100 nm Al on Fe 20kV base case 1000 ch

John Minter

Started 2018-08-07, Last modified: 2018-08-13

Abstract

The objective was to analyze the base case for 100 nm of Al on Fe at 15 kV using the base case from Llovet and Salvat 2018 table 3 case 1 using the Al K-L3 ($K_{\alpha\ 1}$) transition. In this first (baseline) simulation I only collected 1000 channels of data.

Contents

1 Set up for the analysis

Note: This simulation used the same `MSIMPA` parameters for all layers and a `REFLIN` parameter (`5E-1`) to finish.

```
MFNAME Al.mat                          [Material file, up to 20 chars]
MSIMPA 1.0E+3 1.0E+3 1E+3 0 0 0 -1E+2      [EABS(1:3),C1,C2,WCC,WCR]
MFNAME Fe.mat                          [Material file, up to 20 chars]
MSIMPA 1.0E+3 1.0E+3 1E+3 0.2 0.2 1 1      [EABS(1:3),C1,C2,WCC,WCR]
```

First load the libraries we need...

```
library(rpenepma)
library(dplyr)
library(pander)
panderOptions('table.split.table', Inf)
library(here)
library(ggplot2)
```

Next, set the paths and constants that we need...

```
moved_data  <- TRUE  # set to TRUE when sim is done and .rda with data is moved...
e0          <- 15 # kV
int_lo_lim  <- 1.0e-1
int_hi_lim  <- 1.0e4
```

```
reflin_crit <- 5e-1
sim_dir <- here("penepma/penepma16/Al-100nm-on-Fe/Al_100_Fe_15kV_1000_ch")

sim_nam     <- "Al-100nm-on-Fe-base-1000-channels"
res_fi      <- "./penepma-res.dat"
int_fi      <- "./pe-intens-01.dat"
spc_fi      <- "./pe-spect-01.dat"
out_ti      <- sprintf("%s-%gkV", sim_nam, e0)
msa_fi      <- sprintf("./%s-%gkV.msa", sim_nam, e0)
```

2 Get the current simulation time

```
if (moved_data==FALSE){
  # When we have not moved the simulation, measure the simulation time.
  # Note copying the files to another directory seems to change the time
  # Test a reload.
  delta_t <- penepma_measure_simulation_time(sim_dir)
  # print(delta_t)
  save(delta_t, file="./delta_t.rda")
  rm(delta_t)
  load("./delta_t.rda")
  msg <- "Analyzed from simulation directory."
} else {
  # When we have moved the simulation, reload the delta_t.rda
  msg <- "Analyzed from data stored in delta_t.rda."
  load("./delta_t.rda")
}
cat(c(msg, "\nCurrent simulation time:",
      round(as.numeric(delta_t, units="hours"), 3), "hrs"))
```

```
Analyzed from data stored in delta_t.rda.
Current simulation time: 1.26 hrs
```

3 Retrieve the number of showers

We want retrieve the number of showers (trajectories) that our `penepma16` simulation calculated.

```
options(scipen = -6) # force printing in exponential format
num_showers <- penepma_get_number_of_showers(res_fi)
options(scipen = 3) # reset to default
cat("Number of trajectories: ", num_showers)
```

```
Number of trajectories:  3146
```

4 Compute the total maximum intensity for each element

4.1 Retrieve the data

First retrieve the intensity data as a `tibble` and print a preview.

```
tib <- penepma_get_intensities(int_fi)
print(tib)
```

```
# A tibble: 7 x 14
     IZ S0    S1      keV    P.mu    P.se    C.mu    C.se    B.mu    B.se
  <int> <chr> <chr> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1    13 K     L2     1.49 1.06e-5 7.05e-6 0.      0.      0.      0.
2    13 K     L3     1.49 1.83e-5 8.85e-6 1.14e-8 3.42e-8 1.15e-7 1.99e-7
3    13 K     M3     1.56 4.81e-7 1.44e-6 0.      0.      0.      0.
4    26 K     L2     6.39 4.46e-5 2.34e-6 2.28e-7 1.52e-7 1.69e-6 8.31e-7
5    26 K     L3     6.40 8.66e-5 3.52e-6 5.70e-7 2.45e-7 3.31e-6 1.14e-6
6    26 K     M2     7.06 5.26e-6 7.49e-7 1.14e-8 3.42e-8 1.53e-7 2.29e-7
7    26 K     M3     7.06 1.11e-5 1.10e-6 2.28e-8 4.83e-8 3.09e-7 3.15e-7
# ... with 4 more variables: TF.mu <dbl>, TF.se <dbl>, Int.mu <dbl>,
#   Int.se <dbl>
```

What features did we measure?

```
print(names(tib))
```

```
 [1] "IZ"     "S0"     "S1"     "keV"    "P.mu"   "P.se"   "C.mu"
 [8] "C.se"   "B.mu"   "B.se"   "TF.mu"  "TF.se"  "Int.mu" "Int.se"
```

