

# Units in VPIC

A. G. Seaton & S. V. Luedtke

October 4, 2020

Units in VPIC can be confusing. They are user-defined, which can make inheriting a deck confusing for new users. The reason for this flexibility is that common unit systems, such as SI, are not suitable for use in VPIC because of its use of IEEE 754 single precision floating point numbers. In single precision, the smallest number representable with full precision is  $2^{-126} \approx 1.2 \times 10^{-38}$ . As an example, the SI value for  $\hbar$ ,  $1.05 \times 10^{-34}$ , is only four orders of magnitude away from losing precision. Multiply it by the electron mass, and you are hopelessly lost.

To set up a simulation, the user provides VPIC with values for the time step, cell dimensions, speed of light in vacuum, and vacuum permittivity. This is either done by directly assigning to the `grid` struct or by using the functions `define_units`, `define_timestep`, and (for example) `define_periodic_grid`. Usually one or more species are defined too, which each require values for the species mass and charge. These are normally set using the `define_species` function. It is important to understand that VPIC solves the same equations regardless of which values it is given and the chosen unit system.

We will assume here that two unit systems are in use. The first of these we will refer to as the ‘user’ unit system, and is the user’s preferred unit system. This is often used initially to define parameters of interest to the user, such as the temperature and density of the plasma. The second we will refer to as the ‘code’ unit system, which is used for values passed to VPIC itself. This document is concerned primarily with the code unit system.

We will list some common unit systems for values supplied to VPIC later, but for now the primary concerns for choosing a system of units are twofold:

1. The units must be compatible with the SI *equation form* that is solved by VPIC, meaning that Gaussian units cannot be used.
2. Values used by the code (including within calculations) should never exceed  $\pm 1.7 \times 10^{38}$ , and a value smaller than  $\pm 1.2 \times 10^{-38}$  should not be meaningfully different from zero.

## 1 Unit Conversions

In order to calculate values in the code unit system to supply to VPIC, we need to know the unit conversions. Taking the base units as length, time, mass, and charge, we define conversion factors  $L$ ,  $T$ ,  $M$ , and  $Q$  so that a quantity  $f_c$  in the code unit system is converted to a value  $f_u$  in the user unit system using  $f_u = F f_c$ .

Each value supplied to VPIC can be used to constrain the unit system by writing an equation for its conversion from the user unit system to code units in terms of the base unit conversion factors. Therefore by picking a subset of these values, we can fully determine the unit system.

To determine what the four base unit conversion factors are, we need four constraining equations. There are multiple sets of values we could choose. Given the base units, a straightforward choice would be the timestep, cell size, and species mass and charge. Here instead we pick the cell size  $h$ , speed of light  $c$ , vacuum permittivity  $\epsilon_0$ , and the charge-mass ratio  $r_\alpha$  of some species  $\alpha$ . This is motivated by many common unit systems setting  $\epsilon_{0,c} = 1$  or  $c_c = 1$ , as will be seen later.

The constraining equations are then:

$$\begin{aligned} h_c L &= h_u, & c_c \frac{L}{T} &= c_u, \\ \epsilon_{0,c} \frac{Q^2 T^2}{M L^3} &= \epsilon_{0,u}, & r_{\alpha,c} \frac{Q}{M} &= r_{\alpha,u}. \end{aligned}$$

Solving for the conversion factors we find:

$$L = \frac{h_u}{h_c}, \tag{1}$$

$$T = \frac{h_u}{h_c} \frac{c_c}{c_u}, \tag{2}$$

$$M = \frac{h_u}{h_c} \left( \frac{c_u}{c_c} \right)^2 \frac{\epsilon_{0,u}}{\epsilon_{0,c}} \left( \frac{r_{\alpha,c}}{r_{\alpha,u}} \right)^2, \tag{3}$$

$$Q = \frac{h_u}{h_c} \left( \frac{c_u}{c_c} \right)^2 \frac{\epsilon_{0,u}}{\epsilon_{0,c}} \frac{r_{\alpha,c}}{r_{\alpha,u}}. \tag{4}$$

These can then be combined into factors for other quantities. For example, the electric and magnetic field con-

version factors  $E$  and  $B$  are given by

$$E = \frac{ML}{QT^2} = \frac{h_c}{h_u} \frac{r_{\alpha,c}}{r_\alpha} \left( \frac{c_u}{c_c} \right)^2, \quad (5)$$

$$B = E \frac{T}{L} = \frac{h_c}{h_u} \frac{r_{\alpha,c}}{r_{\alpha,u}} \frac{c_u}{c_c}. \quad (6)$$

## 1.1 Exceptions

In addition to calculating the above conversion factors, the user should be aware of how VPIC normalises variables internally.

