Journal of the Arkansas Academy of Science

Volume 49 Article 24

1995

Energy-Loss Particle Identification in 2-D Silicon Drift Detectors

G. Douglas Maudlin
University of Arkansas at Little Rock

A. A. Rollefson
University of Arkansas at Little Rock

Wilfred J. Braithwaite
University of Arkansas at Little Rock

Recommended Citation

Maudlin, G. Douglas; Rollefson, A. A.; and Braithwaite, Wilfred J. (1995) "Energy-Loss Particle Identification in 2-D Silicon Drift Detectors," *Journal of the Arkansas Academy of Science*: Vol. 49 , Article 24.
Available at: http://scholarworks.uark.edu/jaas/vol49/iss1/24

Energy-Loss Particle Identification in 2-D Silicon Drift Detectors

G. Douglas Mauldin, A. A. Rollefson and W. J. Braithwaite
Department of Physics and Astronomy
University of Arkansas at Little Rock
Little Rock, AR 72204

Abstract

A relatively new type of transducer known as the Silicon Drift Detector (SDD) has been fabricated onto thin silicon wafers. SDD operates like a miniature, high-resolution, 2-D Time-projection chamber. One of these devices can detect two dimensions of an ionizing particle's position, and its integrated electrical charge output level is proportional to the particle's energy loss through the silicon. An array of SDD's, arranged in three coaxial cylinders, is being considered as part of an instrument surrounding the beam pipe of highly-relativistic colliding beam facility, where it would be used to simultaneously track individual paths of thousands of charged particles emerging from each primary collision. Energy-loss data from the (x,y) pixels of each track allow individual particle identification as an electron, pion, kaon or proton. CERN's Monte Carlo modeling program, GEANT, is being used to predict energy loss at high statistical accuracy to account for high-energy tailing of the more prevalent pions. GEANT has been installed on a Linux workstation in Little Rock. Speeding up the modeling process is being investigated using parallel virtual memory techniques and groupings of Linux workstations.

Introduction

The Silicon Drift Detector (SDD) is a solid-state device proposed for obtaining trajectory and energy-loss data from ionizing particles such as those emerging from the collision vertex of a high-energy nuclear physics experiment as proposed by the STAR Collaboration (1992) for use at the Relativistic Heavy Ion Collider. The sensitive volume of an SDD detector is a wafer of silicon, and it employs the phenomenon of electron drift through the silicon.

The SDD represents a new detector technology intended as a solid-state version of the well-known time-projection chamber or TPC (Marx and Nygren, 1978). Although not seen as a replacement for the TPC, the SDD could offer greater resolution over a smaller region of space for applications such as precise location of central collision vertices in a collider experiment.

To place the evolution of the SDD in proper perspective, a brief description of the TPC follows. A TPC is an electronically-instrumented ionization chamber. It is essentially a box filled with a gas mixture at atmospheric pressure that is chosen for its ionization characteristics. An electric field is produced along two opposite walls of the chamber between parallel conductive planes. An ionizing particle traversing the gas volume leaves a trail of free electrons, which drift toward the positively-charged side of the chamber, quickly reaching a terminal velocity in the presence of the gas. Some secondary electron emission occurs along the way to the positive plane, which is a matrix of isolated and individually-instrumented plates called pads. The coordinates of the pads provide two spatial dimensions, and analysis algorithms use drift time to infer the third spatial dimension of the track. Typically, energy loss in the TPC is only a fraction of a radiation length, so traversing the TPC has little effect on particle tracks, except at the lowest momenta.

Unlike the TPC, the SDD is a slab of solid silicon. Thin metallic electrodes deposited onto the surfaces of this slab are used for biasing, focusing, and signal readout. Applied bias voltages deplete the silicon of mobile charge carriers. A charged particle passing through the slab will promote atomic electrons of the semiconductor material into its conduction band, thereby creating electron-hole pairs along the path of the particle. The holes are swept away by the focusing electrodes and the electrons drift along a potential channel in the silicon, until they are collected by a segmented readout electrode at one end of the silicon slab and detected by the readout electronics.

To keep this detector from causing appreciable changes to the particles' momenta, the SDD is made very thin, essentially planar, and only two spatial dimensions are measured. A line of pads is used instead of a plane, and time-projection is used to obtain one more dimension. In a practical detector design, an array of SDD's would be used to obtain multiple points to define the helical trajectory of a particle.

