

# FURTHER IMPROVEMENTS IN TRACE 3-D

D.P. Rusthoi, W.P. Lysenko, Los Alamos National Laboratory, Los Alamos, NM,  
and K.R. Crandall, Amparo Corporation, Santa Fe, NM\*

## Abstract

TRACE 3-D, an interactive beam-dynamics program that calculates the envelopes of a bunched beam (including linear space-charge forces) through a user-defined transport system, has undergone several upgrades in physics, coding, and capabilities. Recent modifications include centroid tracking (and misalignment capabilities) and an improved beam description that allows study of some nonlinear effects such as wakefields. The Fortran code has been made portable and runs on numerous platforms. It can be used with a variety of graphics packages. The additional beamline elements, new commands, expanded fitting capabilities, improved beam description, and coding modifications have extended TRACE 3-D's usefulness and applicability to the accelerator community. These changes are documented in the third edition of *TRACE 3-D Documentation*.

## 1 INTRODUCTION

TRACE 3-D, a first-order accelerator design code that was first written as a two-dimensional version in the early 1970's, has been expanded extensively over the years and adapted to run on numerous systems using different versions; however, there is now one official version that is documented and available on most workstations, PC, and Macintosh platforms. Some of the code's newer features are described below.

## 2 NEW CAPABILITIES

Included among new capabilities added to TRACE 3-D are centroid tracking, aperture calculations, quadrupole misalignments, additional matching types, a rotation/translation element, a wiggler element, vertical bend option, graphic scale flexibility, and a number of new commands. The addition of wakefield calculations is described in the following section.

### 2.1 Centroid Tracking

TRACE 3-D now tracks the transverse beam-centroids and plots them in the beam profile box ('+' for the  $x$ -centroid and '\*' for the  $y$ -centroid). The phase-space ellipse centers are also plotted on the corresponding phase-space plots. The code ignores centroid calculations unless an offset is entered or a beamline element causes the centroid to shift off-axis. The example in Fig. 1 shows the typical TRACE 3-D graphic page output with a beam offset generated from random quadrupole misalignments.

### 2.2 Aperture Calculation

The maximum extent of the beam inside each element is now calculated in the two transverse planes by adding to the beam centroids the transverse profiles multiplied by a user-supplied "aperture factor." Given these maximum extents within each element, TRACE 3-D calculates the minimum radius required for a circular beampipe that will transport the beam without scraping. At each point where the minimum radius is largest, the code writes the transverse profiles and the corresponding centroids to an aperture file (when the command is issued). For example, an aperture factor of 3.13 supplied by a user for a TRACE 3-D beam (normally  $\sqrt{5}$  times the rms beamsizes) will yield maximum-extent values for a  $7\text{-}\sigma$  beam.

### 2.3 Quadrupole Misalignments

The quadrupole element, which used to be specified by only two parameters (gradient and length), has been expanded to include three additional parameters specifying misalignment (rotation or translation) in the transverse plane. If translation is specified, transverse offsets in one or both planes may be entered; if rotation is specified, an angle specifying the amount of rotation about the beam axis may be specified. It is also possible to specify random misalignments (either rotation or translation).

### 2.4 Matching Types

Three new matching/fitting options were added, bringing the total of matching types to fourteen. One new matching type finds the values of two variables that produce a round output beam—identical magnitude Twiss parameters in  $x$  and  $y$ —useful when trying to get a circular spot on a beam target. Another new matching type varies the input beam-ellipse parameters (BEAMI) to obtain a specified output beam (BEAMF). A third type enables the user to adjust one or more variables in a periodic transport system to achieve a specified phase advance in one or more phase planes, e.g., setting the quadrupole gradients in a drift-tube linac to get a certain value for the zero-current phase advance in  $x$  and  $y$ . There may be three quadrupoles involved, but all identical, so the user might have "repeat" elements or use the variable-coupling option. If one wants to achieve a specified phase advance with current, an iterative technique is required because this option does not automatically keep the beam matched as it adjusts the variable.

\* Research supported by the U.S. Department of Energy, Office of Energy Research, through the Division of High Energy Physics and Division of Mathematical, Information, and Computational Sciences

## 2.5 Rotate / Translate Element

The rotate element was expanded to provide both a rotation and a translation capability (and is backward compatible with earlier files containing rotation elements). The expanded element allows the user to select a plane (viz.,  $x$ - $y$ ,  $y$ - $z$ ,  $z$ - $x$ ) specifying rotation about the axis normal to that plane (roll, pitch, yaw) and/or translation in that plane.

