RECENT DEVELOPMENTS IN THE ACCELERATOR DESIGN CODE PARMILA*

H. Takeda and J. H. Billen
Los Alamos National Laboratory, Los Alamos, New Mexico 87545 USA

Abstract

The PARMILA code, which originated in the 1960s for designing drift-tube linacs (DTLs), now designs and simulates the performance of many types of rf linear accelerator. The structure types include the DTL, coupled-cavity drift-tube linac (CCDTL) [1], conventional coupled-cavity linac (CCL), and several types of superconducting linac. This new code can handle multiple types of linac structures in a single run. This code features a more logically organized input sequence for the different linac structures and their properties. A PARMILA run can include sequences of beam-transport elements. In this paper, we describe the new user interface, highlighting the implementation of multiple rf structures. Also, we discuss the algorithm used for designing superconducting linac structures.

1 LINAC STRUCTURES

In designing a linac system, one first determines the types of linac (DTL, CCDTL, CCL, etc.). If a linac uses different rf structures, for example, one part of the linac requires a different number of cavities between focusing magnets then another, then we assign a new linac structure for that portion of linac. Starting from the low-energy end of the linac toward higher energy, up to 30 structures are allowed in an input file. The PARMILA code also calculates the beam-dynamics performance of the designed linac.

Each line of the PARMILA input file starts with a keyword, which in this paper we write in lower case letters in double quotes. At the beginning of the input file, we specify the global parameters on the “linac” line. Included parameters are: starting beam energy, bunch frequency, beam current, the beam particle mass and charge state. We specify whether the simulated particle distribution is read from a file created by another code (e.g. PARMTEQ [2] or a previous PARMILA run) with a “readdist” line, or if the distribution should be created anew by PARMILA according to data on the “input” line. After the global parameters for the whole linac are specified, each rf structure type starts with a “structure” line.

Figure 1 shows schematically the sections of a PARMILA input file. At junction between structures, the program keeps track of the ending state of beam characteristics in the linac. For typical room-temperature linacs, the inter-structure spacing is corrected so that the particle enters next structure with a correct phase. While the code designs the linac, it automatically creates a file of TRACE 3-D [3] elements which can be used for beam matching or optical element adjustment. For a multiple structure run, the sequence of rf structures follow the same order in the accelerator. Each structure is terminated as specified by a cell number on the “structure” line. Also, the particle distribution snapshots are stored in a file for later viewing with the DTLPROC postprocessor.

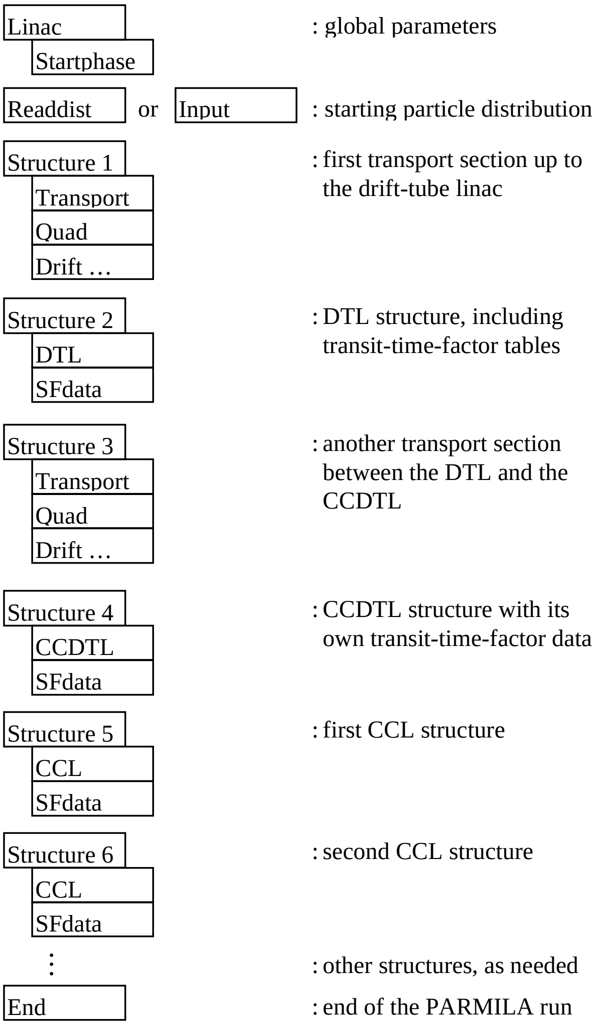
Linac
Startphase
: global parameters
Readdist
or
Input
: starting particle distribution
Structure 1
Transport
Quad
Drift …
: first transport section up to the drift-tube linac
Structure 2
DTL
SFdata
: DTL structure, including transit-time-factor tables
Structure 3
Transport
Quad
Drift …
: another transport section between the DTL and the CCDTL
Structure 4
CCDTL
SFdata
: CCDTL structure with its own transit-time-factor data
Structure 5
CCL
SFdata
: first CCL structure
Structure 6
CCL
SFdata
: second CCL structure
⋮
: other structures, as needed
End
: end of the PARMILA run

Figure 1. Schematic of a typical PARMILA input file.