How many transitions did we measure?

```
print(nrow(tib))
```

```
[1] 7
```

Measure our test transition ("Al K-L3")

```
val <- penepma_get_total_intensity_z_transition(tib, 13, "K", "L3")
print(val)
```

```
# A tibble: 1 x 7
  Symbol    IZ S0    S1       Int.mu     Int.se Int.snr
  <chr>  <int> <chr> <chr>     <dbl>      <dbl>   <dbl>
1 Al        13 K     L3    0.0000184 0.00000887    2.08
```

4.2 Compute the maximum total intensity for each element

1. Compute the maximum total intensity for each element.
2. Stack each row of data together into a tibble.
3. Prepend the sample ID to the data.
4. Print a well-formatted a table using `pander`.
5. Save the final tibble so it can be r-loaded later

```
# 1
Al_t   <- penepma_get_max_total_intensity_z(tib, 13)
Fe_t   <- penepma_get_max_total_intensity_z(tib, 26)
# 2
tot_int <- dplyr::bind_rows(Al_t, Fe_t)
#3
Al_100_Fe_15kV_1000_ch <- prepend_sample_id_max_int_tib(tot_int, out_ti)
# 4
pander(Al_100_Fe_15kV_1000_ch)
```

Sample.ID	Symbol	IZ	S0	S1	Int.mu	Int.se	Int.snr
Al-100nm-on-Fe-base-1000-channels-15kV	Al	13	K	L3	0.00001841	0.00000887	2.076
Al-100nm-on-Fe-base-1000-channels-15kV	Fe	26	K	L3	0.0000905	0.0000038	23.81

```
# 5
save(Al_100_Fe_15kV_1000_ch,
     file="Al_100_Fe_15kV_1000_ch.rda")
# delete and reload be sure it worked...
rm(Al_100_Fe_15kV)
load("./Al_100_Fe_15kV_1000_ch.rda")
```

5 Process the spectra data

5.1 Load the raw data

```
tib <- penepma_read_raw_data(spc_fi)
rownames(tib) <- c()
```

5.2 Summarize the spectrum data

How many rows of data are there?

```
cat(c("Number of measurements: ", nrow(tib)))
```

```
Number of measurements:  282
```

Generate a summary:

```
pander(summary(tib))
```

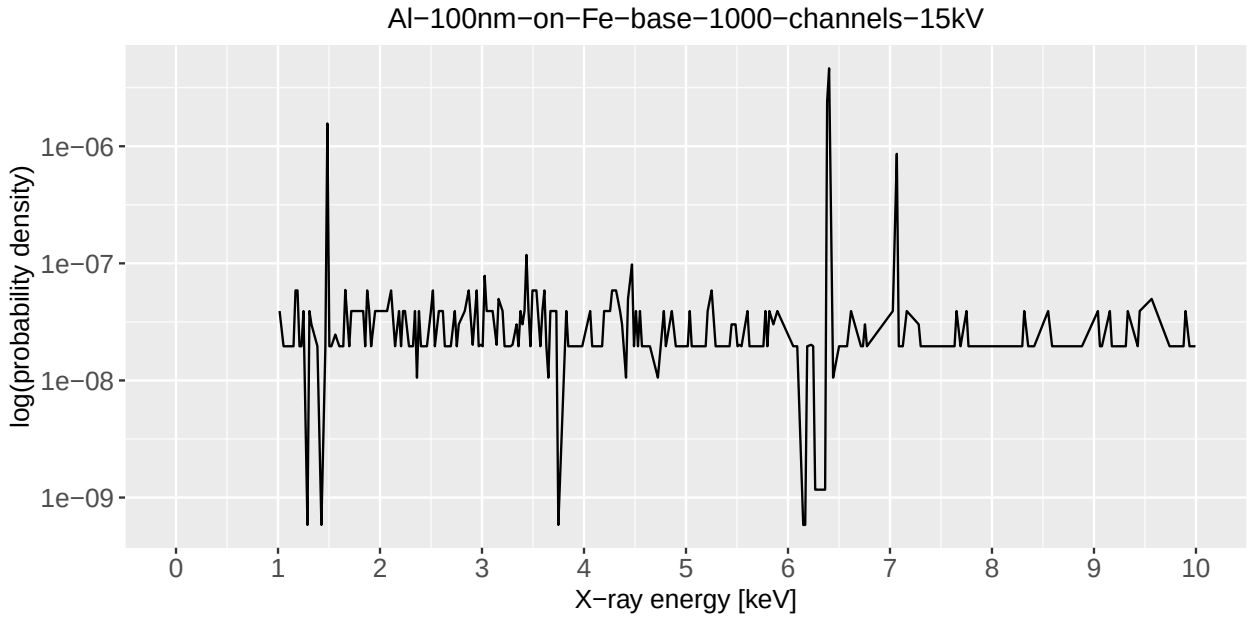
keV	pd.mu	pd.unc
Min. : 1.015	Min. :5.840e-10	Min. :1.751e-09
1st Qu.: 3.055	1st Qu.:1.958e-08	1st Qu.:5.873e-08
Median : 5.115	Median :1.958e-08	Median :5.873e-08
Mean : 6.206	Mean :6.000e-08	Mean :6.838e-08
3rd Qu.: 8.604	3rd Qu.:3.916e-08	3rd Qu.:8.304e-08
Max. :18.389	Max. :4.633e-06	Max. :5.982e-07

