PROJECT REPORT

Summer Student Program 2009

DESY

Supervisors: Winfried Decking, Vladimir Balandin

Lobanova Inna

Lomonosov Moscow State University

28th of August, 2009

Contents

Introduction

There are several codes, developed at DESY, which use EGS4 code system (Electron Gamma Shower), among them are propagation of particles in the slab, scattering after impact in collimators and, the most interesting code, developed during the study of dark current in the TESLA. These codes could play an important role in studies related to the XFEL design, including dark current behavior in the XFEL Linac. But, unfortunately, due to the change of the computer system in DESY, EGS4 code system is not supported anymore. There are two possible ways to bring these codes to the working stage again:

1). to make some archeology in order to find the old version, previously used in DESY and make its reinstallation;

2). to install new, more resent EGS code system version, which for sure was improved over this years in comparison with the old version used in DESY and allows to do more precise modeling of particle transport;

Purpose of my work is connected with the second way and my task was to make a review of available EGS versions together with the review of improvements of coded in them physics; to select the most promising version, install it in DESY and to try to execute old programs with this new installation. It is very important that big part of previously mentioned programs is written in MORTRAN. MORTRAN (More Fortran) is an extension of the Fortran programming language used for scientific computation and it’s code is macro-processed into Fortran code for compilation. It means that we need the version of EGS code system that maintains MORTRAN.

1. EGS4 Code System

The EGS code system is a general purpose package for the Monte Carlo simulation of the coupled transport of electrons and photons in the arbitrary geometry for particles with energies above a few keV up to several TeV (depending on the atomic numbers of the target materials). It was originally developed in the sixties at SLAC (USA) for high energy physics applications and has been extended with the help of NRC (National Research Council Canada) and KEK (High Energy Accelerator Research Organization) to apply for lower energy applications. EGS4 code system is making electromagnetic particles advance in a material media and has on account all the most important processes in which these particles interact loosing their energy.

In the twenty three years, since EGS4 was released in 1985, it has been used in a wide variety of applications, particularly in medical physics, radiation measurement studies, and industrial development.

1.1. History of EGS Code System

The Monte Carlo method was originally suggested by Ulam and von Neumann, and was first used by Goldberger in order to study nuclear disintegrations produced by high-energy particles. The first application of the Monte Carlo technique to study shower production was done by Wilson. Wilson's approach was a simple graphical-mechanical that was described as follows: “The procedure used was a simple graphical and mechanical one. The distance into the lead was broken into intervals of one-fifth of a radiation length (about 1 mm). The electrons or photons were followed through successive intervals and their fate in passing through a given interval was decided by spinning a wheel of chance; the fate being read from one of a family of curves drawn on a cylinder…

A word about the wheel of chance; The cylinder, 4 in. outside diameter by 12 in. long is driven by a high speed motor geared down by a ratio 20 to 1. The motor armature is heavier than the cylinder and determines where the cylinder stops. The motor was observed to stop at random and, in so far as the cylinder is concerned, its randomness is multiplied by the gear ratio…”

from R. R. Wilson

The first use of an electronic digital computer in simulating high-energy cascades by Monte Carlo methods was reported by Butcher and Messel, and independently by Varfolomeev and Svetlolobov. These two groups collaborated in a much publicized work that eventually led to an extensive set of tables describing the shower distribution functions, the so-called “shower book”.

The "seed" for the EGS computer program was brought to SLAC around 1965 by Hans-Hellmut Nagel of Bonn University. Nagel's program was useful in designing elements of accelerator machinery (such as beam stoppers, collimators, and targets) during construction of the two-mile accelerator and beam lines at SLAC. However, Nagel's code proved to be too specific to solve many problems still facing the high-energy physics community.

Two physicists took on the task of redesigning the code from the bottom up to achieve the necessary generalization. Working independently at first, W. Ralph Nelson at SLAC, and later Richard Ford at the Hanson High Energy Physics Laboratory on the Stanford University campus, decided to combine their programming efforts and produced the first version of EGS in 1978. And its companion code was called PEGS (Preprocessor for EGS). This version, written completely in the FORTRAN-IV language, is referred to as Version 1 of the EGS Code System (or more simply EGS1 and PEGS1).

