McDark User's Manual

Timur C. Sahin

M.I.T., 77 Massachusetts Avenue, 26-456

Cambridge, MA 02142-4307

Abstract

This document includes instructions on the installation and execution of the McDark Geant4 simulation for the DMTPC.

1 Installing McDark

1.1 Using the LNS Installation of Geant4 and ROOT

It is possible to use the Geant4 installation on the LNS network. The G4INSTALL path on our network is `/net/cockroach/data01/tcsahin/geant4.8.3`. In order to set up all your environment variables to begin using Geant4, you can execute the bash shell script `/net/cockroach/data01/tcsahin/env_setup_483.sh`. To do this in bash:

```
$ source /net/cockroach/data01/tcsahin/env_setup_483.sh
```

When you see a $, this indicates a command to enter at a bash terminal. If commands differ between bash and tsch, I will try to note it. It is highly recommended you copy this script to your home directory so you can modify as needed. You can then simply copy the binary file McDark into a directory of your choice. The most recent working binary is stored in `/net/cockroach/data01/tcsahin/g4work/tmp`. If you do not wish to modify the source code in any way, you can skip ahead to the Using McDark section. You should go ahead and grab the source files from the CVS repository (see below for how to do this) so you can access the class files needed to read the McDark ouput (see §3).

If you do want to modify the McDark code for you own personal uses, you should specify a working directory for your program by setting the G4WORKDIR environment variable. After running this script, G4WORKDIR points to

`/net/cockroach/data01/tcsahin/geant4.8.3/g4work`. When compiling a Geant4 project, your executables will be produced in `$G4WORKDIR/bin/Linux-g++/`.

1.2 Installing Geant4 on Your Own Machine

It is highly recommended you use the LNS installations of CLHEP, Geant4.8.3, and ROOT. However, if you have a need to install Geant4 on your own machine, the necessary files are located in `/net/cockroach/data01/tcsahin/tars/`. To run McDark, you must install CLHEP, Geant4.8.3, and ROOT. Note that installing CLHEP is a prerequisite for installing Geant4.8.3, and that installing ROOT is a prerequisite for compiling McDark.

Installation for Geant4 is fairly straightforward and well documented: You can follow the instructions here. This document can also direct you to the appropriate documentaton for installing CLHEP. In order to install ROOT, follow the instructions here, or in the readme found in the tar file. Both Geant4.8.3 and Geant4.9.0 installations are available. Our project is designed for Geant4.8.3 and is not currently 100% compatible with later versions of Geant4. We will make the move to later versions of Geant if functionality requirements call for it.

1.3 Acquiring and Compiling the McDark project files

To check out files from the CVS repository:
Make sure environment variables are set up for CVS:

```
$ export CVSROOT=:ext:termite.lns.mit.edu:/export/raid2/cvsroot
export CVS_RSH=ssh
```

Then access the repository

```
$ cvs checkout projects/DarkMatter/McDark
```

This will produce a copy of the current version on your machine. You can then feel free to modify the code as necessary **within your directory**. You can commit changes back to the directory, but these changes will affect everyone who uses the code. Please only do this if make an update that needs to be available to everyone else using McDark. To apply the changes back to the repository:

```
$ cvs commit -m comment
```

If you want to add a wholly new file:

```
$ cvs add <filename>
$ cvs commit -m "comment" <filename>
```

To delete a file from the repository, you must first remove the local copy, then use

the cvs remove command:
Remove local copy

```
$ rm <filename>
$ cvs remove <filename>
```

Then commit changes

```
$ cvs commit
```

If you do not want to make changes, you can simply `rm -r` your personal copy of the directory. In order to compile McDark, you simply need to navigate to the McDark/ directory and and gmake. If you are updating a version of McDark, make sure to gmake clean first.

```
$ gmake clean
```

