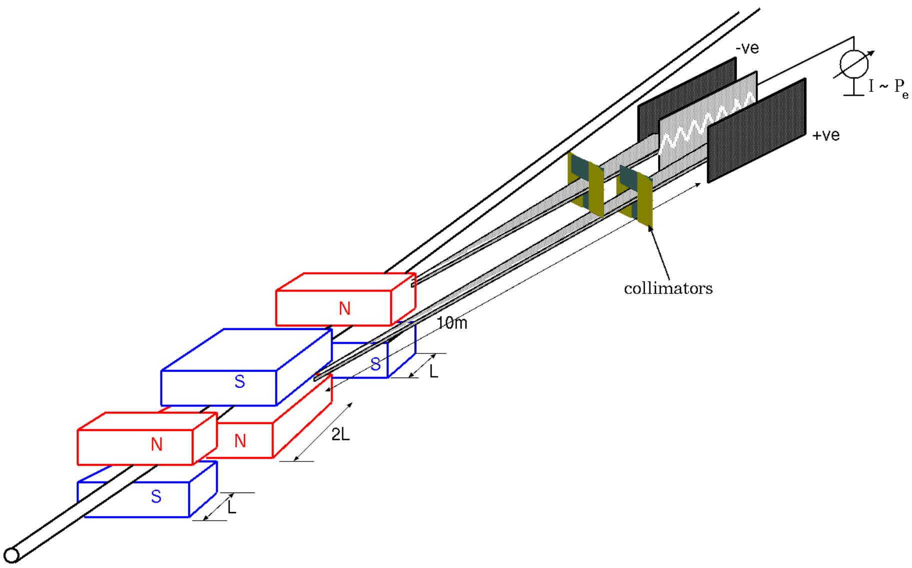
11-25-2012

A novel spin-light polarimeter for the Electron Ion Collider

Prajwal Mohanmurthy

A Novel Spin-Light Polarimeter
for the
Electron Ion Collider

PRAJWAL MOHANMURTHY



Honors Undergraduate Thesis

Nov 25, 2012

1. Reviewer: Dr. Dipangkar Dutta

2. Reviewer: Dr. Seth Oppenheimer

3. Reviewer: Dr. Paul Reimer

Day of the defense: Dec 04, 2012

To,
Amma and *Appa*,
for their dedication and admirable way of life.

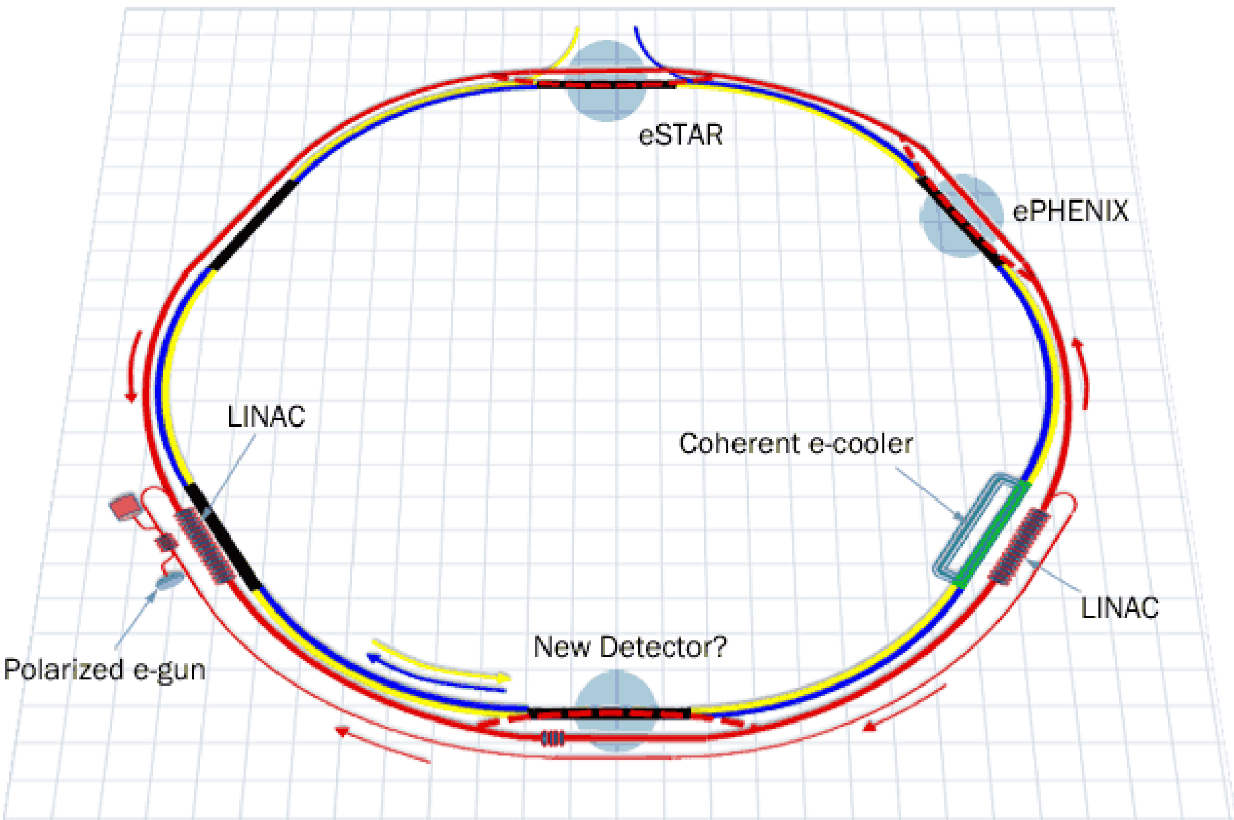
FOREWORD

With Jefferson National Accelerator Facility's (JLAB) 12GeV program in construction phase with a comprehensive set of experiments already planned for the next decade, it is time to think of newer facilities that will further push the boundaries and continue the mission of a premier nuclear physics laboratory to explore the frontiers of fundamental symmetries and nature of nuclear matter. Building of an Electron Ion Collider (EIC) seems to be a natural future step. JLAB is a fixed target laboratory, but at the EIC, the target will also be accelerated thereby providing access to precision physics of quarks and gluons at much higher energies (than 12GeV). JLAB mainly consists of the Continuous Electron Beam Accelerator and 3 halls where the fixed target experiments are performed. Brookhaven mainly consists of the Relativistic Heavy Ion Collider with a number of main collision points on the beam line. There have been two leading proposals for the EIC, *i.e.*

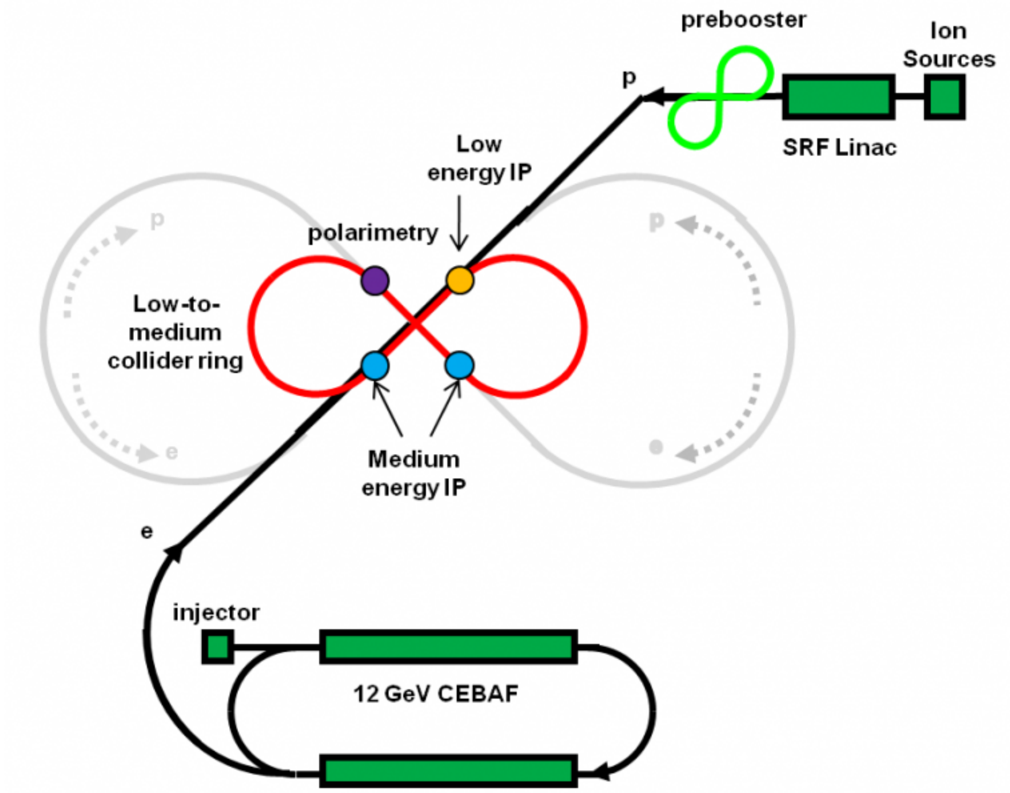
- **eRHIC** : Electron - Relativistic Electron Collider @ Brookhaven National Laboratory, *Upton, NY* [i]

[i] *eRHIC: High Energy Electron-Ion collider,* http://www.bnl.gov/cad/eRhic/, *Retrieved on: Nov 25, 2011*

[ii] *ELIC: Electron Light Ion Collider at CEBAF,* http://casa.jlab.org/research/elic/elic.shtml, *Retrieved on: Nov 25, 2011*



- **ELIC** : Electron - Light Ion Collider @ Jefferson National Accelerator Laboratory, *Newport News, VA* [ii]



Brookhaven already has an ion accelerator and the eRHIC would need addition of an electron accelerator, whereas JLAB already has the electron accelerator and the ELIC would need addition of an ion accelerator.

At JLAB, polarization of the electron beam has played a vital role a number of experiments such as the PVDIS (Parity Violating Deep Inelastic Scattering) and the QWeak (which measured the weak charge of proton). To measure the polarization of the electron beam, JLAB has commissioned Compton and Møller polarimeters which have met the precision demands of JLAB, but the Møller Polarimeter generates a large background as it uses *ee* scattering to measure the polarization. The future demands greater precision in the measurement of polarization of the beam and so at the EIC, it would be convenient to have a second non-invasive polarimeter, besides a Compton Polarimeter, for systematics comparison.

ABSTRACT

A novel precision polarimeter will go a long way in satisfying the requirements of the precision experiments being planned for a future facility such as the Electron Ion Collider. A polarimeter based on the asymmetry in the spacial distribution of the spin light component of synchrotron radiation will make for a fine addition to the existing-conventional Møller and Compton polarimeters. The spin light polarimeter consists of a set of wriggler magnet along the beam that generate synchrotron radiation. The spacial distribution of synchrotron radiation will be measured by an ionization chamber after being collimated. The up-down spacial asymmetry in the transverse plane is used to quantify the polarization of the beam. As a part of the design process, firstly, a rough calculation was drawn out to establish the validity of such an idea. Secondly, the fringe fields of the wriggler magnet was simulated using a 2-D magnetic field simulation toolkit called Poisson Superfish, which is maintained by Los Alamos National Laboratory. This was used to account for beam motion effects and the corresponding correlations were show to be negligible. Lastly, a full fledged GEANT-4 simulation was built to study the response time of the ionization chamber. Currently, this GEANT-4 simulation is being analyzed for variety of effects that may hinder precision polarimetry.

ACKNOWLEDGEMENTS

This work has been generously supported by JSA - Undergraduate Fellowship Program.

Additional, but substantial, travel funding has also been provided by the Shackouls Honors College, MS, USA.

Thanks are due to the Department of Physics and Astronomy at Mississippi State University, MS, USA for providing office & lab space and also the computational infrastructure required for this computational intensive work.

Most importantly, Dr. Dipangkar Dutta, my adviser has been relentlessly at work on this project, guiding and helping the effort at every step.

CONTENTS

I GROUND WORK 1
1 THEORY 3
1.1 Classical SR-Power Law 3
1.2 Quantum Corrections 4
1.3 Spin - Light 6
2 INITIAL DESIGN 7
2.1 Wiggler Magnet 7
2.2 Collimator 8
2.3 Ionization Chambers 8

II PROJECT WORK 11
3 DESIGN CONSIDERATIONS 13
3.1 Spin - Light Characteristics 13
3.2 Wiggler Magnet 14
3.2.1 Effects of wiggler on the beam 16
3.2.2 Effects of realistic dipole magnetic field with fringes 17
3.3 Collimation and Spin-Light fan size 19
3.4 Ionization Chambers 20
3.4.1 Relative Polarimetry 21
3.4.2 Absolute Polarimetry 25
3.4.3 Effects of Extended Beam Size 28
3.4.4 Current and Future Work: GEANT 4 Simulation 30
4 SUMMARY 33
4.1 Systematics 33
4.2 Conclusion 34
A LICENSE 35
B NUMERICAL INTEGRATION CODE 37
B.1 Numerical Integration of the SR - Power Law 37
B.2 LANL Poisson SupeFish Geometry Description 49
B.3 Recursive SR Spectra Adding Code 51
B.4 GEANT4 Geometry File 53

Part I

GROUND WORK

Initial work at proposal time is discussed here. Also included are some glimpses into the theory that motivates this work. This section enumerates the work borrowed from previous work done in this field notably at Novosibirsk, Russia. Also presented are some recent developments in Ionization Chamber Technology.

1

THEORY

1.1 CLASSICAL SR-POWER LAW

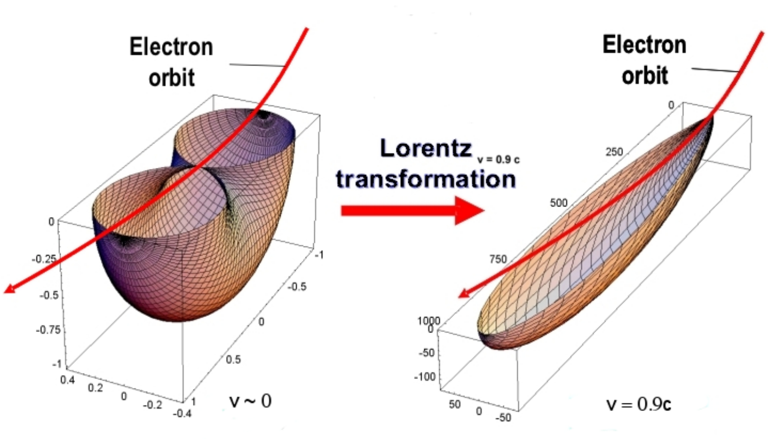


Figure 1: Angular distribution of synchrotron radiation shown for the bottom half of the electron's orbital plane. The left figure is for slow electrons, $\beta \sim 0$ and the right figure is for highly relativistic electrons, $\beta \sim 1$. [1]

[1] *Proposal to the EIC R & D:* https://wiki.bnl.gov/conferences/images/7/74/RD2012-11_Dutta_eic_polarimetry.pdf

[2] *D. D. Ivanenko, I. Pomeranchuk, Ya Zh. Eksp. Teor. Fiz. 16, 370 (1946); J. Schwinger, Phys. Rev. 75, 1912 (1947)*

The total radiative power due to circularly accelerated particles is given by Larmor formula which is $P_{clas} = \frac{2}{3}\frac{e^2\gamma^4 c}{R^2}$ (*i.e.* P is proportional to E^4) {here P_{clas} is the classical total radiative power, e is the electron charge, $\gamma = \frac{E}{m_e c^2}$: the Lorentz boost, c is the speed of light, and R is the radius of trajectory of the electrons}. The angular dependence of radiative power can also be computed via classical electromagnetism.

[3] *I. M. Ternov and V. A. Bordovitsyn, Vestn. Mosk. Univ. Ser. Fiz. Astr. 24, 69 (1983); V. A. Bordovitsyn and V. V. Telushkin, Nucl. Inst. and Meth. B266, 3708 (2008)*

$$\frac{dP_{clas}}{d\Omega} = \frac{e^2\gamma^4 c}{4\pi R^2}\frac{(1-\beta Cos\theta)^2 - (1-\beta^2)Sin^2\theta Cos^2\phi}{(1-\beta Cos\theta)^5} \tag{1}$$

In classical electrodynamics the angular distribution of radiative power from synchrotron (SR) light can be calculated, as indicated in the above eq.(1) [2] {where $d\Omega = d\theta d\phi$, and $\beta = \frac{speedofparticle}{speedoflight}$}. The SR light-cone is spread over a well defined cone with the angular spread - θ in retarded time $\Delta t' \approx \frac{\Delta\theta}{\omega_o}$ {where ω_o is the angular frequency of the photon}. There is no reason to believe that the SR spectrum is mono-energetic. The spectral width of SR radiation can be formulated: $\Delta\omega \approx \frac{1}{\Delta t'(1-\beta)} = \frac{1}{2\gamma^3\omega_o}$. A critical acceleration can be envisioned at which the SR may consist of just 1 photon by solving the equation $\gamma m_e c^2 = \hbar\omega_C$. To achieve this acceleration, a critical uniform magnetic field can be applied on an electron which is $B_c = \frac{m_e^2 c^3}{e\hbar}$ {here m_e refers to the rest mass of an electron}. The energy of the electron at these accelerations can be called the critical energy, $E_C = m_e c^2 \sqrt{\frac{m_e R_c}{\hbar}} \approx 10^3$TeV for R_c of about 10m. It is important to note that even though accelerated electron is under consideration, the equations above don't have acceleration terms. This is because a conventional circular motion is used to calculate the parameters and only the speed is important as the acceleration is a function of speed.

1.2 QUANTUM CORRECTIONS

In the classical theory, the spin does not explicitly appear in the equation. But, in QED, the angular dependence of synchrotron (SR) light can be calculated to a great degree of precision and the spin of the electron involved in SR emission appears explicitly in the power law [3]. The quantum power law for SR was worked out by Sokolov, Ternov and Klepikov as a solution to the Dirac equation [4] and includes the effects introduced by electrons undergoing $j \to j'$ (spin dependence) transitions besides also elaborating on the fluctuations to the electron orbit ($n \to n'$ transitions - linear correction to orbit and $s \to s'$ transitions - quadratic correction to orbit). The power law when integrated over all polarizations and (spacial) angular dependencies, can be written as [5];

[4] *A. A. Sokolov, N. P. Klepikov and I. M. Ternov, JETF 23, 632 (1952).*

[5] *A. A. Sokolov and I. .M. Ternov, Radiation from Relativistic Electrons, A.I.P. Translation Series, New York (1986) ; I. M. Ternov, Physics - Uspekhi 38, 409 (1995).*

$$P = P_{Clas}\frac{9\sqrt{3}}{16\pi}\sum_s \int_0^\infty \frac{y dy}{(1+\xi y)^4} I_{ss'}^2(x) F(y) \quad (2)$$

$$F(y) = \frac{1+jj'}{2}\left[2(1+\xi y)\int_y^\infty K_{\frac{5}{3}}(x)dx + \frac{1}{2}\xi^2 y^2 K_{\frac{2}{3}}(y) - j(2+\xi y)\xi y K_{\frac{1}{3}}(y)\right] + \frac{1-jj'}{2}\xi^2 y^2\left[K_{\frac{2}{3}}(y) + l K_{\frac{1}{3}}(y)\right] \quad (3)$$

{Where $\xi = \frac{3B}{2B_c}\gamma$, 'j' is the spin of the electron, $y = \frac{\omega_o}{\omega_c}$, $x = \frac{3}{4}\frac{\xi\gamma^3 y^2}{(1+\xi y)^2}$, $I_{ss'}(x)$ are Laguerre functions, and $K_n(x)$ are modified Bessel functions}.

