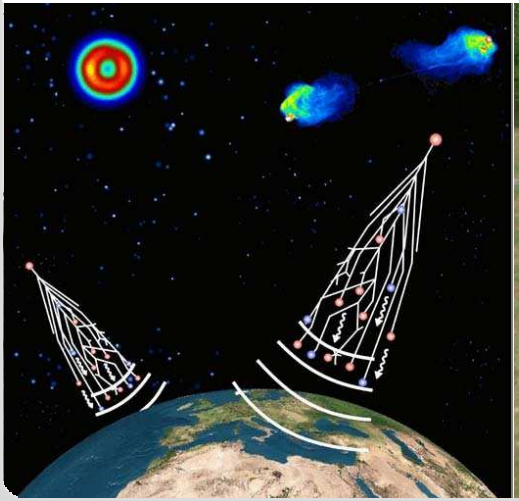
Hands on Exercises: REAS

CORSIKA school 2010, Ooty
Several lines possible.

Marianne Ludwig, Institut für Experimentelle Kernphysik



Outline

- What we want to do today:
  - Installing REAS
  - Processing a simulation chain
  - Using REASplot
- We will start with preparing and running the CORSIKA simulation
- Then we will install REAS and start the REAS simulation, when CORSIKA finished
- While REAS is running, I tell you more about REASPlot and give you examples (on the USB stick there are already finished REAS simulations you can play with)

Notes

- You get this presentation as well as the rings1400m.list from:
  - http://www-ik.fzk.de/~ludwig/
- If you like you can download the presentation as well to have it on your pc
- You can download the files directly with one shell:
  - wget http://www-ik.fzk.de/~ludwig/rings1400m.list
  - wget http://www-ik.fzk.de/~ludwig/LudwigREASexercise.pdf
- I have both on an USB with me

Installing REAS

- You should already know how to install CORSIKA and COAST from the previous exercises (I will repeat this quickly)
- In archives you'll find the reas-3.00 code – copy to the place where you want install REAS, e.g., /home/user/
- Unpack with: **tar –xzvf reas-3.00.tar.gz**
- Before installing make sure that all root-variables are set correctly
- You can use the **setenvironment** file (in exercises) in a bash shell (correct the pathes for your personal settings)

Preparing CORSIKA

- For REAS you will need COAST with the interface THRadio
- Set the environments as yesterday in the COAST exercise (you can use your setEnvironment-script)
- Set the correct root environment
- Compile COAST: **./configure , make install**

```
export ROOTSYS=/usr/local/root
export COAST_DIR=/home/user/coast-v4r1/install
export COAST_USER_LIB=/home/user/coast-v4r1/THRadio
export LD_LIBRARY_PATH=$ROOTSYS/lib:$COAST_DIR/lib:$LD_LIBRARY_PATH
export PATH=$ROOTSYS/bin:$PATH
```

Preparing CORSIKA

- Change your directory where your CORSIKA is installed:
  cd <dir>/corsika-6970
- **./coconut**
- Select SIBYLL and GHEISHA (for the exercise… they are fast)
- Options: Thinning (5), Slant (9) and ROOTRACK (p)
- Finish selection and start compilation (by pressing several enters)

Preparing CORSIKA

- **cd run/**
- Copy the CORSIKA input file from the reas example in the run folder
- **cp <dir>/reas-3.00/examples/RUN000001.inp .**
- Open RUN000001.inp to edit four options (see next slide)
  - ERANGE
  - THETAP
  - THIN and THINH
  - DIRECT (to the reas-3.00/examples folder if you like)

Preparing CORSIKA

```
RUNNR 1
EVTNR 1
SEED    1 0 0
SEED    2 0 0
SEED    3 0 0
PRMPAR     14
ERANGE  1.000E+6 1.000E+6
ESLOPE  0.000E+00
THETAP  40.  40.
PHIP  0.000E+00 0.000E+00
ECUTS 3.000E-01 3.000E-01 4.010E-04 4.010E-04
ELMFLG   T  T
THIN  1.000E-05 1.000E+02 0.000E+00
THINH 1.000E+01 1.000E+01
NSHOW     1
USER  huege
HOST  iklx68
DIRECT  './'
OBSLEV  140000.0
ECTMAP  1.000E+05
STEPFC  1.000E+00
MUMULT   T
MUADDI   T
PAROUT   T  F
MAXPRT      1
MAGNET   18.37 -13.84
LONGI  T    5.  T  T
RADNKG  5.000E+05
DATBAS   F
EXIT
```

Start your CORSIKA simulation

- **./corsika6970Linux_SIBYLL_gheisha < RUN000001.inp > RUN000001.log**
- Your CORSIKA simulation should now take something inbetween 5 to 10 min
- We will use the time to install REAS
- Open a new shell and set again the environment!