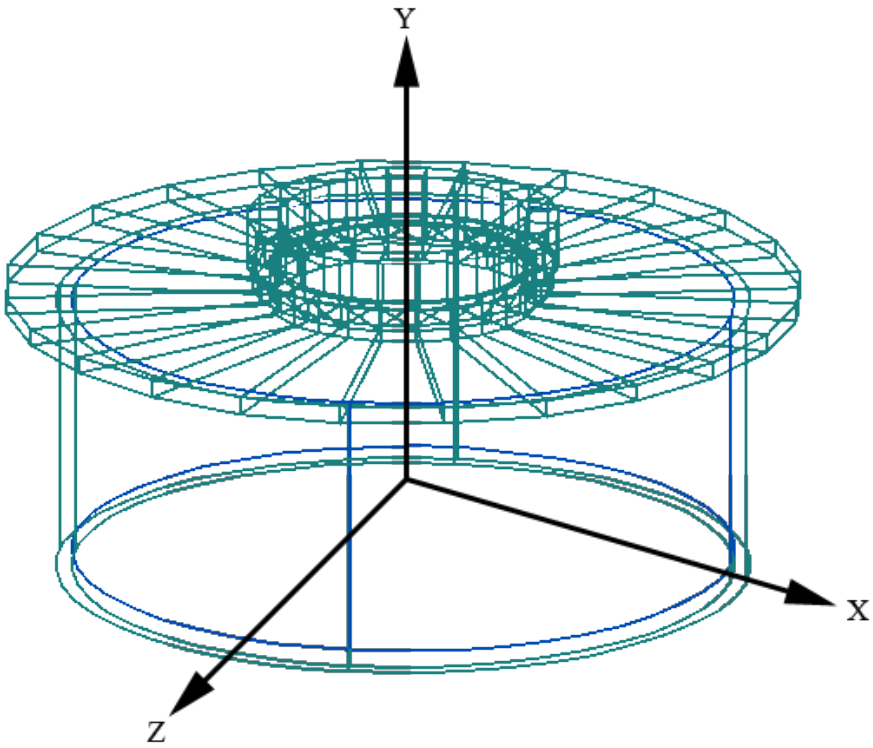
Then run gmake

```
$ gmake
```

2 Running McDark

2.1 Detector Geometry

Currently, the detector geometry in McDark is the 20-cm radius vessel that Dr. Dujmic put together. The coordinates system internally is illustrated below:



With the CCD camera sitting at the window on top parallel to the X-Z plane. Currently, the camera must be on the X-Z plane, but can be displaced on that plane. Information on how to setup and move cameras can be found in the camera setup file. The origin of the system is located in the center of the main cylinder of the vaccuum vessel.

2.2 Environment variables

There are several environment variables that you may need to know about when using McDark. I have briefly described them here:

Variable	*Description*
$G4INSTALL	Specifies directory of Geant4 installation Default: `/net/cockroach/data01/tcsahin/geant4.8.3/`
$G4WORKDIR	Specifies directory for creation of both output and temporary files Default: `/net/cockroach/data01/tcsahin/g4work/`
$SRIMTABLES	Specifies directory for SRIM tables used to calculate stopping powers and quenching factors. Default: `/net/cockroach/data01/tcsahin/srimtables/`
$ROOTOUT	Specifies the directory where ROOT output is created. The ROOT output can grow quite large (850MB for 10k events) so having an output to a separate machine is sometimes necessary. Default: `/net/cockroach/data01/tcsahin/g4work/`

2.3 Geant4 UI

Geant4 has a text-based user interface which is capable of reading and interpreting commands dynamically or with the use of a macro file. To execute McDark, you can: Place McDark in your `$PATH` then run it by entering `McDark`:

```
$ export PATH=$PATH:$G4WORKDIR/bin/Linux-g++/
$ McDark
```

Or you can execute it by navigating to `$G4WORKDIR/bin/Linux-g++` and then entering the following into the command line:

```
$ ./McDark
```

Geant4 will then load up the relevant physics packages (which may take a minute or so). If you have a file named `vis.mac` in your current directory, Geant4 will start executing commands from vis.mac line by line. If you do not have a macro file vis.mac, it will go straight to the Geant4 terminal. You will see a prompt that looks like this:

```
Idle>
```

Geant4 commands resemble Linux commands. For a list of available commands, enter `ls` at the terminal. The output should look like this:

```
Idle> ls
Command directory path :  /
  Sub-directories :
    /control/ UI control commands.
    /units/ Available units.
```

```
   /geometry/ Geometry control commands.
   /tracking/ TrackingManager and SteppingManager control commands.
   /event/ EventManager control commands.
   /run/ Run control commands.
   /random/ Random number status control commands.
   /particle/ Particle control commands.
   /process/ Process Table control commands.
   /material/ Commands for Materials
   /gun/ Particle Gun control commands.
   /generator/ Generator control commands
   /camera/ CCD camera setup commands
   /vis/ Visualization commands.
   /grdm/ Controls for the Radioactive Decay Module.
Commands :
```

Geant4 commands are organized in a directory format. You can also use `ls` to see a description of all the available commands within a directory. For example:

```
Idle> ls gun/
Command directory path :  /gun/
Guidance :
Particle Gun control commands.
  Sub-directories :
  Commands :
    List * List available particles.
    particle * Set particle to be generated.
    direction * Set momentum direction.
    energy * Set kinetic energy.
    position * Set starting position of the particle.
    time * Set initial time of the particle.
    polarization * Set polarization.
    number * Set number of particles to be generated.
    ion * Set properties of ion to be generated.
```