If $\xi << 1$ (given $B_c \approx 4.41 \times 10^9$ Tesla) then Eq. (3) can be Taylor expanded in terms of powers of ξ as follows;

$$P = P_{clas}\left[\left(1 - \frac{55\sqrt{3}}{24}\xi + \frac{64}{3}\xi^2\right) - \left(\frac{1+jj'}{2})(j\xi + \frac{5}{9}\xi^2 + \frac{245\sqrt{3}}{48}j\xi^2\right) + \left(\frac{1-jj'}{2}\right)\left(\frac{4}{3}\xi^2 + \frac{315\sqrt{3}}{432}j\xi^2\right) + ...\right] \quad (4)$$

Eq. (3) and (4) include the a number of effects;

- Classical SR
- Thomas Precession
- Larmor Precession
- Interference between Larmor and Thomas Precession
- Radiation from intrinsic magnetic moment (including anomalous magnetic moment)

Eq. (4) can be re-expressed as a difference between power from unpolarized and polarized electron beam.

$$P_{Spin} = P_{Pol.} - P_{UnPol.} = -j\xi P_{Clas} \int_0^{\infty} \frac{9\sqrt{3}}{8\pi} y^2 K_{\frac{1}{3}}(y)dy \qquad (5)$$

Eq. (5) is essentially the spin light that this project is based on which opens up the possibility of using SR part to measure the polarization of the beam.

The power law in eq. (5), that was derived using QED, has been extensively tested and verified at the Novosibirsk Storage ring over a large range of wavelengths (of SR). For this, the Novosibirsk group used a "snake" shaped wiggler magnet to produce the SR from a relatively low energy electron beam of about $0.5GeV@100\mu A$.

[5] *S. A. Belomesthnykh et al., Nucl. Inst. and Meth. 227, 173 (1984).*

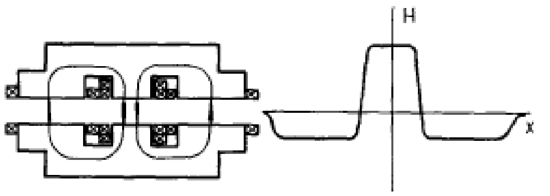


Fig. 1. The field vs the current in the 'snake'. A schematic of the 'snake' and the field distribution along its axis are shown below.

Figure 2: Magnetic snake used at the VEPP-4 as a source of SR to test the spin dependence of SR. [6]

[6] *K. Sato, J. of Synchrotron Rad., 8, 378 (2001).*

Figure 3, clearly demonstrated the power dependence of the SR on beam-electron polarization. It remains to be shows as to how the SR spectra could be measured.

The spin flip probability has a special significance in modern day electron storage rings and is given by the relation [4];

$$W_{\uparrow\downarrow} = \frac{1}{\tau}(1 + j\frac{8\sqrt{3}}{15}) \qquad (6)$$

{where $j = +1$ is spin along the magnetic field (and $j = -1$ is spin against magnetic field), and τ is the time involved in the process}.

As a result of the probability of spin aligning with the magnetic field direction would become very high over long periods of time. A circulating electron beam, such as ones in storage rings self polarize,

[7] *J. Le Duff, P. C. Marin, J. L. Manson, and M. Sommev, Orsay - Rapport Technique, 4-73 (1973).*

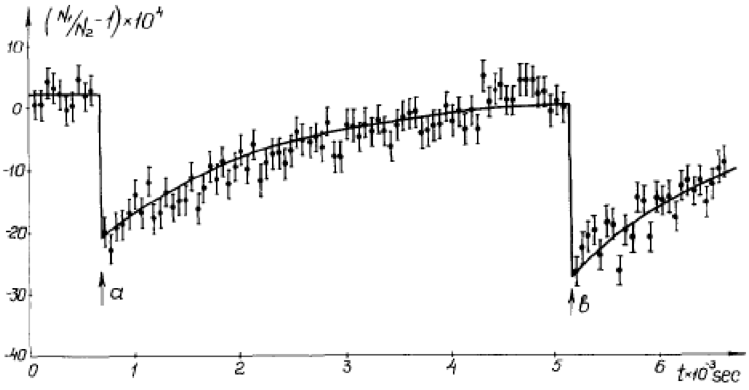


Fig. 12. The measurement results of the SR-intensity as a function of the degree of polarization of the beam. The field in the 'snake' coincides, in direction, with the storage ring guiding field. At points a and b one of the bunches (N_1), was quickly depolarized. The measurement time at a point is 60 s. The bunch polarization time is $\tau_p = 1740 \pm 20$ s ($\zeta = 0.726$).

Figure 3: Results from the experiment showing the increase in intensity of SR as the polarization builds up and then suddenly drops to zero when an RF field is used to depolarize the beam. [6]

and this has been studied in great detail at many storage rings such as ones at DESY, PSI and CESR though it was first observed at the Orsay storage ring [7].

1.3 SPIN - LIGHT

In the Spin-Light polarimeter, the spin-flip term in the power law does not play an important role. The integral power law without the spin-flip term can be written as [4];

$$P_\gamma = \frac{9\eta_e}{16\pi^3}\frac{ce^2}{R^2}\gamma^5 \int_0^{\inf} \frac{y^2 dy}{(1+\xi y)^4} \oint d\Omega (1+\alpha^2)^2 \times \left[K^2_{\frac{2}{3}}(z) + \frac{\alpha^2}{1+\alpha^2} K^2_{\frac{1}{3}}(z) + j\xi y \frac{\alpha}{\sqrt{1+\alpha^2}} K_{\frac{1}{3}}(z) K_{\frac{2}{3}}(z)\right] \quad (7)$$

where $z = \frac{\omega}{2\omega_C}(1+\alpha^2)^{\frac{2}{3}}$ and $\alpha = \gamma\psi$ where ψ is the vertical angle *i.e.* above and below the orbit of the electron. Notice that the last term with a j disappears from the integral over all angles ($-\frac{\pi}{2} \leqslant \psi \leqslant \frac{\pi}{2}$). But for an electron that is polarized, the power below (*i.e.* $-\frac{\pi}{2} \leqslant \psi \leqslant 0$ and above (i.e. $-\frac{\pi}{2} \leqslant \psi \leqslant \frac{\pi}{2}$) are spin dependent. More importantly the difference between the power radiated above and power radiated below is directly spin dependent, which can be directly obtained from Eq. (7) in differential form for circular arcs in the circular cross-section the SR-light cone at an angle θ.

$$\frac{\Delta P_\gamma(j)}{\Delta\theta} = \frac{3}{2}\frac{\hbar c\gamma^3 y}{R} \times \frac{3}{\pi^2}\frac{1}{137}\frac{I_e}{e} j\xi\gamma \int_{y_1}^{y_2} y^2 dy \int_0^{\alpha} \alpha(1+\alpha^2)^{\frac{3}{2}} K_{\frac{1}{3}}(z) K_{\frac{2}{3}}(z) d\alpha \quad (8)$$

2 INITIAL DESIGN

It is obvious now that our setup will have a wiggler magnet which shall be the source of SR and an ionization chamber to measure the power spectra of the SR emitted at the wiggler magnet by the polarized electrons.

2.1 WIGGLER MAGNET

In order to create a fan of SR light (as illustrated in Figure 5), the electron beam could be made to bend in presence of a magnet. An arrangement that would lead to the production of the SR-Cone must look similar to the "snake" magnets that were used by VEPP-4 as described in *Chapter 1*.

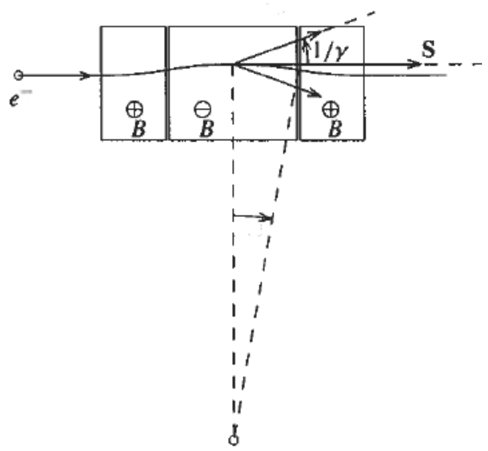


Figure 4: A 3-pole wiggler with central dipole twice the length of end dipoles.

A set of 3 dipoles (3 - pole dipole), each with constant uniform magnetic field would be ideal for this purpose. The central dipole would have twice the pole length as compared to the ones on either side of the central dipole, but with a magnetic field an opposite polarity like in Figure 4. This design will give rise to 4 fans, 2 towards either side of the beam (left and right). Each of the fan will have spacial asymmetry as a function of spin - polarization of the electrons in the vertical plane (up - down the electron beam's orbit which is perpendicular to the plane which contains the 4 fans). The 4 fans help characterize the systematics better since this configuration will flip the sign of the spin - dependent term in Eq. (8) twice essentially returning the sign to the original status.

With this geometry in mind, one could then simulate and calculate the requisite pole strengths and pole lengths appropriate to energies at Electron Ion Collider (EIC). Of course, one would also have to consider time scales at which the statistics would be sufficient to achieve the design requirement ($< 1\%$) of precision. A number of techniques were employed to tackle the above issues. Including issues such as optimizing the distance between each pole were solved through a full fledged GEANT-4 simulation.

2.2 COLLIMATOR

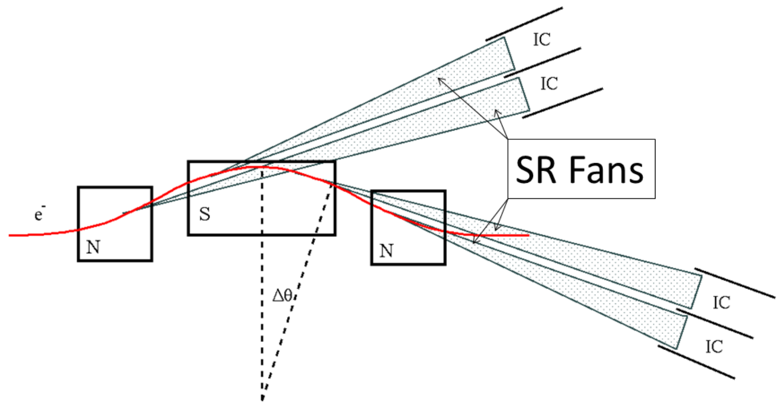


Figure 5: A rough schematic of the 4 fans that shall be created by the Wiggler Magnet

The 4 fans of SR-Light need not be separated from each other. They may in fact overlap and extracting the power - asymmetry information from complicated overlappings would require extensive modeling. This might also introduce new sources of uncertainties. Therefore to uniquely sample each fan at the ionization chamber, collimators would need to be placed on each face of the dipoles to direct and separate the fans of SR. The position of the collimators will be calculated using optimized values of pole - strengths, pole - lengths and relative position of each of the dipoles.

2.3 IONIZATION CHAMBERS

The Ionization Chambers (IC), one on each side (2 fans per IC) of the beam, could be used for measuring the spacial asymmetry in the SR - fans to be used to compute the polarization of the electrons in the beam. One would not expect a large asymmetry in the Spin-Light component (of the order of about 10^{-4}), therefore a high resolution, low noise IC is demanded. The IC will be very close to the beam line and the spin-independent background may be as high as 10^{12}photons/s, therefore the ICs have to be radiation hard. The geometry of the magnets could be changed to deal with SR Spectra with characteristic energy peak in the ranges of about $500\text{keV} - 2.5\text{MeV}$. An IC with Xenon as ionization media operated in the current mode can in principle handle high fluxes and have low noise disturbances [8]. It might be important to note that the Spin-Light asymmetry is spread over the entire spectra of the SR. Sampling radiation over large energy spectrum becomes important. Since Xenon has the lowest ionization energy of about 21.9eV, among non-radioactive nobel gases, it seems to be an ideal candidate. ICs with Xenon under high pressures have already been developed and well tested to perform well in the energy ranges of $50\text{keV} - 2.0\text{MeV}$ [9]. Pressures involved in HPXe ICs exceed $50\text{atm}@0.55\text{g/cc}$ but they work well at room - temperature [10],[11],[12]. One of the bottlenecks was the precision of purity of the Xenon gas in its pristine form. But owing to advances in gas purification techniques [13], a best energy resolution of 2.4% at about 0.662MeV has been shown to be possible when current signals from the shower is used in presence of prompt Xenon scintillation [14]. The results from this attempt has been fairly promising (Figure 6). ICs with $3\% - 4\%$ energy resolution are even being sold commercially by Proportional Technologies Inc. [15].

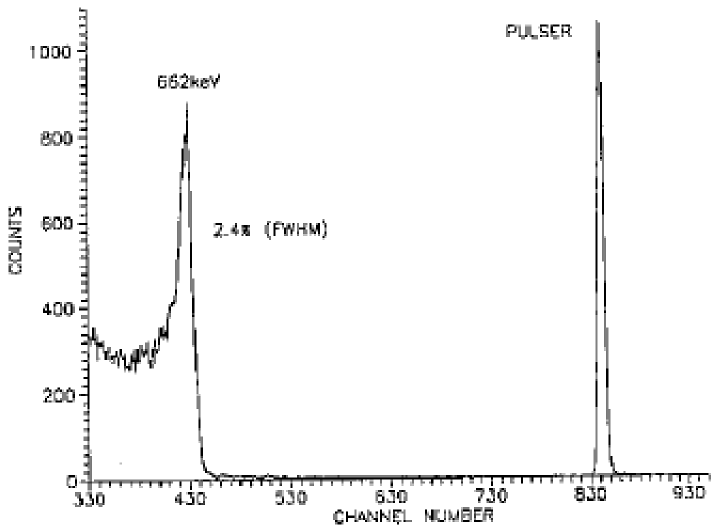


Figure 6: Cs^{137}: $E_i = 1.7\text{kV/cm}$, pulse height spectrum in 57atm at 295K of xenon, $E_i = 1.7\text{kV/cm}$ [14].

[8] *A. E. Bolotnikov and B. Ramsey, Nucl. Inst. and Meth. A396, 360 (1997).*

[9] *T. Doke, Portugal Phys. 12, 9 (1981).*

[10] *V. V. Dmitrenko et al., Sov. Phys.-tech. Phys. 28, 1440 (1983); A. E. Bolotnikov et al., Sov. Phys.-Tech. Phys. 33, 449 (1988)*

[11] *C. Levin et al., Nucl. Inst. and Meth. A332, 206 (1993).*

[12] *G. Tepper and J. Losee, Nucl. Inst. and Meth. A356, 339 (1995).*

[13] *A. E. Bolotnikov and B. Ramsey, Nucl. Inst. and Meth. A383, 619 (1996).*

[14] *G. Tepper and J. Losee, Nucl. Inst. and Meth. A368, 862 (1996).*

[15] *Proportional Technologies Inc., www.proportionaltech.com.*

Part II

PROJECT WORK

In this section, the work is presented with additional studies resulting from discussion with experts. It includes establishing the idea with a "back of the envelop calculation" backed by a full fledged GEANT-4 simulation. Also, related effects such as beam motion were studied and their effects that impact the polarimeter negatively were shown to be minimal. Even though the GEANT-4 simulation is still in the making, a skeletal code is briefly explained here.

3 DESIGN CONSIDERATIONS

3.1 SPIN - LIGHT CHARACTERISTICS

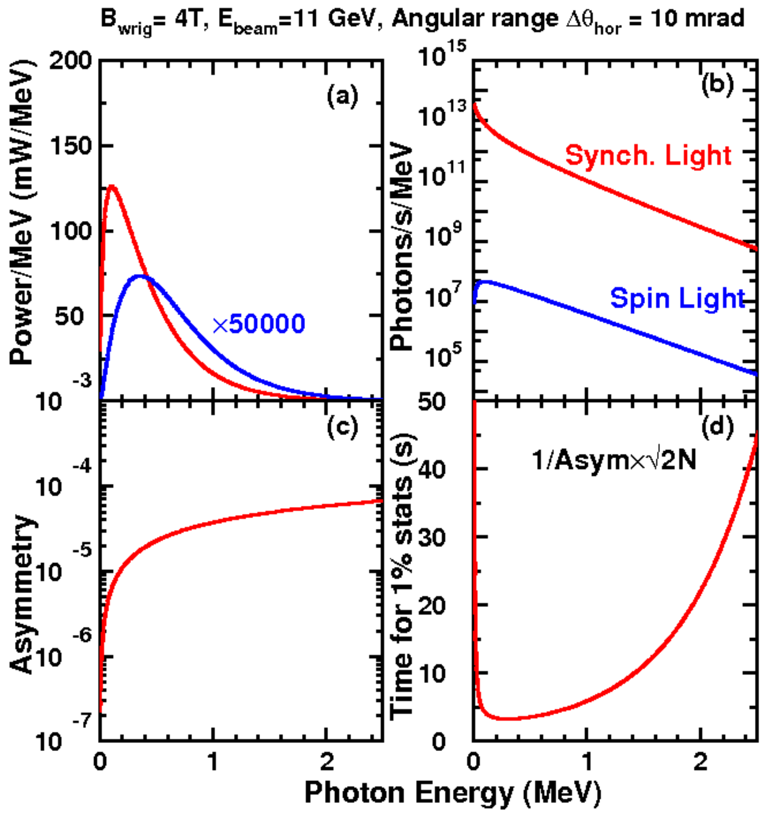


Figure 7: A. Plot of difference between the number of SR and spin light photons that go above and below the orbit of the electron (ΔP_γ) vs. their energy
B. Plot of total number of SR, spin light photons P_γ vs. their energy
C. Plot of the Asymmetry vs. the photon energies
D. Plot of time required to achieve 1% statistics by sampling one wavelength of the spin-light spectra vs. the energy of the photons.

The spin - dependence of the SR can be studied by examining Eq. (6-8) in *Chapter 1*. Using $I_e = 100\mu A$ and $E_e = 11GeV$, Eq. (6-8) were numerically integrated (Appendix B.1) between $-1 < \alpha < 1$ and $\Delta\theta = 10mrad$, for a uniform magnetic field of $B = 4T$ assuming 100% longitudinal polarization.

In Figure 7, the total number of SR and spin light (P_γ) photons radiated is plotted. Also in Figure 7, is a plot the difference between the power of spin light spectra above and below the orbit of the electron

(ΔP_γ). An asymmetry term is defined to be $A = \frac{\Delta P_\gamma}{P_\gamma}$, which was used to nail down the range of energies of the photons which must be measured. Lastly, a plot of sampling time required ($T_s = \frac{\Delta A}{A} = \frac{1}{A\sqrt{2PE_e}}$) to achieve the design precision goals.

It immediately becomes clear that the ionization chamber, which is envisioned to measure the asymmetry (that in turn will be used to compute the electron polarization), will have to be operational at wavelengths corresponding to hard - XRays. Furthermore, the asymmetry plot, Figure 7.D, demands that the sampling be done at the higher energies (and not close to $0.5MeV$) since at higher energies the asymmetry is not rapidly changing, thus making it an ideal high-energy polarimetry technique. It might be important to note that the asymmetry is fairly low but since the integrated power of spin-light is very high, the time required for achieving 1% polarimetry is only of the order of a few seconds. Also, one could plot the asymmetry and SR spectra for different energies to study the trends with change in beam energy. This indicates that there are no suppression effects at higher energies that might hinder effective polarimetry.