In addition to position information, the semiconductor-solid-silicon SDD detector can measure statistically-significant energy loss by a particle traversing its sensitive volume. This measurement can be used to identify particle types (electrons, pions, kaons, protons) for a range of momenta, with the main difficulty in identification being the statistical fluctuation in energy losses for individual particles about the mean value and the likelihood that the energy loss value of a particle could lie in a region characteristic of a different particle type. Obtaining a physically realistic picture of the distribution of energy loss values (traversing silicon) for pions, the predominant particle type anticipated in high-energy collider experiments, is essential for the construction of a suitable identification algorithm which is the focus of future work.

Methods

The passage of charged particles such as pions through matter is governed primarily by atomic physics. Most of the energy loss and scattering effects that occur are due to the particle's Coulomb interactions with bound atomic electrons of the matter in its path. Being much heavier than an electron, a pion suffers a small energy loss and undergoes a small deflection in an interaction with and electron. The electron is excited or ionized by the energy transferred to it in the collision.

Consider a well-collimated beam of pions passing through a slab of silicon. Each pion interacts with a great number of electrons along it path, the small energy losses and deflections add statistically, such that as the beam emerges from the slab, it is no longer monoenergetic and has an angular spread. The energy losses of these particles are distributed about a mean energy loss value.

If the silicon slab is sufficiently thick, some of the pions are stopped. The thickness required to stop a particle is its range in the material. At a thickness R_0, called the mean range, half of the particles are stopped, and at some greater thickness essentially all the particles are stopped. The fluctuation in range is called straggling. Sometimes straggling is used to refer to the fluctuation in energy loss.

Since one of the design goals of a vertex tracking device using SDD detectors is to have minimal effect on the trajectories of the particles it tracks, the energy-thickness of the silicon slabs are chosen to be a fraction of a radiation length for pions, and, thus, well below that which would cause a significant number of pions to be stopped. Thus, pions and other particles can pass through the SDD detectors on their way to other detectors like TPC's which they would minimally affect.

The mean energy loss of a beam of particles traversing an absorber can be expressed as -dE/dx, where -dE is the energy lost over a distance dx in the absorber. An approximate value for -dE/dx can be obtained using the Bethe-Bloch equation. The energy loss is proportional to the electron density of the absorber and the square of the particle charge. The Bethe-Bloch equation breaks down below the energy for which the particle's velocity is less than that of the atomic electrons in the absorber (Frauenfelder and Henley, 1991).

Different statistical distributions of dE/dx are predicted for different particles and energy regimes. These include the Landau Distribution, The Vavilov Distribution and the (generic) Gaussian Distribution. Fig. 1 shows predictions for two different types of Landau distribution, one (flawed) Vavilov distribution, and a Gaussian distribution, using the Center for European Nuclear Research (CERN) detector modeling program GEANT (GEANT, 1994). These plots were generated using an earlier version of GEANT (version 3.15) in which the distribution could be selected via the IMODE variable. This variable is not recognized by the newer versions of GEANT (e.g., version 3.21).

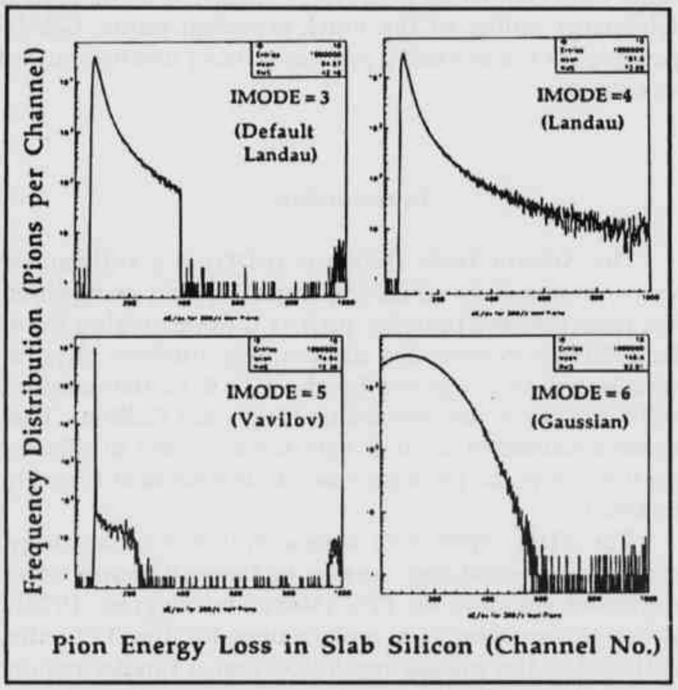

Fig. 1. Energy loss predictions for two different types of Landau distributions, one (flawed) Vavilov distribution, and a Gaussian distribution, using an early version of the CERN detector modeling program (GEANT, 1994).

