35th EPS Conference on Plasma Phys. Hersonissos, 9 - 13 June 2008 ECA Vol.**32D**, P-4.126 (2008)

Interpretation of laser-produced ion beam diagnostics using the PTRACE code

A. Schiavi*

Dipartimento di Energetica, Università di Roma "La Sapienza", Roma, Italy

In recent years many experiments have reported the production and the application as a diagnostic tool of laser-generated ion beams[1]. In particular, in the interaction of an ultraintense laser pulse with a thin solid target, protons and target ions can be accelerated to energies of the order of a few tens of MeV per nucleon. Laser-generated ion beams have typically a wide energy spectrum and contain up to 10^{11-13} ions per shot. The temporal duration of the ion beam source is comparable to that of the laser pulse used to generate it, and typically is about 1 ps long[2]. Moreover these ion beams are almost completely charge neutralised by an accompanying cloud of electrons stripped from the target. As a consequence the expansion of the beam is ballistic, making these ion sources suitable for diagnostic applications, such as the proton imaging technique.

A particle-tracing code, PTRACE, has been originally developed for helping scientists in the interpretation of experimental data, usually collected in the form of two-dimensional cross-sections of the ion-beam used as a probe. The code allows for tracing ions through user-prescribed electromagnetic structures, which can be described by analytical formulas or as field maps obtained from other simulation tools. The code also computes the deposition of the ion beam in nuclear track detectors or radiochromic film stack detectors for direct comparison with actual data.

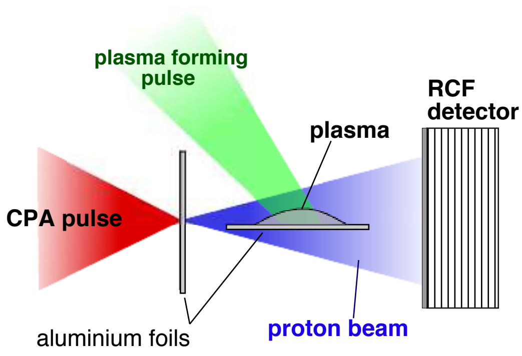

Figure 1: Top view of a typical experimental set-up for proton imaging.

The problem and its geometry

Figure 1 presents a schematic diagram of typical experimental arrangement for the generation of a proton beam using an ultraintense laser pulse. A chirped-pulse-amplified (CPA) pulse is focussed onto an thin foil of solid material (here an Aluminium foil) at an intensity in excess of $5 \cdot 10^{18}$ W/cm^2. On the back side of the irradiated foil the ions of the material are accelerated

*angelo.schiavi@uniroma1.it

via the target normal acceleration effect, resulting in a source of fast charged particles. When hydrocarbon contaminantes are present on the back side of the foil, the emitted beam consists mostly of protons, with energy reaching typically a few tens of MeV. A secondary target is usually placed in the propagation cone of the proton beam, and irradiated by a laser pulse in order to produce different plasma conditions. The intense quasi-static electromagnetic fields arising in the laser-plasma interaction deflect the trajectories of the protons which pass through and around the plasma target. Subsequently the proton beam is stopped in a stack detector, which records the spatial distribution of the protons for different energy intervals. Deconvolution from energy to time-of-flight for each energy component allows for the reconstruction of the temporal evolution of the e.m. field structures generated in the interaction.

The tracer

PTRACE is a particle-tracing code originally developed for simulating the propagation of the fast protons from the source, through the target region, and then up to the detector. There the two-dimensional dose map deposited in each layer of the detector is evaluated and finally matched against the actual experimental data obtained from the stack detector. At the core of PTRACE there is a differential equation solver that computes the trajectory of a particle in presence of electromagnetic fields. The numerical algorithm chosen is a Runge-Kutta fourth-order algorithm coupled with an adaptive stepsize monitoring routine[3]. The adaptive stepsize routine assures that the time steps at which the dynamics is sampled are smaller where the acting forces are larger, so that computational resources are well managed during the simulation. PTRACE is a `C++` code divided into several objects that combine functions and routines with the data structures they are acting upon. Figure 2 shows a schematic diagram of the code. The `Particle` object specifies mass and charge of the particles to trace. The `Source` generates the particles with initial conditions (position and velocity in 3D space) according to parameters such as the axis and divergence of the propagation cone and energy distribution of the particles prescribed via the `Spectrum` object. Particles are passed one at a time to the `Tracer` which integrates the trajectory evaluating at each step the force acting on the particle. Finally the detector response and several diagnostic outputs are evaluated by the `Renderer`.

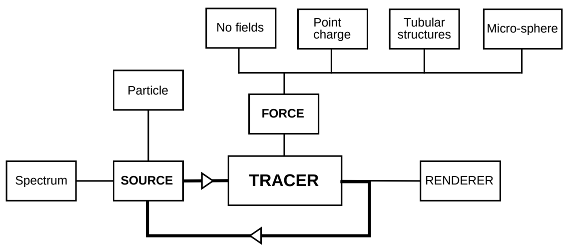

Figure 2: PTRACE flow chart. Main loop is shown together with object components.