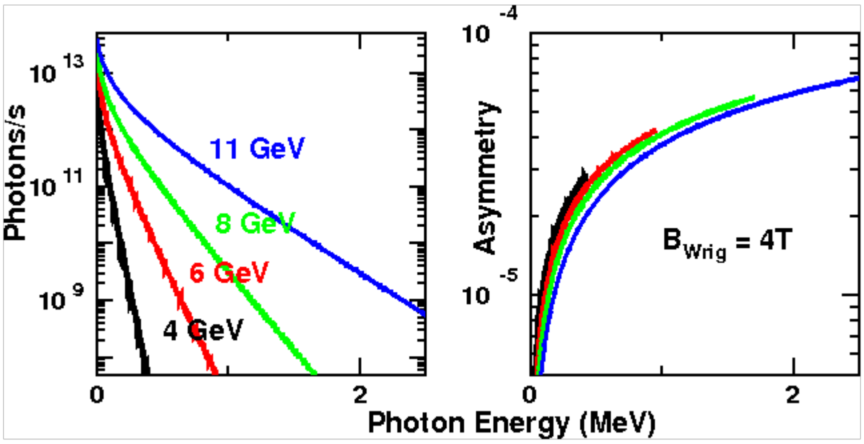


Figure 8: (Left) A. Plot of spin light spectra over photon energies for various electron beam energies ranging from $4GeV - 12GeV$;
(Right) B. Plot of asymmetry over photon energies for various electron beam energies ranging from $4GeV - 12GeV$

3.2 WIGGLER MAGNET

In order to establish the dimensions of the dipoles, the same graphs as in *Section 3.1* Figure 8 were plotted for various pole lengths and magnetic fields. First, the asymmetry increases very slowly with field strength as shown in Figure 9, and the figure of merit (time for 1% statistics) improves very slowly with magnetic fields above 3T as shown in Figure 10B, therefore $B = 4T$ was chosen since 4T wiggler magnets are easily available at light sources around the world. Secondly, an appropriate pole length of $L_p = 10cm$ was selected by

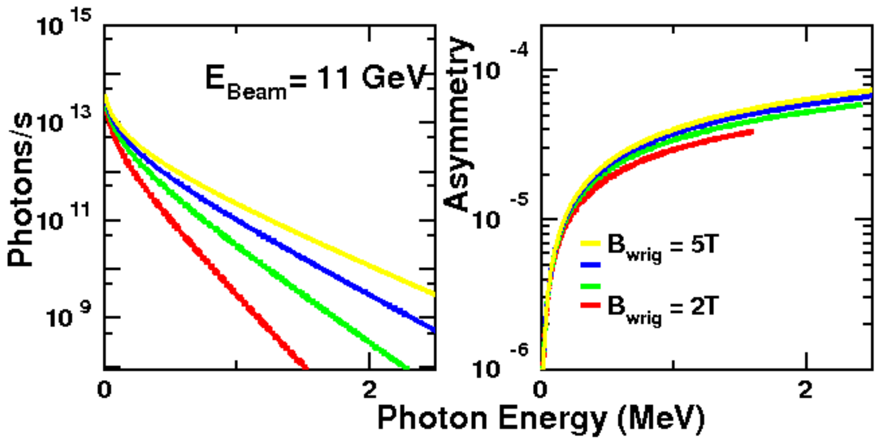


Figure 9: (Left - Right)
A. Plot of total number of spin light photons P_γ vs. their energy for various pole strengths
B. Plot of the newly defined term - Asymmetry vs. the photon energies for various pole strengths

looking at Figure 10 and selecting out the pole length for the pole strength of $4Tesla$. It is noteworthy to see that the plot of pole length as a function of pole strength was done keeping in mind an SR fan - angular spread of about $10mrad$. The plot in Figure 10.B also reassures the reasonable time requirement to achieve the design precision goal of 1%. The last parameter in the wiggler to be fixed is the distance between each dipole.

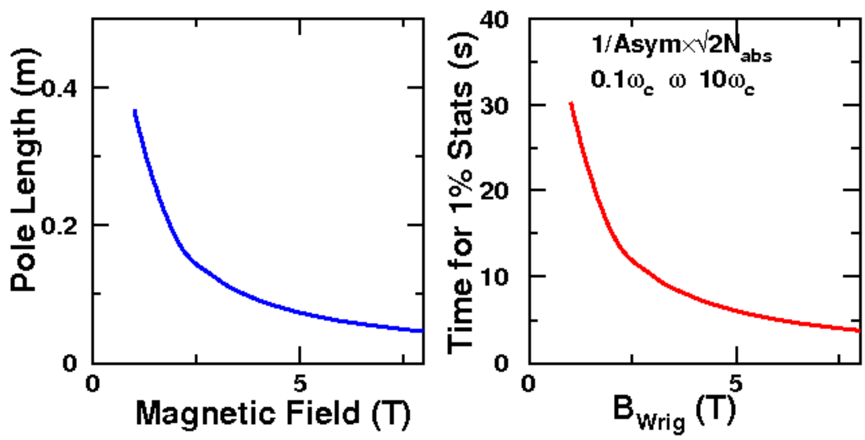


Figure 10: (Left - Right)
A. Pole length required for a $10mrad$ angular spread of SR light fans with $B = 4T$
B. Dependence of time required to achieve 1% statistics by sampling one wavelength of the spin-light spectra with pole strengths.

3.2.1 *Effects of wiggler on the beam*

[16] *B. Norum, CEBAF Technical note, TN-0019 (1985).*

[17] *M. Sands, SLAC Technical note, SLAC-121 (1970).*

A polarimeter must be non - invasive and therefore answering the question of what would be the effects of putting such as polarimeter on a beam line is very important. But the effect of a high energy electron beam emitting SR has been well studied [16]. The number of photons (N) emitted by an electron when it is deviated by a radian, from its initial linear trajectory, when acted upon by a magnetic field is distributed as per the conventional Poisson distribution [17] about a mean value of n;

$$\bar{N}(n) = \frac{n^N e^{-n}}{N!} \tag{9}$$

$$n(E_e) = \frac{5}{2\sqrt{3}} \frac{\gamma}{137} = 20.6E_e \tag{10}$$

The average energy of the SR photons can also be written down as {where E_e is the electron energy};

$$\bar{E}_e = \hbar\bar{\omega} = \frac{3}{2}\frac{\hbar c\gamma^3}{R} = \frac{3}{2}\frac{\hbar E_e^3}{R m_e^3 c^5} \tag{11}$$

In the case of a spin-light polarimeter, the beam energy is about $11GeV$ and we choose pole strength to be about $4T$ in *Section 3.2*. An angular bend of about $10mrad$ of the beam is sufficient for such a polarimeter. Using the values of average number of photons emitted and their average energy, the average energy fluctuation ($\Delta\bar{E}_e$) of the beam can be computed.

$$n = 20.62 \times 11_{GeV} \times .01_{rad} = 2.06 \tag{12}$$

$$\bar{E}_e = \frac{3}{2}\frac{\hbar(11_{GeV})^3}{10_m m_e c^5} = .199MeV \tag{13}$$

$$\frac{\Delta\bar{E}_e}{E_e} = \frac{\sqrt{n}\bar{E}_e}{E_e} \approx 2.5 \times 10^{-5} \tag{14}$$

The energy fluctuations are smaller than the typical precision with which the energy can be measured at an electron accelerator.

Another parameter which needs to be checked before proceeding, is the transverse kicks ($\Delta\theta_e$) received by the electrons when emitting SR photons in the magnets. The transverse kicks can be calculated in terms of angles knowing that the SR power spectrum usually peaks at an angle $\theta_\gamma = \frac{1}{\gamma}$ [17] {where E_γ is the SR - photon energy};

$$\Delta\theta_e = \frac{E_\gamma Sin(\theta_\gamma)}{E_e} \approx 11.3 \times 10^{-9} \frac{E_{e(GeV)}}{R_(m)} \tag{15}$$

$$\bar{\theta}_e = \sqrt{n}\Delta\theta_e \approx 1.5 \times 10^{-8}_{(rad)} \tag{16}$$

It can be clearly seen from Eq. (14) and Eq. (16) that both energy fluctuation and angular kicks shall be negligible. This can be seen for all practical purposes in the GEANT-4 simulation that this work demands. This polarimetry method remains a non-invasive procedure.

3.2.2 *Effects of realistic dipole magnetic field with fringes*

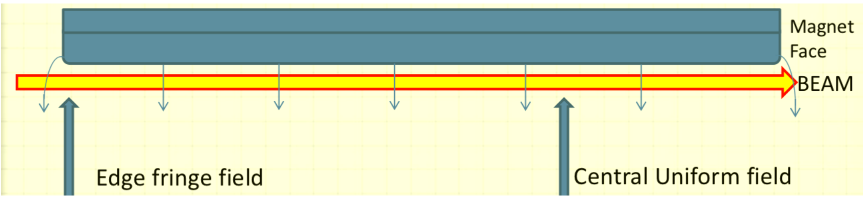


Figure 11: Schematic diagram of the planes at which position the simulation was carried out.

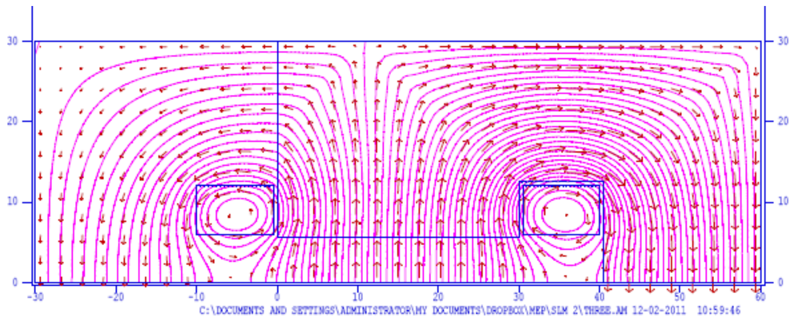


Figure 12: Field map of the dipole face at the center of the dipole.

In *Section 3.2*, while plotting the power spectra and the asymmetry generated by the code in *Appendix B.1* a uniform field was used. But the code can also take a field map. A field map can be generated by solving Maxwell's equations with appropriate boundary conditions. This is essential since field in the transverse plane (perpendicular to the motion of electrons) might distort the SR spectrum and thereby change the asymmetry. In fact there is a custom built suite of programs written by *Los Alamos National Laboratory* to precisely do this called **LANL Poisson SuperFish** [18].

[18] *Poisson SuperFish 2D EM Solver*, `laacg1.lanl.gov/laacg/services/sfu_04_04_03.phtml`, 2007.

In LANL SuperFish, the magnet geometry can be easily defined as is done in *Appendix B.2*. The field map of the magnet can then be plotted. Here, the field map at the edge where the electron beam enters the magnet and at the center of the dipole is presented. In Figures 11 & 13, note that the beam pipe is going at the center below the magnet pole. In Figure 13, the physical taper of the cores can be

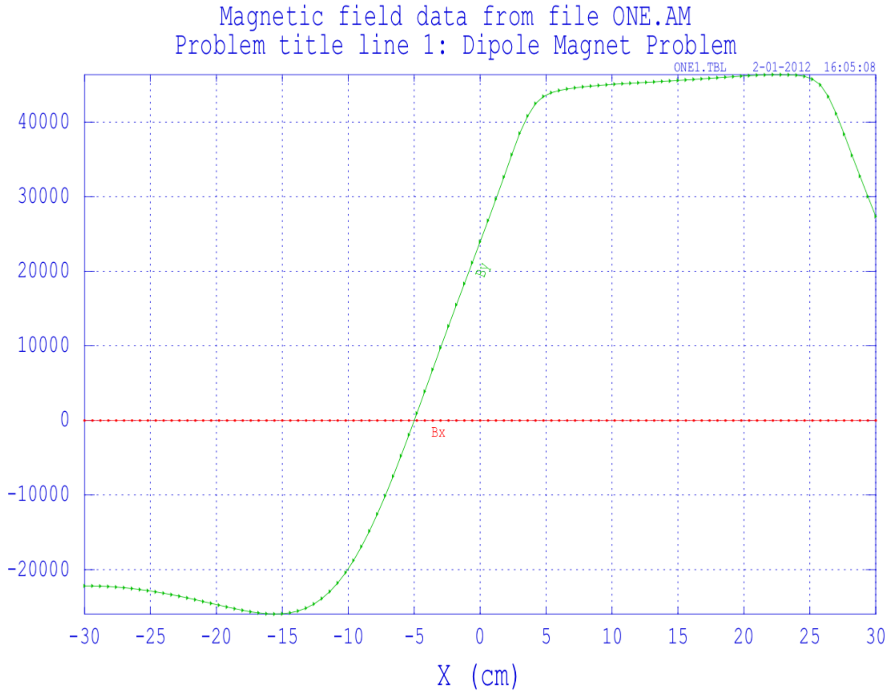


Figure 13: Plot of both the x and y components of the magnetic field on the transverse plane at the the center of the dipole (Beam pipe is centered around 15cm mark along the 'x' axis).

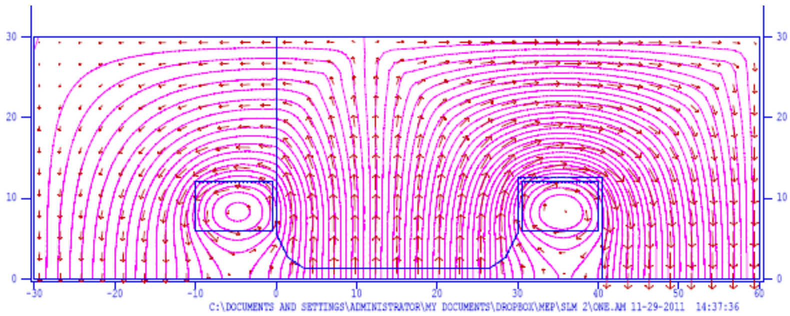


Figure 14: Field map of the dipole face at the edge of the dipole.

notices, since it is at the edge of the magnet face. This taper of the poles is absent in Figure 11, since it is at the center. In Figures 11 & 13, the singularities seen are the areas where the current cuts the plane. Also, it is important to note that the entire 'C' magnet is not visible in the field-map, only the top half of the C magnet is shown in the field map.

X - Axis is to the right and left of the beam and Y - Axis is to the above and below the beam. Also the XY plane is perpendicular to the direction of motion of electron. In Figure 13, it might be important to note that there is no component of the magnetic field. This is because it is at the center of the dipole and there is no fringing of the field. But in Figure 15, there is a non - zero X component to the magnetic field

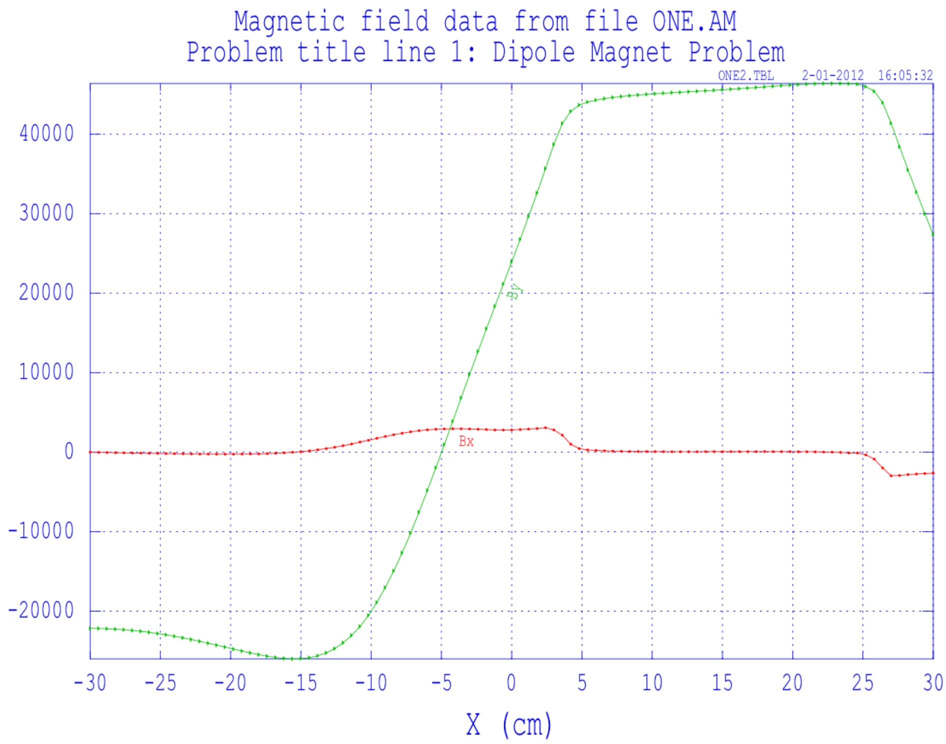


Figure 15: Plot of both the x and y components of the magnetic field on the transverse plane at the the edge of the dipole (Beam pipe is centered around 15cm mark along the 'x' axis).

since it is on the plane at the face of the magnet. A 2D simulation is sufficient since any components along the motion of electrons (Z Axis) will not affect the electrons.

The field map obtained here can be inserted into the numerical integration code (in *Appendix B.1*) and the power spectra and the asymmetry can be obtained. Even though there is a reduction in the total power output of light by introducing a realistic taper for dipole fields, the asymmetry has not changed. This implies that the changes introduces by the realistic dipoles are minimal.

3.3 COLLIMATION AND SPIN-LIGHT FAN SIZE

Even though the distance between the 3 dipoles should in theory not affect the physics involved, it is nevertheless an essential design parameter. A reasonable value of about $1\mathrm{m}$ distance between each dipole was used to start with but this value will be definitely fixed with a full fledged GEANT-4 simulation. A fan with $10\mathrm{mrad}$ spread would then give rise to a spot which is $10\mathrm{cm}$ big in the horizontal plane, $10\mathrm{m}$ from the wiggler where the ionization chambers will be placed. A more important dimension of the SR-spot at the ionization chamber is its height in the vertical direction. An angular spread of

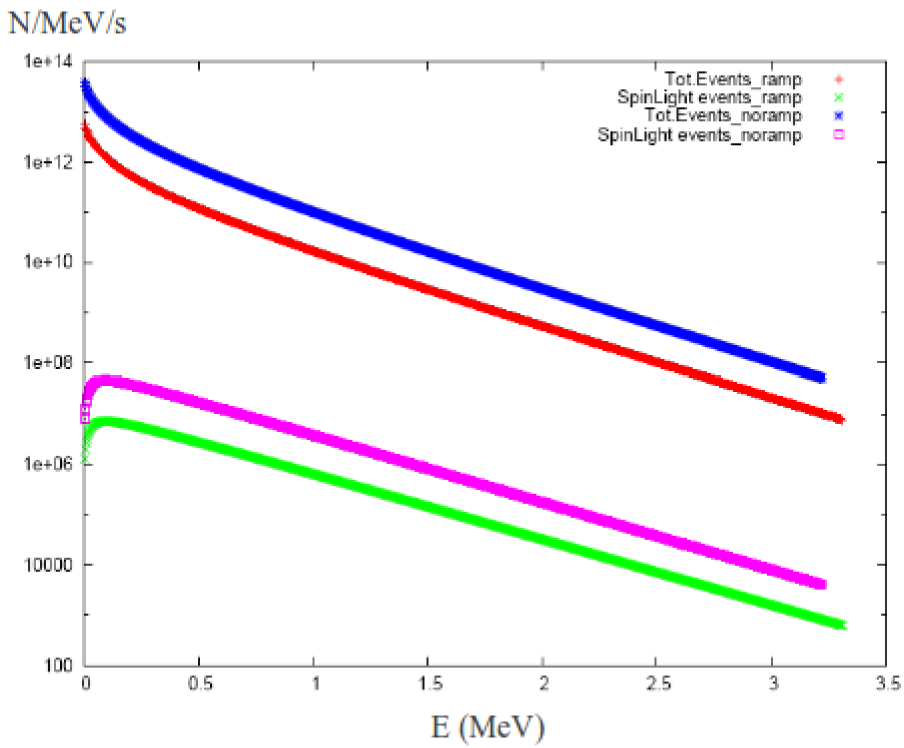


Figure 16: Plot showing the SR - Light (*TotEvents_ramp*) and Spin - Light (*SpinLightEvents_ramp*) power spectra with a realistic taper for the dipoles (Power spectra for uniform magnetic field have also been presented as *_noramp*).

$\Delta\theta = 1/\gamma = 100\mu rad$ would then give rise to a spot which is $1mm$ big in the horizontal plane, $10m$ from the wiggler, where the ionization chambers will be placed.

Figure 18 shows the origin of 4 different fans of SR Light (which contain the spin-light component) that are being created at the wiggler magnet. Corresponding fans of SR light create 4 spots at the ionization chamber which is located $10m$ from the wiggler magnet system. The spots at the ionization chamber as shown in Figure 19, merge with each other and may destroy the spacial asymmetry that contains the polarization information. Therefore collimators may be employed at the face of every dipole to select out a small section of the bigger SR fan as illustrated in Figure 20. After collimation the spots are all uniquely separated and 4 distinct spots can be observed at the ionization chamber (as in Figure 21).

