



# MoNA High Voltage Fitting Code

K. Yoneda, W.A. Peters  
National Superconducting Cyclotron Laboratory

July 13, 2006

## 1 HV Cosmic fitting Code

This program fits a function of a Gaussian peak plus an exponential background to the MoNA cosmic ray spectra. The purpose of this program is to obtain high voltage values proper for the QDC dynamic range you want to set, and to obtain the calibration parameters and QDC threshold values. You need to select one of the two sets of output files available:

1. A high voltage file. The difference between the cosmic ray peak and the pedestal peak is set to be 800 channel in the raw spectrum for each high voltage channel.
2. Calibration files. The pedestal and cosmic ray peak positions are set to be 0 MeV and 22 MeV in equivalent electron energy in the calibrated spectra, respectively. The following describes how to use the program briefly.

### 1.1 High Voltage

According to the plan of the program author, the fitting is done automatically for all the MoNA spectra when you run the program with this shell script.

```
#!/bin/sh
./fit_monaspectra "EOF
monacos.inp
monacosdata/MoNArwQDCcosmics.asc
16
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
test.tdr
EOF
exit
cp HV'cat datetmp.txt'.tcl mona/hv_control/hv_files
In -sf "mona/hv_control/hv_files/HV'cat datetmp.txt'.tcl \
"mona/hv_control/hv_files/QDCfittedHV.tcl
exit
```

This shell script is placed in the same directory with the filename `hv. sh`. You can copy it and modify it as you like. `fit-monospectra` is the name of the executable file of the program. This program requires several inputs. The inputs given in the above sample shell script mean,

- 2: A flag for high voltage file output.
- `monacos.inp`: A file giving initial values, presumable errors, and ranges of the fitting parameters. Behavior of the fitting program is strongly dependent on the initial values of the fitting parameters, so you need to give the values carefully. As far as the author checked, the set of parameter values written in `monacos.inp` worked well for the MoNA cosmic ray spectra fitting, after changing several values using the result of the preliminary peak search in the program. For more details, see the next section.
- `monacosdata/MoNARawQDCcosmics.asc`: An ascii file of the histograms obtained in the DAQ system. The spectrum information is written sequentially for all the MoNA high voltage channels. `monacos-data` is a symbolic link to the directory of the MoNA cosmic ray spectra.
- 1
- 16
- 112 113 114 ... 127: These numbers are to specify which channels you want to skip/analyze. If you want to skip certain channels, you must set the first line 1. The number in the second line specifies the number of channels you want to skip, and the numbers in the last line are the high voltage channels you want to skip. If you want to do fitting only for some certain channels, you must set the first line set 2. The second and third lines give the number of channels and the high voltage channel numbers you want to analyze. If you want to do fitting for all the channels, you must set the first line 0, and the second and the third lines must be eliminated. In this sample, the high voltage channels from 112 to 127 are skipped. Note that you cannot change the line for the list of the channels, even if the number of channels to be listed is very large. In such a case it may be easier to change the source code temporarily. The next section describes briefly how to change the source code.
- `test.tdr`: An output file in which the program writes spectra with fitted curves. This file is formatted so that you can check the spectra and fitted curves with the plotter program `TopDrawer`. When the file already exists, the old one is overwritten. The program creates two output files except for the above spectrum file.
- `datetmp.txt`: A file to record the date and time when the program runs. When the file already exists, the old one is overwritten.
- `hv(date and time).tcl`: A high voltage file written in the format of the `tcl` script. The part of the parentheses (date and time) is filled by twelve digits representing date and time, which is also written in the file `datetmp.txt`. It is strongly recommended to confirm the quality of the fitting by checking the spectra and the fitted curves. In order to see the spectra, type:

```
> ~yoneda/bin/td test.tdr
> gv test.ps
```

then you can check them with a `postscript` file.

If you remove the line `exit` next to the EOF or comment it out with a `#` at the top of the line, the high voltage file is copied to a prescribed directory and a proper symbolic link is also set at the same time

the fitting is done. When you find failure of the fitting, unfortunately, you should try fitting for that channel again with another set of parameters. For the time being several values of parameters are set in the program, so you may have to change the source code of the program. On such an occasion the next section may somewhat help you.

## 1.2 Calibration

The program is designed to create calibration files by fitting the MoNA cosmic ray spectra when you execute the following shell script:

```
#!/bin/sh
,/fit_monaspectra "EOF
monacos.inp
monacosdata/MoNArawQDCcosmics.asc
16
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
test.tdr
EOF
exit
cp THRE'cat datetmp.txt'.tcl SLOOFF'cat datetmp.txt'.tcl \
~mona/config/
In -sf "mona/config/THRE'cat datetmp.txt'.tcl \
"mona/spectcl-tcl/config/QDCfittedTHRE.tcl
In -sf ~mona/config/SLOOFF'cat datetmp.txt'.tcl \
~mona/config/QDCfittedSLOOFF.tcl
exit
```

This shell script is placed in the same directory as "calib.sh". One can copy it and modify it as you like.

