## Pic3P Top-Level Commands

- **ModelInfo**
- **FiniteElement**
- **PRegion**
- **Loading**
- **TimeStepping**
- **Monitor**
- **LinearSolver**
- **CheckPoint**

Note: Refer to **acdtool** command syntax for postprocessing capabilities.

## ModelInfo

```
ModelInfo:
{
 File: ./pillbox.ncdf

 BoundaryCondition:
 {
  Exterior: 6
  Magnetic: 1, 2
  Absorbing: 3, 4
 }

 Material : {
  Attribute: 1
  Epsilon:   1.0
  Mu:       1.0
 }
}
```

**ModelInfo** specifies the model used for pic3p simulation.

**File**: The name of the mesh file in netcdf format. It can be located in a directory specified by the path. If the path is not specified, the default is the directory where the job is submitted.

**BoundaryCondition** specifies the boundary conditions on all the surfaces of the mesh. Every single surface of the mesh has a reference number which is set in Cubit and each reference number is associated with a boundary condition. The types of boundary condition used in pic3p are

- **Electric**: The tangential component of the electric field is zero. It can be used to define a symmetry plane in the model.
- **Magnetic**: The tangential component of the magnetic field is zero. It can be used to define a symmetry plane in the model.
- **Exterior**: The tangential component of the electric field is zero. Computationally it is equivalent to **Electric** boundary condition, but used normally for perfectly conducting cavity surface. For normal conducting surface, it can be used for calculating power loss on the surface using perturbation theory.
- **Absorbing**: Absorbing boundary condition (ABC) allows electromagnetic waves propagating at the speed of light to pass through the boundary without reflection. It is used for the entry and exit planes for a beam excitation, and for coaxial waveguides.
- **Waveguide**: Broadband waveguide boundary condition to account for the dispersive propagation of electromagnetic wave in waveguides.

**Material** specifies the material properties of a region in the model. If not specified, the region is set to vaccum.

- **Attribute**: The ID of the region
- **Epsilon**: Relative permittivity of the region
- **Mu**: Relative permeability of the region

## FiniteElement

```
FiniteElement:
{
  Order: 1
  Curved Surfaces: on
}
```

**FiniteElement** specifies parameters for the finite element method.

**Order**: The order of the finite elements used in the simulation. Order 1-6 have been implemented. The higher the order is, the more accurate the calculation is and the more computational resources are or a fixed mesh. For most applications, using 2$^{nd}$ order is good enough to obtain the solution accuracy.

**Curved Surfaces**: **on** or **off**
**on**: The surfaces of the finite elements on the model surface are represented by curved surfaces to better approximate the geometry.
**off**: The surfaces of the finite elements on the model surface are represented by flat surfaces.

## PRegion

```
FiniteElement:
{
  Order: 0              // finite element order p outside the window
  CurvedSurfaces: on
}

PRegion:
{
  Type: PICCausality
  Order: 1
}

PRegion:
{
  Type: PICDomain
  Order: 1
}
```

**PRegion** enables the use of a causality moving window technique in calculating beam-cavity interaction including space-charge effects. The size of the window in front of and behind the particle bunch is determined by causality. Outside the window, the order of the finite elements is to zero (as shown above). The orders of the finite elements in PRegion are set to higher orders. The finite element order in the region enclosing the particles can be different from the rest of the PRegion.

**Type**: The type of region within the moving window

- **PICCausality**: Causal moving window to account for wakefield effects
- **PICDomain**: Region enclosing particles within the casual moving window for self iteraction.  Normally the order is set to 2 or above for solution accuracy.