3.4 IONIZATION CHAMBERS

[19] *K. Sato, J. of Synchrotron Rad., 8, 378 (2001); T. Gog, D. M. Casa and I. Kuzmenko, CMC-CAT technical report.*

The Spin Light polarimeter detector would consist of a position sensitive ionization chamber to measure the up-down asymmetry in the SR - Light. Such a position sensitive detector that could charecterize

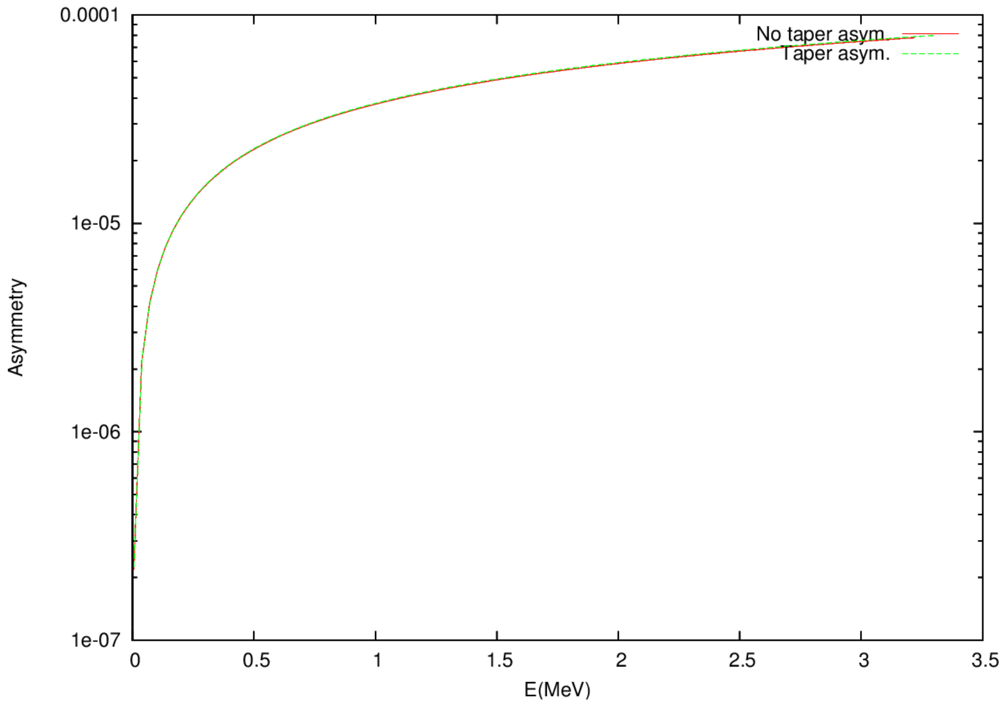


Figure 17: Plot of the assymetry with a realistic taper for the dipoles (*Taper asym.*).

X-Ray spectrum has already been developed at the *Advanced Light Source*, Argonne National Laboratory and at *Sprin-8 Light Source*. This uses a split - plane which essentially divides the ionization chamber into 2 separate chambers but with a common electrode. These have been demonstrated to have a resolution of about 5μm [19]. Subtracting the currents from the top chamber from the bottom chamber will then give a measure of the asymmetry in the SR-Light. A schematic diagram of the protoype is presented in Figure 23.

Using such an ionization chamber, one could easily carryout relative polarimetry. A more challenging but possible option would be to have an absolute polarimeter.

3.4.1 *Relative Polarimetry*

A Xenon media split plate would be an ideal differential ionization chamber. Using Ti windows of sufficient size could in principle cut down on low energy X-Rays (< 50KeV) and Ti has been shown to have a high transparency for hard X-Ray [20]. A schematic diagram of the ionization chambers for the Spin-Light polarimeter is presented in Figure 23. The Spin-Light Polarimeter Ionization chamber shall have 2 compartments into which the 2 collimated fans of SR Light will enter. On each side of the electron beam is one split - plane ionization chamber and therefore all 4 fans of SR - Light, produced at the wiggler magnet, are measured at the 2 ionization chambers. Notice for

[20] *G. Tepper and J. Losee, Nucl. Inst. and Meth. A356, 339 (1995).*

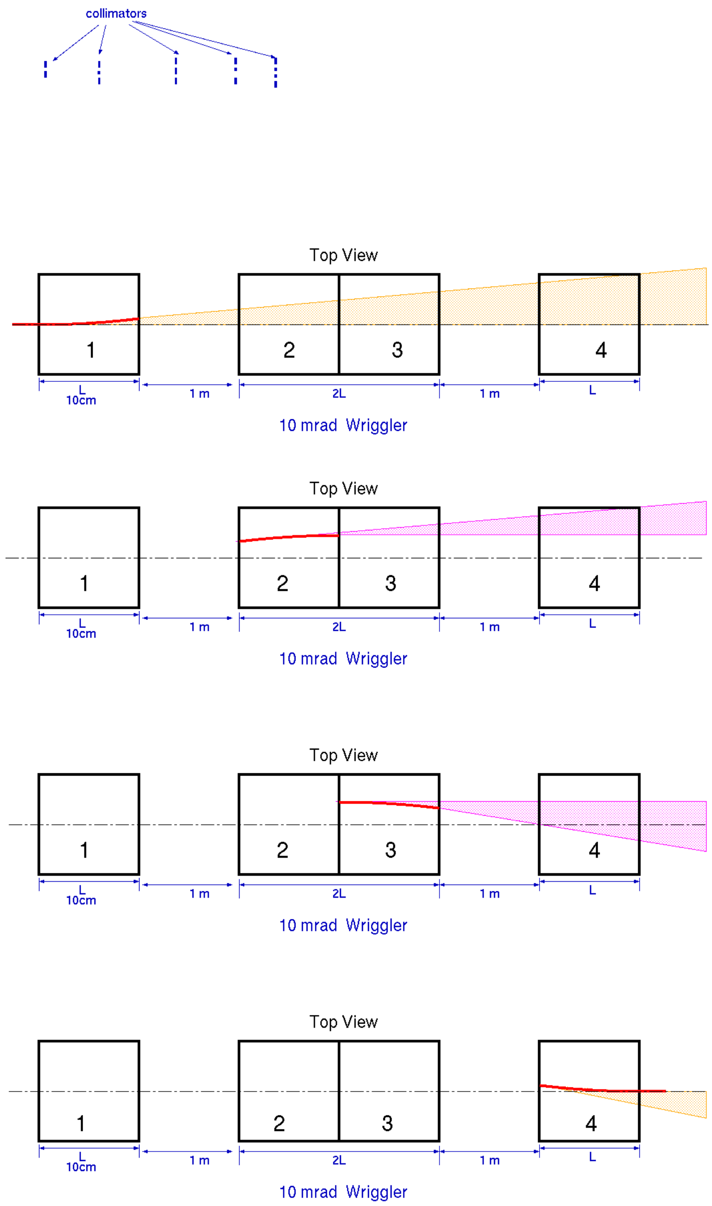


Figure 18: A schematic diagram showing the 4 fans of SR that originate at the wiggler magnet system.

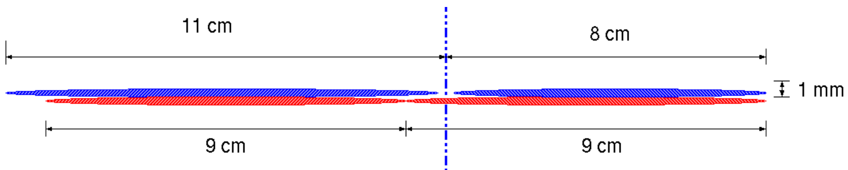


Figure 19: A schematic diagram showing the Spin-Light profile at the Ionization Chamber

polarimetry, just one ionization chamber is required. The 2 separate ICs will provide abundant statistics in a short time.

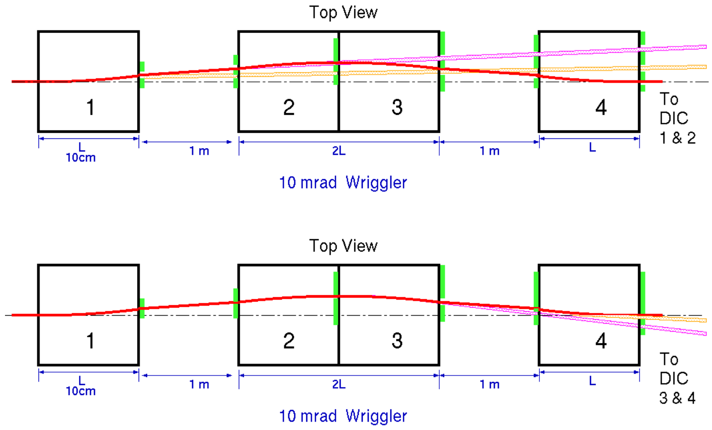


Figure 20: A schematic diagram showing the 4 fans of SR that originate at the wiggler magnet system with collimators.

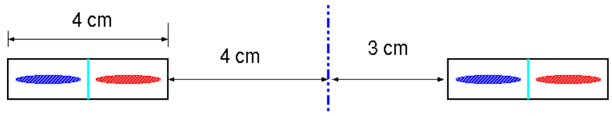


Figure 21: A schematic diagram showing the Spin-Light profile at the Ionization Chamber with collimators.

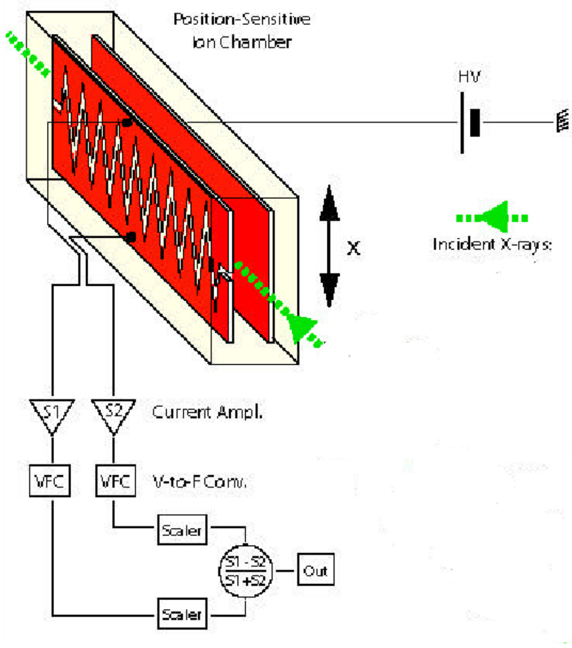


Figure 22: A schematic diagram of a prototype split plate Ionization Chamber.

The signal which will give us a measure of the spacial asymmetry could be measured by subtracting the currents from the UP and DOWN parts of the chamber after being amplified as shown in Figure 24. The spin - light asymmetry shall be of opposite signs on the

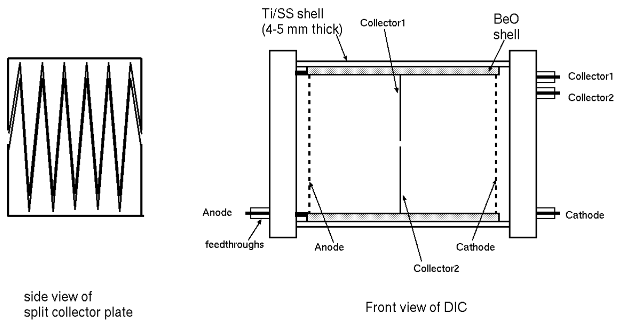


Figure 23: A schematic diagram of the Spin - Light Polarimeter Ionization Chamber.

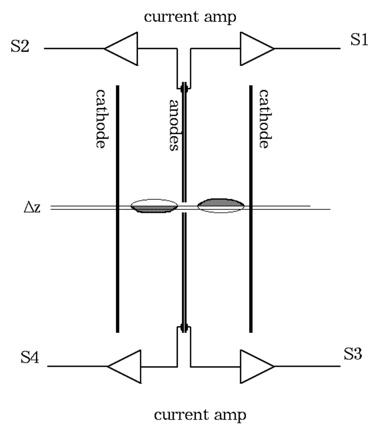


Figure 24: A schematic diagram of signal collection configuration.

LEFT and RIGHT parts of the chamber, since SR fans from adjacent wiggler dipole (which have opposite polarity) enter one on each side of the chamber (L-R). Beam motion effects are nullified as any motion will have have trend (same sign) on both the LEFT and RIGHT sides of the chambers. Each half (T-B) of the split plane collector measures a current proportional to the difference of photon flux between the 2 sides and therefore any vertical beam motion effects cancel out to the first order. The two signals indicated in Figure 24 can be quantified. This definitely shows that the vertical beam motion effects will be canceled out to first order.

$$S_1 = N^l_{SR} + N^l_{spin} + \Delta N^l_z - (N^r_{SR} - N^r_{spin} + \Delta N^r_z) = 2N_{spin} \quad (17)$$

$$S_2 = N^l_{SR} - N^l_{spin} - \Delta N^l_z - (N^r_{SR} + N^r_{spin} - \Delta N^r_z) = -2N_{spin} \quad (18)$$

[21] *G. Tepper and J. Losee, Nucl. Inst. and Meth. A356, 339 (1995).*

{Where $N^{l(r)}_{SR}$ is the number of SR Photons on the left (right) side of the middle split plate, $N^{l(r)}_{spin}$ is the number of spin-light photons and $\Delta N^{l(r)}_z$ is the difference in number of photons introduced by the vertical beam motion}.

Hence $S_1 - S_2 = 4N_{spin}$ is a measure of longitudinal polarization and $S_1 + S_2$ will give a measure of transverse polarization. The ability to measure both transverse and longitudinal polarization makes this a powerful polarimetry technique. The number of photons absorbed in the ionization chamber can be computed by multipling the SR Power equation Eq.(8) with the absorption function (where μ is the absorption coefficient which is material specific and t is the length of the chamber) $A(\lambda, t) = 1 - e^{-\mu(\lambda).t}$. With the help of values of μ obtained from NIST database [21], a plot of photons absorbed in a ionization chamber that is $50cm$ in length and held at $1atm$ pressure is shown in Figure 25. The spectra of number of photons absorbed was used to then calculate the detector response which in this case is asymmetry weighted against absorption.

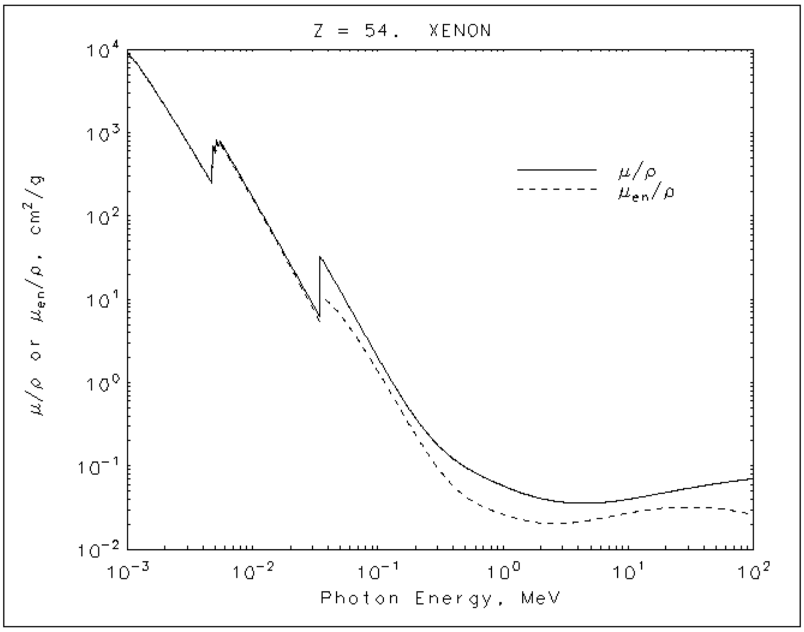


Figure 25: NIST plot of dependence of absorption coefficient of Xenon on the photon energy [21].

3.4.2 *Absolute Polarimetry*

A relative polarimeter could be turned into an absolute polarimeter by making a few modifications to the ionization chamber. for absolute polarimeter, a high resolution ionization chamber is required and so the natural choice would be a high pressure Xenon IC. A cylindrical chamber capable of withstanding $50atm$ of pressure could house the IC setup whereas the rest of the structure would remain unchanged from the relative IC with a few additions. The electrodes could be held in place with thin walled BeO ceramic material which would provide uniform electric field and reduce acoustic noise while being transparent to hard XRays [21]. This design eliminates the need for

[22] *S. Kubota, M. Suzuki and J, Ruan, Phys. Rev. B 21, 2632 (1980).*

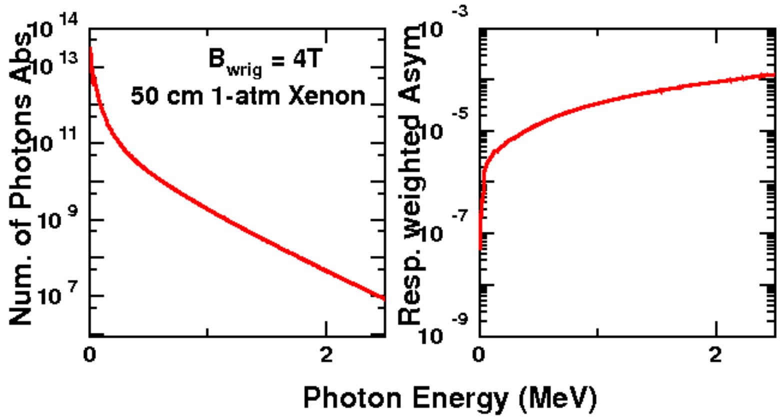


Figure 26: Plot of photons absorption spectra for the ionization chamber.