EGS1, however, could not be readily used to simulate showers in complex geometries. A good example of this was the Crystal Ball detector for which EGS1, under the direction of E. Bloom at SLAC, was modified to handle the particular geometry in question. Thus it was decided that EGS1, which was a one-region, one-medium code, should be generalized in order to handle many-region, many-media, complex, three-dimensional geometries and EGS1 became a subprogram in itself with two user-callable subroutines (HATCH and SHOWER) that require two user-written subroutines (HOWFAR and AUSGAB) in order to define the geometry and do the scoring, respectively. These revisions were completed by the end of 1975 and the new versions of EGS and PEGS appeared (EGS2 and PEGS2).

Comparisons of backscattered photon uence as computed by EGS2 versus unpublished HEPL data, as well as bremsstrahlung angular distribution calculations comparing EGS2 results with those of Berger and Seltzer using ETRAN, suggested that EGS2 might be predicting values in the backward direction that were low by up to a factor of two. In order to implement Nagel's discrete reduced-angle multiple-scattering scheme in a general multi-region environment and in order to achieve greater universality of application (e.g., so that a monochromatic beam of electrons impinging on a very thin slab would have a continuous angular distribution on exit), Ford decided in the summer of 1976 to try to implement a multiple scattering-scheme that would correctly sample the continuous multiple-scattering distribution for arbitrary step lengths. These versions of EGS, PEGS were called EGS3 and PEG3. EGS3 was designed to simulate electromagnetic cascades in various geometries and at energies up to a few thousand GeV and down to cutoff kinetic energies of 0.1 MeV (photons) and 1 MeV (electrons and positrons). However, ever since the introduction of the code in

1978 there had been an increasing need to extend the lower-energy limits (down to 1 and 10 keV for photons and electrons, respectively) and this was a major, but not the only, reason for creating EGS4, which was released in 1985.

1.2. The Use of EGS4

The EGS4 Code System is very flexible. This means that the overall structure can get quite complex. However, each step is straight forward and once familiar with the system, it can be quite user-friendly. EGS4 is written in an extended FORTRAN called Mortran3. The EGS4 includes a Mortran3 pre-processor which takes Mortran3 source code and converts it into FORTRAN.

The user writes an EGS4 User Code in Mortran3 language. Each FORTRAN version of EGS4 is associated with a particular User Code since the latter can and should contain macro-overrides which affect the EGS4 system as applied to the problem at hand (e.g., the number of materials and/or regions that are used, the energies of the particles being tracked, cut-off energies when the particles will be absorbed).

For a user the system is a set of subroutines that implement the modelling of the various interaction processes and the simulation of electron and photon transport. At first the user calls HATCH, which is the EGS4 materials initialisation routine. Then the user makes a series of calls to the transport routine SHOWER (one call per incident particle to be simulated). It uses PEGS4 file and start the run. In fact the main control of the physical process is done by SHOWER. It computes what happens in each step and calls HOWFAR at first, the other user provided subroutine, which determines where and when transported particles cross between the user-defined geometrical regions. Then routine HOWNEAR is called, which is used to determine whether boundary-crossing must be taken into account for PRESTA low-energy electron transport. Further SHOWER calls the user provided subroutine AUSGAB, which collects energy deposited at interactions.

One then compiles and "runs" the FORTRAN code that has been created, which requires as input a file previously prepared by the PEGS4 code. PEGS4 is a program which comes in the same package as EGS4 and contains the electron and photon cross section data for any of the materials appearing in the problem.

1.3. Currently Available Versions of EGS4

For about ten years EGS code system hasn't been supported by SLAC (USA), and nowadays EGS4 package can be obtained in two ways: from NRC and KEK websites. But one has to take into account that both these versions are not identical to each other, as well as they are not identical to the original version of SLAC.

