

# GENERAL PARTICLE TRACER: A NEW 3D CODE FOR ACCELERATOR AND BEAM LINE DESIGN

M.J. de Loos, S.B. van der Geer, FOM-Instituut voor Plasmafysica 'Rijnhuizen',  
P.O. Box 1207, 3430 BE, Nieuwegein, The Netherlands

## Abstract

We report on the particle tracking package General Particle Tracer (GPT). GPT is a new computer code to study charged particle dynamics in electromagnetic fields. To ensure both accuracy and speed, a fifth order Runge-Kutta method with adaptive stepsize control is used to solve the particle's equations of motion in the time domain. Any additional differential equations can be solved while tracing the particles. The code is completely 3D, including the space-charge model. GPT uses an input file language supporting variables and expressions to describe the simulated set-up. Additional features include the complete freedom in initial particle distribution and the flexibility to position and orient all beam line components individually. Separate utility programs are used to calculate macroscopic quantities, produce ascii/graphical output and automate parameter scans. GPT is written in ANSI-C for portability and runs on Unix platforms and PC's. Applications demonstrating the capabilities of GPT are presented.

## 1 INTRODUCTION

Particle tracing is a very powerful tool in the design of an accelerator or a beam line. The computers however used to be too slow to trace the amount of particles needed to obtain good statistics in a reasonable amount of computing time. Therefore the particle tracing method was often abandoned and especially matrix/optical methods became popular. These other methods however have their limitations because they do not simulate an actual beam. Especially when space-charge effects are important, most alternative methods are inaccurate or can not be used at all.

However because of enormous advances in computer technology, it is currently possible to trace thousands of particles through complex electromagnetic fields taking all 3D effects and space-charge forces into account. This offers much more accurate and reliable results than matrix codes.

The General Particle Tracer (GPT) offers all of these capabilities in a very user friendly and easy to customize package. Due to the general character of the code and its flexibility, GPT is suited for a large number of purposes. However GPT was developed to aid in accelerator and beam line design and this aspect will be highlighted here.

Section 2 describes the main features of GPT, and Section 3 analyzes selected cases that highlight the advantages of GPT. Section 4 is a summary and contains helpful information.

## 2 THE GPT CODE

GPT is a software package from Pulsar Physics [1]. Its purpose is to aid in the design of accelerators and beam lines by using modern, particle tracking techniques.

GPT was initially developed for the design of the FELIX accelerator [2] at FOM-Rijnhuizen. During the past years, the program has been intensively used for the design of both the FELIX and FEM accelerators [3].

GPT is a 3D particle tracer designed to simulate the movement of charged particles in the presence of electric and magnetic fields. The code is subdivided into three parts: a preprocessor to define the problem, a processor to solve it and a postprocessor to extract and interpret the results. The following sections explain various properties and specifics of these aspects.

### 2.1 GPT input file

One starts by describing the geometry and the initial particle distribution in a *problem description* file which we shall call a GPT input file. A built-in editor with on-line help is provided in the Windows 95 version. GPT allows the use of variables, functions and algebraic expressions within the input file, thus giving the user great flexibility in his input file.

GPT incorporates all support for automatic parameter scanning. In the near future, an optimizer and root finder will be implemented to further facilitate finding optimal settings for the beam line components.

### 2.2 Elements

The fields of physical structures are represented by so called elements. GPT comes with a large set of standard elements for basic structures such as solenoids, quadrupoles and accelerating structures. However, for specific cases custom elements can be easily added in separate C source files. To aid in their development, all standard elements are provided as an example.

Elements can be positioned anywhere and in any direction in 3D space. This gives the user complete freedom in constructing the experiment and also

components placed off center or rotated can be simulated.

Because GPT uses the actual electromagnetic fields of the beam line components, all fringe-field effects can be calculated automatically when (semi)-analytical expressions for the fields are provided. Also measured fields can be used in the simulation.

### 2.3 Beam generation

Because in reality there are usually billions of particles in a beam, it is virtually impossible to trace all their trajectories. Therefore GPT groups large numbers of particles together to form so called macro particles. These macro particles, representing the actual particles, are traced through the electromagnetic fields.

GPT has a number of functions for generating a set of particles or a particle beam. The most common electron guns are represented and particle distributions generated by other programs can be used. This provides complete freedom to generate an optimal particle distribution.