Installing REAS – check root libraries

- Load with source setenvironment (or as you prefer)

```
export ROOTSYS=/usr/local/root
export COAST_DIR=/home/user/coast-v4r1/install
export COAST_USER_LIB=/home/user/coast-v4r1/THRadio
export LD_LIBRARY_PATH=$ROOTSYS/lib:$COAST_DIR/lib:$LD_LIBRARY_PATH
export PATH=$ROOTSYS/bin:$PATH
```

- Check, if the libraries are correctly set:

```
nbuser@nbhuegex20:/home/ludwig/reas-3.00$ root-config --incdir
/usr/local/include/root
nbuser@nbhuegex20:/home/ludwig/reas-3.00$ echo $ROOTSYS
/usr/local/include/root
nbuser@nbhuegex20:/home/ludwig/reas-3.00$
```

Installing REAS

- In the reas-3.00 folder:
- **./configure CXXFLAGS=-O3** (for optimized compilation)
- On some debian systems you might have to use with some more information (first, try with above one – if your system is not using the ROOTSYS environment)
- **./configure CPPFLAGS=-I/usr/include/root LDFLAGS=-L/usr/lib/root CXXFLAGS=-O3**
- Use in any case the CXXFLAGS option to get an optimized code

Installing REAS – configure

```
nbuser@nbhuegex20:/home/ludwig/reas-3.00$ ./configure CXXFLAGS=-O3
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
```

```
/reas-3.00$ ./configure CPPFLAGS=-I/usr/include/root/ LDFLAGS=-L/usr/lib/root CXXFLAGS=-O3
install... /usr/bin/install -c
ment is sane... yes
ir -p... /bin/mkdir -p
```

Installing REAS – make

- When ./configure finished, just do
- **make**

```
checking how to hardcode library paths into programs... immediate
configure: creating ./config.status
config.status: creating Makefile
config.status: creating shared/Makefile
config.status: creating reasplot/Makefile
config.status: creating reas/Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing depfiles commands
/home/ludwig/reas-3.00>make
make  all-recursive
make[1]: Entering directory `/home/ludwig/reas-3.00'
Making all in shared
```

Installing REAS – make

- If you got no error message, REAS is installed

```
make[2]: Leaving directory `/home/ludwig/reas-3.00/reas'
make[2]: Entering directory `/home/ludwig/reas-3.00'
make[2]: Leaving directory `/home/ludwig/reas-3.00'
make[1]: Leaving directory `/home/ludwig/reas-3.00'
/home/ludwig/reas-3.00>
```

- If you got some error (missing files) you have to use the second option for taking ./configure

- **./configure CPPFLAGS ….** Where your pathes are given for root

```
/reas-3.00$ ./configure CPPFLAGS=-I/usr/include/root/ LDFLAGS=-L/usr/lib/root CXXFLAGS=-03
install... /usr/bin/install -c
ment is sane... yes
ir -p... /bin/mkdir -p
```

- and once again **make**

Using REAS

- You have to edit the event.reas file
- You have only to change the path to your CORSIKA files (the DAT000001.hist.root, … files)
- Your new RUN000001.inp file has to be in the same folder.. So maybe copy it there from your corsika-6970/run folder

```
# parameters specific to CORSIKA based showers - other parameters are read from the CORSIKA input file:

CorsikaFilePath = ./                    ; path to the CORSIKA files (cannot include space characters!)
CorsikaParameterFile = RUN000001.inp          ; specify CORSIKA card file
CorsikaSlantOptionToggle = 1            ; set to 1 if CORSIKA option SLANT is used
SelectedCorsikaShower = 1               ; 0: averaged, i: i-th shower
ShowerEvolutionShift = 0                ; apply slant depth shift to CORSIKA-derived shower evolution,
```

Starting REAS

- In <dir>/reas-3.00/examples:
- Your input parameters have the name: event.reas and all.list
- **../reas/reas event all**
- If your PC allows (or you calculate on a cluster) you can start the same simulation (event.reas) with different antenna files (e.g., east.list)

  (helpful to save some calculation time for one simulation if you have the capacities)
- ../reas/reas event east

Starting REAS

- Your simulation should now run for a while (depending on your system)

```
nbuser@nbhuegex20:/home/ludwig/reas-3.00/examples$ ../reas/reas event all

REAS V3 by Tim Huege & Marianne Ludwig (binary: ../reas/reas)

When publishing results obtained with this code, please cite:

Ludwig, M., Huege, T. 2010, Astroparticle Physics, doi:10.1016/j.astropartphys.2010.10.012

Parameter file event.reas successfully imported!
Antenna list file all.list successfully imported!
CORSIKA file ./RUN000001.inp successfully imported! Importing parameters!
File/Program parameter version: 22/22  -->  match!

Starting calculation:

Random seed is set to: 1292730997
Importing ROOT histograms file ./DAT000001_1.hist.root ... done.
Longfile-Xmax: 480    Rootfile-Xmax: 490.844
Total number of injected particles: 5.707e+09

Warning: Particle weight is only 0.5707! Result will be overly smoothed!

Shower maximum is at slant depth: 490.844 g cm^-2
Geomagnetic angle is: 126.994 degrees
Calculate particles 0 to 1000000 of 10000000000 ...  took 30 s --> ETA: 299970 s (at 30 s of
```