Concerning EGS4 KEK version, it is frozen and has not been developed during last years, due to started development of the EGS5 code system, which will be described in more details in the next section.

EGSnrc distributed by NRC is a version of the EGS4 package, which was developed for applications in radiotherapy and is oriented mainly not to developers of their own application codes, but to users, who will create their applications using supplied graphical user interfaces. The last two updates of EGSnrc were on 13-th of February 2007 (Release V4-2-2-5) and on 10-th of July 2009 (V4-2-3-0).

1.4. EGS5 Code System

EGS5 code system is an extended and improved version of the EGS4 package that contains many changes. Historically, EGS, the simulation code, and PEGS, the data generation code, were completely separate entities. This left open the possibility that a simulation could work on inappropriate data, despite the built-in integrity checks. In EGS5, the two codes have been merged, and now EGS calls PEGS during its initialization procedures. Since the elastic scattering runtime databases can require large amounts of CPU time in certain cases, the capability to reuse PEGS data files still exists, but its use is discouraged unless the accurate low energy elastic scattering option is employed.

Additionally, many of the features and options in EGS4 which were invoked through MORTRAN macro substitutions have been retained in the base shower code in EGS5 and can be “turned on” by user specification of the appropriate flags and parameters.

The last two updates of EGS5 were in August 2005 and on 19-th of January 2009.

2. Installation and Testing of EGS Code System

The survey of variations of EGS code system made in the previous section showed that the latest and most updated version of EGS code system is EGS5, but beginning with version 5 of the EGS code system, MORTRAN is no longer supported. So the presence of big parts of code written in MORTRAN determined the installation of EGS4 KEK code system or EGSnrc code system.

2.1. Installation and testing of EGS4 KEK Code system

The decision to install EGS4 KEK was made because of this version resemblance to the original SLAC version. The installation of EGS4 KEK code system on Linux was made without any problems because instruction for installation is clearly written, the installation script is rather short and it is easy to look into it.

After installation I started checking if old MORTRAN programs work with the new system. The first program tested was simple, but useful program SLAB1MOD, which model a semi-infinite slab of material placed in a vacuum. The slab is in the X-Y plane and the particles are incident in the origin Z=0. In this model user can set up some parameters, for example, the slab thickness, the slab material, charge of particles (0, 1,-1), geometry for recording of energy deposition density and so on. This program in the past was used for choosing the material for manufacturing of XFEL collimators. It was possible to execute this program almost without changes in the source code

To test if the program is working correctly I made a series of runs with different slab parameters and parameters of the incident particles. After every run the energy balance was checked and examples of detailed run protocols can be seen in the Tables 1 and 2.

```
EGS SUCCESSFULLY 'HATCHED' FOR ONE MEDIUM.

-------------------------------------------
INITIAL RANDSEED (IXX)    :   987654321
FINAL   RANDSEED (IXX)    :  2014026277
-------------------------------------------
LENGTH OF MEDIUM          :       3.000 cm
MIN RECORDING POSITION    :       0.000 cm
MAX RECORDING POSITION    :       3.000 cm
HALF-SIZE OF X-BAR        :       0.001 cm
MAX RECORDING X-SIZE      :       0.040 cm
```

```
HALF-SIZE OF Y-BAR          :        0.001 cm
MAX RECORDING Y-SIZE        :        0.040 cm
CUTOFF FOR ELECTRONS        :        1.500 MeV
CUTOFF FOR PHOTONS          :        0.100 MeV
-------------------------------------------
INPUT PARTICLES             :         4000
INPUT ENERGY                : 0.40000E+07 MeV
-------------------------------------------
REFLECTED BACK ENERGY       : 0.30560E+04 MeV (   0.08%)
BY ELECTRONS          : 0.39858E+02 MeV (   1.30%)
BY POSITRONS          : 0.82055E+01 MeV (   0.27%)
BY PHOTONS            : 0.30079E+04 MeV ( 98.43%)
-------------------------------------------
TRANSMITTED ENERGY          : 0.34310E+07 MeV ( 85.77%)
BY ELECTRONS          : 0.88759E+06 MeV ( 25.87%)
BY POSITRONS          : 0.38424E+06 MeV ( 11.20%)
BY PHOTONS            : 0.21591E+07 MeV ( 62.93%)
-------------------------------------------
DEPOSITED ENERGY            : 0.56599E+06 MeV ( 14.15%)
OUTSIDE RECORDING     : 0.33749E+06 MeV ( 59.63%)
INSIDE  RECORDING     : 0.22850E+06 MeV ( 40.37%)
-------------------------------------------
TOTAL CALCULATED ENERGY     : 0.40000E+07 MeV
RELATIVE ERROR IN ENERGY    : 0.11642E-13 %
-------------------------------------------
TOTAL DISCARDED ELECTRONS :      2701549
TOTAL DISCARDED POSITRONS :        16879
TOTAL DISCARDED PHOTONS   :       328405
-------------------------------------------
TRANSMITTED ELECTRONS       :        11883
TRANSMITTED POSITRONS       :         6662
TRANSMITTED PHOTONS         :       132609
```