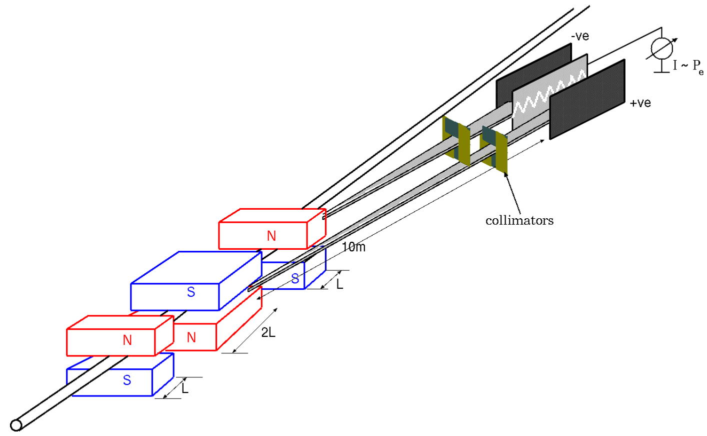


Figure 27: Schematic diagram of the entire Differential Spin Light Polarimeter (The only visible difference between the absolute and relative polarimeters in the schematics is the difference in collector plate bias).

field guide rings which require additional feed throughs and internal voltage dividers. In order to shield against space charge build-up, a wire mesh grid should be placed near the anode which carries a voltage that is intermediate in value to the drift potential (potential between the anode and the cathode). The ratio of the grid field to drift field can be adjusted to maximize the shielding efficiency. The cathodes and the intermediate grids would be build from stainless steel wire mesh to allow the compressed xenon UV scintillation light to be collected by the UV sensitive photomultiplier tubes (PMT). The scintillation signal has a fast component with a decay time of 2.2ns and slow component with a decay time of 27ns [22]. The scintillation light can be used during calibration, to provide a time zero reference for ionization position determination and can also be used for background suppression using pulse shape discrimination and for anti-coincidence Compton suppression. This will help improve the energy resolution and hence aid the determination of the sensitive en-

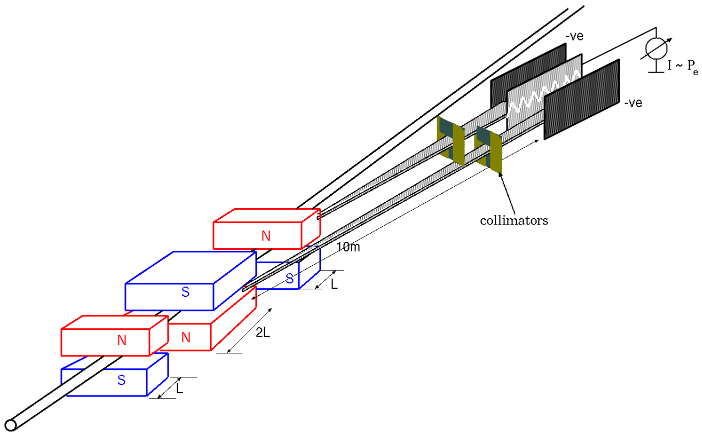


Figure 28: Schematic diagram of the entire Absolute Spin Light Polarimeter (The only visible difference between the absolute and relative polarimeters in the schematics is the difference in collector plate bias).

ergy range of the chamber (during calibration, when the chamber is operated in charge mode). Similar HPXe chamber (without the split anode) have been successfully operated [21] for over a decade now and are also commercially available. A schematic for such an IC is shown in Figure 29. The readout electronics chain would consist of a pre-amplifier and shaping amplifier unlike the current amplifiers used in the current mode ICs. In addition, one would also have to establish the linearity of such an IC given the high flux of photons making the calibration of the IC very challenging. The vertical beam

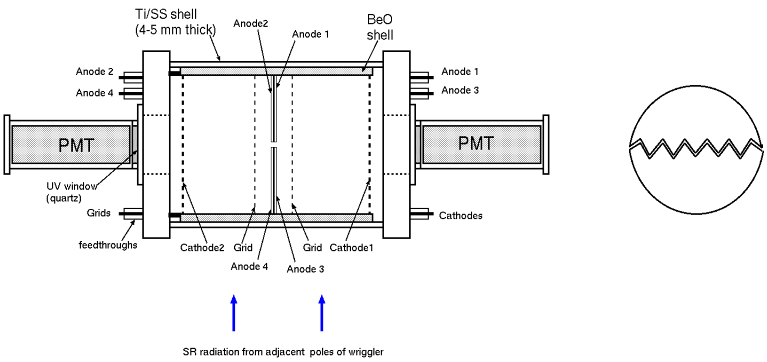


Figure 29: A schematic diagram of the Absolute Spin - Light Polarimeter Ionization Chamber.

motion effects in an absolute IC shall be cancelled to the first order just like in the differential IC. The current signal from each chamber is an integral over the sensitive energy range of the chamber. This energy range convoluted with the detector response function, can be determined by calibrating the chamber at low electron beam currents ($\sim 1nA$), where the photon flux is low enough to operate the chambers in charge mode. The pulse height spectrum from these calibration runs can be used to determine the sensitive energy range and the detector response function. The uncertainty in determining the absolute value of the range of energies integrated (specially the lower bound) is the other major source of uncertainty.

In the case of an absolute IC, 4 different signals involving each part (TOP/BOTTOM parts of split plane) and (LEFT/RIGHT) parts of the chamber can be tapped for analysis as shown below.

$$S_1 = I_{SR} + I_{spin} + \Delta I_z \tag{19}$$

$$S_2 = I_{SR} - I_{spin} + \Delta I_z \tag{20}$$

$$S_3 = I_{SR} - I_{spin} - \Delta I_z \tag{21}$$

$$S_4 = I_{SR} + I_{spin} - \Delta I_z \tag{22}$$

{Where I_{SR} is the current due to all SR Photons and I_{spin} is the current due to just the spin- light photons}.

The signal $(S_1 + S_2) - (S_3 + S_4)$ should always be zero ideally. The longitudinal asymmetry in terms of these 4 signals is given by;

$$A^{long} = \frac{I^{long}_{spin}}{I^{long}_{SR}} = \frac{(S_1 - S_2) - (S_3 + S_4)}{(S_1 + S_2) + (S_3 + S_4)} \tag{23}$$

and the transverse asymmetry in terms of these 4 signals is given by;

$$A^{trans} = \frac{I^{trans}_{spin}}{I^{trans}_{SR}} = \frac{(S_1 + S_3) - (S_2 + S_4)}{(S_1 + S_3) + (S_2 + S_4)} \tag{24}$$

One could in theory come up with many more electrode arrangements.

3.4.3 *Effects of Extended Beam Size*

In *Section 3.2*, the numerical code used a point beam. Therefore the effects of having extended beam size of about 100μm must be studied. To do this the code located in *Appendix B.3* was used. This code essentially superimposes the SR-Power spectra generated by each of a 10^6 such point-cross section beams. The million point - cross section beams together would give a circular beam and each of them was weighted with a Gaussian profile in order to make the extended beam a perfect Gaussian beam. The cumulative spectra can be plotted

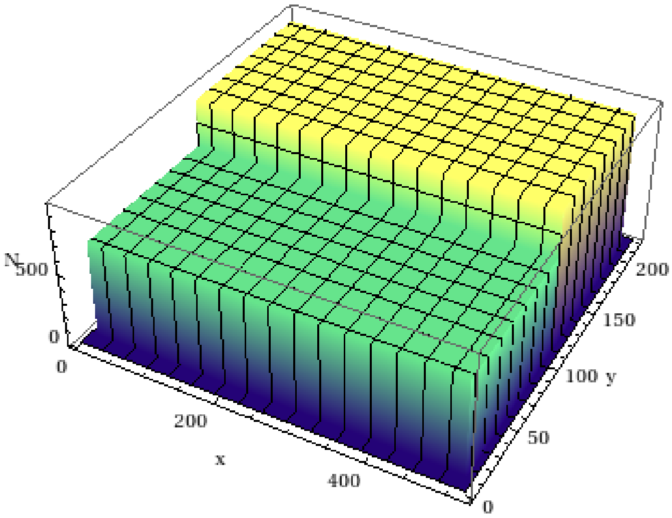


Figure 30: A 3D power spectra of SR Light at the IC due to a point-cross section beam - X,Y(10μm); N($\times 10^{12}$). (The difference between the profile has been enlarged for clarity)

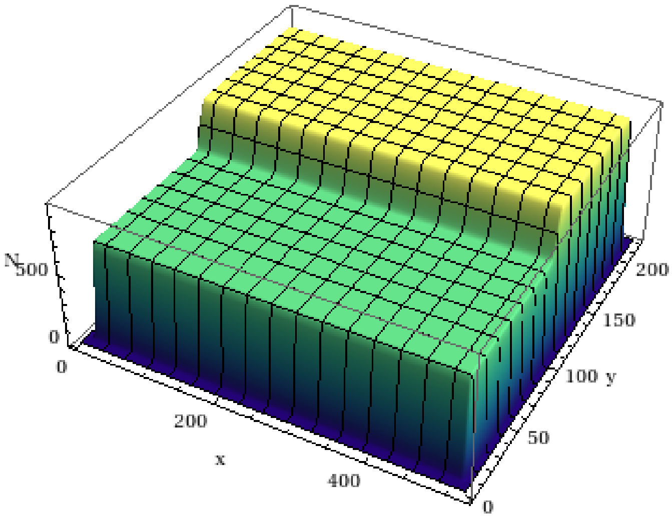


Figure 31: A 3D power spectra of SR Light at the IC due to a real beam of size ($R_{beam} = 100mum$) - X,Y(10μm); N($\times 10^{12}$). (The difference between the profile has been enlarged for clarity)

and one can guess that it should have the same structure as the original spectra for the point - cross section beam. This is so because the size of the beam ($R_{beam} = 100mum$) is small compared to the size of the collimated SR - Light spot which is about $1mm$ big. For the beam with a point cross section, the SR - profile is rather 'box' like at the IC. When an extended beam, that is Gaussian profile, is introduced, the SR - profile gets a taper which is Gaussian in nature too. The graphs inn Figure 30 and 31 show the exact 3D profile correct with position information.

3.4.4 *Current and Future Work: GEANT 4 Simulation*

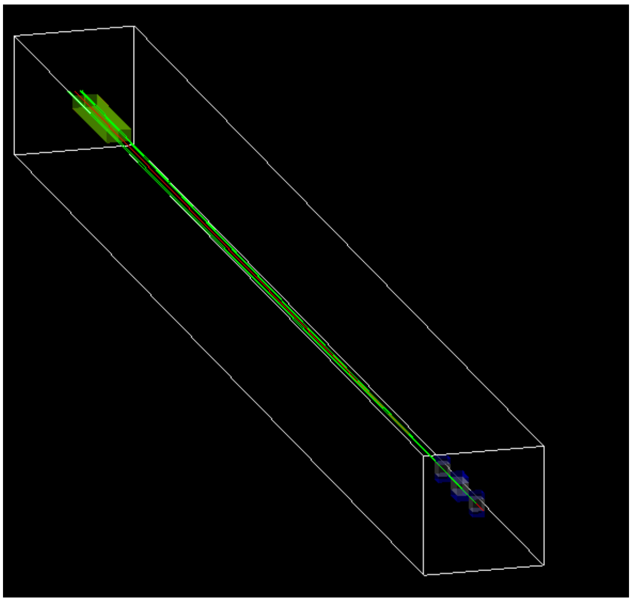


Figure 32: GEANT4 visualization of Spin - Light Polarimeter Setup. The electron beam is red in color and the SR Fans are yellow in color.

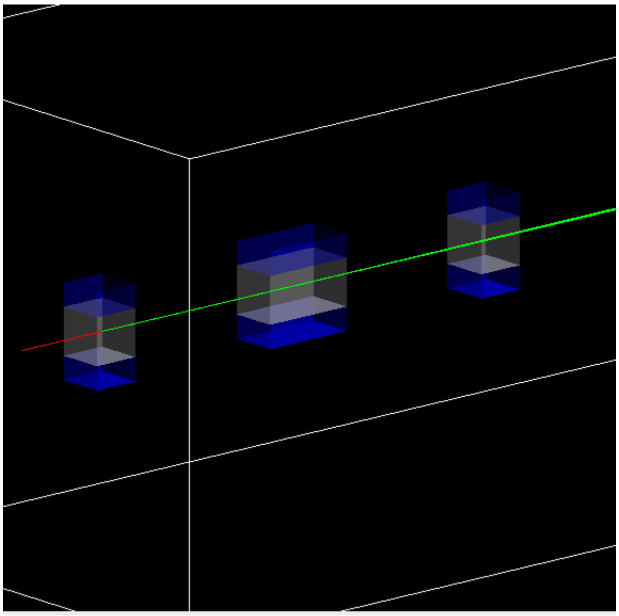


Figure 33: GEANT4 visualization of Wiggler Magnet Setup.

This work demands a GEANT 4 simulation in order to optimize the distance between the dipoles and to optimize the positioning of the collimators. Figure 32 is a GEANT4 visualization of the entire setup. It is important to note that the 2 fans are clearly visible on either side of the beam in the center. The dipoles are the blue blocks on one end

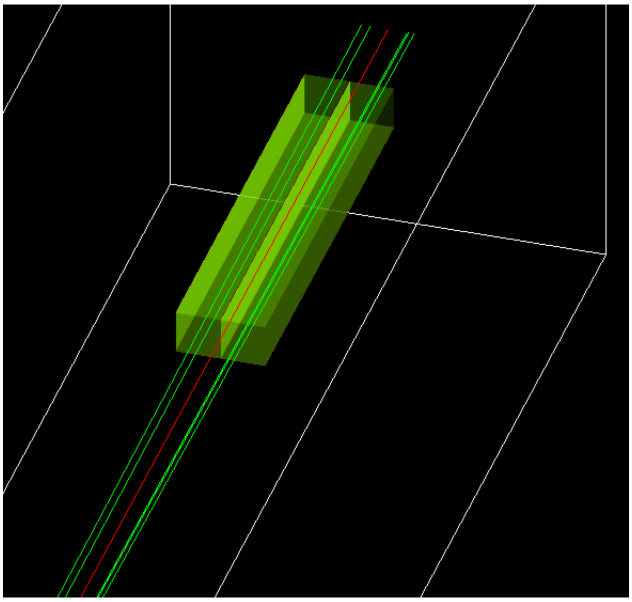


Figure 34: GEANT4 visualization of the 2 Ionization Chamber Setups (One on either side of the beam).

and the ICs are the green blocks on the other end. Work on this part is in progress and will be complete by 2013 May.

The GEANT-4 simulation includes processes such as EM suite including ionization and synchrotron radiation. Primitives are being used to count the number of ionization particles in the IC and the SR spectra produced by the wiggler setup. Further more, histograms are being written into the code to study the recovery time of the IC. Given that the present motion in EIC design is to use a Gatalin gun [23], the recovery time will need to be less than a second if every bunch of electron beam needs to be measured for polarization. This is not an issue since polarimetry can be done as an averaging measurement over many bunches.

[23] *V. N. Litvinenko, Gatling Gun: High Average Polarized Current Injector for eRHIC, EIC BNL Whitepapers (2012)*

4 SUMMARY

4.1 SYSTEMATICS

If the ionization chambers are used in differential mode and have split anodes, the false asymmetries will cancel to first order. Moreover, since the signal used is a differential signal the size of the background must be small compared to the signal. A full simulation is needed to study the background and the asymmetry associated with the background. In the experiment the background can be determined by monitoring the difference in the signal from the chambers with the wriggler magnets turned on and off. The other major source of systematic uncertainty is the lower bound of the integration window used to generate the IC signals . The absolute value of the spin light asymmetry depends on the absolute value of the energy window over which the IC signals are integrated. It is especially sensitive to the lower bound because of the steep fall of the SR intensity with energy. However given the excellent energy resolution that has been demonstrated for HPXe ionization chambers, one should be able to calibrate the chamber and determine the response function and the lower bound of the chamber to better than 2%. A preliminary table of estimated systematic uncertainties is shown in table below.

Source	Uncertainity	$\delta A/A$
Dark Current	pA	$< 0.01\%$
Intensity Fluctuations	$\Delta N \times 10^{-3}$	$< 0.10\%$
Beam Energy	$1.0\times$	$< 0.05\%$
Density of gas in IC	Relative uncertainties	$< 0.05\%$
Length of Chamber	Can be corrected	-
Band - width of X - Rays	2% (for only absolute polarimetry)	1.20%
Background related Dilutions	To be determined if known to 0.5%	$< 0.50\%$
Other dilutions	Cancel to First Order	$< 0.50\%$
Total	Relative Polarimetry	$< 0.68\%$
	Absolute Polarimetry	$< 1.88\%$

Table 1: Systematic uncertainties

4.2 CONCLUSION

Spin light based polarimetry was demonstrated over 30 years ago, but has been ignored since then. The figure of merit for such a polarimeter increases with electron beam energy and the strength of magnetic field used. The $11GeV$ beam at JLab is well suited for testing a spin light polarimetry and such a polarimeter would help achieve the $< 0.5\%$ polarimetry desired by experiments envisioned for the EIC era. A 3 pole wriggler with a field strength of $4T$ and a pole length of $10cm$ would be adequate for such a polarimeter. A dual position sensitive ionization chambers with split anode plates is ideally suited as the X-ray detector for such a polarimeter. The differential detector design would help reduce systematic uncertainties. Locating a reasonable piece of beam-line real estate is however very challenging.

A
LICENSE

GNU GENERAL PUBLIC LICENSE: All the programs are free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details.

