LA-UR-21-28914
Approved for public release; distribution is unlimited.

Title: CpRad User's Manual

Author(s): Sjue, Sky K.
Tang, Elise
Connor, Erin Grace

Intended for: Report

Issued: 2021-09-09

CpRad User's Manual

version 0.0

Sky Sjue, Elise Tang and Erin Connor

Contents

1 Introduction

This is the reference manual for users of the **C** programming language based **p**roton **Rad**iography simulation package **CpRad**, pronounced "see p-rad." These simulations are developed and maintained by Los Alamos National Laboratory personnel. Currently there are threaded versions, versions written for MPI, as well as versions that use GPUs.

The fundamentals of our proton radiography model are well described in a review [1]. The ray tracing in CpRad also includes the incident angle of the protons based on position in the field of view due to the matching conditions.

2 Density models and symmetry

Density models are preferably read via IEEE floating point TIFF files. There are four options for the geometry. We note that any symmetry is invoked in the density model's coordinates.

2.1 Spherical symmetry

If the density geometry in the input file has the following format:
`nxv 100`
`nyv 0`
`nzv 0`,
then it is assumed to be spherically symmetric based on having zero voxels in the y and z directions. The number of x voxels `nxv` corresponds to the radial coordinate.

2.2 Cylindrical symmetry

There are two cylindrically symmetric options. If the density geometry in the input file has the following format:
`nxv 100`
`nyv 100`
`nzv 0`,
then it is assumed to be cylindrically symmetric about the y (vertical in the image) axis. If `xmin` is 0, then later it could be reflected to form a complete

cylindrical geometry. If `xmin` is negative, then independent symmetry will be allowed for the $+x$ and $-x$ half planes.

If the density geometry in the input file has the following format:

```
nxv 100
nyv 0
nzv 100,
```

then it is assumed to be cylindrically symmetric about the z (perpendicular to the image, proton beam direction) axis.

2.3 Three dimensional

If the number of voxels is non-zero for all three dimensions, then the density model will be treated as three dimensional. The sizes in each dimension are arbitrarily set by the user. A cube with volume 1 cm^3 centered at the origin would be created with the following lines:

```
nxv 1
nyv 1
nzv 1
vdx 1.0
vdy 1.0
vdz 1.0
vxmin -0.5
vymin -0.5
vzmin -0.5
```

3 Input options

All of the following options can be used in a text input file.

3.1 xmin

`xmin` is the minimum x coordinate in the image, mapped to the object plane, used within the ray trace.

3.2 xmax

`xmax` is the maximum x coordinate in the image, mapped to the object plane, used within the ray trace.

3.3 nxpix

`nxpix` is the integer number of pixels in the image spanning from $xmin$ to $xmax$

3.4 ymin

`ymin` is the minimum y coordinate in the image, mapped to the object plane, used within the ray trace.

3.5 ymax

`ymax` is the maximum y coordinate in the image, mapped to the object plane, used within the ray trace.

3.6 nypix

`nypix` is the integer number of pixels in the image spanning from $ymin$ to $ymax$.

3.7 lens

`lens` invokes model parameters for magnetic optics associated with one of the magnetic lenses available at LANSCE. Current options are `-I`, `x3` and `x7`. This input sets default values for the parameters `kx`, `ky`, `t126`, `t346`, and `phicut`. To use the `lens` input and change one of these values, its value will need to be set *after* the `lens` input line.

3.8 kx

`kx` sets the x angles within the ray trace. These are based on the matching conditions of the magnetic lens being used. The direction cosine as a function

of x at the object plane within the field of view as

$$\cos(\theta_x) = k_x \cdot x$$

3.9 ky

`ky` sets the y angles within the ray trace. These are based on the matching conditions of the magnetic lens being used. The direction cosine as a function of y at the object plane within the field of view as

$$\cos(\theta_y) = k_y \cdot y$$

3.10 ke0

`ke0` sets the initial kinetic energy (in MeV) of the protons in the ray trace.