The fast ions are charge neutralised by an accompanying cloud of electrons, and their number density is orders of magnitude lower than that of plasma ions in the region close to the target. These two observations suggest that the probing ions propagate independently of each other, under the sole force generated by the e.m. field structures. By neglecting collective effects, the code can be easily parallelised and has been implemented by using multiple threads on multi-core systems or by adopting a master-slaves paradigm for running on distributed-memory clusters.

The modules

The e.m. fields are described in the code using objects which contain information about the spatial and temporal evolution of the fields. Some of them are: the static point charge, the pulsed point charge ($q = q(t)$), the charged wire, the bundle of wires, the pulsed charged sphere with radial filaments, the stopping mesh, the charged mesh, the charged cylinder, the magnetic torus. Typical parameters common to all modules are position and orientation in 3D space, temporal evolution of the E and B fields, spatial orientation of field lines, maximum or reference value for the fields, etc. At each step the net force is obtained as the superposition of all forces associated to each module, and the particle's position is advanced. Since the onset of e.m. fields can be localized to a very small region in space and to a very short interval in time, each module is interrogated by the tracer in order to determine the maximum allowed timestep for a correct sampling of the dynamics.

The detector

The detector is described by a stack of layers of different width and material composition. The energy deposited into each layer is computed using the Transmission Matrix model [4], which is an efficient compromise between the accuracy of a Monte-Carlo tracing code and the computational speed of the continuous slowing-down approximation (CSDA). The input library for this part of the code has been generated using the ion-tracing code `SRIM` [5].

Some applications

`PTRACE` was successfully used in the interpretation of several experiments in ultra-intense laser-plasma interaction using the proton imaging technique. For instance, it was possible to reconstruct for the first time the charging-up and discharge of a laser-irradiated microballoon on a ps timescale [2]. Figure 3 presents a series of snashots of a proton beam propagating through the microballoon: it is possible to see the hole opened up in the leading part of the beam by the positively charged-up sphere at the centre of the beam. The code was also used in the characterisation of the proton source using mesh magnification experiments, and long-lived post-solitonic e.m. structures were identified in the wake of an ultraintense laser beam through underdense plasma combining 1D profiles from PIC simulations with 3D particle-tracing runs.

Future developments

Given the flexibility of the original project, the `PTRACE` code can be used both for accurate data reconstruction and for quick-and-easy experiment planning. The main limitation to the widespread use of the code is that its high-versatility is accompanied by a steep learning curve. In order to overcome this problem, a graphical front-end for the definition of the problem and for the visualization of the results is under-developement. An easy-to-implement scheme for defining custom modules combined with an equation parser for inputting field profiles using simple analytical expression will further simplify the use of this simulation and data analysis tool.

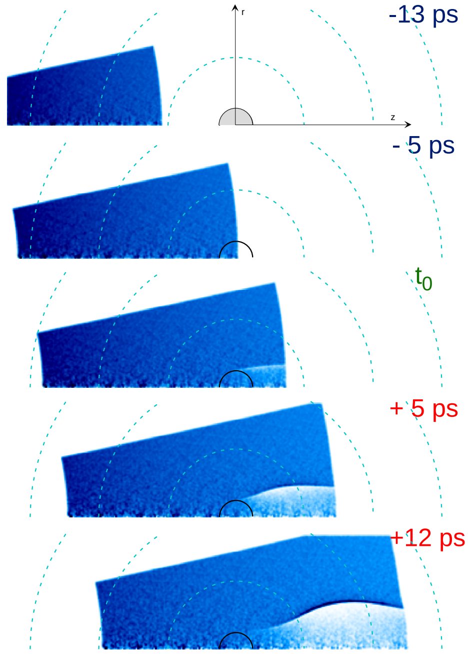

Figure 3: Radial projections of a proton beam with energy between 9.3 and 2.5 MeV.

Acknowledgements

I would like to express my gratitude to M. Borghesi and his collaborators for many fruitful discussions and for suggesting always new ideas and improvements to the code. This work was partially funded by the Italian Ministry of University and Research under the FIRB “BLISS” project.

References

[1] R.A. Snavely, M.H. Key, S.P. Hatchett et al, Physical Review Letters **85** (2000) 2945-2948; M. Hegelich, S. Karsch, G. Pretzler et al, Physical Review Letters **89** (2002) 085002

[2] M. Borghesi, L. Romagnani, A. Schiavi et al, Applied Physics Letters **82** (2003) 1529-1531

[3] W. Press, S. Teukolsky, et al. *Numerical Recipes in C - The Art of Scientific Computing - Second Edition*. Cambridge University Press, USA, 1994.

[4] A. Schiavi, *Computer Physics Communications*, 178(1):35–39, 2008.

[5] J. F. Ziegler, J. P. Biersack, and U. Littmark. The stopping and range of ions in matter, 2003. URL `http://www.srim.org`.