### 2.4 Space-charge

The default space-charge model calculates the space-charge forces from full 3D relativistic particle-particle interaction [4]. This ensures results with a high degree of accuracy. Custom space-charge models are also possible and add to the diversity of the package.

### 2.5 Solver

GPT calculates the three-dimensional relativistic equations of motion of particles as function of time. The equations depend on the particle distribution and the total electromagnetic field generated by the elements and the space-charge. The equations of motion are differential equations which generally cannot be solved analytically and are therefore integrated numerically using small steps.

The solver in GPT uses a fifth-order Runge-Kutta method with adaptive stepsize control [5]. The adaptive stepsize mechanism automatically adjusts the size of the integration steps as function of the electromagnetic field gradients. Thus the computing costs are kept to a minimum while the accuracy is not compromised.

The user can specify the overall accuracy for the position and momentum of the macro particles. The highest achievable accuracy is better than  $10^{-8}$ .

### 2.6 Additional differential equations

Additional differential equations can be solved while tracing the particles. Because the equations of motion of the particles and the user-specified differential equations are solved simultaneously, the results are self-consistent. The equations can be a function of time and/or position of the particles. Using this option provided by GPT, the

beam-loading in a linac and the radiation produced in an undulator have been calculated.

### 2.7 Data-analysis

GPT produces output consisting of all particle coordinates and velocities at specified times or positions, therefore no information about the particles is lost. Data computed by GPT can be saved in a file.

The data-analysis and plotting capabilities of the postprocessors allow the results to be processed and displayed in almost any format. Macroscopic quantities such as bunchlength, emittance and standard energy deviation can be calculated as function of any parameter.

### 2.8 Platforms

GPT runs on all PC's and UNIX machines. The code is written in C/C++ for portability. All input files, output files and custom elements are exchangeable between platforms. Effort was made to make GPT intuitively easy to use and many error checking routines are included to make it robust. Informative messages are displayed if the input is not correct.

Relatively new is a Windows 95 based user interface which incorporates a convenient editor, on-line help about GPT and its elements and simple ways to run GPT.

GPT is fully documented, including a tutorial introduction to the program and a reference of all elements. Due to an extensive network of users and the authors own experience with beam line design, support is fast and covers a large range of areas.

## 3 APPLICATIONS

GPT was the principle tool used for the design of both the FELIX and FEM accelerator. To demonstrate some of the main characteristics of the package, the following cases are examined: FELIX accelerator design, the effect of beam-loading in the constant gradient traveling wave linacs and the influence of radiation and space-charge in the FEM undulator.

### 3.1 Accelerator design

GPT was developed for accelerator design and this example demonstrates a few of its capabilities in this field. The FELIX accelerator consists of a standing wave prebuncher, a traveling wave buncher followed by two rf-linac sections [2]. Because of the low energy of the electrons emitted by the gun (100 keV), space-charge effects are very important in the first section of the beam line.

Figure 1 shows the influence of the buncher phase on the bunchlength, calculated at the entrance of the first linac. The optimal phase of the buncher, resulting in the smallest bunchlength, is obtained fully automatically using the GPT parameter-scan option.

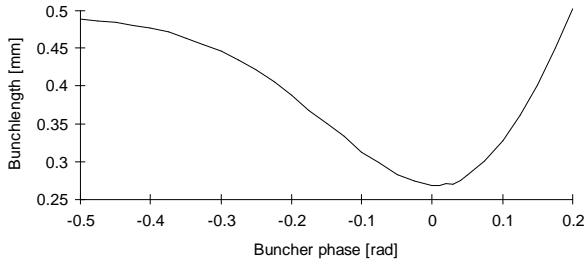


Figure 1: Bunchlength as function of the phase of the buncher.

### 3.2 Linac beam-loading

GPT has the possibility to solve additional differential equations while tracing the particles. Using this option the beam-loading fields in the constant gradient FELIX linacs are calculated directly from the particle trajectories. This method ensures that the beam-loading wave has the correct phase with respect to the particles.