3.11 ketune

`ketune` sets the tune energy of the magnetic lens corresponding the kinetic energy (in MeV) of a proton which will be perfectly focused, referred to as KE_0 elsewhere in this document.

3.12 dphi0

`dphi0`: The incoming protons are assumed to be spread about their matching condition angle with a Gaussian distribution with $\sigma = d\phi_0$ set to `dphi0` in radians.

3.13 phicut

`phicut` is the collimator cut angle in radians, most commonly 10 mrad or 0.01 at LANSCE. If the incoming protons have angular spread $d\phi_0$ as described above, then scatter with a Gaussian distribution characterized by an angle $\sigma = \theta$, the multiple Coulomb scattering contribution to the transmission will be

$$T_{MCS} = \frac{1 - \exp(-\theta_c^2/2(\theta^2 + d\phi_0^2))}{1 - \exp(-\theta_c^2/2d\phi_0^2)}.$$

However, this formula is subject to modification by chromatic effects.

3.14 T126

`t126` is the x portion of the chromatic blur [2, 3]. The chromatic blur in the x direction is given by

$$\delta x = T_{126}\delta\theta_x \frac{KE - KE_0}{KE_0},$$

where $\delta\theta_x$ is the x scattering angle, KE is the kinetic energy and KE_0 is `ketune`. T_{126} is of the order 800 cm in this form for the -I lens at LANSCE.

3.15 T346

`t346` is the y portion of the chromatic blur. The chromatic blur in the y direction is given by

$$\delta y = T_{346}\delta\theta_y \frac{KE - KE_0}{KE_0},$$

where $\delta\theta_y$ is the y scattering angle, KE is the kinetic energy and KE_0 is `ketune`. T_{126} is of the order 800 cm for the -I lens at LANSCE.

3.16 detblur

`detblur` sets a Gaussian σ for uniform blur applied in object plane dimensions. Defaults of 0.019 cm (190 μm) and 0.007 (70 μm) are assumed for the `-I` and `x3` options, respectively.

3.17 reflect

`reflect` will reflect a ray trace for computation of the final image with the following options:

1. `reflect 1` reflect ray trace about $x = 0$.
2. `reflect 2` reflect ray trace about $y = 0$.
3. `reflect 3` reflect ray trace about $x = 0$ and $y = 0$.

3.18 nxv

`nxv` Number of x voxels in the density model.

3.19 nyv

`nyv` Number of y voxels in the density model.

3.20 nzv

`nzv` Number of z voxels in the density model.

3.21 vdx

`vdx` is the x dimension of a voxel in the density model.

3.22 vdy

`vdy` is the y dimension of a voxel in the density model.

3.23 vdz

`vdz` is the z dimension of a voxel in the density model.

3.24 vxmin

`vxmin` is the minimum x coordinate of the density model.

3.25 vymin

`vymin` is the minimum y coordinate of the density model.

3.26 vzmin

`vzmin` is the minimum z coordinate of the density model.

3.27 xshift

`xshift` moves the density model with respect to the ray trace coordinates by the specified amount in cm.

3.28 yshift

`yshift` moves the density model with respect to the ray trace coordinates by the specified amount in cm.

3.29 zshift

`zshift` moves the density model with respect to the ray trace coordinates by the specified amount in cm. While `zshift` does not cause any transverse movement in the image, it can affect magnification and blur.

3.30 roll

`roll` specifies a rotation of the density model within the $x - y$ plane in degrees.

3.31 pitch

`pitch` specifies a rotation of the density model within the $y - z$ plane in degrees.

3.32 yaw

`yaw` specifies a rotation of the density model within the $x - z$ plane in degrees.