For more information about a specific command, you can use the help command:

```
Idle> help <command>
```

For example:

```
Idle> help /gun/energy
Command /gun/energy
Guidance :
Set kinetic energy.
Parameter :  Energy
  Parameter type :  d
  Omittable :  True
  Default value :  taken from the current value
Parameter :  Unit
  Parameter type :  s
  Omittable :  True
  Default value :  GeV
  Candidates :  eV keV MeV GeV TeV PeV J electronvolt kiloelectronvolt
megaelectronvolt gigaelectronvolt teraelectronvolt petaelectronvolt joule
```

2.4 Running an Event

To run an event, first, specify the kind of particle you wish to track. You can do this by using the `/gun/particle` command. For electrons, use

```
Idle> /gun/particle e-
```

A complete set of valid particles can be accessed in the Geant4 User's Manual, or by running the command

```
Idle> /gun/List
```

Ions are handled differently than other particles. In order to set up an ion, first specify that you wish to shoot ions:

```
Idle> /gun/particle ion
```

Then specify the exact ion using the `/gun/ion` command:

```
Idle> /gun/ion Z A Q
```

Where Z is the atomic number, A is the atomic mass, and Q is the charge (in units of e). For example, to select an alpha particle, you would use:

```
Idle> /gun/ion 2 4 2
```

Then you must specify the type of distribution you want in your particles. The event generator is controlled using the `/gun/` and `/generator/` commands. Currently, there are three distributions available to you: gun, isotropic, and spergel. The gun distribution fires a number of particles from the same position in the same direction. The isotropic distribution fires particles with random incoming directions. It also allows for flat spatial distribution of events and flat distribution of energy. To switch between distributions, use the `/generator/distribution` command. **Note that you must set the particle using the /gun/particle command regardless of the distribution you use.**

2.4.1 Gun Distribution

To enable the gun distribution, use the command:

```
Idle> /generator/distribution gun
```

The gun comes as a part of the Geant4 package and is controlled with the `/gun/` commands. The important commands available to you then are:

/gun/direction Set momentum direction

/gun/direction [x] [y] [z]

e.g. /gun/direction 1 0 -1

/gun/energy Set kinetic energy

The Geant4 particle gun will do a Poisson shoot around this value.

/gun/energy [value] [unit]

e.g. /gun/energy 250 keV

A complete list of units is available by using the help command.

/gun/position Set position of gun

/gun/position [x] [y] [z] [unit]

e.g. /gun/position -5 0 0 cm

A complete list of units is available by using the help command.

After you set the desired parameters, use the `/run/initialize` and `/run/beamOn` commands to run the events. For example:

```
Idle> /run/initialize
Idle> /run/beamOn 10
```

Will run 10 events. The output will be generated in `$ROOTOUT` in a file named `mcdark.root`. Information on manipulating the ROOT file can be found in §3. Below is an example where 10 alpha particles are fired from (-1 0 0) cm in the positive y direction (straight up) with an energy of 50 keV.

```
Idle> /gun/particle ion
Idle> /gun/particle 2 4 2
Idle> /gun/position -1 0 0 cm
Idle> /gun/direction 0 1 0
Idle> /gun/energy 50 keV
Idle> /run/initialize
Idle> /run/beamOn 10
```

2.4.2 Isotropic distribution

For image reconstruction and analysis purposes, it's useful to use an isotropic distribution instead of a collimated beam. For this purpose, select the isotropic distribution:

```
Idle> /generator/distribution isotropic
```

You then have a range of parameters you must set:

/generator/isotropic/minEnergy Set minimum energy of flat distribution

/generator/isotropic/minEnergy [value] [unit]

e.g. /generator/isotropic/maxEnergy .1 MeV

/generator/isotropic/maxEnergy Set maximum energy of flat distribution

/generator/isotropic/maxEnergy [value] [unit]

e.g. /gun/energy 2.0 MeV

/generator/isotropic/spatiallyDistributed

Toggle whether or not particles are generated flat spatially, as opposed to all originating from the same point.