Table 1

```
EGS SUCCESSFULLY 'HATCHED' FOR ONE MEDIUM.

-------------------------------------------
INITIAL RANDSEED (IXX)      :    987654321
FINAL   RANDSEED (IXX)      :   -857813275
-------------------------------------------
LENGTH OF MEDIUM            :        3.000 cm
MIN RECORDING POSITION      :        0.000 cm
MAX RECORDING POSITION      :        3.000 cm
HALF-SIZE OF X-BAR          :        0.001 cm
MAX RECORDING X-SIZE        :        0.399 cm
HALF-SIZE OF Y-BAR          :        0.001 cm
MAX RECORDING Y-SIZE        :        0.399 cm
CUTOFF FOR ELECTRONS        :        1.500 MeV
CUTOFF FOR PHOTONS          :        0.100 MeV
-------------------------------------------
INPUT PARTICLES             :           30
```

```
INPUT ENERGY              : 0.15000E+05 MeV
-------------------------------------------
REFLECTED BACK ENERGY     : 0.18569E+02 MeV (  0.12%)
      BY ELECTRONS        : 0.00000E+00 MeV (  0.00%)
      BY POSITRONS        : 0.00000E+00 MeV (  0.00%)
      BY PHOTONS          : 0.18569E+02 MeV (100.00%)
-------------------------------------------
TRANSMITTED ENERGY        : 0.11834E+05 MeV ( 78.89%)
      BY ELECTRONS        : 0.23672E+04 MeV ( 20.00%)
      BY POSITRONS        : 0.11177E+04 MeV (  9.44%)
      BY PHOTONS          : 0.83488E+04 MeV ( 70.55%)
-------------------------------------------
DEPOSITED ENERGY          : 0.31477E+04 MeV ( 20.98%)
      OUTSIDE RECORDING   : 0.36648E+03 MeV ( 11.64%)
      INSIDE  RECORDING   : 0.27813E+04 MeV ( 88.36%)
-------------------------------------------
TOTAL CALCULATED ENERGY   : 0.15000E+05 MeV
RELATIVE ERROR IN ENERGY  : 0.00000E+00 %
-------------------------------------------
TOTAL DISCARDED ELECTRONS :       15109
TOTAL DISCARDED POSITRONS :          82
TOTAL DISCARDED PHOTONS   :        1800
-------------------------------------------
TRANSMITTED ELECTRONS     :          49
TRANSMITTED POSITRONS     :          26
TRANSMITTED PHOTONS       :         682
```

Table 2

In the program SLAB1MOD there are possibilities to store run results in a special format in ASCII files. These ASCII files in the next tern can be used as an input for specially written postscript drivers, producing different plots: particle trajectories inside the medium, graphics of energy deposition density and etc. It was checked that output files have the correct format for postscript drivers, and resulting examples can be seen in Figures 1-4.