*This work sponsored by the Division of Materials Sciences, US department of Energy, under contract number DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation for Oak Ridge National Laboratory.

2 INDIVIDUAL STRUCTURES

The keyword “structure” identifies and delimits the current linac structure. It includes the structure identification number, particle-dynamics termination cell number, frequency of the linac and the frequency used for the transit-time factors calculated by SUPERFISH [4]. All the phase coordinates of the initial particle distribution can be displaced by a specified amount if the rf module is driven by a separate power supply. Normally, one designs each linac structure one or more segments beyond the point where the beam-dynamics simulation ends. (Segments refer to contiguous sections of rf cavities, usually separated by focusing magnets.) This approach allows the code to calculate the distance between structures correctly. This can be specified by a design end energy parameter. For a CCDTL or CCL, one can design linac segments either with individual cell lengths proportional to particle velocity βc or with each segment made of cavities of identical length. In the latter case, the cell lengths are designed for a particle having the velocity β_gc, where the subscript g stands for geometrical.

One may specify several rf structure types such as “ccdtl”, “ccl”, “sc1”, and “sc2”. These keywords define parameters of a coupled-cavity drift-tube linac (CCDTL), a conventional coupled-cavity linac (CCL), or two particular types of superconducting cavity layout. For each of these structure types, PARMILA requires one or more tables of transit-time-factor data headed by one of the keywords “sfdata,” “sfdatae,” or “sfdataq.” The three tables correspond to different boundary conditions imposed upon the fields in the bore tube at the edge of the cavity. The “scheff” keyword (for space-charge effects) specifies the initial space charge mesh size in each rf structure section. Once the particle dynamics starts (after a “begin” line), PARMILA adjusts the mesh size automatically to fit the extent of the particle distribution. An “output” keyword directs the code to store particle distribution snapshots at specified cell positions. The multiple structure run stops after encountering an “end” keyword.

3 TRANSPORT STRUCTURE

The “transport” structure type contains only beam optical elements. The code does not design an rf cavity layout for this sections an input file. A file can contain several “transport” sections, for example to match the beam’s phase-space properties between accelerator sections. The user specifies transport elements such as “drift” and “quad” elements. After a “begin” keyword the code does the beam dynamics simulation through the specified optical elements. To assist designing the linac, PARMILA includes the transport elements in the TRACE 3-D input file that it automatically creates for the entire linac. A PARMILA run can consists only of a transport structure to simulate the beam dynamics. For a “transport” structure, the user can specify the starting design phase , if necessary.

4 CCDTL AND CCL STRUCTURE

The “ccdtl”, or ”ccl” line defines an rf structure of the same name. Both lines include data that specify the design termination energy, starting design phase angle, starting cavity field E_0, number of drift tubes per cavity (for the CCDTL), number cavities per segment, segment spacing, default quadrupole-lens parameters and other transverse focusing lattice settings. Other input lines (“extquad,” “change”) appearing after the “ccl” line or “ccdtl” line can be used to modify these default settings for specific rf cavities or magnets. There is considerable flexibility in the methods for defining the focusing magnet layout. For example, singlet or doublet quadrupoles can be placed a fixed distance from the end of each cavity segment, or each magnet location can be set individually. Also, their positions can be gradually moved within successive magnet spaces according to the beam velocity. The cavity field and amplitude can be set globally or automatically ramped using the “setramp,” “e0ramp,” and “phaseramp” lines. If one wants to maintain a constant real estate E_0T (accelerating field averaged over a transverse focusing period) across the segments, the “etfixed” line will compute the required acceleration.

After quadrupole and rf cavity definitions in the input file are the transit-time-factor tables. These tables list as a function of beam velocity βc several integrals computed by the SUPERFISH postprocessor SFO for a number of representative cavities. The “sfdata” table corresponds to symmetric cells, the “sfdatae” table is for cells at the edge of an internal cavity facing another cavity operated in the π mode, and the “sfdataq” table is for cell at the end of a segment where the electric field penetrates into the bore tube. The presence of these tables completes the linac design.