To simulate the statistical fluctuations in energy loss suffered by particles traversing matter, CERN's program GEANT uses a Monte Carlo method. Monte Carlo refers to a numerical technique that accounts for randomness in a physical process through the use of random numbers, and is named for the well-known casino resort city where the element of chance plays a key role.

In a Monte Carlo program, random numbers are passed through a modeling algorithm to produce simulated data. These values can then be analyzed as if they were produced by the physical system being modeled. For example, they can be arranged into an occurrence distribution and graphed as a histogram. The extent to which such a simulation represents the modeled system is affected both by the validity of the algorithm chosen and the quality of the random number supply.

In practice, detector modeling programs are complicated, but well-connected researchers need not write their simulation software entirely from scratch. The management of CERN, the European Organization for Nuclear Research, makes it software library, known as Cernlib, available for use to members of the High Energy Physics (HEP) community. The Cernlib subroutines for detector descriptions are found in a Cernlib package known a GEANT, whose name is said to be a contraction of "geometry and tracking." Perhaps not by happenstance, GEANT is also the French word for "giant." These and other subroutines in Cernlib, along with various necessary user-generated subroutines, can be used to handle the many aspects of a practical detector simulation, including particle tracking, physics processes, histogram booking, and other functions. Strictly speaking, GEANT comprises only the parts of a simulation that deal with detector description, but it is common to speak of the entire program as a "GEANT simulation."

Cernlib software consists primarily of libraries of subroutines but also includes some complete programs, most notably the Physics Analysis Workstation (PAW, 1994) for data manipulation and presentation. Cernlib is made available for a variety of platforms ranging from mainframe computers to workstations, including PC-clone systems running Linux, which is a clone of the UNIX operating system (Johnson, 1994). Such a system can be assembled for a fraction of the cost of other kinds of platforms used for this purpose. The emergence of this no-cost UNIX that runs on low-cost hardware prompted a group of physics researchers in the former Soviet Union to "port" (move to a different computing platform) a version of GEANT and other essential Cernlib subroutines and programs to run on this kind of system. GEANT programs under Linux are functionally equivalent to those on other platforms, including mainframes and supercomputers. Being well received by the worldwide high-energy physics community, the Linux port is now an officially-sanctioned part of the CERN Program Library.

Most of the simulation work presented was performed on a Linux system constructed and administered by the first author. The choice to use a personal system instead of a University-owned computer involved a trade-off of computing speed for control. Being the administrator of one's own system enables one to quickly make needed changes that would be at best time-consuming and difficult if not impossible to obtain on a system administered by others. Furthermore, a personal computer with one user can devote essentially all its resources to the one simulation program. Dedicating a Linux system to one task effectively narrows the real-time performance gap between the Linux system, and they can be slower, depending on the load imposed by other users. Since total control over the system could be had for an acceptable sacrifice in performance, the choice of a Linux system for this project was clear.

The computer system used for this work was assembled from components chosen for optimal performance under Linux. The Intel 80486DX-50 microprocessor was chosen over the only high-speed alternative at the time, the 80486DX2-66, due to reports by other Linux users that its overall performance is better. For Linux use, the faster CPU of the DX2-66 is more than offset by the slower 33-MHz bus rate when compared with the DX-50 which runs at 50 MHz both internally and on its bus. The Pentium was not considered because no compiler optimized for it had been developed for use under linux. Primary storage consists of eight megabytes of random-access memory (RAM) and a 256-Kilobyte memory cache. Secondary storage is provided by a 1080-megabyte Western Digital fixed-disk drive. A Colorado Memory Systems tape dive in QIC (quarter-inch cassette) format is used for off-line storage and backup. A local-bus video board with two megabytes of video RAM drives a 14-inch color display with resolution of 1024 by 768 picture elements (pixels) which is adequate for the graphics requirements of CERN's Physics Analysis Workstation software.

There have been recent developments in parallel processing techniques using clusters of UNIX workstations. One implementation of this is called Parallel Virtual Machine, or PVM (Beguelin, et al., 1993). This consists of a suite of functions that can be called from FORTRAN or C programs and is portable to a number of platforms including Linux. Since our GEANT simulation consists of many independent iterations of the algorithm, it appears to be a good candidate for a paralled processing method such as PVM. We have begun to implement this by preparing a second Linux machine and an Ethernet link to connect the two and acquiring the needed additional software, but a final version of a PVM system had not been implemented at the time of this writing. We intend to pursue this enhancement in the future.