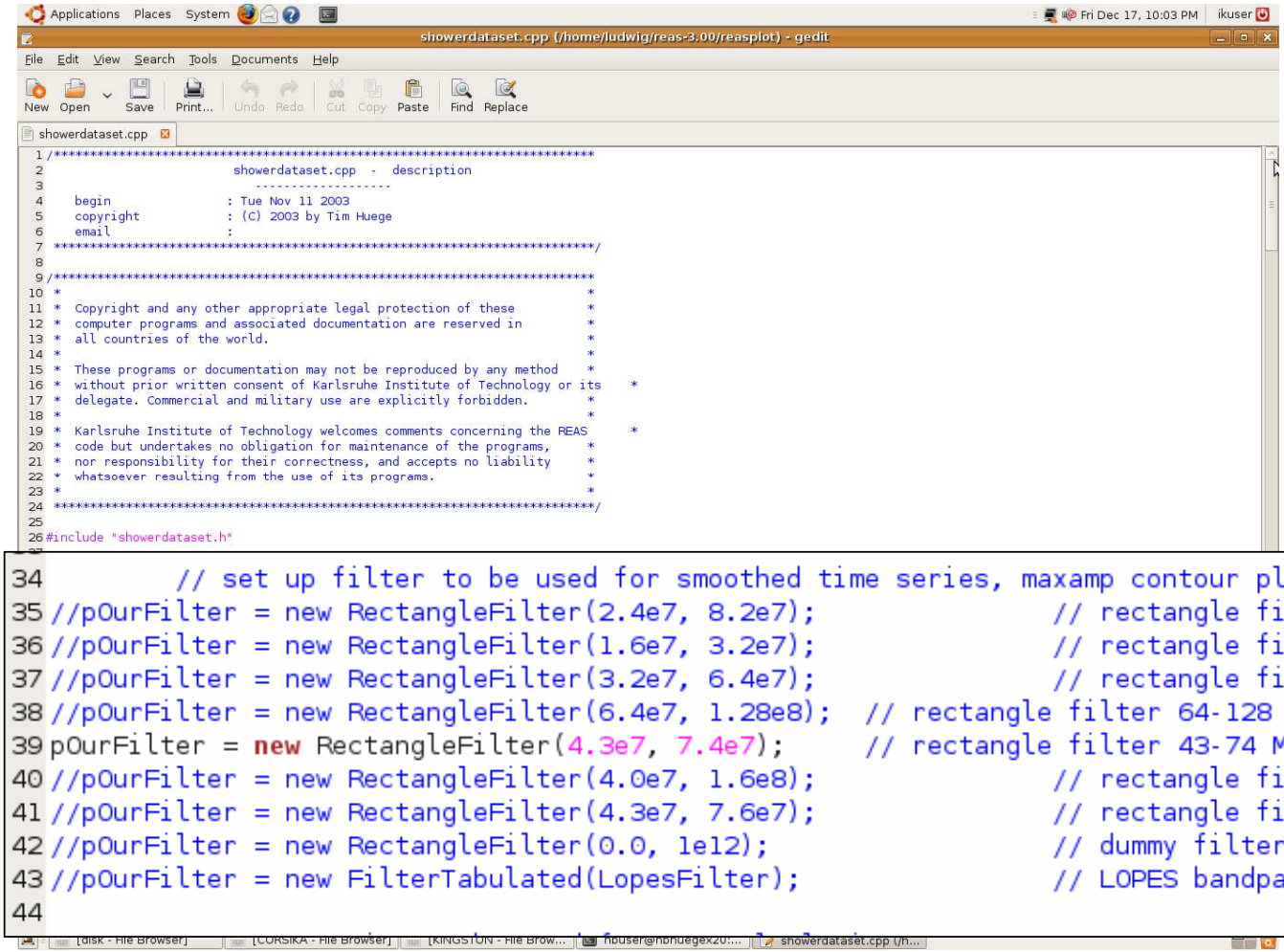
Using REASPlot

- Open a new shell and set your environment correctly
- There is an extra folder on the USB stick: exercises
- Go there: **tar –xzvf reas-exercises.tar.gz**
- You have there two finished REAS simulations (a vertical shower (SIM000001.reas) and a inclined shower (SIM000007.reas)
- Copy the **rings1400m.list** in this folder
- You get it with: wget http://www-ik.fzk.de/~ludwig/rings1400m.list