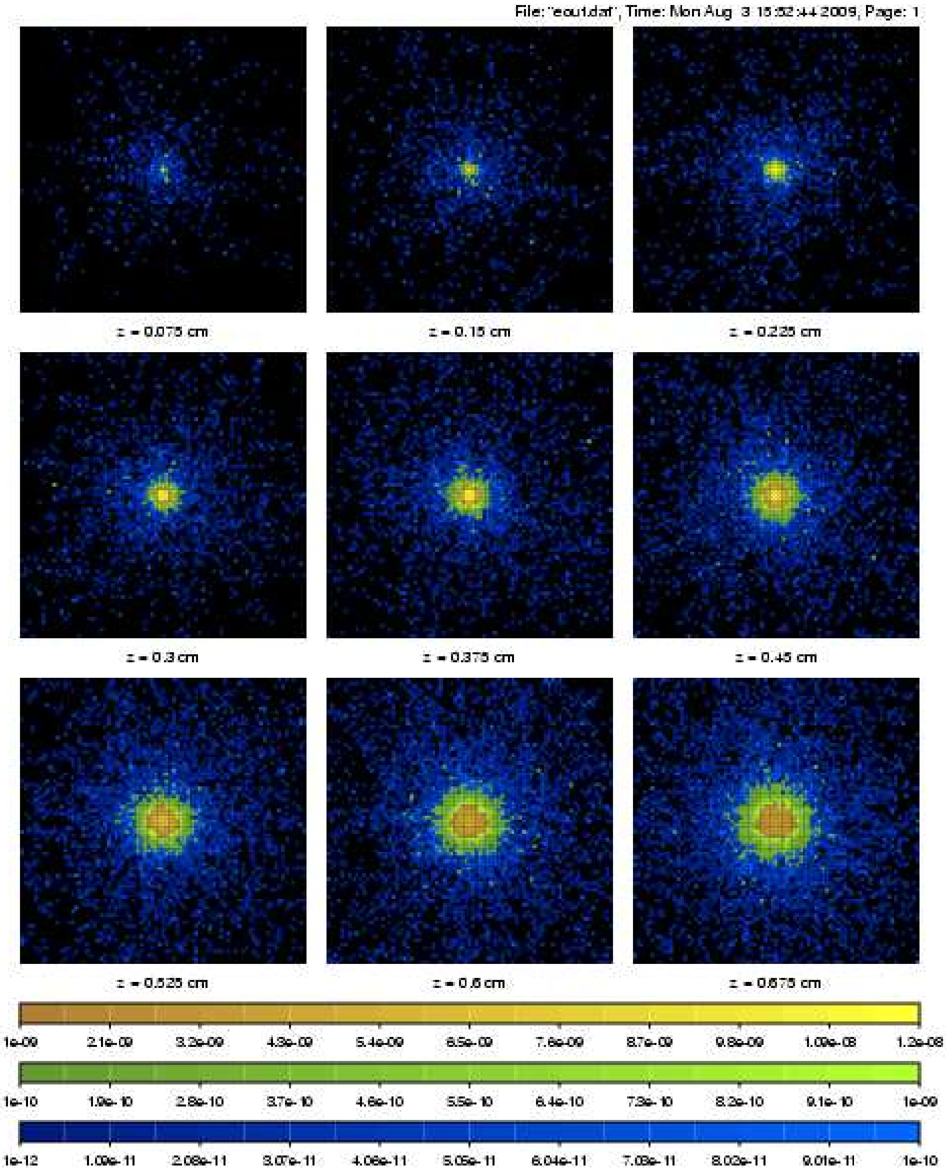
File: "eout.dat", Time: Mon Aug 3 15:52:44 2009, Page: 1
z = 0.075 cm
z = 0.15 cm
z = 0.225 cm
z = 0.3 cm
z = 0.375 cm
z = 0.45 cm
z = 0.525 cm
z = 0.6 cm
z = 0.675 cm
1e-09 2.1e-09 3.2e-09 4.3e-09 5.4e-09 6.5e-09 7.6e-09 8.7e-09 9.8e-09 1.09e-08 1.2e-08
1e-10 1.9e-10 2.8e-10 3.7e-10 4.6e-10 5.5e-10 6.4e-10 7.3e-10 8.2e-10 9.1e-10 1e-09
1e-12 1.09e-11 2.08e-11 3.07e-11 4.06e-11 5.05e-11 6.04e-11 7.03e-11 8.02e-11 9.01e-11 1e-10