3.33 matname

`matname` specifies the name of a material. It is used to obtain default values for the relevant material model parameters. These can also be specified independently. The following elements are currently recognized: H, He, Li, Be, B, C, N, O, F, Na, Mg, Al, Si, P, S, Cl, Ar, K, Ca, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn, Ga, Ge, As, Kr, Zr, Nb, Y, Mo, Ag, Cd, In, Sn, Sb, Te, I, Xe, La, Hf, Ta, Re, Os, Ir, W, Pt, Au, Pb, Ce, U, and Pu. In addition to these elements, the following compound materials have entries: polyethylene, polystyrene, polycarbonate, PTFE, POM, delrin, B4C, Ti-6Al-4V. It is relatively simple to add a new material upon request. Alternatively, a user can specify the parameters to model a material via `rholnuc`, `rholrad`, `dedx` and `zovera`. These parameters must be specified between `matname` and `fileloc` entries.

Multiple densities for different materials within the same geometry can be specified by repeating `matname` and `fileloc` in the input file.

3.34 fileloc

`fileloc` gives the location of a TIFF image representing a density model. In addition to TIFF images, flat binary files written from C floating point arrays will work. (An example creating and using such a format is included in this document.) It should be specified after `matname`.

3.35 rholnuc

`rholnuc` is the nuclear attenuation length used in the simulation in units of g/cm^2. For a single material with `rholnuc`= λ, the transmission after nuclear attenuation for an areal mass of ρL is

$$T_N = \exp(-\rho L/\lambda).$$

This parameter must be specified after `matname` and before `fileloc`.

3.36 rholrad

`rholrad` is the radiation length for a material used in the simulation in units of g/cm^2. This parameter must also be specified after `matname` and before `fileloc`.

3.37 dedx

`dedx` is the stopping power of the material in units of MeV/(g/cm^2). The default values refer to a proton kinetic energy of 800 MeV. This parameter must also be specified after `matname` and before `fileloc`.

3.38 zovera

`zovera` is the effective value of Z/A, the atomic number divided by the atomic mass, either for an element averaged over its isotopes or for a compound material averaged over its constituent elements. Practically, `zovera`=0.5 is a good enough approximation, as this is only used to estimate the width of the straggling distribution, which in turn is only used to set a minimum value on

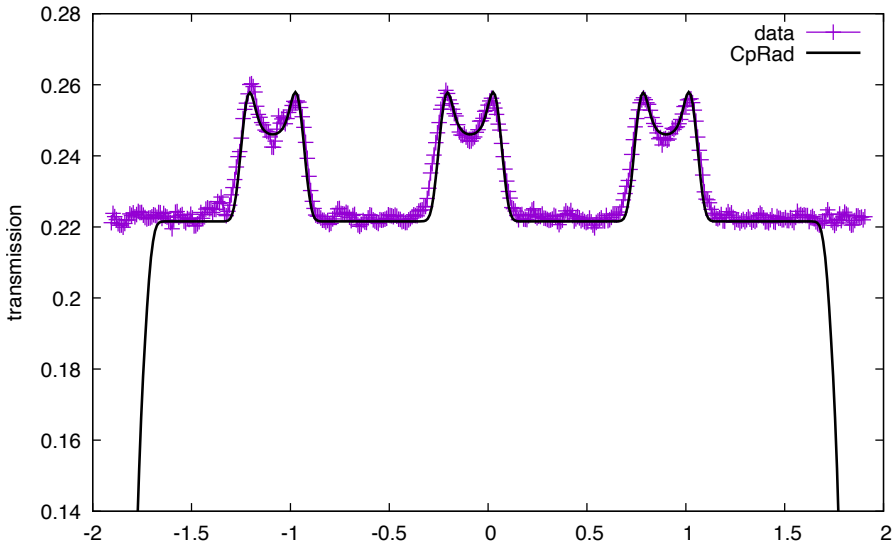
the chromatic terms involving T_{126} and T_{346} if $KE \approx KE_0$. The straggling width used to set this minimum value is given in Equation 13 of [4].