/generator/isotropic/spatiallyDistributed [bool]

e.g. /generator/isotropic/spatiallyDistributed false

/gun/position Set position of gun

/gun/position [x] [y] [z] [unit]

e.g. /gun/position -5 0 0 cm

Note that you only need to set a definite position if you've disabled the spatial distribution with the previous command

/generator/isotropic/minXpos Set minimum x position of flat spatial distribution

/generator/isotropic/minXpos [value] [unit]

e.g. /generator/isotropic/minXpos -76.2 mm

/generator/isotropic/maxXpos Set maximum x position of flat spatial distribution

/generator/isotropic/maxXpos [value] [unit]

e.g. /generator/isotropic/maxXpos 76.2 mm

Note: there exist analogous commands minYpos, maxYpos, minZpos, maxZpos.

Below is an example to generate 1000 electrons between .5 MeV and 1 MeV with random directions, with $x \in [-50, 50]$ mm, $y \in [10, 50]$ mm, $z \in [-50, 50]$ mm:

```
Idle> /gun/particle e-
Idle> /generator/distribution isotropic
Idle> /generator/isotropic/minEnergy 0.5 MeV
Idle> /generator/isotropic/maxEnergy 1.0 MeV
Idle> /generator/isotropic/spatiallyDistributed true
Idle> /generator/isotropic/minXpos -50 mm
Idle> /generator/isotropic/maxXpos 50 mm
Idle> /generator/isotropic/minYpos 10 mm
Idle> /generator/isotropic/maxYpos 50 mm
Idle> /generator/isotropic/minZpos -50 mm
Idle> /generator/isotropic/maxZpos 50 mm
Idle> /run/initialize
Idle> /run/beamOn 1000
```

2.4.3 Spergel Distribution

The Spergel Distribution is currently in development. Feel free to experiment with it, but it may not be 100% accurate. An update will be posted onto Hypernews when the code for this is complete.

2.5 Camera Manipulation

By default, McDark is setup to take images from one CCD camera (1024 pixels squared, 4 by 4 binning),centered at (0,0,0), with a view field of 15.25 by 15.25 centimeters. To change these parameters, use the /camera/setupCamera command:

```
Idle> /camera/setupCamera [cameraNumber] [pixels_per_side]
[pixels_per_bin_side] [viewField_x] [viewField_y] [displacement_x]
[displacement_y] [units]
```

Cameras are numbered sequentially starting from 0. viewField_x and viewField_y specify the view of the camera. **Note that x and y here are relative to the camera's internal coordinate system, NOT Geant4's universal coordinate system.** Future versions of McDark will allow for specification of a camera using a vector to the camera, a normal vector to the plane of the CCD, and information about the lens (aperture, etc.). For now, cameras are restricted to the X-Z plane, in the manner described in the Detector Setup section.

Example: to set up camera #0 as a 1024 by 1024 pixel camera with 2 by 2 binning, taking an image of a 10 cm by 10 cm area, with a displacement in the (global) X direction of 15 cm and displacement in the (global) Z direction of -10 cm, you would use the command:

```
Idle> /camera/setupCamera 0 1024 2 10 10 15 10 cm
```

Note that the displacement_y is *positive* 10, as it refers to the displacement along the direction of the camera's internal coordinate system.

To add additional cameras beyond the first, use the addCamera command:

```
Idle> /camera/addCamera [number_of_cameras]
```
e.g.
```
Idle> /camera/addCamera 5
```

would add five cameras. Their parameters need to be set up individually using the `/camera/setupCamera` command.

The CCD noise can be set individually for each camera. It is represented by a Gaussian with a mean of zero. You can set the sigma of this Gaussian using the `/camera/setNoise` command:

```
Idle> /camera/setNoise [cameraNumber] [noise_sigma]
```
e.g.
```
Idle> /camera/setNoise 1 6.3
```

would set the standard deviation of this distribution of the noise on camera #1 to 6.3.

You can also toggle noise on or off entirely across all cameras:

```
Idle> /camera/toggleNoise [bool]
```
For example,
```
Idle> /camera/toggleNoise false
```

disables all CCD noise.

3 Manipulating the ROOT file

You can access the ROOT file like any other file. An example is provided in `McDark/Root/mcdarkscript.c`. You can run mcdarkscript.c in ROOT using the `.x` command:

```
root [] .x mcdarkscript.c
```

If you want to read the ROOT file as part of a rootcint session or a ROOT script, use the method outlined in the script. To read the ROOT file within a compiled project, simply include and link the McDarkEvent class and header file as you would any other class.
The McDarkEvent object has the following functions: GetX_i(), GetY_i(), and GetZ_i() all return a double with the initial x, y, and z coordinates of the track in mm. GetX_f(), GetY_f(), and GetZ_f() return the final x, y, and z coordinates in mm. GetP_x(), GetP_y(), and GetP_z() return the initial momentum. GetE() returns a double with kinetic energy in units of MeV. You can also use GetID() to get the Particle Data Group ID of an event.