**Order**: Order of finite elements

## Loading

```
Loading:
{
 SymmetryFactor: 4
 SymmetryOrigin: 0 0 0

 Particles:
 {
  TrackingOnly: no
  z0: -6.0325e-2
  TotalCharge: -1e-9

  Distribution:
  {
   Beta: 0 0 0.003
   NumberOfParticles: 1e3
   Radial:
   {
    rMin: 0
    rMax: 1e-3
   }
   Longitudinal:
   {
    Type: Flattop
    DeltaT: 10e-12
    RiseTime: 2e-12
   }
  }

  Monitors:
  {
   DumpInterval: 2e-11
   DumpPhases: 18 64
   DumpPositions: 0 1e-2
   ObserverPlaneZPositions: -2e-2 2e-2
  }
 }

 ConstantField:
 {
  E: 0 0 0
  B: 0 0 0
 }

 DriveField:
```

```
 {
  FieldValueAtPhase0: -120e6
  FieldComponent: 2
  SamplePoint: 0,0,-6.0325e-2
  CentroidPhaseBeforeCrest: -58

  Omega3PMap:
  {
   Directory: omega3p_results
   ModeNumber: 1
  }
 }

}
```

**Loading** specifies the information for the excitations and the parameters of the particle bunch in pic3p. It includes the initial particle distribution, the external electromagnetic fields, and the parameters for monitoring particles.

**SymmetryFactor**: The fraction of the full model simulated by taking advantage of symmetry of the model. For example, **SymmetryFactor** is 2 when a half of the model is simulated; **SymmetryFactor** is 4 when a quarter of the model is simulated.

**SymmetryOrigin**: x, y, z
The coordinates of the point to which the symmetry is referenced [units in m]

**Particles**: Specify initial particle distribution and monitoring of particle distribution during pic3p run.

- **TrackingOnly**:
  - **yes**: space charge is off
  - **no**: space charge is on

- **z0**: z location where particles are injected into the computational model [unit in m]

- **TotalCharge**: Total charge of the particle distribution [unit in C]. Note that the electron charge is negative.

- **Distribution**: Initial particle distribution, either specified by the distribution **Type** in pic3p or as a user-specified distribution. For pic3p built-in distributions, they are described by the following.
  - **Beta**: $\beta_x$, $\beta_y$, $\beta_z$
    Relativistic parameters (v/c) of particles in x, y and z directions
  - **NumberOfParticles**: Number of particles
  - **Radial**: Uniform, defined by radial dimensions

  - **rMin**: Minimum radial distance [unit in m]
  - **rMax**: Maximum radial distance [unit in m]
- **Longitudinal**: Longitudinal distribution
  - **Type**: Type of distribution
    - **Flattop**: Flattop with rise time at both sides
    - **Gaussian**
    - **Ellipsoid**

  For **Flattop**, specify the following parameters
  - **DeltaT**: Duration of the flattop [unit in s]
  - **Risetime**: Rise time or drop time [unit in s]

  For **Gaussian**, specify the following parameters
  - **SigmaT**: Temporal sigma of the Gaussian [unit in s]
  - **NumSigma**: Number of sigmas on both sides of the Gaussian peak

  For **Ellipsoid**, specify the following parameters
  - **DeltaT**: Total duration [unit in s]

For a user-defined distribution, it is input using a file in netcdf format:
- **File**: Name of the file containing the particle distribution. First, specify the initial particle distribution in ascii format as follows.

  <number of particles>
  <x> <px> <y> <py> <z> <pz>
  . . .   . . .

  The first line is the number of particles in the distribution. Each of the following lines lists the position and momentum of a particle, with (x, y, z) being the position [units in m] and (px, py, pz) the normalized momentum $\gamma\boldsymbol{\beta}$. Then use acdtool to convert the ascii format to netcdf format:

  *acdtool postprocess pic3pconvert <filename>*

  where *filename* is the name of the ascii file in the format of, say, *mydistribution.txt*. The output netcdf file is *mydistribution.ncdf*.