One part of learning to write GEANT programs is understanding the complex flow of control among the many subroutines. The "main" program, supplied by the user, calls a set of subroutines in the proper order, some supplied by the GEANT library and some by the user, which call other subroutines, some supplied by the GEANT library but some supplied by the user. The GEANT manual provides a crude diagram attempting to show these relationships but falls short of adequate.

Other GEANT users have constructed diagrams which are more readable and intuitive (Roetzel and Braithwaite, 1993). A diagram showing subroutine calls made in the program used in this study is presented in Fig. 2. The arrows represent subroutine calls, each with its tail at the calling subroutine and its head at the called subroutine. Execution begins and ends in the main program. Each called subroutine returns control to the calling subroutine when finished. The time flow of the diagram is generally top to bottom. Not all available user subroutines had to be provided in the present program; those not needed are not shown.

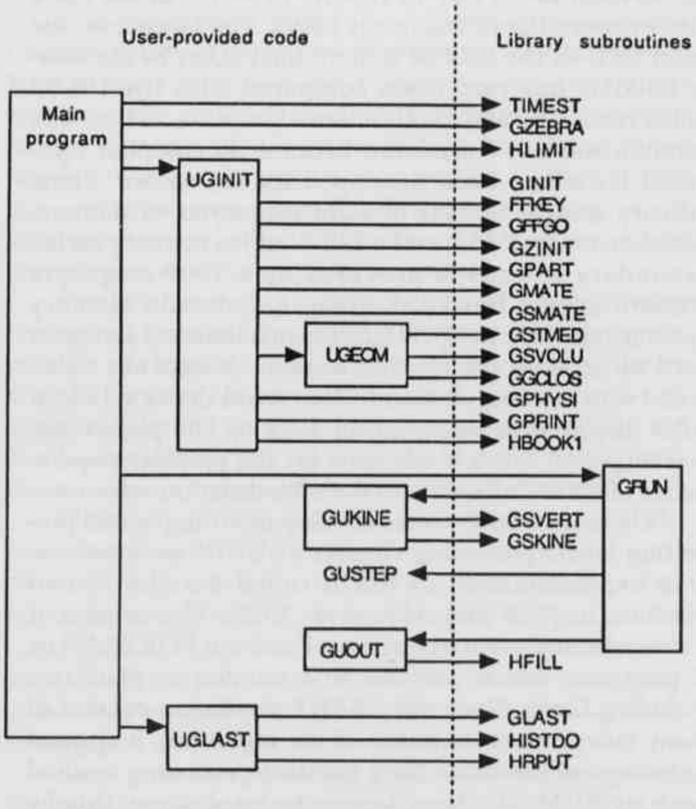

Fig. 2. Subroutine calls in currently used GEANT simulation Program.

In reading the diagram in Fig. 2, one may notice naming conventions for GEANT subroutines. All GEANT library subroutines begin with the letter "G." User-supplied subroutines begin with " UG" or "GU." For historical reasons, everything in FORTRAN is in uppercase type, and that convention is seen in the names of subroutines in the diagram and elsewhere in this work. Although most user-supplied subroutines are mandatory, some are optional and have default subroutines in the library which will be used if one is not supplied by the user.

An aspect of Cernlib software that is important to understand is the liberal use of COMMON blocks. The COMMON statement was provided by the designers of FORTRAN to add versatility. COMMON blocks are used by Cernlib to work around limitations in FORTRAN 77, to provide global variable space and dynamic memory allocation.

COMMON blocks are inherently dangerous because there is no protection against the uncoordinated alteration of a block's contents by different parts of a program. This can lead to undetected errors in the results of the program. In a complex programming environment such as GEANT, multiple programmers sometimes use the same COMMON for different purposes, thereby unwittingly damaging each other's data.

In addition to the choice of distribution, the program can take interactions with nuclei into account. The program simulates a particle's trajectory as a series of steps and calculates the probability of an interaction occurring at each step. The choice of step size is a tradeoff between accuracy and computing speed. The program calculates a step size, but the user can impose an upper limit.

The various user-selectable parameters available to the programmer in the version of GEANT current at the time this project was begun raised the issue of determining which permutations of these parameters would give realistic results. We proposed to evaluate these choices and validate the applicability of GEANT to the problem of measuring dE/dx of pions on silicon, comparing our results with experimental data which was to become available.