The inputs are basically the same as the case of the high voltage files, except for the first input "3", which is a flag to output calibration files. The program also create the file "datetmp.txt" as well. Two additional files are created,

- THRE(date and time).tcl: A file for the threshold setting of QDC.
- SLOOFF(date and time).tcl: A file for the slope and offset for the QDC calibration.

Both can be used as tcl scripts. The way to check the fitting is the same as the case of a high voltage file. By removing "exit" next to the EOF, the files are copied to a right place and symbolic lines are also set properly.

## 2 Changing The Program

This section describes what is written in the source code of the fitting program. Several parts of the program are still written to deal with some specific cases, so there must be some occasions that users need to change a part of the source code of the program. This section would help users to figure out where and how they should change. The following files are used as fortran source files:

```
fit-monaspectra.f
fopenmsg.f
open_prmfile.f
fcn.f
read-data.f
calculation.f
min_funct.f
peak-finding.f
ped-finding.f
result.f
newhv.f
```

The program uses a function minimization and error analysis code MINUIT, which is provided by cernlib. The program MINUIT searches values of parameters to minimize a value calculated in subroutine fcn in fcn.f (in the present case  $\chi^2$ ). The uncertainties of the parameters are also calculated at the same time. The flow of program is roughly as follows:

- Read the mode of calculation
- Read fitting parameters from a file
- Open a file of spectra
- 288 loops of
  - Read HV channel, previous HV value, the name of the channel
  - Read values of a spectrum
  - Search pedestal position (call ped\_finding(HVchannel))
  - Judge if the present channel is skipped or not
  - Search cosmic ray peak position (call peak\_finding(vstrt(2)))
  - Change several initial values of fitting parameters
  - Run procedures of MINUIT to obtain the optimal parameter values

For HV output

- Calculate new HV value for each channel and write it into a file For Calibration output
- Calculate QDC threshold and QDC calibration parameters for each channel and write it into a file For several important processes additional description is given below. Note that you need to type  
> make

in order to compile the program and make the change effective. Read fitting parameters from a file A subroutine open-prmfile is called to read fitting parameters. The subroutine is written in open\_prmfile.f. For a typical format of the parameter file, see for example monacos.inp. In this file, only the part

```
1. 'Yield' 5.D3 1.D3 1.D3 3.D4
```

2. 'Position' 800. 1. 0. 0.
3. 'Sigma' 100. 0.1 0. 500.
4. 'bg-a0' 50. 10. 0. 0.
5. 'bg-a1' 200. 1. 0. 0.
6. 'bg-a2' 0. 0.0 0. 1.

is effective. The other parts are kept for the case of running MINUIT in other modes (see the MINUIT manual for details). The first column is the number of the parameters needed in MINUIT, the second column is the name of the parameters but not so meaningful, the third column is the initial value of the parameter, the fourth column is the error of the parameter, the fifth and sixth column are the lower and upper limit of the parameter if one wants to give, respectively. If the error is set 0., the parameter is treated as a constant value, and if both upper and lower limits are set 0., it means the parameter does not have any limit. These 6 parameters are used in the fitting function, which is written in min-funct.f, as par(1)-par(6). It is written like this.

```
yield = PAR(1)
pos = PAR(2)
width = PAR(3)
a0 = Par(4)
a1 = Par(5)
a2 = Par(6)
sig = yield / sqrt(2.*PI) /width
&      * exp(-1.*(x-pos)**2/2/width**2)
bak = a0 * exp(-x/a1) + a2
all = sig + bak
```

As one can see, the parameter all is the total of the gaussian part sig and an exponential background bak, and this all is used as a value of the fitting function. The user can change the functional form here when it is needed.

## 2.1 Skipping a Channel

The channels to skip or to be analyzed are given when you run the program. But there may be a case that you feel more comfortable to give the channels in the source code, for example when they are too many to list, or when you always skip certain channels. In order to specify channels explicitly in the source code, you need to modify this part of the program.

```
if (skipmode .eq. 1) then
  skipping = .false.

do iii = 1,nskipchannel
  if (hvchannel.eq.skipchannel(iii))
  & skipping = .true.
enddo
```

```

else if (skipmode .eq. 2) then
  skipping = .true.
  do iii = 1, nskipchannel

  if (hvchannel.eq.skipchannel(iii))
    & skipping = .false.
  enddo

else
  skipping = .false.
end if

if (skipping) goto 1405

c      if (nhist .ne. 49) goto 1405
c      if ((nhist .ne. 48) .and. (nhist .ne. 49)) goto 1405
c      if (nhist .gt. 1) goto 1405
c      if ((nhist .ge. 225) .and. (nhist.le.240)) goto 1405
c      if (nhist .eq. 155) goto 1405

```

As you can see easily, what you need to do is just to goto 1405 when you want to skip the channel. If you add a statement like the ones commented out, you

can skip the channels. Note that the number of channels given when you run the program is the number you will give at that time, not including the channels skipped explicitly in the source code.

## 2.2 Cosmic Ray Peak Position

A preliminary peak search is carried out in the subroutine `peak-finding.f`. In the subroutine the data are summed up for 300 channels in order to remove the effect of statistical uncertainties. Until now this subroutine works well, but when it fails changing the re-binning size may fix it.