Figure 1

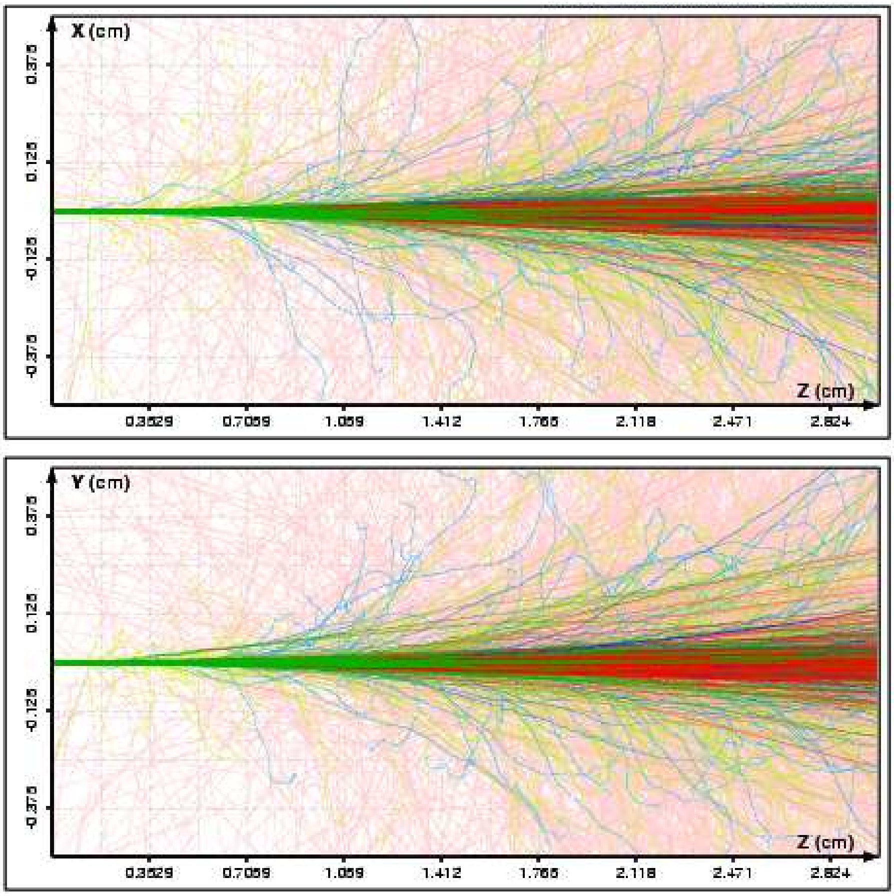

Figure 2

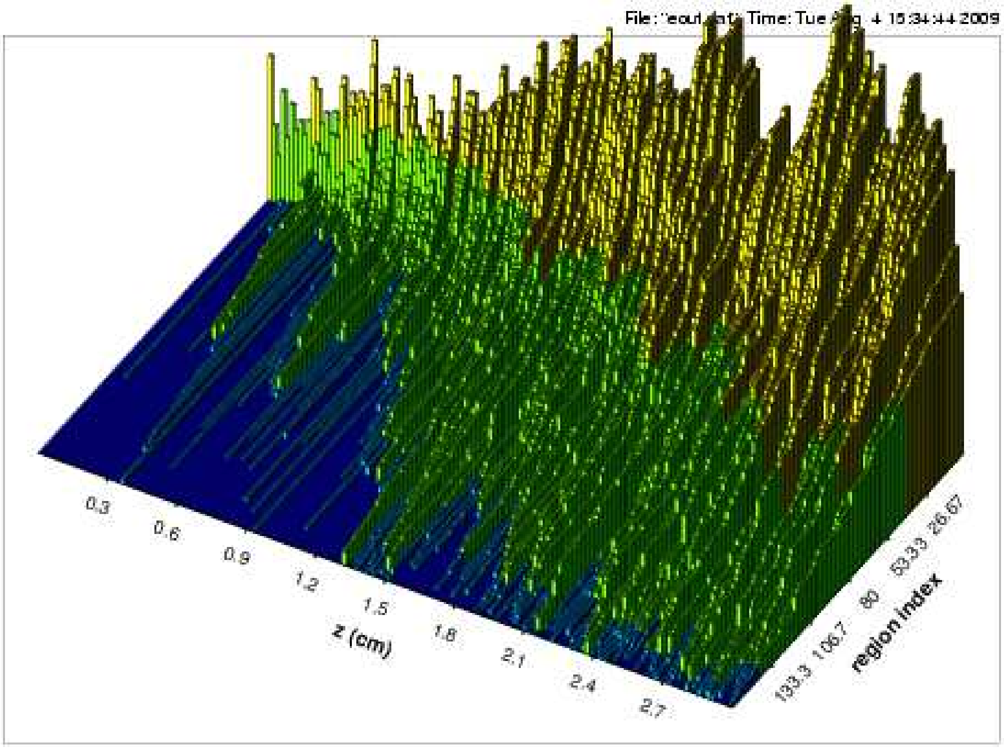

Figure 3

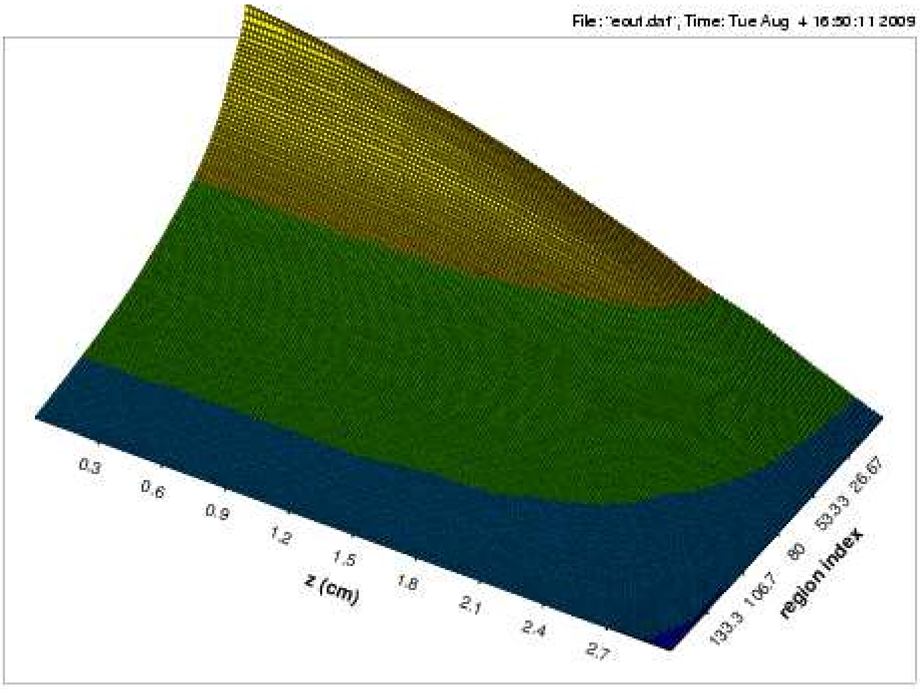

Figure 4

2.2. Problem with EGS4 KEK Version

Everything worked with the programs that have relatively small size of the source codes, but when we tried to execute the program for simulation of dark current in the Tesla Main Linac, it occurred that the capacity of this program is bigger than the internal storage capacity in MORTRAN and the execution failed already on the step of production of FORTRAN code. That’s why the decision to install EGSnrc was made.