B

NUMERICAL INTEGRATION CODE

B.1 NUMERICAL INTEGRATION OF THE SR - POWER LAW

```
C
C       PROGRAM Ngamma.f  (WRITTEN BY D. Dutta 1/7/2010)
C       and
C       Ngammaspectra.f by Prajwal Mohanmurthy (Sept 2011),
C       prajwal@jlab.org, Mississippi State University
C
C       To include effects from the Real Magnetic field
C       with non-zero gradient taper.
C
C       This program calculates the total number (and
C       difference in number above and below the orbital
C       plane) of Synchrotron photons emitted by longitudinally
C       polarized electrons over a horizontal angular range of
C       dTheta and verticle angular range of;
C                         +/-alpha = +/-gamma*Psi
C       where gamma is the Lorentz boost, i.e.;
C                         +/-alpha approx = +/-1
C       when traversing a 3 pole wriggler magnet with a field
C       stength of Bwg tesla and a pole length of Lwg.
C
C        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
        IMPLICIT real*8(A-H,O-Z)
        external fn01
        external fn02
        external gn01
        external gn02
        real*8 xheeng(48),xhemu(48),couB(15)
        real*8 couxn,coudn

        parameter (xMe=0.510998902) !electron mass in MeV/c^2
        parameter (GeV2MeV=1000.0)
        parameter (hbarc=197.3269602) ! MeV*fm
        parameter (xmuB=5.788381749E-11) ! Bohr magnetron MeV/T
        parameter (c=299792458) ! m/s
        parameter (pi=3.141592654)
        parameter (qe=1.602176462E-19) ! coulomb
        parameter (n1=10) ! number of times the integration alg
             is compounded
        parameter (n2=10)

        Common gamma,y

c        write(6,*)'Enter Ebeam (GeV) and current (micro A)'
c        read(5,*)Ebeam,xIe
c        write(6,*)'Wriggler B-field (T) and Pole length (m)'
c        read(5,*)Bwg,xLwg
```

```
      do i = 1,15
        couB(i)=4.55
      enddo
      couB(1)=4.11
      couB(2)=4.38
      couB(14)=4.38
      couB(15)=4.11
      open(unit=10,file="spinlight_gydep4.dat",status="
          unknown")
      open(unit=11,file="xenon.dat",status="old")
      do i =1,48
        read(11,*)xheeng(i),xhemu(i)
      enddo
      close(11)
      ymin = 0.01 ! min fractional photon freq (W/Wc)
      ymax=0.02 !initialize
c      Ebeam=4.0+(i-1)*1.0 ! e-beam enengy GeV
      Ebeam=11.0 ! e-beam enengy GeV
      xIe=100.0 ! e-beam current micro A
      do i=1,1000
       do j=1,15
        couxn=0.0
        coudn=0.0
        Bwg=couB(j)     ! B-field in T
c      Bwg=1.0+(i-1)*1.0
c      xLwg=0.066 ! pole length in m
c      write(6,*)'Ebeam =',Ebeam, 'GeV'
        gamma = Ebeam*GeV2MeV/xMe  ! Lorentz boost = E/(Me*c
            ^2)
        R_bend = gamma*hbarc*1.0E-15/(2.*xmuB*Bwg) !bending
            radius in m
        Omega_o = c/R_bend  ! betatron freq.
        Omega_c = 1.5*gamma**3*Omega_o ! central photon
            frequency
        E_cent=(Omega_c*hbarc*1.0E-15/c)*1000. ! central
            photon energy in keV
        xlambda_c=2.*pi*c/Omega_c
        ymax = 0.02+(i-1)*0.01 ! max fractional photon freq (
           W/Wc)
        y_cent= (ymin+ymax)/2.
        E_min = (ymin*Omega_c*hbarc*1.0E-15/c)*1000. ! min
            photon energy in keV
        E_max = (ymax*Omega_c*hbarc*1.0E-15/c)*1000. ! max
            photon energy in keV
        E_cent =(y_cent*Omega_c*hbarc*1.0E-15/c) ! photon
            energy in MeV
        xLwg = 0.0135 ! pole length in m
        dTheta = xLwg/R_bend ! horizontal angular range
        ij=2
        ik=1
        ift=1
        do ii=1,48
          if(E_cent.lt.xheeng(ii).and.ift.eq.1)then
            ij=ii
            ik=ii-1
            ift=ift+1
          endif
```

```
          enddo
          xrayabs=xhemu(ik)+
     > (((E_cent-xheeng(ik))/(xheeng(ij)-xheeng(ik)))*
     >                          (xhemu(ij)-xhemu(ik)))
c         absconst=0.023*(2./4.)*(2.*xlambda_c*1.0E+10/y_cent)
     **2.78
c         if (absconst.lt.0.1) absconst=0.1
c         write(6,*)E_cent,xrayabs
          absconst=xrayabs*(5.9/1000.)
          Amut=(1.0-exp(-absconst**0.5))
c         write(6,*)'Wriggler B-field and pole lengths =',Bwg,
     xLwg
c         write(6,*)'boost, central freq and bend radius=',gamma
     ,Omega_c,
c     1           R_bend
c         write(6,*)'vertical ang. range, min and max photon
     energy=',
c     1 dTheta, E_min, E_max,xlambda_c,absconst,y_cent,Amut,E_
     cent
c         Psi_min = -asin(1./gamma) ! min vertical angle
c         Psi_max = asin(1./gamma) ! max vertical angle
c         alpha_min = gamma*Psi_min ! boosted min vertical angle
c         alpha_max = gamma*Psi_max ! boosted max vertical angle
c         z_min = (ymin/2.)*(1.+alpha_min**2.)**1.5 ! z=(y/2)
     *(1+alpha)^1.5
c         z_max = (ymax/2.)*(1.+alpha_max**2.)**1.5
c         write(6,*)alpha_min,alpha_max,z_min,z_max
          xNe=xIe*1.0E-06/qe ! # of electrons
          xi=1.5*gamma**2*hbarc*1.0E-15/xMe/R_bend ! crit
              parameter
          xHbyHo=gamma*hbarc*1.0E-15/xMe/R_bend
          tau=(8.*sqrt(3.)/15.)*(hbarc*1.0E-15/xMe/c) ! use
              hbarc/qe^2 =137
     1        *(1./xHbyHo)**3*(1./gamma**2.)*137.0
          sflip=hbarc*(xLwg/c)*(1.+8.*sqrt(3.)/15.)*0.5*1./tau
c         write(6,*)'sflip probability',sflip
          const1=3.*xNe*gamma*dTheta/(4.*pi**2.*137.)
          const2=4.*const1*xi
          call p3pgs(ymin,ymax,n1,fn01,gn01,vint1) !
              integrations for N
          call p3pgs(ymin,ymax,n2,fn02,gn02,vint2) ! integraton
              for Delta N
          xn=const1*vint1
          dn=const2*vint2
          xa=(dn/xn)*sqrt(2.*xn)
          xpow=xn*E_cent*1.6E-19*1.0E+6! power released in W
          dpow1=dn*E_cent*1.6E-19*1.0E+10 ! power of spin light
              in W
c         write(6,*)'edep',xdpow !Ebeam,gamma,dTheta,gamma*
     dTheta,const1,vint1
c         write(6,*)'photon energy, #of photons, up/dn diff,
     assym,
c     1 analyzing pwr, photons abs'
c         write(6,15)(E_min+E_max)/2000.,xn,dn,dn/xn,xa,xpow,
     Amut*xn
c         y_cent = E/E_max (do not add), E_cent: bi central
     energy(do not add), xn: tot. num. events (add),dn:Events
```

```
    from Spin light(add), dn/xn = ratio, xa= events above -
    events below/ sum, xpow: integrated per bin, Amut*xn:
    signal size
          couxn = couxn + xn
          coudn = coudn + dn
        enddo
        xn = couxn
        dn = coudn
        write(10,15)y_cent,E_cent,xn,dn,dn/xn,xa,xpow,Amut*xn
        ymin=ymax
      enddo
      close (10)
 15   format(1x,f6.3,1x,f6.3,1x,e15.3,1x,e15.3,1x,e15.3,1x,e
   15.3,1x,
   1        e15.3,1x,e15.3)
      END

      SUBROUTINE IKV(V,X,VM,BI,DI,BK,DK)
C
C
    ====================================================
C       Purpose: Compute modified Bessel functions Iv(x) and
C                Kv(x), and their derivatives
C       Input :  x --- Argument ( x > 0 )
C                v --- Order of Iv(x) and Kv(x)
C                      ( v = n+v0, n = 0,1,2,..., 0 < v0 < 1 )
C       Output:  BI(n) --- In+v0(x)
C                DI(n) --- In+v0'(x)
C                BK(n) --- Kn+v0(x)
C                DK(n) --- Kn+v0'(x)
C                VM --- Highest order computed
C       Routines called:
C            (1) GAMMA for computing the gamma function
C            (2) MSTA1 and MSTA2 to compute the starting
C                point for backward recurrence
C
    ====================================================
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION BI(0:*),DI(0:*),BK(0:*),DK(0:*)
        PI=3.141592653589793D0
        X2=X*X
        N=INT(V)
        V0=V-N
        IF (N.EQ.0) N=1
        IF (X.LT.1.0D-100) THEN
           DO 10 K=0,N
              BI(K)=0.0D0
              DI(K)=0.0D0
              BK(K)=-1.0D+300
 10           DK(K)=1.0D+300
           IF (V.EQ.0.0) THEN
              BI(0)=1.0D0
              DI(1)=0.5D0
           ENDIF
           VM=V
           RETURN
```

```
          ENDIF
          PIV=PI*V0
          VT=4.0D0*V0*V0
          IF (V0.EQ.0.0D0) THEN
             A1=1.0D0
          ELSE
             V0P=1.0D0+V0
             CALL GAMMA(V0P,GAP)
             A1=(0.5D0*X)**V0/GAP
          ENDIF
          K0=14
          IF (X.GE.35.0) K0=10
          IF (X.GE.50.0) K0=8
          IF (X.LE.18.0) THEN
             BI0=1.0D0
             R=1.0D0
             DO 15 K=1,30
                R=0.25D0*R*X2/(K*(K+V0))
                BI0=BI0+R
                IF (DABS(R/BI0).LT.1.0D-15) GO TO 20
15           CONTINUE
20           BI0=BI0*A1
          ELSE
             CA=DEXP(X)/DSQRT(2.0D0*PI*X)
             SUM=1.0D0
             R=1.0D0
             DO 25 K=1,K0
                R=-0.125D0*R*(VT-(2.0D0*K-1.0D0)**2.0)/(K*X)
25              SUM=SUM+R
             BI0=CA*SUM
          ENDIF
          M=MSTA1(X,200)
          IF (M.LT.N) THEN
             N=M
          ELSE
             M=MSTA2(X,N,15)
          ENDIF
          F2=0.0D0
          F1=1.0D-100
          DO 30 K=M,0,-1
             F=2.0D0*(V0+K+1.0D0)/X*F1+F2
             IF (K.LE.N) BI(K)=F
             F2=F1
30           F1=F
          CS=BI0/F
          DO 35 K=0,N
35           BI(K)=CS*BI(K)
          DI(0)=V0/X*BI(0)+BI(1)
          DO 40 K=1,N
40           DI(K)=-(K+V0)/X*BI(K)+BI(K-1)
          IF (X.LE.9.0D0) THEN
             IF (V0.EQ.0.0D0) THEN
                CT=-DLOG(0.5D0*X)-0.5772156649015329D0
                CS=0.0D0
                W0=0.0D0
                R=1.0D0
                DO 45 K=1,50
```

```
                     W0=W0+1.0D0/K
                     R=0.25D0*R/(K*K)*X2
                     CS=CS+R*(W0+CT)
                     WA=DABS(CS)
                     IF (DABS((WA-WW)/WA).LT.1.0D-15) GO TO 50
 45                  WW=WA
 50              BK0=CT+CS
             ELSE
                 V0N=1.0D0-V0
                 CALL GAMMA(V0N,GAN)
                 A2=1.0D0/(GAN*(0.5D0*X)**V0)
                 A1=(0.5D0*X)**V0/GAP
                 SUM=A2-A1
                 R1=1.0D0
                 R2=1.0D0
                 DO 55 K=1,120
                     R1=0.25D0*R1*X2/(K*(K-V0))
                     R2=0.25D0*R2*X2/(K*(K+V0))
                     SUM=SUM+A2*R1-A1*R2
                     WA=DABS(SUM)
                     IF (DABS((WA-WW)/WA).LT.1.0D-15) GO TO 60
 55                  WW=WA
 60              BK0=0.5D0*PI*SUM/DSIN(PIV)
             ENDIF
         ELSE
             CB=DEXP(-X)*DSQRT(0.5D0*PI/X)
             SUM=1.0D0
             R=1.0D0
             DO 65 K=1,K0
                 R=0.125D0*R*(VT-(2.0*K-1.0)**2.0)/(K*X)
 65              SUM=SUM+R
             BK0=CB*SUM
         ENDIF
         BK1=(1.0D0/X-BI(1)*BK0)/BI(0)
         BK(0)=BK0
         BK(1)=BK1
         DO 70 K=2,N
             BK2=2.0D0*(V0+K-1.0D0)/X*BK1+BK0
             BK(K)=BK2
             BK0=BK1
 70          BK1=BK2
         DK(0)=V0/X*BK(0)-BK(1)
         DO 80 K=1,N
 80          DK(K)=-(K+V0)/X*BK(K)-BK(K-1)
         VM=N+V0
         RETURN
         END


         SUBROUTINE GAMMA(X,GA)
C
C        ==================================================
C        Purpose: Compute gamma function a(x)
C        Input :  x  --- Argument of a(x)
C                        ( x is not equal to 0,-1,-2,uuu)
C        Output:  GA --- a(x)
C        ==================================================
```

```
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        DIMENSION G(26)
        PI=3.141592653589793D0
        IF (X.EQ.INT(X)) THEN
           IF (X.GT.0.0D0) THEN
              GA=1.0D0
              M1=X-1
              DO 10 K=2,M1
10               GA=GA*K
           ELSE
              GA=1.0D+300
           ENDIF
        ELSE
           IF (DABS(X).GT.1.0D0) THEN
              Z=DABS(X)
              M=INT(Z)
              R=1.0D0
              DO 15 K=1,M
15               R=R*(Z-K)
              Z=Z-M
           ELSE
              Z=X
           ENDIF
           DATA G/1.0D0,0.5772156649015329D0,
     &          -0.6558780715202538D0, -0.420026350340952D-1,
     &          0.1665386113822915D0,-.421977345555443D-1,
     &          -.9621971527877D-2, .7218943246663D-2,
     &          -.11651675918591D-2, -.2152416741149D-3,
     &          .1280502823882D-3, -.201348547807D-4,
     &          -.12504934821D-5, .11330272320D-5,
     &          -.2056338417D-6, .61160950D-8,
     &          .50020075D-8, -.11812746D-8,
     &          .1043427D-9, .77823D-11,
     &          -.36968D-11, .51D-12,
     &          -.206D-13, -.54D-14, .14D-14, .1D-15/
           GR=G(26)
           DO 20 K=25,1,-1
20            GR=GR*Z+G(K)
           GA=1.0D0/(GR*Z)
           IF (DABS(X).GT.1.0D0) THEN
              GA=GA*R
              IF (X.LT.0.0D0) GA=-PI/(X*GA*DSIN(PI*X))
           ENDIF
        ENDIF
        RETURN
        END


        INTEGER FUNCTION MSTA1(X,MP)
C
C       ===================================================
C       Purpose: Determine the starting point for backward
C                recurrence such that the magnitude of
C                Jn(x) at that point is about 10^(-MP)
C       Input :  x     --- Argument of Jn(x)
C                MP    --- Value of magnitude
```

```
C       Output:  MSTA1 --- Starting point
C       ==================================================
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        A0=DABS(X)
        N0=INT(1.1*A0)+1
        F0=ENVJ(N0,A0)-MP
        N1=N0+5
        F1=ENVJ(N1,A0)-MP
        DO 10 IT=1,20
           NN=N1-(N1-N0)/(1.0D0-F0/F1)
           F=ENVJ(NN,A0)-MP
           IF(ABS(NN-N1).LT.1) GO TO 20
           N0=N1
           F0=F1
           N1=NN
 10        F1=F
 20     MSTA1=NN
        RETURN
        END


        INTEGER FUNCTION MSTA2(X,N,MP)
C
C       ==================================================
C       Purpose: Determine the starting point for backward
C                recurrence such that all Jn(x) has MP
C                significant digits
C       Input :  x  --- Argument of Jn(x)
C                n  --- Order of Jn(x)
C                MP --- Significant digit
C       Output:  MSTA2 --- Starting point
C       ==================================================
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        A0=DABS(X)
        HMP=0.5D0*MP
        EJN=ENVJ(N,A0)
        IF (EJN.LE.HMP) THEN
           OBJ=MP
           N0=INT(1.1*A0)
        ELSE
           OBJ=HMP+EJN
           N0=N
        ENDIF
        F0=ENVJ(N0,A0)-OBJ
        N1=N0+5
        F1=ENVJ(N1,A0)-OBJ
        DO 10 IT=1,20
           NN=N1-(N1-N0)/(1.0D0-F0/F1)
           F=ENVJ(NN,A0)-OBJ
           IF (ABS(NN-N1).LT.1) GO TO 20
           N0=N1
           F0=F1
           N1=NN
10         F1=F
20      MSTA2=NN+10
```

```
          RETURN
          END

          REAL*8 FUNCTION ENVJ(N,X)
          DOUBLE PRECISION X
          ENVJ=0.5D0*DLOG10(6.28D0*N)-N*DLOG10(1.36D0*X/N)
          RETURN
          END


       SUBROUTINE P3PGS ( A, B, N, FN, GN, VINT )

c
     *********************************************************************72

C
C   THIS SUBROUTINE USES THE PRODUCT TYPE THREE-POINT GAUSS-
C   LEGENDRE-SIMPSON RULE COMPOUNDED N TIMES TO APPROXIMATE
C   THE INTEGRAL FROM A TO B OF THE FUNCTION FN(X) * GN(X).
C   FN AND GN ARE FUNCTION SUBPROGRAMS WHICH MUST BE SUPPLIED
C   BY THE USER.  THE RESULT IS STORED IN VINT.
C
       DOUBLE PRECISION A, AG, AM(2,3), B, F(2), FN, G(3),
     & GN, H, VINT, X(2), Y(2), DBLE

       DATA AM(1,1), AM(2,3) / 2 * 1.718245836551854D0 /,
     & AM(1,2), AM(2,2) / 2 * 1.D0 /, AM(1,3), AM(2,1)
     & / 2 * -.2182458365518542D0 /

       H = ( B - A ) / DBLE ( FLOAT ( N ) )
       X(1) = A + .1127016653792583D0 * H
       X(2) = A + .8872983346207417D0 * H
       Y(1) = A + H / 2.D0
       Y(2) = A + H
       VINT = 0.D0
       G(3) = GN ( A )
       DO 3 I = 1, N
         AG = FN ( Y(1) )
         G(1) = G(3)
         DO 1 J = 1, 2
           F(J) = FN ( X(J) )
           G(J+1) = GN ( Y(J) )
           X(J) = X(J) + H
1          Y(J) = Y(J) + H
         VINT = VINT + AG * 4.D0 * G(2)
         DO 3 J = 1, 2
           AG = 0.D0
           DO 2 K = 1, 3
2            AG = AG + AM(J,K) * G(K)
3          VINT = VINT + F(J) * AG
       VINT = H * VINT / 9.D0

       RETURN
       END
```

```
      function fn01(x)
      implicit none

      integer n

      double precision fn01
      double precision x

      fn01 = x

      return
      end

      function fn02(x)
      implicit none

      integer n

      double precision fn02
      double precision x

      fn02 = x**2.

      return
      end

      function gn01(x)
      implicit real*8(A-H,O-Z)

c      real*8 Psi_min,Psi_max,alpha_min,alpha_max,y,vint3
c      real*8 gamma

c      integer n
      external fn03
      external gn03

c      double precision gn01
c      double precision x

      parameter (n=20)

      common gamma,y

      Psi_min = -asin(1./gamma) ! min vertical angle
      Psi_max = asin(1./gamma) ! max vertical angle
      alpha_min = 2.*gamma*Psi_min ! boosted min vertical
         angle
      alpha_max = 2.*gamma*Psi_max ! boosted max vertical
         angle
      alpha_cutoffm=-0.16
      alpha_cutoffp=0.16
      y=x
      call p3pgs(alpha_min,alpha_max,n,fn03,gn03,vint31)
c      call p3pgs(alpha_min,alpha_cutoffm,n,fn03,gn03,vint31)
c      call p3pgs(alpha_cutoffp,alpha_max,n,fn03,gn03,vint32)
      gn01 = vint31 !+vint32
```

```
      return
      end

      function gn02(x)
      implicit none

      real*8 Psi_min,Psi_max,alpha_min,alpha_max,y,vint4
      real*8 gamma

      integer n
      external fn04
      external gn04

      double precision gn02
      double precision x
      parameter (n=20)
      common gamma,y

      Psi_min = -asin(1./gamma) ! min vertical angle
      Psi_max = asin(1./gamma) ! max vertical angle
      alpha_min = 0. !16 !for the diff the int is from alpha_
          cutoff to alpha_max
      alpha_max = 2.*gamma*Psi_max ! boosted max vertical
          angle
      y=x
      call p3pgs(alpha_min,alpha_max,n,fn04,gn04,vint4)
      gn02 = vint4

      return
      end

      function fn03(x)
      implicit none


      integer n

      double precision fn03
      double precision x


      fn03 = (1+x**2.)**2.

      return
      end

      function fn04(x)
      implicit none


      integer n

      double precision fn04
      double precision x


      fn04 = x*(1+x**2.)**1.5
```

```
      return
      end

      function gn03(x)
      implicit real*8 (A-H,O-Z)

c       real*8 z,v,k23,k13,gamma,y,vm
c       dimension BI(0:*),DI(0:*),BK(0:*),DK(0:*)

c       integer n

      double precision gn03
      double precision x
      common gamma,y
      COMMON BI(0:250),DI(0:250),BK(0:250),DK(0:250)

      xk23=0.
      xk13=0.
      z=(y/2.)*(1+x**2)**1.5 ! z= (omega/2omega_c)*(1+alpha^2)^3/2
c       write(6,*)'gn03 gamma,y,x,z',gamma,y,x,z
      v=2./3.
      CALL IKV(V,z,VM,BI,DI,BK,DK)
      xk23=BK(0)

      v=1./3.
      CALL IKV(V,z,VM,BI,DI,BK,DK)
      xk13=BK(0)
c       write(6,*)'g03 k23 k13',xk23,xk13
      gn03 = xk23**2. + x**2.*xk13**2/(1+x**2.)

      return
      end

      function gn04(x)
      implicit real*8(A-H,O-Z)

c       real*8 z,v,k23,k13,y,vm,gamma
c       Dimension BI(0:*),DI(0:*),BK(0:*),DK(0:*)

c       integer n

c       double precision gn04
c       double precision x
      Common gamma,y
      COMMON BI(0:250),DI(0:250),BK(0:250),DK(0:250)

      z=(y/2.)*(1+x**2)**1.5
      v=2./3.
      CALL IKV(V,z,VM,BI,DI,BK,DK)
      xk23=BK(0)

      v=1./3.
      CALL IKV(V,z,VM,BI,DI,BK,DK)
      xk13=BK(0)
```

```
      gno4 = xk23*xk13

      return
      end
```

