Instructions for the muon simulation program

Tom Lancaster
t.lancaster1@physics.ox.ac.uk

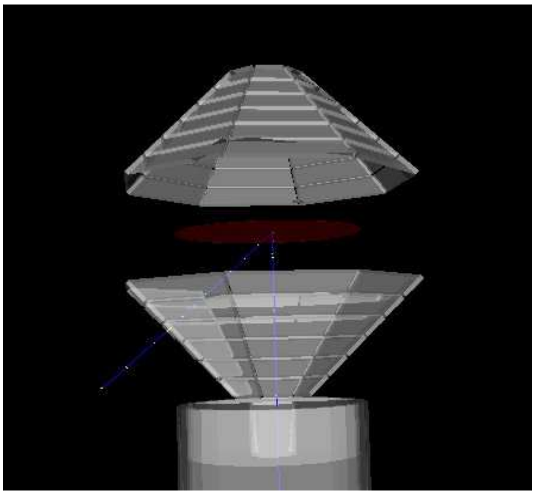

Figure 1: A sample geometry created by the simulation programs.

1 Introduction

These are the instructions for the μ^+SR spectrometer simulation package. The program is based on the GEANT4 C++ libraries.

1.1 Where are we going?

The package includes everything required to implement a simple geometry, generate the initial conditions, run the simulations and record the results.

1.2 What's included?

The files split into two classes. The source code and the run-time files. Both are required to use the programs.

- The source code (musr.tar.gz). This includes a make file for easy compilation

The run-time files are for use with the executable you'll create from the sources.

- Three Perl programs. readconfig.pl is used to run the programs and process.pl is used for some data processing. The file visconfig.pl is useful for visualising the geometry before you start a simulation.
- A configuration file (config.txt) which may be edited by the user.
- An example magnetic field map (emu20.dat) and TURTLE input-file (DEVA+10s20.dat).
- A simple setup program (setup.pl) which copies the runtime files into a sensible directory structure.
- A sample visualisation file vis.mac for use with visconfig.pl

These are all included in the tar-ball runtime.tar.gz

2 Installation

2.1 System requirements

The CLHEP and GEANT4 libraries are required to run the simulations (the software requires at least GEANT4.7.0.). Supported platforms and detailed installation procedures are given at: http://geant4.web.cern.ch/geant4/support/index.shtml. Be warned that a little fiddling will probably be required to get it working. I've now had it running on various flavours of Linux and on my Mac.

Once GEANT4 has been successfully installed (you should check this by running one of the examples) you may compile the simulation code. Remember to set your environment variables before attempting a compilation (usually done by running env.sh or something similar).

2.2 Installation

Unpack the source tarball into a suitable directory

```
tar -xvzf musr.tar.gz
```

Change to the new directory and run the makefile

```
cd musr
make
```

This should produce an executable called “musr” in the directory you selected during the GEANT4 installation.

Copy the runtime tar-ball into your GEANT4 executable directory and unpack it.

```
tar -xvzf runtime.tar.gz
```

Since several files will be used in the course of the simulation it is expected that you will keep field-maps, TURTLE files and configuration files in other directories. For example, my running directory contains the musr executable along with readconfig.pl and process.pl. It also contains the folders:

- config/ (containing config.txt)
- fieldprofiles/ (containing emu20.dat)
- turtle/ (containing DEVA+10s20.dat)
- macfiles/ (where batch files will be stored)

These choices are declared in the configuration file.

For ease of use the file setup.pl will copy the files into a structure like that suggested above.