### 1.1.1 Particle Momenta

VPIC stores relativistic momentum rather than velocity (which would cause numerical issues) for each particle. The momentum is stored in dimensionless form given by

$$p_{\text{norm}} = \frac{p}{m_\alpha c}, \quad (7)$$

where  $m_\alpha$  is the mass assigned to species  $\alpha$  and  $c$  is the speed of light in a vacuum. This is what must be supplied by the user when performing the particle load.

### 1.1.2 Magnetic Fields

For performance reasons, and due to the use of single-precision floating point numbers, VPIC stores the scaled magnetic field  $c\mathbf{B}$  rather than  $\mathbf{B}$ .

## 2 Species & Particle Weighting

To define a particle species in VPIC, the user specifies the species mass  $m_s$  and charge  $q_s$ . Additionally, each particle is individually assigned a weighting factor  $w$ . These must be consistent with each other and the above unit system to produce the desired behavior.

### 2.1 Choosing Charge Mass Ratio

The motion of a physical particle with charge  $q_p$  and mass  $m_p$  in the EM fields only has explicit dependence on the charge-to-mass ratio  $r \equiv q_p/m_p$ . This is also the case for macroparticles in VPIC's particle push. Therefore the corresponding macroparticle species must be defined such that its charge  $q_s$  and mass  $m_s$  satisfy  $q_s/m_s = r$ .

### 2.2 Choosing Macroparticle Weight

When calculating the fields generated by a macroparticle, the code uses the species charge and macroparticle weight to calculate a total charge for the macroparticle:

$$q_{\text{MP}} = wq_s. \quad (8)$$

For this to result in an amount of current deposition consistent with the desired physical particle density, the macroparticle weight should be defined such that the charge contained in volume  $V$  is the same for the macroparticles as for physical particles, i.e.

$$\begin{aligned} q_p n_p V &= q_{\text{MP}} n_{\text{MP}} V \\ &= w q_s n_{\text{MP}} V, \end{aligned}$$

where  $n_p$  and  $n_{\text{MP}}$  are the physical particle and macroparticle number densities (macroparticles per unit volume) respectively. So, the particle weight must satisfy

$$w = \frac{q_p n_p}{q_s n_{\text{MP}}} = \frac{\rho_p}{\rho_s}. \quad (9)$$

This illustrates that there is a degree of freedom available in that the species charge defined in the deck does not have to be the same as the physical species charge.

### 2.3 Consequences of $q_s \neq q_p$

The possibility of  $q_s \neq q_p$  means that the code has no way of calculating numbers of physical particles since it doesn't store the true particle charge. This for example is why the hydro dumps contain charge density rather than number density. Users should therefore also ensure that any custom diagnostics do not assume  $q_s = q_p$ .

## 3 Examples of Common Unit Systems

Various choices are often made for the unit system, and we discuss a few of them here. In general, aside from ensuring that values do not exceed the limits for single-precision floating point numbers, it is often desirable to choose a unit system that eases interpretation of simulation results. Many of these unit systems allow the governing equations to be written in simpler form, which are convenient in theoretical analysis, and we will illustrate some of these cases here. As discussed above, the unit system has no bearing on the calculations performed by VPIC – it always solves the full system of equations.

### 3.1 Electrostatic Plasma Units

In an electrostatic problem simulating an electron plasma, the natural length and time-scales are the Debye length  $\lambda_D$  and inverse plasma frequency  $\omega_{pe}^{-1}$ . These are also the typical scales for the grid and time-step. Then choosing the electron charge/mass ratio so that  $e/m_e$  becomes unity reduces the equation of motion of an electron to

$$\ddot{\mathbf{r}} = \mathbf{E}.$$

Finally setting  $\varepsilon_0 = 1$  means that Gauss' law is also simplified

$$\nabla \cdot \mathbf{E} = \rho.$$

In the deck we would specify

$$\begin{aligned} h_c &= \frac{h_u}{\lambda_D}, & c_c &= \frac{c}{v_{th}}, \\ \varepsilon_{0,c} &= 1, & r_{e,c} &= -1, \end{aligned}$$

giving the unit conversions as

$$\begin{aligned} L &= \lambda_D, & T &= \omega_{pe}^{-1}, & M &= N_D m_e, \\ Q &= N_D e, & E &= \frac{m_e}{e} v_{th} \omega_{pe}, & B &= \frac{m_e}{e} \omega_{pe}, \end{aligned}$$

where  $N_D \equiv n_e \lambda_D^3$  is the number of electrons in a 'Debye cube'.