REASPlot

- You can select different bandwidths for your filter
- For this you have to open with an editor the **showerdataset.cpp** in reasplot with an editor you like:
- **gedit <dir>/reas-3.00/reasplot/showerdataset.cpp**
- You can keep the filter which is selected for the first exercise with REASPlot

Slecting the filter




```
34         // set up filter to be used for smoothed time series, maxamp contour plots and snr contou
35 //pOurFilter = new RectangleFilter(2.4e7, 8.2e7);                // rectangle filter 24-82 MHz
36 //pOurFilter = new RectangleFilter(1.6e7, 3.2e7);                // rectangle filter 16-32 MHz
37 //pOurFilter = new RectangleFilter(3.2e7, 6.4e7);                // rectangle filter 32-64 MHz
38 //pOurFilter = new RectangleFilter(6.4e7, 1.28e8);  // rectangle filter 64-128 MHz
39 pOurFilter = new RectangleFilter(4.3e7, 7.4e7);     // rectangle filter 43-74 MHz
40 //pOurFilter = new RectangleFilter(4.0e7, 1.6e8);                // rectangle filter 40-160 MHz
41 //pOurFilter = new RectangleFilter(4.3e7, 7.6e7);                // rectangle filter 43-76 MHz
42 //pOurFilter = new RectangleFilter(0.0, 1e12);                   // dummy filter
43 //pOurFilter = new FilterTabulated(LopesFilter);                 // LOPES bandpass filter
44
```

Selecting filter

- In the same file you can also slect for which frequencies you want the spectral field strengths to be written out

```
48
49          // set up sample frequencies (in ascending order!) for spectral contour plots and azimuth c
50 //itsSpectrumFrequencies.push_back(1.0e4);                                    // 10 kHz       ---
   <10 kHz !!!
51          itsSpectrumFrequencies.push_back(1.0e7);                                      // 10 MHz
52          itsSpectrumFrequencies.push_back(3.0e7);                                      // 30 MHz
53          itsSpectrumFrequencies.push_back(6.0e7);                                      // 60 MHz
54 //itsSpectrumFrequencies.push_back(7.5e7);                                    // 75 MHz
55          itsSpectrumFrequencies.push_back(1.0e8);                                      // 100 MHz
56
```

- Keep everything as it is for the beginning

Processing data with REASPlot

- You have to specify a folder where you want to write your output, e.g., SIM000001_filtered_43to74MHz

- **<dir>/reas-3.00/reasplot/reasplot SIM000001_filtered_43to74MHz SIM000001 rings1400m**

- The same you can do for SIM000007

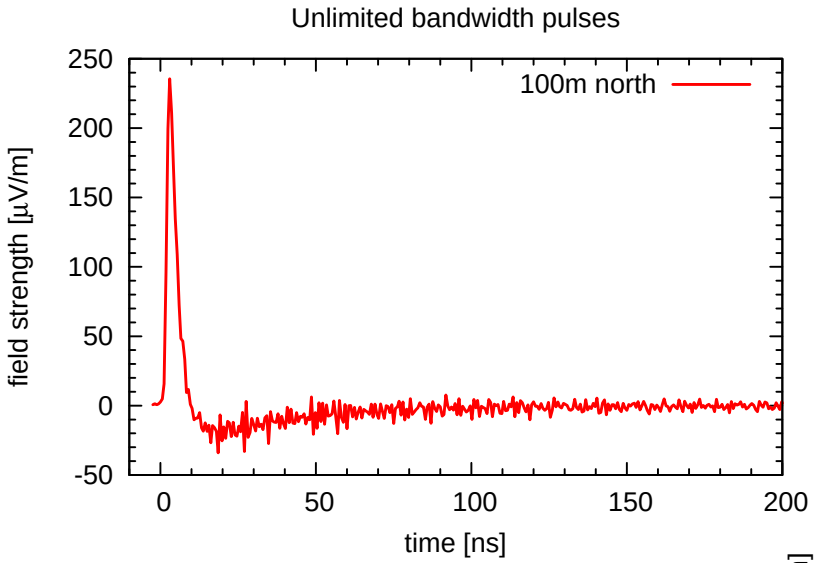
```
../reas-3.00/reasplot/reasplot SIM000007_filtered_43to76MHz SIM000007 rings1400m
```