Simply run the file as follows

```
perl setup.pl
```

3 The simulation

The simulation inputs the configuration file config.txt, an example of which is shown below:

```
Number of radial pixels set to <<8
Number of length pixels set to <<8
Near Radius set to (mm)        <<150
Far Radius set to (mm)         <<10
Detector separation set to (mm)<<100
Array length set to (mm)       <<150
Detector Thickness set to (mm) <<1
TargetMaterial                 <<Air
TargetPosition (mm)            <<0
TargetThickness (mm)           <<1
Muon-target distance (cm)      <<99
Positron test mode (0/1)       <<1
Beam input from file (0/1)     <<0
Beamline input file            <<DEVA+10s20.DAT
BeamFWHM (mm)                  <<40
Muon momentum (GeV/c)          <<0.029
Default number of muons        <<100000
Beampipe clearance (cm)        <<30
Collimator radius  (mm)        <<25
Field from file (0/1)          <<1
Name of fieldmap file (if used)<<emu20.dat
StartField (T)                 <<0
Endfield (T)                   <<5
Number of field points         <<6
Folder where results will go   <<results
Config file folder             <<config
Macfile folder                 <<macfiles
TURTLE folder                  <<turtle
Fieldmaps folder               <<fieldprofiles
```

These instructions are discussed below.

3.1 Geometry

At present, the geometry includes a sample muon beampipe with lead collimator, a target (the material of which may be varied) a detector array. A shield surrounds the apparatus and annihilates particles that impinge on it. This reduced runtime significantly.

3.1.1 Dimensions

The following lines of the configuration file refer to the geometry of the detectors and target.

```
Number of radial pixels set to <<8
Number of length pixels set to <<8
Near Radius set to (mm)        <<150
Far Radius set to (mm)         <<50
Detector separation set to (mm)<<100
Array length set to (mm)       <<150
TargetMaterial                 <<ArgonGas
TargetPosition (mm)            <<0
Beampipe clearance (cm)        <<30
Collimator radius  (mm)        <<25
```

The geometry is symmetrical with identical forward and backward detector arrays with cylindrical symmetry. The geometry is shown in figure 2. The arrays may be segmented in equal divisions along their length (length pixels) and about the cylindrical axis (radial pixels). The other required parameters are defined in the figure.

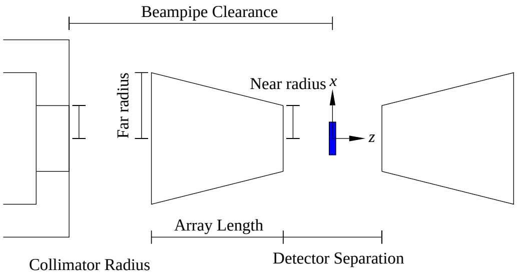

Figure 2: Geometry for the simulations.

3.1.2 Materials

By default the space in which the experiment takes place (the world) is made of air. The muons move down an aluminium tube filled with argon (at a very low pressure, so effectively a vacuum) and pass through a gap in a lead collimator. The detector elements are made of plastic scintillator.

At present the target material may be set to ArgonGas (again, at a very low pressure, so effectively a vacuum), Scintillator, Silver, Aluminium, Air or Lead. Note the capitalisation - the input is case sensitive.

3.2 Beam input

The incoming muon beam may be implemented in one of two ways. Either through a TURTLE input file, or through a parallel beam with a Gaussian intensity distribution.

The initial distance of the particle from the magnet centre is chosen with the

```
Muon-centre distance
```

command. It requires a positive number.

3.2.1 TURTLE

The expected input is

```
x xang y yang p w Pz
```

where

- x = horizontal spatial coordinate of particle at sample position in zero field (cm)
- xang = horizontal angular coordinate of particle (mrad)
- y = vertical spatial coordinate of particle (cm)
- yang= vertical angular coordinate of particle (mrad)
- p = momentum of particle (GeV/c)
- w = weight of particle (default = 1). Not used at present(!)
- Pz= longitudinal polarisation of particle (default = 0)

These parameters must be specified for each incoming muon. The simulation will run a single muon for each complete line.

In this configuration the commands:

```
BeamFWHM (mm)
Muon momentum (GeV/c)
Default number of muons
```

are ignored.

3.2.2 Parallel beam

In this configuration you must set the following commands:

```
BeamFWHM (mm)
Muon momentum (GeV/c)
Default number of muons
```

This allows the user to run these parameters for as many muons as desired. Initial positions in the x-y plane are randomly chosen to give a Gaussian intensity distribution.

3.2.3 Positron test mode

The line

```
Positron test mode (0/1)
```

when switched to 1 allows the user to start with an ensemble of stopped muons at the magnet centre. In this configuration you must specify the FWHM. This is useful for testing detector arrays as a function of field.