B.2 LANL POISSON SUPEFISH GEOMETRY DESCRIPTION

Listing 1: LANL Poisson SupeFish Geometry Description

```
Dipole Magnet Problem
DIPOLE 6 Simulation of Spin Light Chicane Magnet

; Copyright 1987, by the University of California.
; Unauthorized commercial use is prohibited.
; Author: Prajwal Mohanmurthy; Dec, 2011; (prajwal@jlab.org)

&reg kprob=0,         ! Poisson or Pandira problem
mode=0,               ! Some materials have variable
    permeability
nbsup=0,              ! added by gbf  !
nbslf=0,
rhogam=0.001          !try this gbf
xreg1=10.0,kreg1=82,! Physical and logical coordinates of x
xreg2=25.0,kreg2=98,
xreg3=48.0,kreg3=150,

yreg1=14.0,lreg1=65,! Y line regions
yreg2=18.0,lreg2=70,
yreg3=21.0,lreg3=74,
lmax=80 &

&po x= −30.0000,y= 0.0000 &
&po x=60.000,y= 0.0000&
&po x=60.000,y=30.0000 &
&po x= −30.0000,y=30.0000 &
&po x= −30.0000,y= 0.0000 &

&reg mat=1,cur=−1040000.0 &
&po x=30.5000,y=6.000 &
&po x=40.0000,y=6.000 &
&po x=40.0000,y=12.000 &
&po x=30.5000,y=12.000 &
&po x=30.5000,y=6.000 &

&reg mat=1,cur=1040000.0 &
&po x=−10.0000,y=6.000 &
&po x=−0.5000,y=6.000 &
&po x=−0.5000,y=12.000 &
&po x=−10.000,y=12.000 &
&po x=−10.000,y=6.000 &

&reg mat=3,mtid=−2, mshape=0 &
&po x= 0.00,y=  5.62 &
&po x= 1.50,y=  2.62 &
&po x= 3.50,y=  1.27 &
```

```
&po x=26.50,y=  1.27 &
&po x=28.50,y=  2.65 &
&po x=30.00,y=  5.62 &
&po x=30.00,y= 12.50 &
&po x=40.50,y= 12.60 &
&po x=40.50,y=  0.00 &
&po x=60.00,y=  0.00 &
&po x=60.00,y= 30.00 &
&po x= 0.00,y= 30.00 &
&po x= 0.00,y=  5.62 &

&mt mtid=1
bgam=0.00000   0.0017513135
900      0.001747079
950      0.001741742
1000     0.001735498
1050     0.001728309
1100     0.00172014
1150     0.001710963
1200     0.001700753
1250     0.001689494
1300     0.001677174
1350     0.001663786
2800     0.001080694
2850     0.001068051
2900     0.001056142
2950     0.001044912
3000     0.001034309
3050     0.001024289
3100     0.001014809
3150     0.001005828
3200     0.000997312
3250     0.000989226
3300     0.000981539
3350     0.000974222
4000     0.000904952
4500     0.000856798
5000     0.000818493
5500     0.000788085
6000     0.000764202
6500     0.000745863
7000     0.000732376
7500     0.000723261
8000     0.000718209
8500     0.000717054
9000     0.000719758
9500     0.000726411
10000    0.000737231
10500    0.000752594
10578    0.0007562580
11319    0.0007951022
11940    0.0008375209
12451    0.0008834703
12912    0.0009293680
13313    0.0009764671
13654    0.0010253255
13935    0.0010764263
```

```
14216    0.0011254924
14447    0.0011767475
14618    0.0012313603
14789    0.0012846865
15020    0.0013315579
15131    0.0013879251
15252    0.0014423770
15432    0.0014912019
15594    0.0015389351
15705    0.0015918497
16180    0.0018542555
16840    0.0023752969
17150    0.0029154519
17360    0.0034566194
17620    0.0039729837
17830    0.0044863167
18200    0.0054945055
18950    0.0079176564
19500    0.0102564103
20200    0.0148588410
20650    0.0193798450
20950    0.0238663484
21600    0.0370370370
21900    0.0456621005
23000    0.0869565217
23386    0.1002810000
23850    0.1181630000
24408    0.1387420000
25079    0.1622460000
25885    0.1888580000
26854    0.2186950000
28019    0.2517840000 &
```

B.3 RECURSIVE SR SPECTRA ADDING CODE

Listing 2: LANL Poisson SupeFish Geometry Description

```
!Recurssive SR - Spectra Adding Code
!Author: Prajwal Mohanmurthy, Dec 2011 (prajwal@jlab.org)
! Mathematica 7 File
up=30;
down=25;
asym=5;
sigup=up/E;
sigdown=down/E;
width=200; (*100+100*)
mid=width/2;
motion=10;
scale=0;
If[up-asym!= down,Print["Check Var : asym, down, up"]]
sr=Table[Table[0,{k,1,width+(2 motion)}],{kk,1,21}]; (*Set 'X'
    width =21*)
(*Print["--------------------"];*)
For[xx=1,xx<=21,xx++,
count=0;
```

```
count2=0;
x=xx-1;
For[y=0,If[x<=10,y<= x,y<=10-Abs[10-x] ],y++,
count++;
(*Print[x, ":x | y:",y];*)
For[k=motion+1+y,k<=motion+mid+y,k++,
sr[[xx]][[k]]+=Floor[(down*E^(-(((x-10)^2+(y)^2)/sigdown)))];
scale+=E^(-(((x-10)^2+(y)^2)/sigdown));
(*Print["Adding ",Floor[(down*E^(-(((x-10)^2+(y)^2)/sigdown)))
    ]," to ",k];*)
];
For[k=motion+mid+1+y,k<=motion+width+y,k++,
sr[[xx]][[k]]+=Floor[(up*E^(-(((x-10)^2+(y)^2)/sigup )))];
scale+=E^(-(((x-10)^2+(y)^2)/sigdown));
(*Print["Adding ",Floor[(up*E^(-(((x-10)^2+(y)^2)/sigup )))],"
     to ",k];*)
];
];
(*Print["------------------------"];*)
For[y=0,If[x<= 10,y< x,y<10-Abs[10-x] ],y++,
count2++;
(*Print[x, ":x | y:",-(y+1)];*)
For[k=motion+1+-(y+1),k<=motion+mid-(y+1),k++,
sr[[xx]][[k]]+=Floor[(down*E^(-(((x-10)^2+(y+1)^2)/sigdown)))
    ];
scale+=E^(-(((x-10)^2+(y)^2)/sigdown));
(*Print["Adding ",Floor[(down*E^(-(((x-10)^2+(y+1)^2)/sigdown)
    ))]," to ",k];*)
];
For[k=motion+mid+1-(y+1),k<=motion+width-(y+1),k++,
sr[[xx]][[k]]+=Floor[(up*E^(-(((x-10)^2+(y+1)^2)/sigup )))];
scale+=E^(-(((x-10)^2+(y)^2)/sigdown));
(*Print["Adding ",Floor[(up*E^(-(((x-10)^2+(y+1)^2)/sigup )))
    ]," to ",k];*)
];
];
(*Print["------------------------"];*)
(*Print[count,"\t",count2,"\t",count+count2];*)
(*Print["------------------------"];*)
];
N[scale]
(*ListPlot3D[sr,PlotRange->Full]*)
400.023
nsize=520;
nwm=nsize-(2*motion);
Dimensions[sr];
sr2=Table[Table[0,{k,1,220}],{k,1,nsize}];
Dimensions[sr2]
Dimensions[sr2];
For[i2=1,i2<=21,i2++,
For[j2=1,j2<=220,j2++,
For[k2=i2,k2<nwm+i2,k2++,
sr2[[k2]][[j2]]+=sr[[i2]][[j2]];
];
];
];
```

```
Export["3dcontour_manypt.png",ListPlot3D[Transpose[sr2],
    ColorFunction->"BlueGreenYellow",AxesLabel->{"x","y","N"},
    PlotRange->Full]]
{520,220}
3dcontour_manypt.png
sr3=Table[Table[0,{k,1,220}],{k,1,nsize}];
Dimensions[sr3];
For[i2=11,i2<=110,i2++,
For[j2=11,j2<=nwm+motion,j2++,
sr3[[j2]][[i2]]=sr2[[nsize/4]][[50]];
];
];
For[i2=111,i2<=210,i2++,
For[j2=11,j2<=nwm+motion,j2++,
sr3[[j2]][[i2]]=Max[sr2];
];
];
Export["3dcontour_onept2.png",ListPlot3D[Transpose[sr3],
    AxesLabel->{"x","y","N"},ColorFunction->"BlueGreenYellow",
    PlotRange->Full]]
3dcontour_onept2.png
Clear[write2]
write2=OpenWrite["srXYdisc1.dat"]
WriteString[write2,"# x (micro m)" ,"|" ,"y (micro m)","|","N"
    ,"\n"];
For[i=1,i<=220,i++,
For[j=1,j<=nsize,j++,
WriteString[write2,Floor[20 j] ,"\t" ,Floor[5 i],"\t",Floor[
    sr2[[j]][[i]]],"\n"];
];
]
Close[write2]
OutputStream[srXYdisc1.dat,36]
srXYdisc1.dat
Clear[write1]
write1=OpenWrite["srXYdiscone1.dat"]
WriteString[write1,"# x (micro m)" ,"|" ,"y (micro m)","|","N"
    ,"\n"];
For[i=1,i<=220,i++,
For[j=1,j<=nsize,j++,
WriteString[write1,Floor[20 j] ,"\t" ,Floor[5 i],"\t",Floor[
    sr3[[j]][[i]]],"\n"];
];
]
Close[write1]
OutputStream[srXYdiscone1.dat,37]
srXYdiscone1.dat
```