Once the linac is designed, one can modify the operating phase, amplitude, or quadrupole settings for a dynamics calculation. The “linout” line writes details of the designed linac to the output file. The “start” and “stop” lines tell the program the cell numbers on which to start and stop the beam-dynamics simulation. The “output” line specifies cell numbers for storing beam snapshots for later display by the DTLPROC postprocessor. If one simulates a well-matched 0-mA beam and saves the pseudo-particle coordinates at the center of each quadrupole magnet of a FODO lattice or a doublet lattice, the postprocessor DTLPROC can use the data to calculate the phase advance per focusing period along the linac.

5 SUPERCONDUCTING STRUCTURES

The “sc1” line defines a particular type of superconducting linac structure, which is characterized by a missing segment at regular intervals. The “sc1” line includes number of cells per segment, missing segment interval, the energy gain per segment, the design end energy, design phase, segment separation and default quadrupole settings. The “sc2” line defines a superconducting linac structure, which is characterized by groups of cavity segments with quadrupole magnets between groups. The “sc2” line includes the energy gain per segment, segment spacing, number of segments between quadrupoles, and the lattice type: either a “singlet” lattice or “doublet” lattice. Like the room-temperature structures, the quadrupole settings for both superconducting layouts can be customized by “extquad” lines. The transit-time-factor tables used in the superconducting cavities are calculated by the SUPERFISH code for a number of beam particle velocities βc for the cavities with an identical cell length β_gc/2.

6 SUPERCONDUCTING LINAC DESIGN ALGORITHM

There are important differences between conventional room-temperature linacs and superconducting (SC) proton linacs now being proposed for some projects. In a SC linac of the type being designed for the Accelerator Production of Tritium project, multiple segments driven by the same power supply deliver equal power to the beam. Thus, each equally powered segment gives the same energy gain. Also, the segment spacing is fixed for large portions of the linac. Many cavity segments have identical lengths so that only a few different cavity shapes are manufactured for the entire linac. Unlike the conventional CCL, most cell lengths in the SC linac will differ considerably from $\beta\lambda/2$ (the distance traveled by a particle of velocity βc in one rf period). Instead of the usual phase difference of 180 degrees from cell to cell seen in a CCL, the phase changes may be tens of degrees larger or smaller than 180 degrees. By using active phase shifters, the rf power for each SC cavity segment can be driven independently. Within a segment, which consists of a series of coupled cells operated in the π mode, the relative rf phase between cells is fixed. Because of the usual large bore diameter in SC cavities, the fields in the end cavities extend out into the bore tube. This field penetration results in an effective shift of the electrical center from its geometrical center. The shift can be as large as ~2 cm for a 700-MHz cavity. These phase shifts are all included in the PARMILA code.

Because the lengths and positions of the of the SC linac cavities are predetermined, the design of the linac in PARMILA means finding the correct rf phase and the rf power at which to operate each segment. Because of the large phase slip from cell to cell, we trace a test particle through each segment recording the center-of-gap phases and energy gains of the particle. The code then requires two conditions simultaneously: 1) the energy gain must equal the design energy gain for the segment, and 2) the average phase must equal the design phase. This iterative process is performed by a two-dimensional optimization routine in PARMILA. After calculating the amplitude of rf electric field applied on one segment, and the entry and the exit phases of that segment, the code can compute the rf drive phase for the next cavity segment based upon the distance between segments and the difference between the exit phase of the upstream segment and the entry phase of the downstream segment.

7 SUMMARY

Format of PARMILA code input file was upgraded. The code can design and perform beam dynamics simulation of multiple linac structures in a single run including DTL, CCDTL, CCL, and superconducting linacs. Each linac structure is logically separated in the input file and the state of the beam at the end of each structure is transferred automatically to next structure type in the beam-dynamics simulation. Currently, a comprehensive PARMILA code documentation is in preparation.

8 REFERENCES

[1] J. H. Billen et al., “A New Rf Structure for Intermediate Velocity Particles,” Proceedings of the 1994 Linear Accelerator Conference (August 21-26, 1994 Tsukuba, Japan).

[2] K. R. Crandall et al., RFQ Design Codes, Los Alamos National Laboratory report LA-UR-96-1836 (Revised May 20, 1998).

[3] K. R. Crandall and D. P. Rusthoi, TRACE 3-D Documentation, Los Alamos National Laboratory report LA-UR-97-886.

[4] J. H. Billen and L.M. Young, “POISSON SUPERFISH,” Los Alamos National Laboratory report LA-UR-96-1834 (Revised April 22, 1997).