Results and Discussion

The problem of making suitable choices for the GEANT parameters affecting the results of a simulation involving energy-loss fluctuations was evidently a concern to other researchers, as evidenced by the changes that were made to GEANT by the CERN programmers during the time period of our work. In the newer releases of Cernlib and GEANT, the task of choosing a distribution of energy-loss fluctuations has been automated. The current GEANT program also makes automatic selection of other parameters such as step size.

Automatic selections override the user's input parameters, as evidenced by the lack of discernible difference in the resulting histograms from successive runs of our simulation program in which some of these parameters had been changed over their entire range of values. An example of these histograms is presented in Fig. 3.

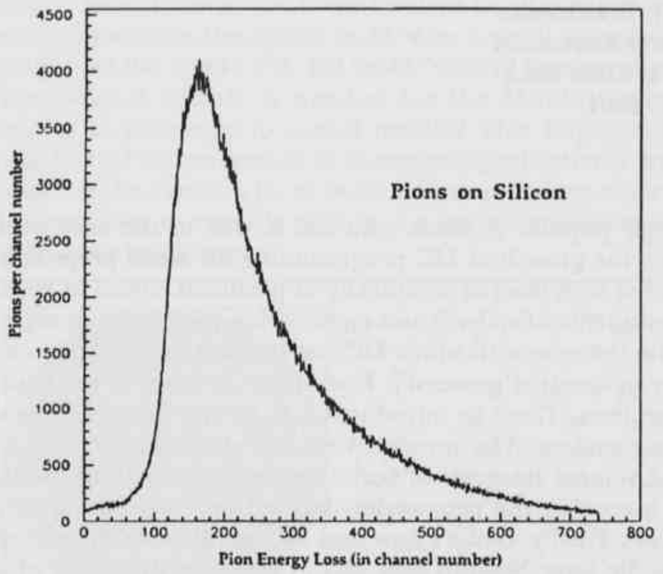

Fig. 3. A sample histogram of energy loss for one-million 300 MeV/c pions passing through 300 microns of silicon.

Issues addressed by this study have been recognized independently by the CERN programmers, confirming our concerns were valid. The focus of our study now shifts to the issue of whether the results produced with our simulation using these automated selection processes are realistic. As the production of experimental data for comparison has been delayed, this question remains to be answered.

Plans for future work include searching for means to override the automatic features in the newer versions of GEANT, re-running our simulations to prepare dE/dx distributions using the various permutations of variable parameters, awaiting experimental data for comparison with our simulation results, and speeding processing time with a parallel cluster of workstations using PVM.

ACKNOWLEDGMENTS.—This work was supported by grants funded by the Arkansas Science and Technology Authority, the U. S. Department of Energy (EPSCoR), and the Office of High Energy and Nuclear Physics, Division of Nuclear Physics of the U. S. Department of Energy under grant DE-FG05-92ER40753. This work recognizes the help of the STAR Collaboration (1992).

Literature Cited

Beguelin, A., J.J. Dongarra, G.A. Geist, W. Jiang, R. Manchek, K. Moore and V.S. Sunderam. 1993. The PVM Project. Available free from "ftp:netlib2.cs.utk.edu/PVM3/writeup.ps'.

Frauenfelder, Hans, and Ernest M. Henley. 1991. Subatomic Physics, 2nd ed. Prentice-Hall, New Jersey.

GEANT 1994. GEANT user's guide: Detector Description and Simulation Tool, CERN Program Library Internal document, CERN Data division, Geneva, CH.

Johnson, Michael K. 1994. Linux Information Sheet. Available free from "ftp:tsx11.mit.edu/pub/linux/docs/INFO-SHEET".

Marx, J.N., and D.R. Nygren. 1978. The Time Projection Chamber, Physics Today, (October 1978), P. 46-53

PAW 1994. PAW user's guide: Physics Analysis Workstation (PAW manual), CERN Program Library Internal Document, CERN Data Division, Geneva, CH.

Roetzel, D.L. and W.J. Braithwaite. 1993. User-interface coding for the CERN/ GEANT Nuclear Physics Program. Proc. Arkansas Acad. of Sci. 47:98-101.

Stallman, Richard. 1993. The Gnu Manifesto. Free Software Foundation, Inc. Available free from "ftp:prep.ai.mit.edu/pub/gnu/GNUinfo/Gnu-Manifesto".

STAR Collaboration. 1992. STAR Conceptual Design Review (original version). Internal document. PUB-5347. Lawrence Berkely Laboratory Technical Information Department, Berkeley, California 94720.