3.39 new geometry

`new geometry` indicates that a new density model geometry will be specified. The ray trace will loop over geometries until they are exhausted. A new geomery requires the following inputs to be repeated: `nxv`, `nyv`, `nzv`, `vxmin`, `vymin`, `vzmin`, `vdx`, `vdy`, `vdz`, `fileloc` and `matname`.

4 Example 1: three dimensional

Here we include a three dimensional example that is well documented [5, 6]. This example was run from the command line as follows: `./pradsim sim.inp`. A plot showing the output compared to data is shown below. Note that the simulation output transmission falls toward the edges. It is recommended to simulate a larger region of interest than desired, then trim any edge effects before making statistical comparisons. This example uses a three dimensional input geometry with two materials, with different lengths in each dimension. Source code written in C to generate the density models used in the simulation is also included.



Comparison of CpRad simulation with data based on following inputs.

4.1 Input file

The following input uses only one y pixel, which is treated as a special case in the chromatic blur calculations for convenience.

```
xmin  -1.8
xmax   1.8
ymin   0.0
ymax   0.01
nxpix  360
nypix   1
ke0    800.0
lens   -I
ketune  763.0
phicut  0.01
dphi0   0.003
vxmin  -2.0
vymin  -0.5
vzmin  -0.3
vdx     0.04
vdy     1.0
vdz     0.1
nxv     100
nyv     1
nzv     43
xshift  -0.115

matname Al
fileloc  almodel.flt

matname Ta
fileloc  tamodel.flt
```

4.2 Density model

This short C program generates the density models needed. It uses a flat, floating point binary file format, which is an alternative option to TIFF files.

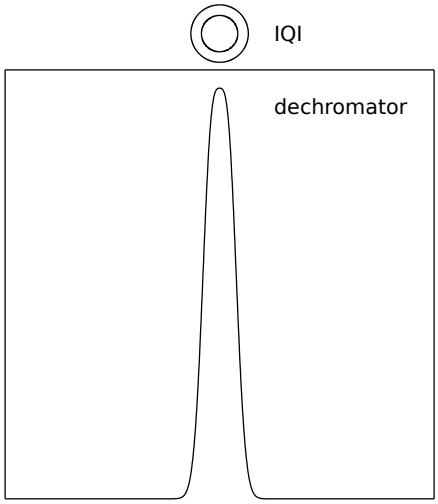
```
#include<stdio.h>
```

```
#include<stdlib.h>
#include<string.h>
#include<math.h>

int main(){
 int ix,iy,iz,i;
 int nx,ny,nz;
 int nxy;
 int nvoxels;
 float x,y,z;
 FILE *ta, *al;
 float densta, densal, dens0;
 densta = 16.69;
 densal = 2.70;
 dens0 = 0.0;
 nx = 100;
 ny = 1;
 nz = 43;
 float dx, dy, dz;
 dx = 4.0/((float)nx);
 dz = 4.3/((float)nz);
 nvoxels = nx*ny*nz;
 printf("nvoxels: %d\n", nvoxels);
        ta=fopen("tamodel.flt", "w");
        al=fopen("almodel.flt", "w");

 for(i = 0;i<nvoxels;i++){
  x = -2.0 + 4.0*(i%nx)/(1.0*nx);
  z = 0.0 + 4.3*(i/(nx*ny))/(1.0*nz);
  //check if in aluminum or not, >0.3 cm
  if(z > 0.3){
   fwrite(&densal, sizeof(float), 1, al);
   fwrite(&dens0, sizeof(float), 1, ta);
  }//now all remaining values are for Ta sheet
  else if(x > -1.02 && x < -0.98){
   //in left slit
   fwrite(&dens0, sizeof(float), 1, al);
   fwrite(&dens0, sizeof(float), 1, ta);
```

```
  }
  else if(x > -0.02 && x < 0.02){
   //in middle slit
   fwrite(&dens0, sizeof(float), 1, al);
   fwrite(&dens0, sizeof(float), 1, ta);
  }
  else if(x > 0.98 && x < 1.02){
   //in right slit
   fwrite(&dens0, sizeof(float), 1, al);
   fwrite(&dens0, sizeof(float), 1, ta);
  }
  else{
   //in Ta
   fwrite(&dens0, sizeof(float), 1, al);
   fwrite(&densta, sizeof(float), 1, ta);
  }
 }
 fclose(ta);
 fclose(al);
 return 0;
}
```