**Monitors** specifies the output of particle data. The files are stored in the directory *<jobname>/OUTPUT/Particles*, where the default *jobname* is *pic3p_results*. The output is specified by the following parameters.
- **DumpInterval**: Time interval for writing [unit in s]
- **DumpPhases**: <phi1> <phi2> . . .
  Phases of drive field at which data are output [units in degree]
- **DumpPositions**: <z1> < z2> . . .
  z positions where data are output [ units in m]
- **ObserverPhaneZPositions**: <z1> <z2> . . .
  z positions where temporal transits of particles are recorded [units in m]

**ConstantField** allows the input of external constant fields. The electric and magnetic fields are specified by

- **E**: <Ex> <Ey> <Ez>
  Electric field components [units in V/m]
- **B**: <Bx> <By> <Bz>
  Magnetic field components [units in T]

**DriveField** allows the input of external rf fields. Multiple **DriveField**'s can be specified. The solution of external rf fields can be solved using omega3p. The normalization of the drive field is specified by the following parameters.

- **FieldValueAtPhase0**: The electric field gradient at zero rf phase of the drive field [unit in V/m]. The sign matches that specified in **TotalCharge**.
- **FieldComponent**: The component of the field used for normalization – 0 for x, 1 for y and 2 for z
- **SamplePoint**: <x>, <y>, <z>
  The position where the field gradient is normalized [units in m]
- **CentroidPhaseBeforeCrest**: Phase before the crest of the rf when the bunch centroid is injected [unit in degree]

RF field maps can be read in from Omega3P and Superfish. The field map provided by omega3p is specified by

- **Omega3PMap** specifies the cavity mode calculated by omega3p using the following parameters
  - **Directory**: <jobname>
    jobname is the directory containing the solution from omega3P. The default is *omeg3p_results*.
  - **ModeNumber**: The mode number associated with the cavity mode to be input, starting from 0

The field map provided by Superfish is specified by

- **PoissonMap** specifies the cavity mode calculated by omega3p using the following parameters
  - **File**: <filename>
    jobname is the file containing field map calculated using Superfish.
  - **Frequency**: Frequency of mode [unit in Hz]

**Solenoid** uses the magnetic field map calculated by Poisson by specifying the following parameters

- **SolenoidStrength**: Strength of magnetic field [unit in T]
- **Component**: The component of the field used for normalization – 0 for x, 1 for y and 2 for z
- **SamplePoint**: <x>, <y>, <z>
  The position where the field gradient is normalized [units in m]
- **PossionMap**: specifies the Poisson map file
  - **File**: <filename>
    jobname is the file containing field map calculated using Poisson

**FieldEmisson** features self-consistent Fowler-Nordheim instead of **Distribution**. Specify the following parameters.

- **BoundaryID**: Reference number of surface for field emission
- **Work Function**: Work function [unit in eV]
- **FieldEnhancement**: Field enhancement factor
- **MaxCurrentDensity**: Maximum current density allowed for particle emission [unit in $A/m^2$]

**ThermionicEmisson** features self-consistent thermionic emission instead of **Distribution**, with emission current density governed by J(E)=A*exp(B*sqrt(E)). Specify the following parameters.

- **BoundaryID**: Reference number of surface for thermionic emission
- **A**: Temperature dependent zero field current density [unit in $A/m^2$]
- **B**: Coefficient from Schottky emission formula [unit in $(V/m)^{-1/2}$]
- **MaxCurrentDensity**: Maximum current density allowed for particle emission [unit in $A/m^2$]

## TimeStepping

```
TimeStepping:
{
  MaximumTime: 3.0e-9
  DT: 1e-12
}
```

**TimeStepping** sets the total time and the time step of t3p simulation.

**MaximumTime**: Total simulation time [unit in s]

**DT**: Time step for advancing the Maxwell equation numerically in the time domain [unit in s]

## Monitor

```
Monitor:
{
 Type: Point
 Name: mon1
 Coordinate: 0.01, 0.00, 0.001
}

Monitor:
{
 Type: Volume
 Name: mymon
 TimeStart: 0e-9
 TimeEnd:   3e-9
 TimeStep:  2.5e-11
}

Monitor:
{
 Type: WakeField
 Name: wakefield
 StartContour: -0.03
 EndContour:  0.03
 Smax: 0.5
}
```

**Monitor** provides diagnostics for the electromagnetic field in a time domain simulation. Several types of monitors have been implemented in t3p, namely, **Point**, **Volume**, and **WakeField**, which can also be used by pic3p. In pic3p applications, normally only **Point** and **Volume** type will be used.