For the complete program, refer to `mohanmurthy.com/a/SpinIC.gz.tar`

B.4 GEANT4 GEOMETRY FILE

Listing 3: GEANT4 Geometry File

```
//
// ****************************************
```

```
// * License and Disclaimer *
// * *
// * The Geant4 software is copyright of the Copyright Holders of *
// * the Geant4 Collaboration. It is provided under the terms and *
// * conditions of the Geant4 Software License, included in the file *
// * LICENSE and available at http://cern.ch/geant4/license . These *
// * include a list of copyright holders. *
// * *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work make any representation or warranty, express or implied, *
// * regarding this software system or assume any liability for its *
// * use. Please see the license in the file LICENSE and URL above *
// * for the full disclaimer and the limitation of liability. *
// * *
// * This code implementation is the result of the scientific and *
// * technical work of the GEANT4 collaboration. *
// * By using, copying, modifying or distributing the software (or *
// * any work based on the software) you agree to acknowledge its *
// * use in resulting scientific publications, and indicate your *
// * acceptance of all terms of the Geant4 Software license. *
// ****************************************
//
//
// $Id: Em10DetectorConstruction.cc,v 1.32 2007-07-27 17:52:04 vnivanch Exp $
// GEANT4 tag $Name: geant4-09-04-patch-01 $
//
//

#include "Em10DetectorConstruction.hh"
#include "Em10DetectorMessenger.hh"
#include "Em10CalorimeterSD.hh"
```

```
#include "Em10Materials.hh"

#include "G4Material.hh"
#include "G4Box.hh"
#include "G4LogicalVolume.hh"
#include "G4PVPlacement.hh"
#include "G4UniformMagField.hh"
#include "G4FieldManager.hh"
#include "G4PropagatorInField.hh"
#include "G4TransportationManager.hh"
#include "G4SDManager.hh"
#include "G4GeometryManager.hh"
#include "G4RunManager.hh"
#include "G4UserLimits.hh"

#include "G4MagneticField.hh"
#include "G4Mag_UsualEqRhs.hh"
#include "G4ChordFinder.hh"
#include "G4ClassicalRK4.hh"
#include "G4RKG3_Stepper.hh"
#include "G4VisAttributes.hh"

#include "G4Region.hh"
#include "G4RegionStore.hh"
#include "G4PhysicalVolumeStore.hh"
#include "G4LogicalVolumeStore.hh"
#include "G4SolidStore.hh"
#include "G4ProductionCuts.hh"

#include "G4VisAttributes.hh"
#include "G4Colour.hh"

#include "G4UnitsTable.hh"
#include "G4ios.hh"
#include "G4Element.hh"

/////////
//
// Vacuum
  G4double density     = universe_mean_density;
                    //from PhysicalConstants.h
  G4double pressure    = 1.e-19*pascal;
  G4double temperature = 0.1*kelvin;
  G4double a = 1.01*g/mole;
  G4double z = 1.;
  G4Material* vacuum = new G4Material("vacuum", z, a, density,
       kStateGas,temperature,pressure);

/////////////////////////////////
// Magnets Description

void Em10DetectorConstruction::magnets()
{

  // Magnets Generic variables
  G4Material* fFe = fMat->GetMaterial("Iron");
  G4bool propagateToDaughters = true;
```

```
G4double fMinStep = 0.001*mm; // minimal step of 1 mm is default
G4double stdfield = 1.0*tesla;
G4VisAttributes* colorends = new G4VisAttributes(G4Colour(0.0, 0.0, 1., 0.5));
G4VisAttributes* colormid = new G4VisAttributes(G4Colour(1.0, 1.0, 1.0, 0.3));
colorends->SetForceSolid(true);
colormid->SetForceSolid(true);

//####### 1st magnet############

G4Box*               fmagnetonetop;
G4LogicalVolume*     fLogicmagnetonetop;
G4VPhysicalVolume*   fPhysicsmagnetonetop;

G4Box*               fmagnetonebot;
G4LogicalVolume*     fLogicmagnetonebot;
G4VPhysicalVolume*   fPhysicsmagnetonebot;

G4Box*               fmagnetonemid;
G4LogicalVolume*     fLogicmagnetonemid;
G4VPhysicalVolume*   fPhysicsmagnetonemid;

G4UniformMagField*  magFieldone;
G4FieldManager*  fieldMgrone;
G4Mag_UsualEqRhs*  fLocalEquationone;
G4MagIntegratorStepper*  fLocalStepperone;
G4ChordFinder*  fLocalChordFinderone(0);

// top end plates

fmagnetonetop = new G4Box("magnet1t", 5.0*cm, 2.5*cm, 5.0*cm);

fLogicmagnetonetop = new G4LogicalVolume(fmagnetonetop, fFe, "magnet1t");

fPhysicsmagnetonetop = new G4PVPlacement(0, G4ThreeVector(0,7.5*cm,-5.80*m), "magnet1t", fLogicmagnetonetop, fPhysicsWorld, false,    0 );

// bottom end plates

fmagnetonebot = new G4Box("magnet1b", 5.0*cm, 2.5*cm, 5.0*cm);

fLogicmagnetonebot = new G4LogicalVolume(fmagnetonebot, fFe, "magnet1b");

fPhysicsmagnetonebot = new G4PVPlacement(0, G4ThreeVector(0,-7.5*cm,-5.80*m), "magnet1b", fLogicmagnetonebot, fPhysicsWorld, false,   0 );

//Middle field area geometry
fmagnetonemid = new G4Box("magnet1m", 5.0*cm, 5.0*cm, 5.0*cm);
```

```
fLogicmagnetonemid = new G4LogicalVolume(fmagnetonemid,
    vacuum, "magnet1m");

fPhysicsmagnetonemid = new G4PVPlacement(0, G4ThreeVector
    (0,0,-5.80*m), "magnet1m", fLogicmagnetonemid,
    fPhysicsWorld, false, 0 );

//Middle-local field manager

magFieldone = new G4UniformMagField(G4ThreeVector(0.,
    stdfield,0.));

fieldMgrone = new G4FieldManager();
fieldMgrone->SetDetectorField(magFieldone);

fLogicmagnetonemid->SetFieldManager(fieldMgrone,
    propagateToDaughters);

fLocalEquationone = new G4Mag_UsualEqRhs(magFieldone);
fLocalStepperone = new G4ClassicalRK4(fLocalEquationone);
if(fLocalChordFinderone) delete fLocalChordFinderone;
fLocalChordFinderone = new G4ChordFinder( magFieldone,
    fMinStep, fLocalStepperone);
fieldMgrone->SetChordFinder( fLocalChordFinderone );

//Colors
fLogicmagnetonemid->SetVisAttributes(colormid);
fLogicmagnetonetop->SetVisAttributes(colorends);
fLogicmagnetonebot->SetVisAttributes(colorends);

//####### 2nd magnet#############

G4Box*              fmagnettwotop;
G4LogicalVolume*    fLogicmagnettwotop;
G4VPhysicalVolume*  fPhysicsmagnettwotop;

G4Box*              fmagnettwobot;
G4LogicalVolume*    fLogicmagnettwobot;
G4VPhysicalVolume*  fPhysicsmagnettwobot;

G4Box*              fmagnettwomid;
G4LogicalVolume*    fLogicmagnettwomid;
G4VPhysicalVolume*  fPhysicsmagnettwomid;

G4UniformMagField* magFieldtwo;
G4FieldManager* fieldMgrtwo;
G4Mag_UsualEqRhs* fLocalEquationtwo;
G4MagIntegratorStepper* fLocalSteppertwo;
G4ChordFinder* fLocalChordFindertwo(0);

// top end plates

fmagnettwotop = new G4Box("magnet2t", 5.0*cm, 2.5*cm, 10.0*
    cm);
```

```
fLogicmagnettwotop = new G4LogicalVolume(fmagnettwotop, fFe,
    "magnet2t");

fPhysicsmagnettwotop = new G4PVPlacement(0, G4ThreeVector
    (0,7.5*cm,-5.30*m), "magnet2t", fLogicmagnettwotop,
    fPhysicsWorld, false,    0 );

// bottom end plates

fmagnettwobot = new G4Box("magnet2b", 5.0*cm, 2.5*cm, 10.0*
    cm);

fLogicmagnettwobot = new G4LogicalVolume(fmagnettwobot, fFe,
    "magnet2b");

fPhysicsmagnettwobot = new G4PVPlacement(0, G4ThreeVector
    (0,-7.5*cm,-5.30*m), "magnet2b", fLogicmagnettwobot,
    fPhysicsWorld, false,   0 );

//Middle field area geometry
fmagnettwomid = new G4Box("magnet2m", 5.0*cm, 5.0*cm, 10.0*
    cm);

fLogicmagnettwomid = new G4LogicalVolume(fmagnettwomid,
    vacuum, "magnet2m");

fPhysicsmagnettwomid = new G4PVPlacement(0, G4ThreeVector
    (0,0*cm,-5.30*m), "magnet2m", fLogicmagnettwomid,
    fPhysicsWorld, false,      0 );

//Middle-local field manager

magFieldtwo = new G4UniformMagField(G4ThreeVector(0.,-
    stdfield,0.));

fieldMgrtwo = new G4FieldManager();
fieldMgrtwo->SetDetectorField(magFieldtwo);

fLogicmagnettwomid->SetFieldManager(fieldMgrtwo,
    propagateToDaughters);

fLocalEquationtwo = new G4Mag_UsualEqRhs(magFieldtwo);
fLocalSteppertwo = new G4ClassicalRK4(fLocalEquationtwo);
if(fLocalChordFindertwo) delete fLocalChordFindertwo;
fLocalChordFindertwo = new G4ChordFinder( magFieldtwo,
    fMinStep, fLocalSteppertwo);
fieldMgrtwo->SetChordFinder( fLocalChordFindertwo );

//Colors
fLogicmagnettwomid->SetVisAttributes(colormid);
fLogicmagnettwotop->SetVisAttributes(colorends);
fLogicmagnettwobot->SetVisAttributes(colorends);

//####### 3rd magnet##################

G4Box*              fmagnetthreetop;
G4LogicalVolume*    fLogicmagnetthreetop;
```

```
G4VPhysicalVolume* fPhysicsmagnetthreetop;

G4Box*             fmagnetthreebot;
G4LogicalVolume*   fLogicmagnetthreebot;
G4VPhysicalVolume* fPhysicsmagnetthreebot;

G4Box*             fmagnetthreemid;
G4LogicalVolume*   fLogicmagnetthreemid;
G4VPhysicalVolume* fPhysicsmagnetthreemid;

G4UniformMagField* magFieldthree;
G4FieldManager* fieldMgrthree;
G4Mag_UsualEqRhs* fLocalEquationthree;
G4MagIntegratorStepper* fLocalStepperthree;
G4ChordFinder* fLocalChordFinderthree(0);

// top end plates

fmagnetthreetop = new G4Box("magnet2t", 5.0*cm, 2.5*cm, 5.0*cm);

fLogicmagnetthreetop = new G4LogicalVolume(fmagnetthreetop, fFe, "magnet2t");

fPhysicsmagnetthreetop = new G4PVPlacement(0, G4ThreeVector(0,7.5*cm,-4.80*m), "magnet2t", fLogicmagnetthreetop, fPhysicsWorld, false,          0 );

// bottom end plates

fmagnetthreebot = new G4Box("magnet2b", 5.0*cm, 2.5*cm, 5.0*cm);

fLogicmagnetthreebot = new G4LogicalVolume(fmagnetthreebot, fFe, "magnet2b");

fPhysicsmagnetthreebot = new G4PVPlacement(0, G4ThreeVector(0,-7.5*cm,-4.80*m), "magnet2b", fLogicmagnetthreebot, fPhysicsWorld, false,          0 );

//Middle field area geometry
fmagnetthreemid = new G4Box("magnet2m", 5.0*cm, 5.0*cm, 5.0*cm);

fLogicmagnetthreemid = new G4LogicalVolume(fmagnetthreemid, vacuum, "magnet2m");

fPhysicsmagnetthreemid = new G4PVPlacement(0, G4ThreeVector(0,0*cm,-4.80*m), "magnet2m", fLogicmagnetthreemid, fPhysicsWorld, false,  0 );

//Middle-local field manager

magFieldthree = new G4UniformMagField(G4ThreeVector(0., stdfield,0.));

fieldMgrthree = new G4FieldManager();
```

```
  fieldMgrthree->SetDetectorField(magFieldthree);

  fLogicmagnetthreemid->SetFieldManager(fieldMgrthree,
      propagateToDaughters);

  fLocalEquationthree = new G4Mag_UsualEqRhs(magFieldthree);
  fLocalStepperthree = new G4ClassicalRK4(fLocalEquationthree)
      ;
  if(fLocalChordFinderthree) delete fLocalChordFinderthree;
  fLocalChordFinderthree = new G4ChordFinder( magFieldthree,
      fMinStep, fLocalStepperthree);
  fieldMgrthree->SetChordFinder( fLocalChordFinderthree );

  //Colors
  fLogicmagnetthreemid->SetVisAttributes(colormid);
  fLogicmagnetthreetop->SetVisAttributes(colorends);
  fLogicmagnetthreebot->SetVisAttributes(colorends);
}

////////////////////////////////
// Ionization Chamber Description

void Em10DetectorConstruction::ic()
{
  //IC Generic Variables
  G4Material* fAr = fMat->GetMaterial("Argon");
  G4Material* fAl = fMat->GetMaterial("Al");
  G4VisAttributes* coloricout = new G4VisAttributes(G4Colour
      (1.0, 1.0, 0.0, 0.5));
  G4VisAttributes* coloricin = new G4VisAttributes(G4Colour
      (0.0, 1.0, 0.0, 0.2));
  coloricout->SetForceSolid(true);
  coloricin->SetForceSolid(true);

  //IC1 Geometry Variables
  G4Box*             ficoutone;
  G4LogicalVolume*   fLogicicoutone;
  G4VPhysicalVolume* fPhysicsicoutone;

  G4Box*             ficinone;
  G4LogicalVolume*   fLogicicinone;
  G4VPhysicalVolume* fPhysicsicinone;

  //IC1 Geometry
  ficoutone = new G4Box("icout1", 5.1*cm, 5.1*cm, 50.0*cm);

  fLogicicoutone = new G4LogicalVolume(ficoutone, fAl, "icout1
      ");

  fPhysicsicoutone = new G4PVPlacement(0, G4ThreeVector(5.2*cm
      ,0,5.20*m), "icout1", fLogicicoutone, fPhysicsWorld,
      false,       0 );

  ficinone = new G4Box("icin", 5.0*cm, 5.0*cm, 50.0*cm);

  fLogicicinone = new G4LogicalVolume(ficinone, fAr, "icin1");
```

```
  fPhysicsicinone = new G4PVPlacement(0, G4ThreeVector(5.2*cm
      ,0,5.20*m), "icin1", fLogicicinone, fPhysicsWorld, false
      ,  0 );

  //IC2 Geometry Variables
  G4Box*              ficouttwo;
  G4LogicalVolume*    fLogicicouttwo;
  G4VPhysicalVolume* fPhysicsicouttwo;

  G4Box*              ficintwo;
  G4LogicalVolume*    fLogicicintwo;
  G4VPhysicalVolume* fPhysicsicintwo;

  //IC2 Geometry
  ficouttwo = new G4Box("icout1", 5.1*cm, 5.1*cm, 50.0*cm);

  fLogicicouttwo = new G4LogicalVolume(ficouttwo, fAl, "icout1
      ");

  fPhysicsicouttwo = new G4PVPlacement(0, G4ThreeVector(-5.2*
      cm,0,5.20*m), "icout1", fLogicicouttwo, fPhysicsWorld,
      false,       0 );

  ficintwo = new G4Box("icin", 5.0*cm, 5.0*cm, 50.0*cm);

  fLogicicintwo = new G4LogicalVolume(ficintwo, fAr, "icin1");

  fPhysicsicintwo = new G4PVPlacement(0, G4ThreeVector(-5.2*cm
      ,0,5.20*m), "icin1", fLogicicintwo, fPhysicsWorld, false
      , 0 );

  //IC Colors
  fLogicicoutone->SetVisAttributes(coloricout);
  fLogicicinone->SetVisAttributes(coloricin);
  fLogicicouttwo->SetVisAttributes(coloricout);
  fLogicicintwo->SetVisAttributes(coloricin);

  if( fRegGasDet != 0 ) delete fRegGasDet;
  if( fRegGasDet == 0 )         fRegGasDet = new G4Region("
      XTRdEdxDetector");
                                fRegGasDet->
                                    AddRootLogicalVolume(
                                    fLogicicintwo );
                                fRegGasDet->
                                    AddRootLogicalVolume(
                                    fLogicicinone );

}

/////////////////////////////////
// Collimator Description

void Em10DetectorConstruction::collimators()
{

  //G4Material* fPb = fMat->GetMaterial("Lead");
```

```
}

//////////////////////////////////
//
//


Em10DetectorConstruction::Em10DetectorConstruction():fSetUp("
    simpleprajwal")
{
  fDetectorMessenger = new Em10DetectorMessenger(this);
  fMat               = new Em10Materials();
  //userLimits         = new G4UserLimits();
}

///////////////////////////////
//
//

Em10DetectorConstruction::~Em10DetectorConstruction()
{
  delete fDetectorMessenger;
  delete fMat;
}

///////////////////////////////
//
//

G4VPhysicalVolume* Em10DetectorConstruction::Construct()
{
  return ConstructDetectorXTR();
}


//////////////////////////////
//
//

G4VPhysicalVolume* Em10DetectorConstruction::
    ConstructDetectorXTR()
{
 // Cleanup old geometry

  G4GeometryManager::GetInstance()->OpenGeometry();
  G4PhysicalVolumeStore::GetInstance()->Clean();
  G4LogicalVolumeStore::GetInstance()->Clean();
  G4SolidStore::GetInstance()->Clean();

  if( fSetUp == "simpleprajwal" )
  {
    return SetUpprajwal();
  }
  else
  {
    G4cout<<"Experimental setup is unsupported. Check /
        XTRdetector/setup "<<G4endl;
```

```
    G4cout<<"Run default: prajwal"<<G4endl;
    return SetUpprajwal();

    //  return 0;
  }
}

void Em10DetectorConstruction::SetMagField(G4double fieldValue
    )
{
  //apply a global uniform magnetic field along Z axis
  G4FieldManager* fieldMgr
   = G4TransportationManager::GetTransportationManager()->
       GetFieldManager();

  if (magField) delete magField;        //delete the existing
      magn field

  if (fieldValue!=0.)                   // create a new one if
       non null
    {
      magField = new G4UniformMagField(G4ThreeVector(0.,0.,
          fieldValue));
      fieldMgr->SetDetectorField(magField);
      fieldMgr->CreateChordFinder(magField);
    }
   else
    {
      magField = 0;
      fieldMgr->SetDetectorField(magField);
    }
}

void Em10DetectorConstruction::SetMaxStepLength(G4double val)
{
  // set the maximum length of tracking step
  //
  if (val <= DBL_MIN)
    { G4cout << "\n --->warning from SetMaxStepLength: maxStep
         "
              << val  << " out of range. Command refused" <<
                  G4endl;
      return;
    }
  G4TransportationManager* tmanager = G4TransportationManager
      ::GetTransportationManager();
  tmanager->GetPropagatorInField()->SetLargestAcceptableStep(
      val);
}

///////////////////////////////////////
//
// Simplified setup for SpinLight Polarimeter (~2012).
// Runs by  : TestEm10 SpinIC.mac
// Author   : Prajwal Mohanmurthy (prajwal@mohanmurthy.com)
// Adopted from GEANT-4 example suite example TestEm10
// available under foler '~/examples/extended/electromagnetic'
```

```
G4VPhysicalVolume* Em10DetectorConstruction::SetUpprajwal()
{
  fWorldSizeZ = 12.*m;
  fWorldSizeR = 50.*cm;

  // Radiator and detector parameters

  fRadThickness = 0.020*mm;
  fGasGap       = 0.250*mm;
  foilGasRatio  = fRadThickness/(fRadThickness+fGasGap);

  fFoilNumber   = 220;

  fAbsorberThickness = 38.3*mm;

  fAbsorberRadius    = 100.*mm;
  fAbsorberZ         = 136.*cm;

  fWindowThick     = 51.0*micrometer ;
  fElectrodeThick = 10.0*micrometer ;
  fGapThick        =  10.0*cm ;


  fDetThickness =  40.0*mm ;
  fDetLength    = 200.0*cm  ;
  fDetGap       =   0.01*mm ;


  fStartR       = 40*cm   ;
  fStartZ       = 100.0*mm  ;

  fModuleNumber = 1       ;

  // Preparation of mixed radiator material
  G4Material* Mylar = fMat->GetMaterial("Mylar");
  G4Material* Al   = fMat->GetMaterial("Al");

  G4double foilDensity =  1.39*g/cm3; // Mylar // 0.91*g/cm3;
      // CH2 0.534*g/cm3; //Li
  G4double gasDensity  =  1.2928*mg/cm3; // Air // 1.977*mg/
     cm3; // CO2 0.178*mg/cm3; // He
  G4double totDensity  = foilDensity*foilGasRatio + gasDensity
     *(1.0-foilGasRatio) ;

  G4double fractionFoil =  foilDensity*foilGasRatio/totDensity
      ;
  G4double fractionGas  =  gasDensity*(1.0-foilGasRatio)/
     totDensity ;

  G4Material* radiatorMat = new G4Material("radiatorMat"   ,
     totDensity , 2);
  //radiatorMat->AddMaterial( Mylar, fractionFoil ) ;
  radiatorMat->AddMaterial( vacuum, fractionFoil ) ;
  radiatorMat->AddMaterial( vacuum, fractionGas   ) ;
```

```
// default materials of the detector and TR radiator

fRadiatorMat = radiatorMat;
fFoilMat     = Mylar; // CH2; // Kapton; // Mylar ; // Li ; // CH2 ;
fGasMat      = vacuum; // CO2; // He; //

fWindowMat     = Mylar ;
fElectrodeMat = Al ;

fAbsorberMaterial = fMat->GetMaterial("Xe15CO2");


fGapMat           = fAbsorberMaterial;

fWorldMaterial     = vacuum; // CO2 ;

fSolidWorld = new G4Box("World", fWorldSizeR,fWorldSizeR, fWorldSizeZ/2.);

fLogicWorld = new G4LogicalVolume(fSolidWorld, fWorldMaterial, "World");

fPhysicsWorld = new G4PVPlacement(0, G4ThreeVector(), "World",fLogicWorld, 0, false, 0);

//%%%%%%%MAGNETS%%%%%%%%%%%%%%%%

magnets();

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

//%%%%%%%Collimators%%%%%%%%%%%%

collimators();

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

//%%%%%%%Ionization chamber%%%%%%

ic();

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
// TR radiator envelope

fRadThick = fFoilNumber*(fRadThickness + fGasGap) - fGasGap + fDetGap;

fRadZ = fStartZ + 0.5*fRadThick ;

//fSolidRadiator = new G4Box("Radiator",1.1*fAbsorberRadius , 1.1*fAbsorberRadius, 0.5*fRadThick );

//fLogicRadiator = new G4LogicalVolume(fSolidRadiator, fRadiatorMat,"Radiator");
```

```
  //fPhysicsRadiator = new G4PVPlacement(0, G4ThreeVector(0,0,/*fRadZ*/4.95*m), "Radiator", fLogicRadiator, fPhysicsWorld, false, 0 );

  // create region for window inside windowR for

  if( fRadRegion != 0 ) delete fRadRegion;
  if( fRadRegion == 0 )        fRadRegion = new G4Region("XTRradiator");
                               //fRadRegion->AddRootLogicalVolume(fLogicRadiator);

  fWindowZ = fStartZ + fRadThick + fWindowThick/2. + 15.0*mm ;

  fGapZ = fWindowZ + fWindowThick/2. + fGapThick/2. + 0.01*mm ;

  fElectrodeZ = fGapZ + fGapThick/2. + fElectrodeThick/2. + 0.01*mm;
/*
  // Absorber

  fAbsorberZ = fElectrodeZ + fElectrodeThick/2. + fAbsorberThickness/2. + 0.01*mm;

  fSolidAbsorber = new G4Box("Absorber", fAbsorberRadius, fAbsorberRadius, fAbsorberThickness/2.);

  fLogicAbsorber = new G4LogicalVolume(fSolidAbsorber, fAbsorberMaterial,"Absorber");

  fPhysicsAbsorber = new G4PVPlacement(0, G4ThreeVector(0.,0., fAbsorberZ),  "Absorber", fLogicAbsorber, fPhysicsWorld, false,  0);

  if( fRegGasDet != 0 ) delete fRegGasDet;
  if( fRegGasDet == 0 )        fRegGasDet = new G4Region("XTRdEdxDetector");
                               fRegGasDet->AddRootLogicalVolume(fLogicicintwo );
                               fRegGasDet->AddRootLogicalVolume(fLogicicinone );
*/
  // Sensitive Detectors: Absorber

  G4SDManager* SDman = G4SDManager::GetSDMpointer();

  if(!fCalorimeterSD)
  {
    fCalorimeterSD = new Em10CalorimeterSD("CalorSD",this);
    SDman->AddNewDetector( fCalorimeterSD );
  }
  //if (fLogicAbsorber)  fLogicAbsorber->SetSensitiveDetector(fCalorimeterSD);
```

```
  PrintGeometryParameters();

  //Uniform Magnetic field for all of world volume is defined
      here

  G4double fieldValue = 0.0*tesla;
  G4UniformMagField* magField = new G4UniformMagField(
      G4ThreeVector(0.,fieldValue,0.));

  G4FieldManager* fieldMgr = G4TransportationManager::
      GetTransportationManager()->GetFieldManager();
  fieldMgr->SetDetectorField(magField);

  fieldMgr->CreateChordFinder(magField);


  return fPhysicsWorld;
}

////////////////////////////////
//
//

void Em10DetectorConstruction::PrintGeometryParameters()
{
  G4cout << "\n The  WORLD   is made of "
         << fWorldSizeZ/mm << "mm of " << fWorldMaterial->
             GetName() ;
  G4cout << ", the transverse size (R) of the world is " <<
      fWorldSizeR/mm << " mm. " << G4endl;
  G4cout << " The ABSORBER is made of "
         << fAbsorberThickness/mm << "mm of " <<
             fAbsorberMaterial->GetName() ;
  G4cout << ", the transverse size (R) is " << fAbsorberRadius
      /mm << " mm. " << G4endl;
  G4cout << " Z position of the (middle of the) absorber " <<
      fAbsorberZ/mm << "  mm." << G4endl;

  G4cout <<"fRadZ = "<<fRadZ/mm<<" mm"<<G4endl ;

  G4cout <<"fStartZ = "<<fStartZ/mm<<" mm"<<G4endl ;

  G4cout <<"fRadThick = "<<fRadThick/mm<<" mm"<<G4endl ;
  G4cout <<"fFoilNumber = "<<fFoilNumber<<G4endl ;
  G4cout <<"fRadiatorMat = "<<fRadiatorMat->GetName()<<G4endl
      ;
  G4cout <<"WorldMaterial = "<<fWorldMaterial->GetName()<<
      G4endl ;
  //  G4cout<<"fAbsorberZ = "<<fAbsorberZ/mm<<" mm"<<G4endl;
  G4cout << G4endl;
}

////////////////////////////////
//
//
```

```
void Em10DetectorConstruction::SetAbsorberMaterial(G4String
    materialChoice)
{
  // get the pointer to the material table
  const G4MaterialTable* theMaterialTable = G4Material::
      GetMaterialTable();

  // search the material by its name
  G4Material* pttoMaterial;

  for (size_t J=0 ; J<theMaterialTable->size() ; J++)
  {
    pttoMaterial = (*theMaterialTable)[J];

    if(pttoMaterial->GetName() == materialChoice)
    {
      fAbsorberMaterial = pttoMaterial;
      fLogicAbsorber->SetMaterial(pttoMaterial);
        // PrintCalorParameters();
    }
  }
}
////////////////////////////////
//
//

void Em10DetectorConstruction::SetRadiatorMaterial(G4String
    materialChoice)
{
  // get the pointer to the material table

  const G4MaterialTable* theMaterialTable = G4Material::
      GetMaterialTable();

  // search the material by its name

  G4Material* pttoMaterial;
  for (size_t J=0 ; J<theMaterialTable->size() ; J++)
  {
    pttoMaterial = (*theMaterialTable)[J];

    if(pttoMaterial->GetName() == materialChoice)
    {
      fRadiatorMat = pttoMaterial;
      fLogicRadSlice->SetMaterial(pttoMaterial);
      // PrintCalorParameters();
    }
  }
}

/////////////////////////////////
//
//

void Em10DetectorConstruction::SetWorldMaterial(G4String
    materialChoice)
{
```

```
  // get the pointer to the material table
  const G4MaterialTable* theMaterialTable = G4Material::
      GetMaterialTable();

  // search the material by its name
  G4Material* pttoMaterial;

  for (size_t J=0 ; J<theMaterialTable->size() ; J++)
  {
    pttoMaterial = (*theMaterialTable)[J];

    if(pttoMaterial->GetName() == materialChoice)
    {
      fWorldMaterial = pttoMaterial;
      fLogicWorld->SetMaterial(pttoMaterial);
       //  PrintCalorParameters();
    }
  }
}

///////////////////////////////
//
//

void Em10DetectorConstruction::SetAbsorberThickness(G4double
    val)
{
  // change Absorber thickness and recompute the calorimeter
      parameters
  fAbsorberThickness = val;
  //  ComputeCalorParameters();
}

///////////////////////////////
//
//

void Em10DetectorConstruction::SetRadiatorThickness(G4double
    val)
{
  // change XTR radiator thickness and recompute the
      calorimeter parameters
  fRadThickness = val;
  // ComputeCalorParameters();
}

///////////////////////////////
//
//

void Em10DetectorConstruction::SetGasGapThickness(G4double val
    )
{
  // change XTR gas gap thickness and recompute the
      calorimeter parameters
  fGasGap = val;
  // ComputeCalorParameters();
```

```
}

/////////////////////////////////
//
//

void Em10DetectorConstruction::SetAbsorberRadius(G4double val)
{
  // change the transverse size and recompute the calorimeter
      parameters
  fAbsorberRadius = val;
  // ComputeCalorParameters();
}

////////////////////////////////
//
//

void Em10DetectorConstruction::SetWorldSizeZ(G4double val)
{
  fWorldChanged=true;
  fWorldSizeZ = val;
  // ComputeCalorParameters();
}

///////////////////////////////
//
//

void Em10DetectorConstruction::SetWorldSizeR(G4double val)
{
  fWorldChanged=true;
  fWorldSizeR = val;
  // ComputeCalorParameters();
}

//////////////////////////////////
//
//

void Em10DetectorConstruction::SetAbsorberZpos(G4double val)
{
  fAbsorberZ  = val;
  // ComputeCalorParameters();
}

//////////////////////////////////
//
//

void Em10DetectorConstruction::UpdateGeometry()
{
  G4RunManager::GetRunManager()->DefineWorldVolume(
      ConstructDetectorXTR());
}

//
```

```
//
/////////////////////////////////
```