

# Physics of High Intensity Laser Plasma Interactions

## Part V: Tutorial on Particle-in-Cell Simulation

20–25 June 2011 | Paul Gibbon

# Lecture 5: Tutorial on Particle-in-Cell Simulation

The PIC code BOPS

Prerequisites

Installation

Running BOPS

Project I: Laser wakefield accelerator

Project II: Ion acceleration – TNSA vs. RPA

Useful formulae

# The PIC code BOPS

- BOPS is a one-and-three-halves (1 spatial, 3 velocity coordinates: 1D3V) particle-in-cell code originally created by Paul Gibbon and Tony Bell in the Plasma Physics Group of Imperial College, London.
- Based on standard algorithm for a 1D electromagnetic PIC code from Birdsall & Langdon.
- Optionally employs a Lorentz ‘boost’ along the target surface to mimic 2D, periodic-in-y geometry, with big savings in compute time.
- Applications: absorption, electron heating, high harmonic generation, ion acceleration

# Prerequisites

- Current version: BOPS 3.4
- Download site:  
<https://trac.version.fz-juelich.de/bops>
- Compiler: sequential code written in Fortran 90 (Intel ifort or GNU's gfortran)
- Run scripts provided are designed for generic Unix systems: Ubuntu, SuSE, RedHat and also MacOS.

## Linux and MAC users:

Unpack the tar file (name may differ) with:

```
tar xvfz bops3.4.tar.gz
```

and cd to the installation directory bops.

# Windows users

**1** First install CYGWIN ([www.cygwin.com](http://www.cygwin.com)). This creates a fully-fledged Unix environment emulator under Windows. In addition to the default tools/packages offered during the installation you will also need:

- Gnu compilers gcc, g77, g95 etc. (*devel* package)
- make (*devel* package)
- vi, emacs (*editors* package)
- X11 libraries if you want to generate graphics directly under cygwin (eg using gnuplot)

If you forget anything first time, just click on the Cygwin installer icon to locate/update extra packages.

**2** Download and unpack the bops3.4.tar.gz file with an archiving tool (eg PowerArchiver: [www.powerarchiver.com](http://www.powerarchiver.com)). Put this in your 'home' directory under CYGWIN, e.g.:

```
C:\cygwin\home\<username>\
```

**3** Open a Cygwin terminal/shell and 'cd' to the bops directory.

# Installation

The directory structure resulting from unpacking the tar files should look like this:

<code>src/</code>	...	containing the fortran90 source code
<code>doc/</code>	...	some documentation in html and ps
<code>tutorial/</code>	...	scripts and files for current tutorial
<code>benchmarks/</code>	...	additional sample scripts for running the code
<code>tools/</code>	...	postprocessing tools

## To compile:

Go to the source directory `src`, and edit the `Makefile`: adjust and tune the flags to match your machine type (`FC=ifort` or `gfortran` etc). Then do:

```
make
```

## Running BOPS

Once the code has compiled, go back up to the base (or top) directory and enter the `tutorial` directory. Here you will find the run scripts (suffix `.sh`) which launch the example simulations. These can be edited directly or copied as needed.

To run from this directory, just type eg:

```
./wake.sh
```

Or if you prefer to stay in the top directory – adjust paths in `wake.sh` first:

```
tutorial/wake.sh
```

Further examples can be found in benchmarks

# Project I: Laser wakefield accelerator

- 1 Edit and run the script: `./wake.sh`. Note that the input parameters are normalised to laser wavelength and wavenumber.
- 2 Look at the printed output to check the actual laser and plasma parameters (are they what you intended?)

## Plot files

These are in run directory (`foil_tnsa1`) in ASCII format and have a suffix `NN.xy`, where `NN` is the snapshot number. See `bops.oddata` for complete list

<code>ezsi</code>	EM field $E_z$ (S-polarized wave)
<code>nenc</code>	Electron density
<code>exsi</code>	Electrostatic field
<code>pxxe</code>	Electron momenta ( $x - p_x$ phase space)

- 3 Examine the field and particle phase space plots at successive time intervals (e.g.:  $t=100$ ,  $t=200$ ). The gnuplot script `wake.gp` should help you get started.
- 4 How can the plasma wave amplitude  $|E_x|_{\max}$  be optimised? Hint: try changing the electron density and/or pulse length.
- 5 Increase the laser amplitude to ( $a_0 = 3$ ) and observe the difference in the electron phase space and plasma wave. How does the matching condition need to be altered in this regime? Why is the amplitude of the latter reduced towards the back of the wake?
- 6 (advanced) Set up a plasma equivalent to twice the dephasing length (Eq. 37) and determine the maximum energy of electrons trapped in the plasma wave. Can you beat the scaling predicted by Eq. 38?



# Project II: Ion acceleration – TNSA vs. RPA

## 1 TNSA regime.

a) The script `foil.sh` sets up a  $2\ \mu\text{m}$  ‘foil’ out of frozen hydrogen ( $Z = A = 1$ ), with density  $n_e/n_c = 36$ . This is irradiated by a linearly polarized 50 fs pulse with intensity  $10^{19}\ \text{Wcm}^{-2}$ .

Run the script and inspect the following plots at the snapshot times (0, 50, 100, 150) fs:

### Plot files

```
nenc, ninc Electron & Ion densities
exdc      Cycle-averaged electrostatic field
pxxe, pxxi Electron & ion momenta
fuep, fuip Electron, ion energy spectra
uinc, ubac Incoming and outgoing wave energy
```

b) Try varying, eg: pulse amplitude or duration and compare the scaling of the maximum ion energy against the theoretical prediction of Eq. (15).

## 2 HB regime.

a) Copy the script to a new file (eg: `hb.sh`), switch the polarization from linear to circular (set `cpolzn='C'`), change the run directory – eg: `RUN=hb1` and rerun. (What phase space variables can you check the laser polarization with?) Compare the results to the TNSA case.

b) To compare with HB theory, it is convenient to specify a ‘square’ pulse profile. This can be done with, eg: `ilas=1`, `trise=5`, `tpulse=150`. Another important constraint in this case is to avoid **numerical heating** arising from an underresolved (cold) electron Debye length. Adjust the grid size/resolution to ensure that  $\Delta x < 0.5\lambda_D$ . Rerun and compare the ion shock velocity with Eq. (63).

## 3 Light sail regime.

a) Use the matching condition Eq. (64) to determine the foil width (using the same laser and target material) for which the light sail regime is reached. Set `ilas=4` with `tpulse=50`, `tfall=5`, and ensure that the new foil is resolved by the grid. Create a new script with these parameters and rerun. NB: you may have to increase the parameter `uimax` for this case!

b) Investigate the scaling behaviour of the monoenergetic ion feature with intensity, pulse length etc.

c) Now repeat with a Gaussian or  $\sin^2$  pulse shape. Compare the hot electron spectrum and phase space with the flat-top case at early times: what causes this difference? What could you change to improve the far-field ion sail stability for realistic pulse forms?

d) Now try the same thing with a **carbon** foil. Here, you will need higher (solid) number densities and multiply charged ions (see Eq. 41). What happens to the max. ion energy? How does the energy/nucleon compare with the hydrogen case?