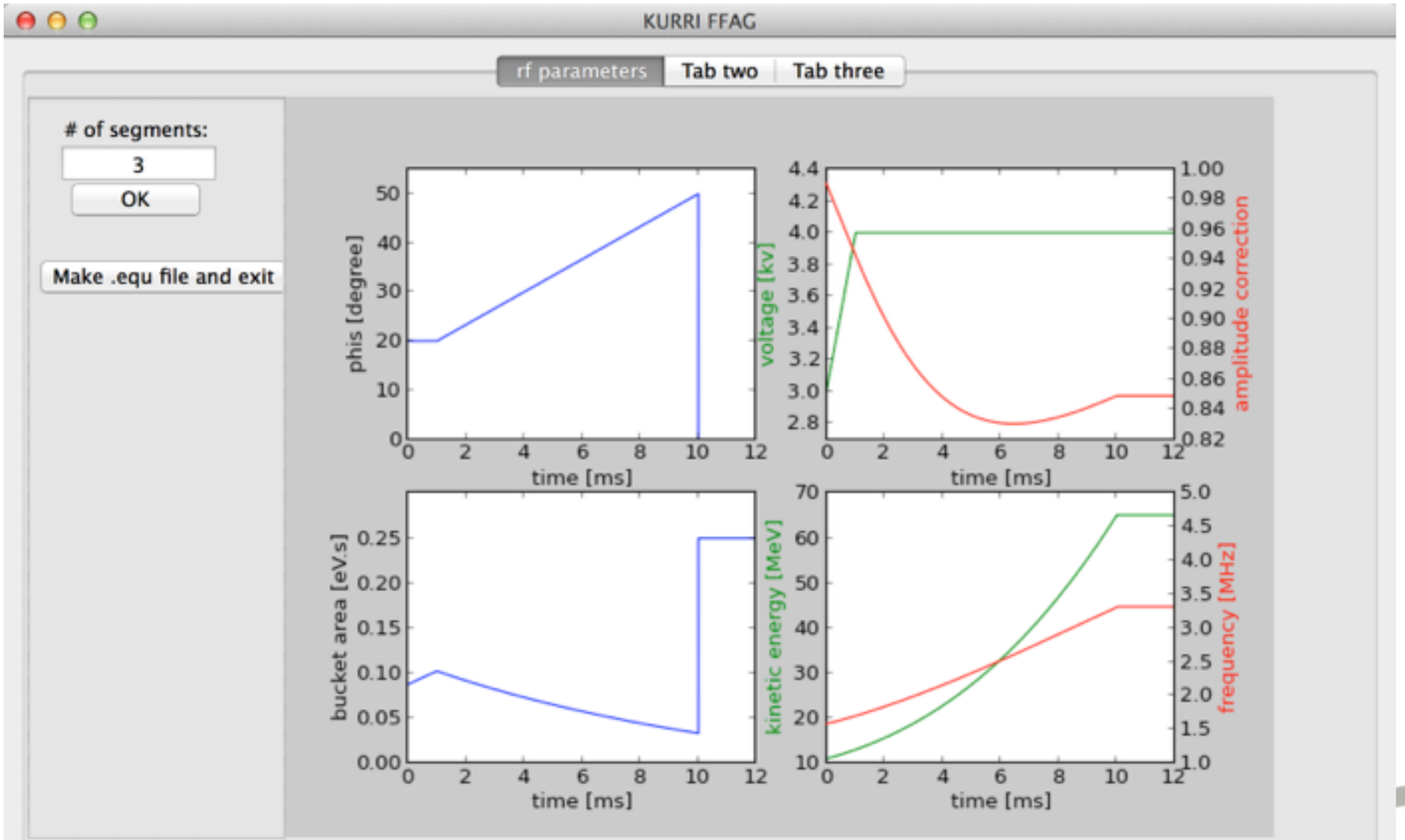
For the storage region (10-12 ms), phis is 0 deg., voltage is constant with 4 kV. Push “OK”.



Science & Technology Facilities Council

Step 4

After specifying parameters for three regions, you will see bucket area, voltage correction factor, frequency vs time. If this is the pattern you want to upload, push button “make .equ file and exit”.



Step 5

.equ file looks like this.

```
'**********          **********
'    tmp_all.equ
'
'
'    by Python script
'    Shinji Machida
'**********          **********

clock = 40000000 ' = 40 M

dT = 0.1e-3
Tini = 0.1e-3
Tfin = 0.1e-3

'***** Initial blank pattern *****
size = Tini*clock
"tmp_ini.wfm"=0.

'***** Final blank pattern *****
size = Tfin*clock
"tmp_fin.wfm"=0.

'***** Coefficients An for phase and Dn for voltage *****
Tacc1 = 0.001
size = Tacc1*clock
A0 = 1582831.0
A1 = 49994128.6096
A2 = 5186179472.94
A3 = -107363763136.0
A4 = 1.14253737138e+13
A5 = -4.29518048922e+15
D0 = 0.9904433
D1 = -47.2126901374
D2 = -1914.9882814
D3 = 1528565.41689
D4 = -38868735.3975
D5 = -18669473806.7
phoff = 0.0
"tmp_acc1.wfm"=sin(2*3.14159265359*(A0*(time-dT)+A1*(time-dT)^1+phoff+A2*(time-dT)^2+phoff+A3*(time-dT)^3+phoff+A4*(time-dT)^4+phoff+A5*(time-dT
)^5+phoff))*(D0+D1*(time-dT)^1+D2*(time-dT)^2+D3*(time-dT)^3+D4*(time-dT)^4+D5*(time-dT)^5)
```

Remarks

Amplitude correction factor is calculated from the previous examples and assuming that it is a function of rf frequency only.

In order to connect region, we may need to add some phase when \phis changes rapidly. It is not included yet.

Whether index k is constant or variable is hardwired in the code at the moment (default is variable k).

When you first run the script, it takes time to finish the first step because it creates intermediate files. It is faster at the second time.

I am sure we need debugging to make it work properly! Please try and tell me the problem.