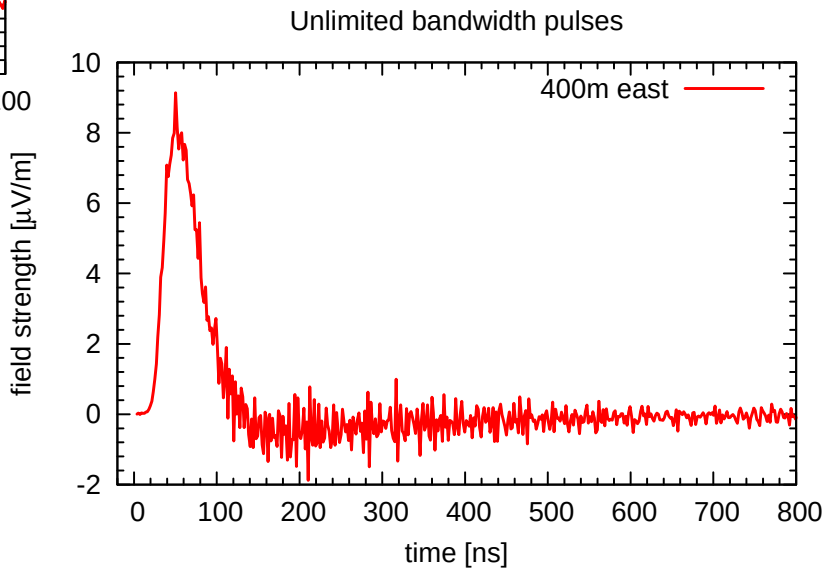
```
REASPlot V1.0 by Tim Huege

Processing SIM000007_rings1400m/raw_pole_50m_0deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_45deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_90deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_135deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_180deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_225deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_270deg.dat ...
Processing SIM000007_rings1400m/raw_pole_50m_315deg.dat ...
Processing SIM000007_rings1400m/raw_pole_100m_0deg.dat ...
```

Exercises

- Now you know how to run REAS and REASPlot
- Remember the data format (time, x=north, y= west, z=vertical)
- Let‘s start with some exercises to use the output:
- 1. Create a plot with the raw pulses of SIM00001 for the EW-polarisation
  - For an observer 100m in the North (0deg)
  - For an observer 400m in the East (270deg)
  - Convert the: x-axis into ns and the y-axis into µV/m
  - For cgs to µV/m you will need the factor: 2.99792458e10

Results


Unlimited bandwidth pulses
100m north
field strength [µV/m]
time [ns]
250
200
150
100
50
0
-50
0
50
100
150
200


Unlimited bandwidth pulses
400m east
field strength [µV/m]
time [ns]
10
8
6
4
2
0
-2
0
100
200
300
400
500
600
700
800