In this case, the commands

```
Muon-centre distance (cm)
Beam input from file (0/1)
Beamline input file
Muon momentum (GeV/c)
```

are ignored.

3.3 Magnetic field

You may select to use a field specified in a text file using

```
Field from file (0/1)
```

If set to 0, the field will be uniform in the z direction.

See the included field profile file for an example of the expected format of the input.

For multiple runs at different fields the following variables must be set

```
StartField (T)                 <<0
Endfield (T)                   <<5
Number of field points         <<6
```

3.4 File structure

Several directories must be declared

- Folder where results will go: This will be created by the program
- Config file folder: Where config.txt is stored
- Macfile folder: Where the .mac files will be stored
- TURTLE folder: Where the turtle input files are stored
- Fieldmaps folder: Where the fieldmaps are stored

4 Running the simulation

Once the configuration file has been edited the simulation is ready to run. It is strongly recommended to run the simulations from the same directory where the "musr" executable is stored and the subdirectories have been put.

4.1 Normal use

In the intended mode of use, the simulation is run using the perl runtime files. These files take the input from config.txt and create batch (.mac) files in your chosen directory. The program is then automatically run from these batch files. The perl programs then handle the basic data processing.

4.1.1 Preliminary visualisation (optional, but useful!)

If you wish to have a look at the geometry before running the simulation run the command

```
perl visconfig.pl
```

You may then go back and edit the configuration file if required. This will not run any events.

4.1.2 Running the full simulation

The simulation proper is run using the command

```
perl readconfig.pl
```

This will implement the same geometry as you saw using visconfig.pl, but will now run the events set up in the configuration file.

4.2 Advanced use

The main program may also be run from the command line thus:

```
./musr
```

This will load the program with some default settings (the file config.txt will be ignored.). Typing

```
help
```

will give you some simple menus listing what can be done from the terminal. To use the commands specific to this program, select option 10 (called "musr").

The manipulation of the program from the command line is, at present, beyong the scope of this document. There's nothing you can do at the command line that you can't do with readconfig.pl. If, however, you are planning to use the command line, please note that after making any changes in the geometry, you must issue the command

```
musr/det/update
```

for any of the changes to come into effect.

5 Understanding the Results

The simulation creates raw results files in your executable folder. These are

- muposition.dat: Listing the field at the sample position (in T) along with the coordinates for each muon (x, y, z) where it impinged on the target.

- mupol.dat: Listing the field, coordinates (x, y, z) on impact with the target and the polarization (P_x, P_y, P_z) at that position.
- iforward.dat and ibackward.dat. Files for positron events at the forward and backward detector arrays. The format is field, coordinates (x, y, z), time, detector segment, whether this is a multiple event (0/1) (i.e. if this positron was detected in another detector you get a 1)
- cforward.dat, cbackward.dat, fmult.dat bmult.dat: some intermediate files used by the program.

The final results are copied by readconfig.pl into the folder that was specified in config.pl. The directory will contain the following:

- nfXt.dat : numbers detected in forward array at X tesla for each pixel
- nbXt.dat : numbers detected in backward array at X tesla for each pixel
- nbXt_tot.dat : numbers detected in backward array summed over a ring (ie for fixed z).

The detectors are numbered xy, where x is the number of the ring numbered (closest to)-(furthest from sample) and y is the detector number in a particular ring.

- multbXt.dat : number of multiple hits in backward array
- asymmetryXt.dat : asymmetry for each detector pair
- fomXt : figure of merit of each pixel

Also included are files like bmultlistX.dat which list the detector element in which each multiple hit occurs. This allows positron tracking through the array.

The file rad.dat gives information on the average muon beam properties at the sample position. The format is

- Field (T)
- average radius (cm)
- error in average radius (cm)
- average x-coordinate (cm)
- average y-coordinate (cm)
- RMS x-coordinate (cm)
- RMS $y$coordinate (cm)

6 The end

The programs are available at http://users.ox.ac.uk/∼pemb0809/simulations.html New versions will be made available regularly. Please check you have the most recent version before submitting an error report.

That said, I'm grateful for feedback. Please report corrections to me at the email address given in the title.