5.3 Plot the spectrum.

Start with a log-intensity plot.

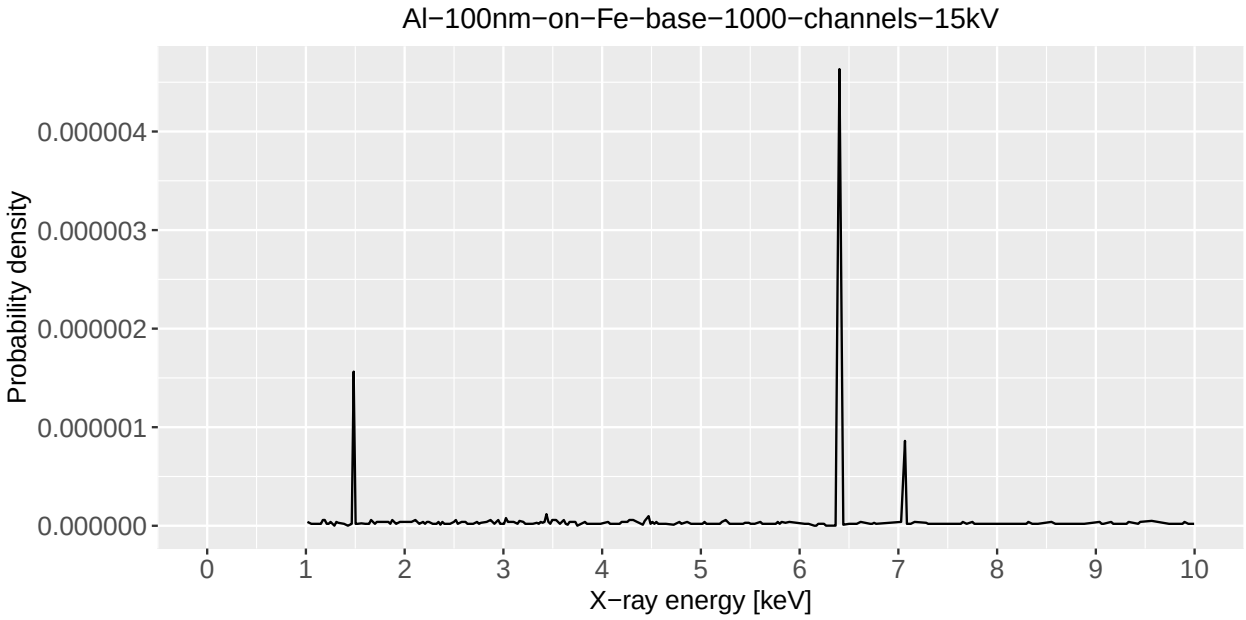
```
m_t <- max(pretty(tib$mu))

plt <- ggplot(tib, aes(x = keV, y = pd.mu)) +
       geom_line() +
       scale_x_continuous(breaks = seq(from = 0, to = e0-5, by = 1),
```

```
                          limits = c(0,e0-5)) +
       scale_y_log10() +
       xlab(label="X-ray energy [keV]") +
       ylab(label="log(probability density)") +
       # (1/(eV*sr*electron)") +
       ggtitle(out_ti) +
       theme(axis.text=element_text(size=12),
             axis.title=element_text(size=12),
             # center the title
             plot.title = element_text(hjust = 0.5))

print(plt)
```



And then on a linear intensity scale...

```
plt <- ggplot(tib, aes(x = keV, y = pd.mu)) +
       geom_line() +
       scale_x_continuous(breaks = seq(from = 0, to = e0-5, by = 1),
                          limits = c(0,e0-5)) +
       scale_y_continuous() +
       xlab(label="X-ray energy [keV]") +
       ylab(label="Probability density") +
       # (1/(eV*sr*electron)") +
       ggtitle(out_ti) +
       theme(axis.text=element_text(size=12),
             axis.title=element_text(size=12),
             # center the title
             plot.title = element_text(hjust = 0.5))

print(plt)
```



5.4 Write a spectrum file in MSA format

```
penepma_to_msa(spc_fi, msa_fi,e0, out_ti)
```