This unit system allows for some convenient sanity checks. As mentioned above, the cell size and time-step must be around 1. Additionally, if the electron mass in these units is not significantly smaller than 1, then the plasma is not ideal, and simulating it with VPIC is not appropriate. Interpretation of simulation results may also be more convenient. For example, the normalized electric field  $E_c$  is the electron quiver velocity in a plasma wave divided by the thermal velocity – an important parameter for determining whether wavebreaking will occur. Similarly, the wavenumber of a plasma wave is normalised such that  $k_c = k \lambda_D$ , which determines the Landau damping rate.

### 3.2 Electromagnetic Plasma Units

For an electromagnetic problem, the electrostatic units above might be modified so that the length scale becomes the electron skin depth  $l_e = c/\omega_{pe}$ . The Lorentz force and Maxwell's equations become

$$\ddot{\mathbf{r}} = \mathbf{E} + \mathbf{v} \times \mathbf{B},$$

and

$$\begin{aligned} \nabla \times \mathbf{E} &= -\partial_t \mathbf{B}, \\ \nabla \times \mathbf{B} &= (\mathbf{J} + \partial_t \mathbf{E}). \end{aligned}$$

The deck assignments become:

$$\begin{aligned} h_c &= h_u \frac{\omega_{pe}}{c}, & c_c &= 1, \\ \varepsilon_{0,c} &= 1, & r_{e,c} &= -1, \end{aligned}$$

so that the conversion factors are now

$$\begin{aligned} L &= \frac{c}{\omega_{pe}}, & T &= \frac{1}{\omega_{pe}}, \\ M &= n_e \frac{c^3}{\omega_{pe}^3} m_e, & Q &= n_e \frac{c^3}{\omega_{pe}^3} e, \\ E &= \frac{m_e}{e} \omega_{pe} c, & B &= \frac{m_e}{e} \omega_{pe}. \end{aligned}$$

Note that by removing  $\lambda_D$  from the unit system, there is no longer any dependence on the electron temperature used to calculate it.

### 3.3 Laser-Plasma Units

For laser-plasma interaction problems, the unit system can again be modified by replacing the time-scale with the inverse laser frequency, so that the unit system no longer depends on the electron density used to calculate  $\omega_{pe}$  and is instead dependent purely on the laser frequency. In the deck the assignments are

$$\begin{aligned} h_c &= h_u k_0, & c_c &= 1, \\ \varepsilon_{0,c} &= 1, & r_{e,c} &= -1, \end{aligned}$$

where  $k_0 \equiv \omega_0/c$  is the vacuum laser wavenumber.

The conversion factors then become

$$\begin{aligned} L &= \frac{1}{k_0}, & T &= \frac{1}{\omega_0}, \\ M &= \frac{n_{cr}}{k_0^3} m_e, & Q &= \frac{n_{cr}}{k_0^3} e, \\ E &= \frac{m_e}{e} \omega_0 c, & B &= \frac{m_e}{e} \omega_0, \end{aligned}$$

where  $n_{cr} \equiv m_e \varepsilon_0 \omega_0^2 / e^2$  is the critical density. In this unit system the simulation cell size and time step should satisfy  $\Delta x \ll 1$  and  $\Delta t \ll 1$  in order for the field solver to accurately model the laser. It is also necessary to verify that the cell size resolves the Debye length, which is often a more restrictive condition. Additionally, the normalized electric (or magnetic) field is the parameter  $a_0$ . For  $a_0 \ll 1$ , this is the electron quiver velocity divided by the speed of light, while  $a_0 \gtrsim 1$  indicates a relativistic field intensity which will lead to effects such as relativistically induced transparency and relativistic self-focusing.