Figure 2 shows the increase in energy when the beam passes through the buncher and the two linac sections. A quadrupole triplet is located between the linacs. Without taking the beam-loading effect into account, the energy would increase linearly in the structures. Figure 2 shows that the linacs no longer produce a constant acceleration due to the beam-loading field, which opposes the field of the RF fed to the accelerator.

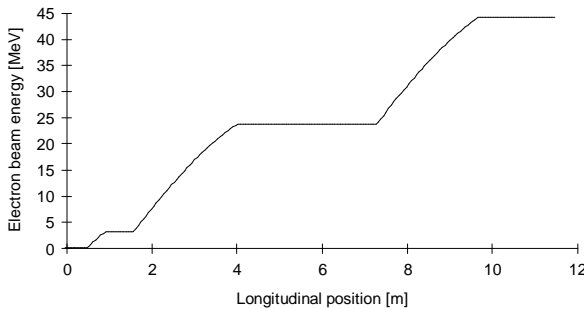


Figure 2: Simulation of the energy of a bunch travelling through the complete FELIX accelerator.

### 3.3 Undulator model

Also using the possibility of solving additional user-specified differential equations while tracing the particles, the trajectories in the undulators of the FOM Fusion FEM [3] were examined. This way, the effect of the microwaves on the particle trajectories is calculated and the output spectrum is obtained simultaneously. The method was used to optimize both beam transport and output power.

The differential equations for the power and phase of the various frequencies and modes in the waveguide were added. This enables multi-mode, multi-frequency examination of the modes in the FEM cavity and their influence on the particle trajectories.

Figure 3 shows the bunching of the electrons in the undulator. This effect is caused by the generated radiation fields in combination with the undulator fields. The 3D space-charge effect opposes this bunching and is therefore included in the simulation.

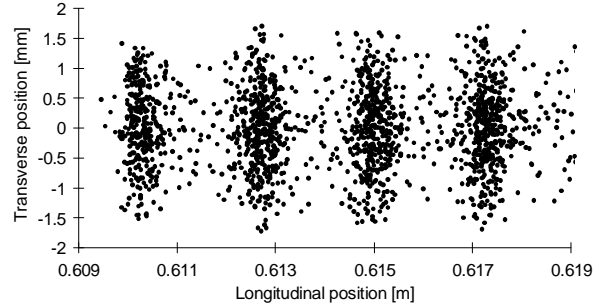


Figure 3: Example of bunching in the FEM undulator.

The most challenging part in the design of the FEM is the beam recovery system. The final beam energy distribution is crucial for the design of the collector. Figure 4 shows the energy distribution of the electron beam leaving the second FEM undulator. Most electrons have lost energy to the radiation fields, but the total energy distribution is relatively broad.

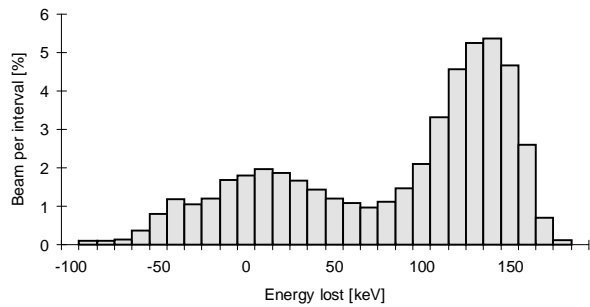


Figure 4: Energy spread of the beam after the undulators.

## 4 CONCLUSION

The main features of GPT were quickly reviewed. The high accuracy, the possibility to add custom code, the large number of standard elements and the user friendly interface, all make GPT an attractive analysis tool. Also the possibility to add self-consistent differential equations makes GPT a powerful tool for the design of accelerators and beam lines, and even FEL's. More information about GPT can be obtained by e-mail: [gpt@pulsar.nl](mailto:gpt@pulsar.nl).

## 5 REFERENCES

- [1] GPT User Manual, Pulsar Physics, Flamingostraat 24, 3582 SX Utrecht, The Netherlands.
- [2] P.W. van Amersfoort et al, Nucl. Instrum. Methods A 318, 42 (1992).
- [3] A. van der Linden, et al., Proc. of EPAC 94, London, 1994, pp. 849.
- [4] Ch. Bourat, CGR-MeV, Thesis 1988.
- [5] W.H. Press, et al., Numerical Recipes in C, Cambridge University Press, 2nd edition, 1992, Chapter 16.