## 2.6 Wiggler Element

A new element was added to simulate a wiggler by using a series of alternating bends and drifts (bend-drift-bend-drift), where odd-numbered bends are defined as positive and even-numbered bends as negative. The user supplies the overall length of the wiggler, a field strength and dipole gap to be applied to each bend, plus the length of and distance between each bend.

## 2.7 Graphic Scales Flexibility

When input and output beams vary significantly in size, one can enter different scales for the input and output phase-space plots. This can be seen in the example of Fig. 1 where both transverse and longitudinal phase-space plots have different input and output scales. Additionally, phase-space plots can be generated at any point along the beamline by entering values for NEL1 and NEL2 (displayed at top of phase-space plots). Another option, useful for viewing portions of long transport files, is the profile close-up plot, generated by specifying values for the profile box, NP1 and NP2 (also displayed in Fig. 1).

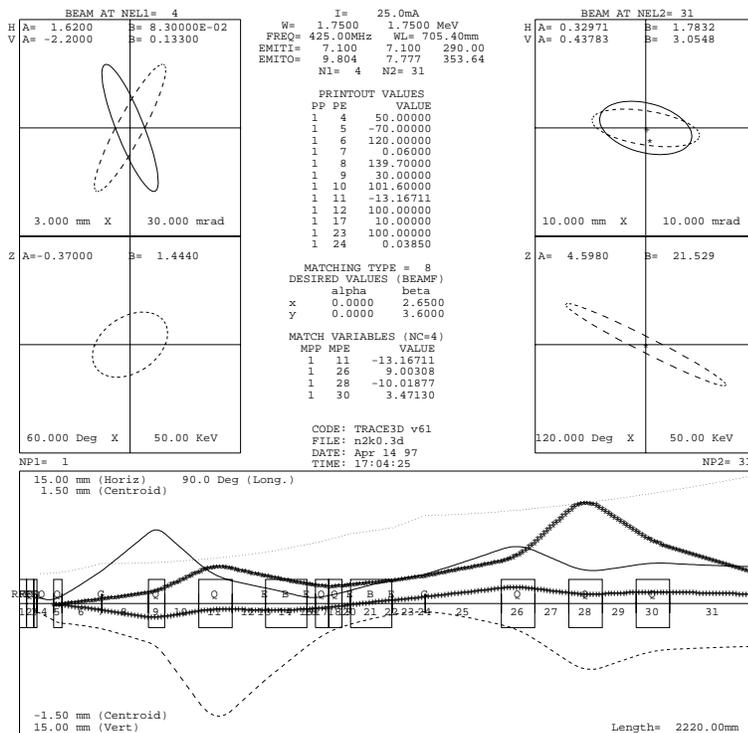


Fig. 1. Typical TRACE 3-D output showing beam envelopes in three dimensions and beam centroid profiles in  $x$  and  $y$  due to random quadrupole misalignments.

These plotting options do not affect beam-dynamics calculations.

## 2.8 New Commands

Several new commands were added, some of which are mentioned here. The *aperture* command, previously discussed, was modified not only to calculate the minimum radius of a circular beampipe required to transport the beam without scraping, but also to calculate the maximum beam extent for both transverse planes within (inside) each beamline element. An *unmatch* command was added to allow a user to conveniently return to pre-match values of matching variables should the results of a matching calculation provide an unacceptable solution. To check the parameter values of a particular element in the transport, a *query* command was added to obtain a display of parameter values of a specific element. Now that TRACE 3-D tracks beam centroids, the *centroid* command was added to enable a user to display the centroid at any point. A *mini-help* command is now available to assist when adding new beamline elements, and a *newfile* command brings in new datafiles to allow for comparisons.

## 3 NEW PHYSICS

### 3.1 Wakefield Calculation

There is a parallel version of TRACE 3-D that can simulate some nonlinear effects, such as wakefields, related to the variation of the beam bunch along the longitudinal direction [1,2]. This code is compatible with the standard version and will produce the same results for problems not involving wakefields. The two codes may be merged in the future.

In the wakefield code, the beam bunch is divided into a number of slices longitudinally. We describe each slice by its 6-D centroid and 6x6 sigma matrix, and follow the collection of centroids and sigma matrices down the beamline. Whenever the code needs to generate output or compute space charge, it combines the slices into a single 6-D centroid and 6x6 sigma matrix and uses the existing code structure. The bunch-slice centroids and sigma matrices are transported as usual by the various elements in the beamline. However, space charge and the new wakefield elements get a modified treatment. In the case of space charge, we take into account the effect of space charge on the slice centroids. Of course, the overall centroid is not affected by space charge. Wakefield

effects are taken into account with new zero-length elements, similar to how bend edges are handled. The wakefield element parameters specify wakefield functions, which determine forces acting on a given bunch slice caused by slices ahead of the given bunch slice. There are three wakefield elements: one each for monopole, dipole, and quadrupole wakefields. These wakefield elements are user-specified elements and may easily be changed to model other effects that depend on longitudinal position within a beam bunch. Unlike some existing particle codes, this approach allows for the distance between bunch slices to vary throughout the simulation. This is useful for studying wakefield effects in a regime where space charge is important.