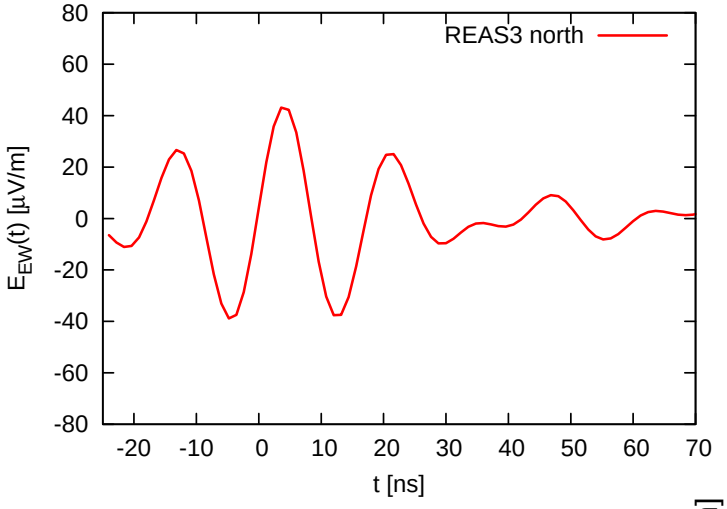
Exercise 2

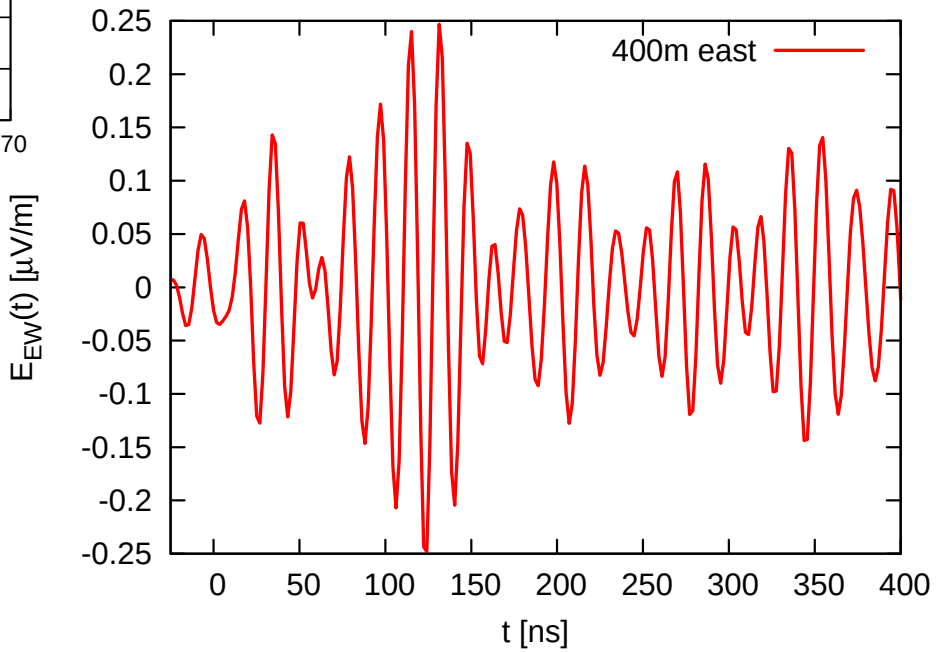
- Now create the plots for the filtered pulses
- Again for 100m north and 400m east

- You will need the **smooth_pole_100m_0deg.dat**
- You find it in the folder of your REASPlot output

- Keep the axis as in exercise 1

Exercise 2 - results



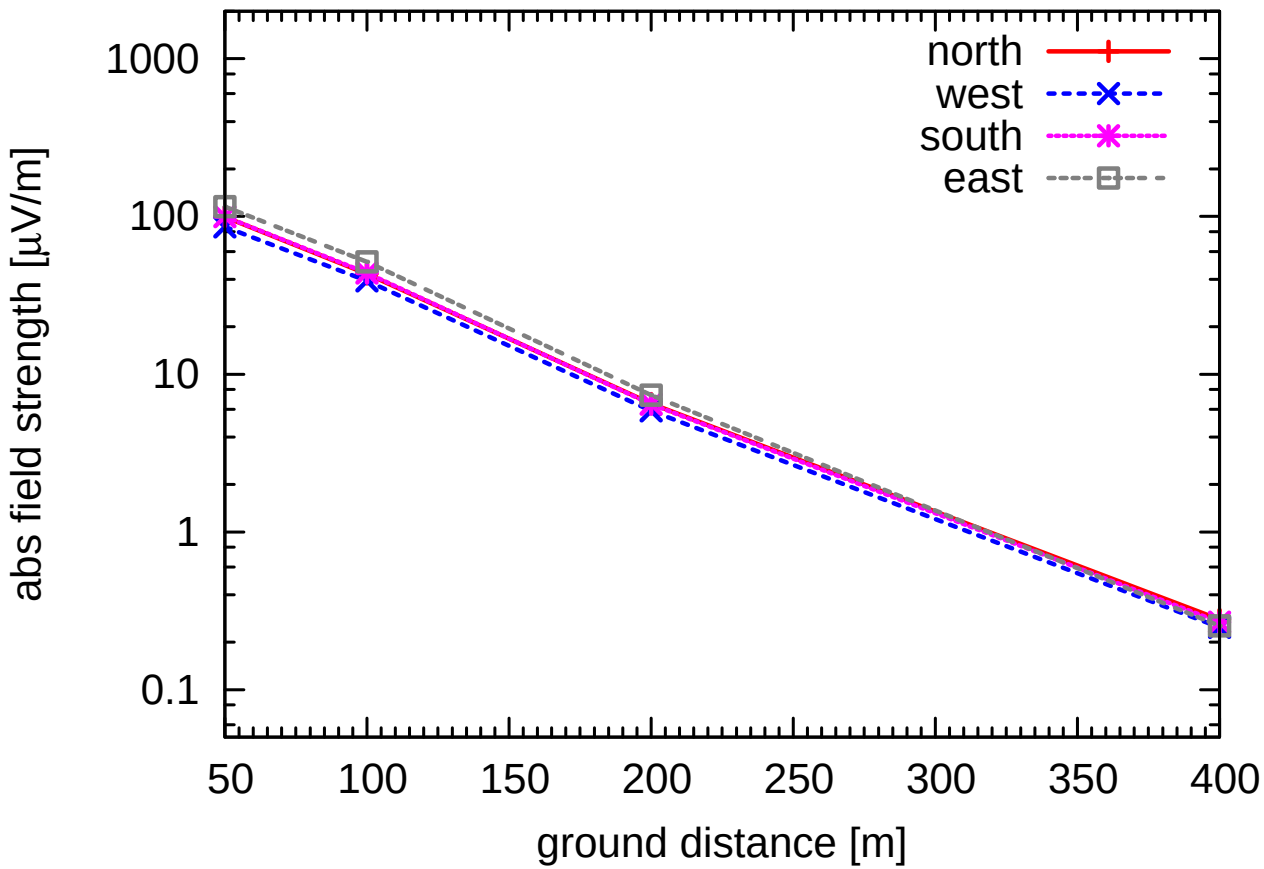


Exercise 3 – lateral distribution

- Create a plot for the lateral distribution of the radio signal for the four directions: north, west, south, east
- Plot it for the absolute field strength
- Convert the axis again in ns and µV/m
- You should use: **maxamp_*deg.dat**

