

In the following, a general format for the input file of MITHRA is presented. The red icons or groups can be repeated in the text. *int* stands for an integer number, *real* represents a real value, and *string* denotes a string of characters. The reference directory in the path locations is the path where the simulation is started. In other words, “./” points to the location where the project is called.

```

MESH
{
  length-scale
    = < real |
      METER |
      DECIMETER |
      CENTIMETER |
      MILLIMETER |
      MICROMETER |
      NANOMETER |
      ANGSTROM >

  time-scale
    = < real |
      SECOND |
      MILLISECOND |
      MICROSECOND |
      NANOSECOND |
      PICOSECOND |
      FEMTOSECOND |
      ATTOSECOND >

  mesh-lengths
    = < ( real, real, real ) >
  mesh-resolution
    = < ( real, real, real ) >
  mesh-center
    = < ( real, real, real ) >
  total-time
    = < real >
  total-distance
    = < real >
  bunch-time-step
    = < real >
  mesh-truncation-order
    = < 1 | 2 >
  space-charge
    = < true | false >
  solver
    = < NSFD | FD >
  optimize-bunch-position
    = < true | false >
  initial-time-back-shift
    = < real >
  lorentz-factor
    = < real >
}

BUNCH
{
  bunch-initialization
  {
    type
      = < manual |
        ellipsoid |
        3D-crystal |
        file >

    distribution
      = < uniform | gaussian >
    file-name
      = < string >
    charge
      = < real >
    number-of-particles
      = < int >
    gamma
      = < real >
    beta
      = < real >
    direction
      = < ( real, real, real ) >
    position
      = < ( real, real, real ) >
    sigma-position
      = < ( real, real, real ) >
    sigma-momentum
      = < ( real, real, real ) >
    numbers
      = < ( int, int, int ) >
    lattice-constants
      = < ( real, real, real ) >
  }
}

```

```

transverse-truncation
  = < real >
longitudinal-truncation
  = < real >
bunching-factor
  = < real between 0 and 1 >
bunching-factor-phase
  = < real >
shot-noise
  = < true | false >
}

bunch-sampling
{
  sample
    = < true | false >
  directory
    = < /path/to/location >
  base-name
    = < string >
  rhythm
    = < real >
}

bunch-visualization
{
  sample
    = < true | false >
  directory
    = < /path/to/location >
  base-name
    = < string >
  rhythm
    = < real >
}

bunch-profile
{
  sample
    = < true | false >
  directory
    = < /path/to/location >
  base-name
    = < string >
  time
    = < real >
  rhythm
    = < real >
}

FIELD
{
  field-initialization
  {
    type
      = < plane-wave |
        confined-plane-wave |
        gaussian-beam >

    position
      = < ( real, real, real ) >
    direction
      = < ( real, real, real ) >
    polarization
      = < ( real, real, real ) >
    radius-parallel
      = < real >
    radius-perpendicular
      = < real >
    signal-type
      = < neumann | gaussian |
        secant-hyperbolic |
        inverse-gaussian |
        flat-top >

    strength-parameter
      = < real >
    offset
      = < real >
    pulse-length
      = < real >
    wavelength
      = < real >
    rising-cycles
      = < int >
    CEP
      = < real >
    sigma-inverse-gaussian
      = < real >
  }

  field-sampling
    = < real >
}

```

```

{
  sample
    = < true | false >
  type
    = < over-line | at-point >
  field
    = < Ex | Ey | Ez |
      Bx | By | Bz |
      Ax | Ay | Az | F >

  directory
    = < /path/to/location >
  base-name
    = < string >
  rhythm
    = < real >
  position
    = < ( real, real, real ) >
  line-begin
    = < ( real, real, real ) >
  line-end
    = < ( real, real, real ) >
  number-of-points
    = < int >
}

field-visualization
{
  sample
    = < true | false >
  type
    = < in-plane | all-domain >
  plane
    = < xy | yz | xz >
  position
    = < ( real, real, real ) >
  field
    = < Ex | Ey | Ez |
      Bx | By | Bz |
      Ax | Ay | Az | F >

  directory
    = < /path/to/location >
  base-name
    = < string >
  rhythm
    = < real >
}

field-profile
{
  sample
    = < true | false >
  field
    = < Ex | Ey | Ez |
      Bx | By | Bz |
      Ax | Ay | Az | F >

  directory
    = < /path/to/location >
  base-name
    = < string >
  rhythm
    = < real >
  time
    = < real >
}

UNDULATOR
{
  static-undulator
  {
    undulator-parameter
      = < real >
    period
      = < real >
    length
      = < int >
    polarization-angle
      = < real >
    offset
      = < real >
    distance-to-bunch-head
      = < real >
  }

  static-undulator-array
  {
    undulator-parameter
      = < real >
    period
      = < real >
    length
      = < int >
  }
}

```

```

polarization-angle = < real >
gap = < real >
number = < int >
tapering-parameter = < real >
distance-to-bunch-head = < real >
}

optical-undulator
{
    beam-type = < plane-wave |
        truncated-plane-wave |
        gaussian-beam |
        super-gaussian-beam |
        standing-plane-wave |
        standing-truncated-plane-wave |
        standing-gaussian-beam |
        standing-super-gaussian-beam >
    position = < ( real, real, real ) >
    direction = < ( real, real, real ) >
    polarization = < ( real, real, real ) >
    radius-parallel = < real >
    radius-perpendicular = < real >
    order-parallel = < int >
    order-perpendicular = < int >
    signal-type = < neumann | gaussian |
        secant-hyperbolic |
        inverse-gaussian |
        flat-top >
    strength-parameter = < real >
    offset = < real >
    pulse-length = < real >
    wavelength = < real >
    rising-cycles = < int >
    CEP = < real >
    sigma-inverse-gaussian = < real >
    distance-to-bunch-head = < real >
}

EXTERNAL-FIELD
{
    electromagnetic-wave
    {
        beam-type = < plane-wave |
            truncated-plane-wave |
            gaussian-beam |
            super-gaussian-beam |
            standing-plane-wave |
            standing-truncated-plane-wave |
            standing-gaussian-beam |
            standing-super-gaussian-beam >
        position = < ( real, real, real ) >
        direction = < ( real, real, real ) >
        polarization = < real >
        radius-parallel = < int >
        radius-perpendicular = < real >
        order-parallel = < int >
        order-perpendicular = < int >
        signal-type = < neumann | gaussian |
            secant-hyperbolic |
            inverse-gaussian |
            flat-top >
        strength-parameter = < real >
        offset = < real >
        pulse-length = < real >
        wavelength = < real >
        rising-cycles = < int >
        CEP = < real >
        sigma-inverse-gaussian = < real >
    }
}

FEL-OUTPUT
{
    radiation-power
    {
        sample = < false | true >
        type = < at-point | over-line >
        directory = < /path/to/location >
        base-name = < string >
        plane-position = < real >
        line-begin = < real >
        line-end = < real >
        number-of-points = < int >
        normalized-frequency = < real >
        minimum-normalized-frequency = < real >
        maximum-normalized-frequency = < real >
        number-of-frequency-points = < int >
    }
    power-visualization
    {
        sample = < false | true >
        directory = < /path/to/location >
        base-name = < string >
        plane-position = < real >
        normalized-frequency = < real >
        rhythm = < real >
    }
    bunch-profile-lab-frame
    {
        sample = < false | true >
        directory = < /path/to/location >
        base-name = < string >
        position = < real >
    }
}

```