### 3.2 Space Charge

It was found that when one tried to design an achromatic bend with current, the beam did not remain achromatic downstream. The problem was discovered in the space-charge calculation used when the beam ellipsoid was tilted with respect to the coordinate axes. The beam is first “stretched” in the longitudinal direction by multiplying the  $z$ -coordinate by  $\gamma$ . The beam is then rotated to be upright, the space-charge impulses applied, the inverse rotation applied, and the  $z$ -coordinate divided by  $\gamma$ . The beam “stretching” and “unstretching” transformations are now symplectic.

## 4 NEW CODING

### 4.1 Standard FORTRAN 77

Previously, different versions of the code were maintained in different formats for various computer platforms, resulting in machine dependencies and versions differing in capabilities and documented features. TRACE 3-D has now been rewritten using Standard FORTRAN 77 (with exception of the NAMELIST extension—now standard in Fortran 90)<sup>†</sup>, and is capable of being compiled on almost any system. This standard version of the code is maintained by the Los Alamos Accelerator Code Group and corresponds to the published documentation.

### 4.2 Plotting Routines

All external plotting calls have been placed in short subroutines at the end of the source code. The code can still be used with PLOT10 on systems using the Tektronix terminal or systems emulating a Tektronix 4000-series terminal. On Unix systems, we supply a set of replacement routines for the PLOT10 library that produce PostScript and run a PostScript viewer. More recently, we compiled TRACE 3-D with a set of “freeware” graphics available off the Internet from CalTech (<http://astro.caltech.edu/~tjp/pgplot/>) called

PGPLOT. The PGPLOT graphics routines have drivers written for X-Windows, the Macintosh, Windows95, Windows NT, and a host of other platforms. The code comes with two sets of plotting subroutines: one for PLOT10 users and/or PostScript replacement calls, and one for PGPLOT users.

## 5 CODE / DOCUMENTATION AVAILABILITY

*TRACE 3-D Documentation* [3], now in its third edition (May 1997), includes a description of the code, the transport elements and their transfer matrices, the dynamics calculations, and complete instructions for the user, including a number of examples. Fifteen appendices give detailed information on the physics and coding. The documentation is available on-line as a Microsoft Word document or as a PostScript file. Both code and documentation may be accessed by anonymous FTP via connection to the Los Alamos Accelerator Code Group FTP Server, *PC-AOT-1.ATDIV.LANL.GOV* (IP Address 128.165.32.190), with username *TRACEUSER* and password *ftptrace* (password is in lower case). The TRACE directory contains readme files and subdirectories for documentation, the source code, and downloadable executables for various platforms. A hardcopy of the documentation is provided upon request.

A special interface for the Macintosh [4], developed by the MACcelerator Project in Del Mar, California, has been written and is commercially available using the standard version of TRACE 3-D under a license agreement with the Los Alamos National Laboratory. Among other advantages, there is a minimal amount of alpha-numeric (keyboard) input used for setting parameter values, and no text or command-line input is required.

## 6 ACKNOWLEDGMENTS

The authors express their appreciation to members of the Los Alamos Accelerator Code Group who help maintain and distribute TRACE 3-D and its documentation, and provide technical assistance to users as needed.

## REFERENCES

- [1] W.P. Lysenko, D.P. Rusthoi, K.C.D. Chan, G.H. Gillespie, and B.W. Hill, “Wakefields in the TRACE 3-D Code,” Proceedings of the 18th International Linac Conference held August 26-30, 1996 in Geneva, Switzerland.
- [2] W.P. Lysenko, D.P. Rusthoi, K.C.D. Chan, G.H. Gillespie, and B.W. Hill, “TRACE 3-D Code Improvements,” Proceedings of the 1996 Computational Accelerator Physics Conference held September 24-27, 1996 in Williamsburg, Virginia.
- [3] TRACE 3-D Documentation, K.R. Crandall and D.P. Rusthoi, LA-UR-97-886, Los Alamos National Laboratory Report, May 1997.
- [4] G.H. Gillespie and B.W. Hill, “A Graphical User Interface for TRACE 3-D Incorporating Some Expert System Type Features,” 1992 Linear Accelerator Conference Proceedings (Ottawa) AECL-10728, 787-789.

<sup>†</sup> Old TRACE 3-D datafiles may need slight modification to be read by NAMELIST extensions on some Windows and Macintosh systems.