  (data format: radial distance from shower core, x=north, y=west, z=vert, time stamp (where the pulse maximum was reached))

Exercise 3 - result



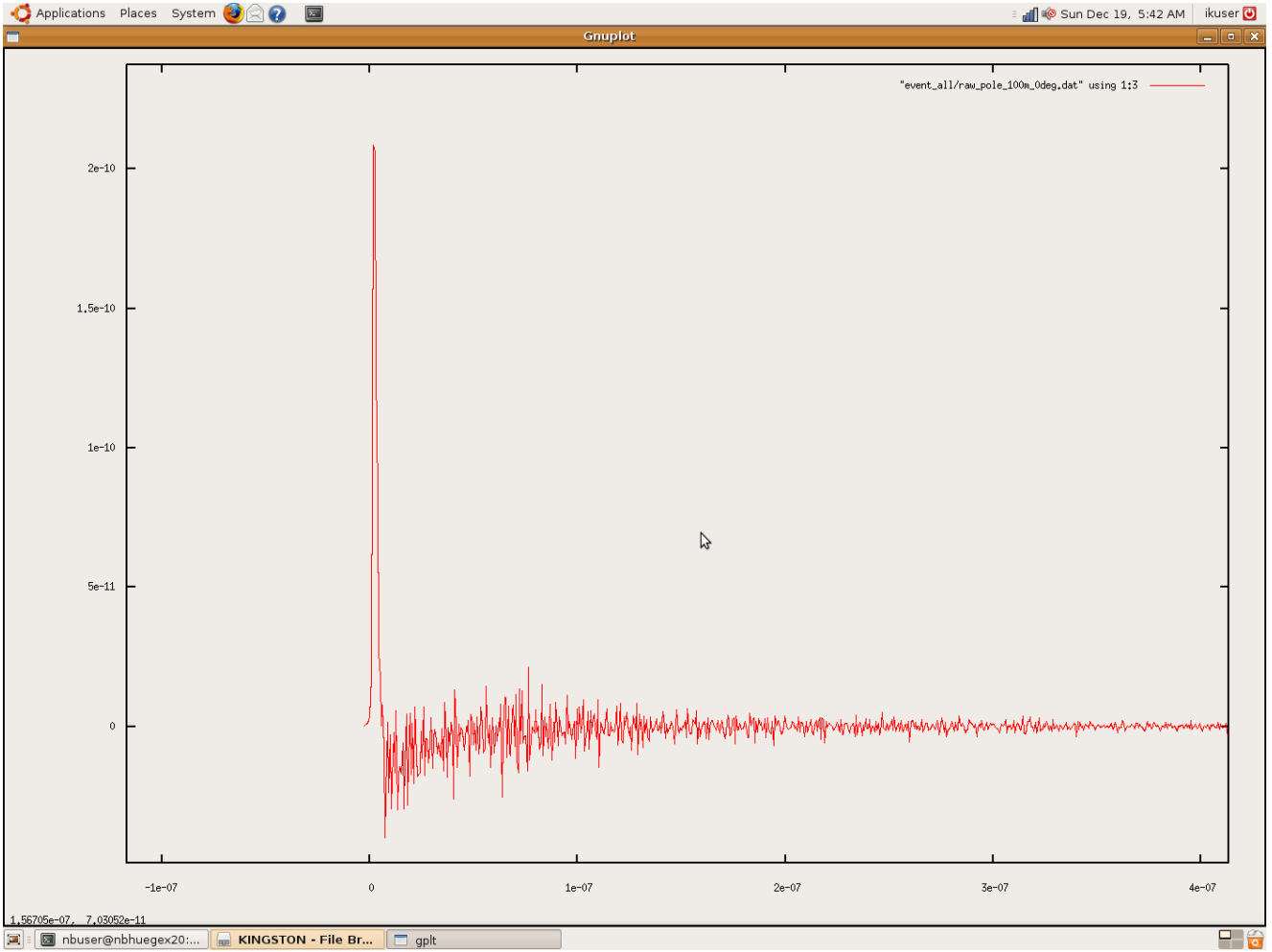
Exercise 5 – filter options

- Chose a different filter for REASPlot
- Do **make** and process the data once again
- Do the exercises for the new filter selection and compare with the first one
- You can also create some frequency spectra,
- For transformation of cgs units you can check the spectra.gnu file which is given in the reas-3.00/examples - folder