5 Example 2: multiple geometries

Here we include input to reproduce the simulations shown in [7], which illustrates the use of multiple geometries. This setup includes a cylindrical geometry followed by a spherically symmetric Image Quality Indicator (IQI) geometry by invoking `new geometry`.



The input below implements this geometry, with the dechromator being cylindrically symmetric along a vertical axis and the IQI being a sphere. The proton beam enters at the bottom and exits at the top.

5.1 Input file with multiple geometries

```
xmin    -2.0
xmax     2.0
ymin    -2.0
ymax     2.0
nxpix    800
nypix    800
lens     x3
dphi0    0.0035
phicut   0.01
ke0      800.0
ketune   769.0
vxmin   -3.0
vymin    0.0
vzmin   -16.8
vdx      0.01
vdy      0.01
```

```
vdz       0.01
nxv       600
nyv       0
nzv       1170

matname polycarbonate
fileloc dechrompc.flt

new geometry
vxmin   0.0
vymin   0.0
vzmin   0.0
vdx     0.01
vdy     0.01
vdz     0.01
nxv     100
nyv     0
nzv     0

matname Au
fileloc iqiau.flt

matname Ti-6Al-4V
fileloc iqiti.flt
```

5.2 Density models

The following C program generates the cylindrical geometry.

```
#include<stdlib.h>
#include<stdio.h>
#include<math.h>

#define XMAX  3.0
#define XMIN -3.0
#define DXPIX 0.01
#define NXPIX ((float)((float)XMAX-XMIN)/((float)DXPIX))
#define ZMIN 0.0
#define ZMAX 11.7
```

```
#define NZPIX ((float)((float)ZMAX-ZMIN)/((float)DXPIX))
#define DENSPC 1.2

int main(){
 int i,j;
 int nx,nz;

 nx=(int)NXPIX;
 nz=(int)NZPIX;

 float * lexan;

 printf("\nmake-lexan.c\n");
 printf("nxpix %d\n",nx);
 printf("nzpix %d\n",nz);

 lexan=(float *)malloc(nx*nz*sizeof(float));
 for(i=0; i<nx*nz; i++)lexan[i]=0.0;

 float x,z;

 int nsub=5;
 float ksub=0.2;
 float gfactor;
 int ix,iz;
 float sig,frac,gmax;

 for(j=0; j<nz*nsub; j++){
 for(i=0; i<nx*nsub; i++){
  iz=j/nsub;
  ix=i/nsub;
  x=XMIN+0.5*DXPIX*ksub+i*ksub*DXPIX;
  z=ZMIN+0.5*DXPIX*ksub+j*ksub*DXPIX;
  sig=0.265;
  frac=0.95;
  gmax=10.8;
  gfactor=gmax*(1.0+x*x/2.0/sig/sig*frac)
                *exp(-x*x/2.0/sig/sig);
```

```
  if(z > gfactor)lexan[ix+nx*iz]+=ksub*ksub*DENSPC;
 }}

 //convenient gnuplot format
 FILE * fout;
 fout=fopen("lex-check.txt","w");
 for(j=0; j<nz; j++){
 fprintf(fout, "\n");
 for(i=0; i<nx; i++){
  fprintf(fout,"%f  %f  %f\n",
                    XMIN+i*DXPIX+0.5*DXPIX,
                    j*DXPIX+0.5*DXPIX,
                    lexan[i+nx*j]);
 }}
 fclose(fout);

 fout=fopen("dechrompc.flt","w");
 fwrite(lexan, sizeof(float), nx*nz, fout);
 fclose(fout);

 free(lexan);

 return 0;
}
```