2.3. An Attempt to use EGSnrc Code system

Installation of EGSnrc on Linux is rather complicated in comparison with the installation of EGS4 KEK on Linux, because the installation script is large, not well commented and, additionally, does not recognize the shell on DESY Linux computers. So, as it was the most important to check the capabilities of MORTRAN version included in EGSnrc, I installed EGSnrc on Windows because of the simplicity. What was needed is just to download certain files from the EGSnrc home page and the self-extracting installation wizard script and then run this script.

After installation I defined some environment variables and used egs_gui program (a graphical user interface) for testing. Good news is that their MORTRAN was able to produce

FORTRAN output from the code used for Tesla Main Linac simulation. But unfortunately, I was not able to run this program in the EGSnrc environment. Accurate analysis of the EGSnrc documentation showed me that probably it is not possible at all, because EGSnrc developers decided not to support what they consider as “bad coding practices”, especially replacement of EGS internal common blocks or subroutine COMINs. As, for example, we can’t simulate dynamics of electrons in the TESLA acceleration cavities without having laboratory time as a particle coordinate and this coordinate was introduced in EGS internal common block STACK, the program requires many changes in order to be executed in EGSnrc environment, and it is not even clear now if such changes are possible at all.

As the next step I tried to combine MORTRAN source file from EGSnrc with EGS4 KEK, but it didn’t work either, because changes in the source codes of both MORTRANs are different. In principle, it could be possible to make needed changes in the KEK MORTRAN, but taking into account that there are no comments in the source code it seems to be extremely complicated. For example, after changing in the KEK MORTRAN sizes of the buffers to the same values as used in NRS MORRAN version, the code was able to produce FORTRAN output, but with the errors in FORTRAN syntax.

3. Summary

The survey of available now versions of EGS code system showed that they can be obtained from KEK and NRC websites. The most updated version of EGS KEK code system, EGS5, can’t be used wih the old programs, but both EGS4 KEK code system and EGSnrc code system were installed. Whereas EGS4 KEK code system works with programs, that have relatively small size of the source codes, it fails to execute programs of bigger capacity. For EGSnrc code system the size of the program doesn’t play a big role, but there a lot of other difficulties connected with EGS internal common blocks and the program requires many changes in order to be executed in EGSnrc environment.

Nevertheless EGS4 KEK was checked on relatively small programs and can be used for not very complicated codes, for example for propagation of particles in the slab or scattering after impact in collimators.

Concerning the TESLA Main Linac code, I think, that it would be better to install EGS5 code system, but it will take much time to rewrite MORTRAN code to FORTRAN code.

4. Acknowledgements

I would like to thank my supervisors: Winfried Decking and Vladimir Balandin for a brilliant support during my stay at DESY, tolerance and understanding. And of course my gratitude to Professor Meyer for the possibility to come and work in DESY.

My time as a summer student was a great experience that provided me with the skills and knowledge.

It was a pleasure to work here.

5. References

1. **V. Balandin, R. Brinkmann, K. Flottmann, N. Golubeva, *Studies of Electromagnetic Cascade Showers Development in the TESLA Main Linac Initiated by Electron Field Emission in RF Cavities*, DESY TESLA 2003-10, 2003.**
2. **Kawrakow, E. Mainegra-Hing1, D.W.O. Rogers2, F. Tessier1 and B.R.B. Walters1, *The EGSnrc Code System: Monte Carlo Simulation of Electron and Photon Transport*, NRCC Report PIRS-701, 2009**
3. **Hideo Hirayama,Yoshihito Namito, Alex F. Bielajew, Scott J. Wilderman, *The EGS5 code system*, SLAC-Report-730, 2007**
4. **Kawrakow, E. Mainegra-Hing, D. W. O. Rogers, *EGSnrcMP: the multi-platform environment for EGSnrc*, NRCC Report PIRS-877, 2006**
5. **http://rcwww.kek.jp/research/egs/kek/**
6. **http://www.irs.inms.nrc.ca/EGSnrc/EGSnrc.html**