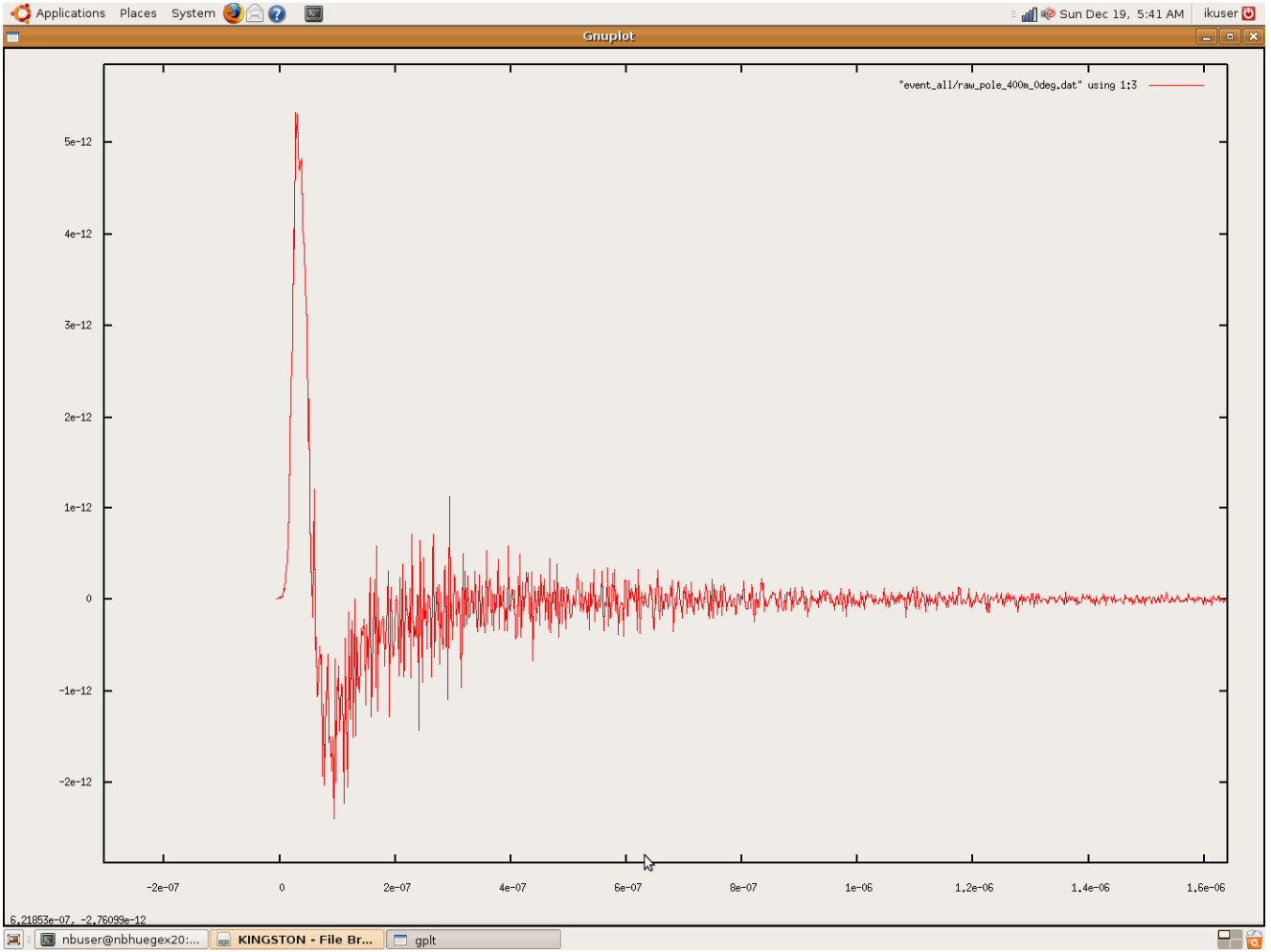
Your REAS simulation

- If your REAS simulation from the beginning is still running:
- You can stop the simulation with CTRL-C
- The calculations which were done so far will be written in the event_all folder
- Your pulses might be noisy – but to get an idea if everything is working they are already good enough
- Just check the raw pulses and you will see the quality
- In my case: after ~40min calculation time for event.reas and all.list

Raw pulse 100m north (ew, in cgs units)



Raw pulse 400m north (ew, in cgs units)



… the end

- There is still more output you get with REAS(plot)
- If you are interested in radio emission, please read the manual where all the output and the possibilities are explained
- If you are interested in REAS please register at the webpage or give me your email adress (that you can get information on updates,…)

Thank you!