This C program creates the one dimensional spherical geometry.

```
#include<stdlib.h>
#include<stdio.h>
#include<math.h>

#define XMAX 1.0
#define XMIN 0.0
#define DXPIX 0.01
#define NXPIX ((float)((float)XMAX)/((float)DXPIX))
#define IRAU 0.0
#define ORAU 0.5
#define ORTI 0.79
#define RAU  19.3
```

```
#define RTI  4.42

int main(){
 int i,j;
 int kx,ky,k;
 int nx;

 nx=(int)NXPIX;

 float * gold;
 float * titanium;
 float * rsum;

 printf("\nmake-iqi.c\n");
 printf("nxpix %d\n",nx);

 gold=(float *)malloc(nx*sizeof(float));
 titanium=(float *)malloc(nx*sizeof(float));
 rsum=(float *)malloc(nx*sizeof(float));
 for(i=0; i<nx; i++)gold[i]=0.0;
 for(i=0; i<nx; i++)titanium[i]=0.0;
 for(i=0; i<nx; i++)rsum[i]=0.0;

 float x,r;

 for(i=0; i<nx; i++){
  x=XMIN+0.5*DXPIX+i*DXPIX;
  r=x;
  if(r >= IRAU && r < ORAU){
   gold[i]=RAU;
   rsum[i]+=RAU;
  }
  if(r >= ORAU && r < ORTI){
   titanium[i]=RTI;
   rsum[i]=RTI;
  }
 }
```

```
 //convenient gnuplot format again
 FILE * fout;
 fout=fopen("iqi-check.txt","w");
 for(i=0; i<nx; i++){
  fprintf(fout,"%f  %f\n", i*DXPIX+0.5*DXPIX, rsum[i]);
 }
 fclose(fout);

 fout=fopen("iqiau.flt","w");
 fwrite(gold, sizeof(float), nx, fout);
 fclose(fout);

 fout=fopen("iqiti.flt","w");
 fwrite(titanium, sizeof(float), nx, fout);
 fclose(fout);

 free(rsum);
 free(titanium);
 free(gold);

 return 0;
}
```

References

[1] C. L. Morris, N. S. P. King, K. Kwiatkowski, F. G. Mariam, F. E. Merrill, and A. Saunders, "Charged particle radiography," *Reports on Progress in Physics*, vol. 76, p. 046301, mar 2013.

[2] K. L. Brown, "A First and Second Order Matrix Theory for the Design of Beam Transport Systems and Charged Particle Spectrometers," *Adv. Part. Phys.*, vol. 1, pp. 71–134, 1968.

[3] C. T. Mottershead and J. D. Zumbro, "Magnetic optics for proton radiography," in *Proceedings of the 1997 Particle Accelerator Conference (Cat. No.97CH36167)*, vol. 2, pp. 1397–1399 vol.2, May 1997.

[4] S. Sjue, R. George, and D. Mathews, "Analytic saddlepoint approximation for ionization energy loss distributions," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 407, pp. 270 – 275, 2017.

[5] S. K. L. Sjue, F. G. Mariam, F. E. Merrill, C. L. Morris, and A. Saunders, "High order magnetic optics for high dynamic range proton radiography at a kinetic energy of 800 MeV," *Review of Scientific Instruments*, vol. 87, no. 1, p. 015110, 2016.

[6] A. E. Schach von Wittenau, M. Aufderheide, and G. Henderson, "Predicting image blur in proton radiography: Comparisons between measurements and Monte Carlo simulations," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 652, no. 1, pp. 901 – 904, 2011. Symposium on Radiation Measurements and Applications (SORMA) XII 2010.

[7] S. K. Sjue, M. C. Anderson, and Z. Tang, "Image quality indicator and dechromator setup for static LANSCE proton radiography experiment," *LA-UR-20-26337*.