## 2.3 CHanging Initial Values

The fitting process is pretty sensitive to the initial values of the fitting parameters. The MINUIT gives us a wrong fitting result easily when the initial values are too far from the right values. In order to decrease this risk, the initial values, limits, and the region of the parameters are changed by using the result of the peak search and the sum of the spectrum. Note that the author added statements again and again when the fitting did not work well, so this part is rather "patched". For the moment this set of initial values works fine, but the author is not sure that it is always good. When users meet continuous failure of the fitting, they should consider to change this part. The array `vstr` is the initial values of the fitting parameters, `vlwim` and `vuplm` are the lower and upper limits of the parameters, `ixlowlim` and `ixhighlim` represent the lower and upper limit of the region of the data used for fitting.

## 2.4 Minuit Procedures

Here describes the processes of the MINUIT used in the program briefly. For more details, see the MINUIT manual.

- CALL `mnparm(nprm(iii), pnam(iii), vstrt(iii), ...` This subroutine gives a set of parameter name, value, error, lower and upper limits, to MINUIT.
- CALL `MNSETI(title)` This subroutine gives the name of the title to MINUIT. This does not affect the result of the fitting.
- CALL `MNEXCM(FCN,'FIX', ARGLIS ,1,IERFLG,0)`  
This subroutine gives the command `fix` to MINUIT. Arguments of the command are given by the array `ARGLIS`, and the next 1 means only the first component of `ARGLIS` is valid. In this case `arglis(1)` is set to 2., so this subroutine make the 2nd parameter fixed, in the following processes until the command release is given.
- CALL `MNEXCM(FCN,'MIGRAD', ARGLIS ,0,IERFLG,0)`  
This subroutine gives the command `migrad` to MINUIT. The command `migrad` searches the optimal values and errors symmetrical for upper and lower sides. In this case, this subroutine gives the optimal value for parameters 1,3,4,5 and errors with the parameter 2 fixed.
- CALL `MNEXCM(FCN,'RELEASE', ARGLIS ,1,IERFLG,0)` This subroutine gives the command `release` to MINUIT. Here the parameter 2 is set unfixed.
- CALL `MNEXCM(FCN,'FIX', ARGLIS ,2,IERFLG,0)`
- CALL `MNEXCM(FCN,'MIGRAD', ARGLIS ,0,IERFLG,0)` This time, the parameters 4 and 5 are fixed, i.e., the background is fixed, and for the parameters 1, 2, and 3 the optimal values and errors are calculated.
- CALL `MNEXCM(FCN,'RELEASE', ARGLIS ,2,IERFLG,0)`
- CALL `MNEXCM(FCN,'MIGRAD', ARGLIS ,0,IERFLG,0)`
- CALL `MNEXCM(FCN,'MINOS', ARGLIS ,0,IERFLG,0)`  
This time, the fitting is performed for the 5 free parameters. The command `minos` is given. This command performs a more precise fitting with unsymmetrical errors for upper and lower side of the parameter.
- CALL `MNPOUT(iii,CHNAM(iii),VAL(iii),....`
- CALL `MNERRS(iii,EPLUS(iii).EMINUS(iii),....`

These subroutine provides the present value of the parameters and their errors.

## 2.5 Calculate new HV values

New HV values are calculated in the function `newhv` in `newhv.f`. Now the new HV value is calculated so that the difference between the cosmic ray peak and pedestal is 800 channel in the raw spectrum. Users can change this value by modifying this function.

## 2.6 Calculate QDC calibration parameters

The QDC threshold and QDC calibration parameters are calculated in `fit-monaspectra.f`,

```
threpos(iii) = pedpos(iii)/16+2
slope(iii) = 22./(peakpos(iii) - pedpos(iii))
offset(iii) = pedpos(iii)
```

Users can change the methods of calculation by modifying this part.

## References

- [1] R. Fox, *SpecTcl - User's Guide* October 28, 2003. NSCL.  
[http://docs.nsl.msui.edu/daq/spectcl/users\\_guide.htm](http://docs.nsl.msui.edu/daq/spectcl/users_guide.htm)
- [2] A. Ratkiewicz, W.A. Peters, *Time Sorting Pseudos* 2005. NSCL.  
[/projects/proj1/mona/reports](#)
- [3] J. Miller, M. Strongman, L. Elliott, D.B. Hecksel, M.M. Kleber, P.J. Voss, T. Pike, R. Pepin, A. Ratkiewicz, W.A. Peters, *MoNA Calibration* 2005. NSCL.  
[/projects/proj1/mona/reports](#)
- [4] A. Ratkiewicz, W.A. Peters, *MoNA SpecTcl Guide* 2006. NSCL.  
[/projects/proj1/mona/reports](#)
- [5] W.A. Peters, *Tandem SpecTcl Guide* 2006. NSCL. [/projects/proj1/mona/reports](#)
- [6] K. Yoneda, W.A. Peters, *MoNA Fitting Codes* 2004. [/projects/proj1/mona/reports](#)