T3P Sample Inputs

Table of Contents

ModelInfo

Tell T3P which mesh file to load and what boundary conditions are used for the different side sets in the mesh file (default: Electric)

```
ModelInfo:  {
 File: coarse.ncdf
 BoundaryCondition:   {
   Electric: 2
   Magnetic: 3 4
   Absorbing: 5 6
  }
}
```

MeshPartitioning

To specify the method to partition the mesh

```
  MeshPartitioning:  {
       Method: PARMETIS   //the other option is ZOLTAN
       Zoltan: {          //if the main method is ZOLTAN, this container will provide further zoltan specific
options
          Method: RCB
          Dimension: 1
          Partition Direction: Z
       }
  }
```

Normal finite element parameters

```
 FiniteElement: {
   Order: 2               // global order of basis functions (can be 1...6, 2 is recommended)
   CurvedSurfaces: on
 }
```

P-window for short-range wakefield

- set the basis order to be 0 (p=0).

  ```
  FiniteElement: {
    Order: 0            //p=0 outside of the window
    CurvedSurfaces: on
  }
  ```

- set an automatic moving window that following with the beam

```
PRegion: {
  Type: AutomaticMovingWindow
  Order:  2           //inside the window, p=2 (basis function order)
  Back:  0.01         //back pudding is 0.01m
  Front: 0.1          //front pudding is 0.1m
  StructureEnd: 1.0   //the maximal z.
}
```

Moving-window with mesh refinement for short-range wakefield

- set the basis order to be 0 (p=0).

```
FiniteElement: {
  Order: 0            //p=0 outside of the window
  CurvedSurfaces: on
}
```

- set an automatic moving window that following with the beam

```
MeshRefinement: {
  Order:  2          //inside the window, p=2 (basis function order)
  Back:  0.01        //back pudding is 0.01m
  Front: 0.1         //front pudding is 0.1m
  Subdivision: 1     //subdivide each element inside window once
  StructureEnd: 1.0  //the maximal z.
}
```

Gaussian beam going through a cavity

- The first step is to provide beam information:

```
LoadingInfo:  {
   Bunch: {
     Type: Gaussian
     Sigma: 2e-3          //Sigma (RMS) size of the bunch
     Nsigmas: 5           //beam occupies the location from -5 sigma to +5 sigma, total of 10 sigmas
     Charge: 1.           //charge
   }
   SymmetryFactor: 4       //factor by which to reduce the charge to account for symmetry conditions
(monopole on axis: use 4, dipole at X (or Y) offset: use 2 in connection with proper electric boundary
conditions in one plane)
   StartPoint: 0. 0. 0.    //StartPoint is the position where the beam enters the structure (typically
at low Z values)
   Direction: 0. 0. 1.     //Direction along which the bunch will move, at the speed of light (should
be the direction of the normal of the face with BoundaryID)
   BoundaryID: 5           //The boundary ID (sidelist number from Cubit), specifies the boundary
through which the bunch enters the structure (should be a flat surface, containing StartPoint)
 }
```

- Optional: Force analytical BeamBoundaryLoading (can be used if the beampipe is cylindrical). Not required. Default is OFF.

```
  Loading: {
    Type: BeamBoundaryLoading
    Analytical: on
    // Specify the right-handed coordinate system with its Z-axis along the beamline ( CrossProduct(X, Y) = Z =
Direction specified above)
    Origin: 0.0 0.0 0.0
    XDirection: 1.0 0.0 0.0     //this is the direction of the beam offset, if any
    YDirection: 0.0 1.0 0.0
    Beampipe radius: 0.04
    Beam offset: 0              //offset in x-direction of the local 2D coordinate system (value needs to be
consistent with StartPoint specified above)
  }
```

Time Integration Parameters

```
  TimeStepping: {
     MaximumTime: 10.e-10  //the maximal time to step
     DT: 2e-12             //delta T
  }
```

Wakefield Monitor

```
   Monitor: {
    Type: WakeField      // Weiland method (not for protruding structures, beam pipe radius must be the same on
left and right side)
    Name: wake
    Start contour: 0.05  // z-position at which the beampipe-cavity transition starts
    End contour: 0.10    // z-position at which the beampipe-cavity transition ends
    Smax: 0.3            // the longitudinal wake potential will be recorded from s=0 to s=Smax
  }
```

Point Monitor

To record the field values at specified location

```
   Monitor: {
     Type: Point             //point monitor
     Name: monA              //an output file called monA.out will be generated
                             //it contains: t Hx Hy Hz Ex Ey Ez
     Coordinate:  0.00002, 0.02, 0.1495  //the location
   }
```

Power Monitor

```
  Monitor: {
    Type: Power
    ReferenceNumber: 4     //which reference surface to monitor
    Name: mymon2
    TimeStart: 0           //when power monitor starts
    TimeEnd:   30.0e-9     //when it ends
    TimeStep:  0.125e-11   //how often it records power density
  }
```

Volume Monitor

```
  Monitor: {
    Type: Volume
    Name: vol
    TimeStart: 10.e-9          //when volume monitor starts
    TimeEnd:   500.e-9       //when it ends
    TimeStep:  50.e-9         //how often it records volume fields
  }
```

After T3P finished runs, users should run acdtool to generate mode files for each records of the volume fields using the following command:
`acdtool postprocess volmontomode t3pinput <jobname>`

The mode files generated can be viewed using paraview.

CheckPoint

request T3P code to checkpointing itself every certain timesteps so that one can restart T3P.

```
  CheckPoint: {
    Action: restart         //default should be restart. If there is no data available, it will have fresh
start.
    Ntimesteps: 100         //every 100 times steps, code will checkpoint itself
    Directory: CHECKPOINT   //the default directory to store checkpointing data
  }
```

LinearSolver

The options for linear solvers in the implicit timestepping.

```
  LinearSolver: {
        Solver:              CG          //other options include MUMPS (direct solver, faster for less than 32
CPUs) if it is compiled in
        Preconditioner:          CHOLESKY       //other options include DIAGONAL
        PrintFrequency:      50          //if you want print solver convergence history
        QuietMode:           1                 //Set it to 1 if you do not want to print anything
        Tolerance:           1e-10       //relative tolerance
        MaxIterations:       3000        //maxima number of iterations before CG quits
  }
```

Load a TEM waveguide mode on a coax port

```
Loading: {
 Type: PortModeLoading  //loading type
 Port:  {
   ReferenceNumber: 3   //port is at reference surface 3
   Origin: 0.0 0.0 -0.011
   XDirection: 1.0 0.0 0.0
   YDirection: 0.0 1.0 0.0
   ESolver: {
     Type: Analytic
     Mode:  {
       WaveguideType: Coax
       ModeType: TEM
       A: 0.0011
       B: 0.0033
     }
   }
 }
 Excitation: {
   Power: 1.
   Pulse: {
     Type: Monochromatic
     Frequency: 10.5e9
     Rise periods: 150
     Fall periods: 150
     T0: 0.
     TMax: 100.e-9
   }
 }
}
```