**Type**: Type of monitor

- **Point**: Monitors the fields at a location as a function of time. The output is stored in an ascii file with the format (t Hx Hy Hz Ex Ey Ez) with units in SI.
- **Volume**: Dumps snapshots of volumetric fields. The output is stored in netcdf format which can be visualized using ParaView.
- **WakeField**: Calculates the wakefield due to beam excitation.

(1) For **Point** monitor, specify the following.

- **Name**: The name of the file storing the fields as a function of time. The output is stored in an ascii file with the format (t Hx Hy Hz Ex Ey Ez), where the units are SI.
- **Coordinate**: x, y, z
  Location where the fields are monitored. The coordinates (x, y, z) are in untis of m.

(2) For **Volume** monitor, specify the following.

- **Name**: The name of the files storing the snapshots of fields at a regular interval.
- **TimeStart**: Start time for dumping fields [unit in s]
- **TimeEnd**: End time for dumping fields [unit in s]
- **TimeStep**: Time interval for dumping fields [unit in s]

(3) For **Wakefield** monitor, specify the following.

- **Name**: The name of the file storing the wakefield as a function of s, which is the distance from the front of the bunch and greater than 0. The output is stored in an ascii file with the format (s W(s), I(s)), where s is the distance in m, W(s) wakefield in V/C, and I(s) the bunch density in A/m, respectively.
- **StartContour**: The position [unit in m] where wakefield integration starts. If not specified, it is set to the beginning of the model in the z direction.
- **EndContour**: The position [unit in m] where wakefield integration ends. If not specifies, it is set to the end of the model in the z direction.
- **Smax**: The maximum distance for which the wakefield is calculated. It cannot be larger than that set by **MaximumTime**, the total bunch length and the model geometry to ensure the wakefield catch up with the beam. It is determined as follows.
  Let L = total model length and l = length of downstream beampipe, then **Smax** = c***MaximumTime –** (L-l).

Note that the wakefield integration used in this monitor is the Weiland wakefiled integration. It only applies to structures where on can see only vacuum from the cross section of the upstream and downstream beampipe. The integration can be reduced along the vacuum gap (specified by **StartContour** and **EndContour**) along the beampipe boundary. If **StartContour** and **EndContour** are not specified, they will be set to the start and end of the model in the z direction.
The wakefield will be automatically calculated at the transverse coordinates specified by (x, y) in **Bunch**: **StartPoint**. For other transverse coordinates, the wakefield can be calculated using acdtool as follows.

*acdtool postprocess wake_new t3p_results <x1, y1>*

where *<x1, y1>* are the transverse coordinates in units of m. Because the Laplace solver is mesh-based, it returns the wakefield at a mesh point closed to *<x1, y1>* indicated in the wakefield output file. The output file is stored as t3p_results/OUTPUT/wakefield.out, where the file name "wakefield" has been specified in **Monitor**.

## LinearSolver

```
LinearSolver: {
  Solver:            MUMPS
}

LinearSolver: {
 Solver:           CG
 Preconditioner:      CHOLESKY
}
```

**LinearSolver** specifies the solver for solving the linear system used in t3p simulation.

**MUMPS** is a direct solver and used for small problems.

**CG** is an iterative solver and used for large problems. **Preconditioner** for the iterative solver includes **CHOLESKY** and **DIAGONAL**.

## CheckPoint

```
CheckPoint:
{
  Action: restart
  Ntimesteps: 700
}
```

**CheckPoint** enables the writing of all relevant information so restart of time domain run can be restarted. This is useful when not enough CPU time is specified for a run or more simulation time is required. In the former case, one can just re-submit the job. In the latter case, one needs to change the **MaximumTime** in **TimeStepping** to a larger value.

**Action**: Option for checkpoint

- **none**: no checkpoint
- **restart**: default, with checkpoint

**Ntimesteps**: The interval in number of time steps when the information is dumped for the subsequent restart. The default is 1000.