Dielectric Wakefield Calculator and Solver

Currently:

* Python: beam before and after DLW
  + Make sure the directory of where you run Python code from is the same folder that contains beam\_tools.py
  + Need to have fastkde installed (“pip install fastkde”)
* C++: Field calculation
  + Kick the beam using fields if only 1 step (i.e. assume no change in beam position over the DLW)
* Mathematica: beam envelope for more than one step

# Making a Beam

* Manipulating beams pre-post DLW uses the Python class and library
  + Example: HDF5\_BeamMaker.py
    - This file also meshes the same beam
* Define beam parameters:
  + For each beam (in SI unless stated otherwise)
    - N\_MACROS: number of macros
      * Not a beam variable but is used when generating the macroparticles
    - “pz\_MeV”: Mean longitudinal momentum in MeV/c
    - “eps\_x\_N”, “eps\_y\_N”: Normalised horizontal and vertical emittance
    - “sig\_x\_0”, “sig\_y\_0”: RMS horizontal and vertical beam size
    - “sig\_z\_0”: RMS bunch length in m (in the example file the value is set for SIGMA\_T)
    - “sig\_pz\_0”: RMS variation in longitudinal momentum (in MeV/c)
    - “lCorr\_Fac”: Arbitary but will be converted to a chirp value in future
    - “x\_0”, “y\_0”, “z\_0”: Average horizontal, vertical, and longitudinal position
    - “xp\_0”, “yp\_0”: Average horizontal and vertical momentum in MeV/c
    - “charge\_per\_macro”: formally this is the charge of each macroparticle
      * In the example file it is where the total charge is set since N\_MACROS is defined
    - “LongitudinalProfile”: must be set to one of a few options
      * ‘Gaussian’
      * ‘SkewGaussian’
        + “skew”: skewness value (alpha in standard PDF)
      * ‘Uniform’: This is flat-top profile
      * ‘Plateau’
        + “plat\_rise”: The time for a flat-top distribution to reach constant current

Set as a Gaussian with the first half at the head of the bunch and second half at the tail

* + - * ‘DoubleGauss’
        + “sig\_z\_2”: RMS length of the second Gaussian (in meters)
        + “offset”: Offset from the first Gaussian mean longitudinal position

Offset > 0

* + - * + “rel\_amp”: Relative amplitude of the second Gaussian compared to the first

Given the need for Offset > 0, use this value to set which of the two Gaussians are bigger

* + N.B. when using skew-Gaussian:
    - The mean shifts compared to the standard PDF: the way the beam is defined is so the mean longitudinal position is at t=0
    - Mean Shift =
      * Only matters if plotting the distribution to compare with fields generated
    - SigmaT is the SigmaT in the standard skew-Gaussian PDF and is not the RMS bunch length
      * Variance =
* Manipulating envelope before:
  + Beam is defined at a waist (i.e. will diverge from this point given the emittance and beam width)
  + Can drift the beam to change position: i.e. if a beam is called myBeam
    - myBeam.driftBeam(-1.5) drifts the beam -1.5m
  + Different vertical and horizontal envelopes is awkward
    - Need to define two beams, drift each by their respective distances, save the x and x’ of one of the beams, set x and x’ of the other beam to be as saved
    - Avoid doing this unless absolutely necessary

# Meshing an Existing Beam

* If the beam is external, read in the beam using specific function
  + readWakeCodeBeam()
* Example:
  + MeshCreator.py
  + This is for a beam which has the field solved for
* Inputs:
  + Inputs
* Most the time this doesn’t need changing
  + Widths and bunch lengths measured (BeamName.cpx BeamName.cpy and BeamName.cpz)
  + Set the length of the mesh to be some number of these widths (normally choose 6 for sigmaz and 5 for sigmax and sigmay)
    - Length of each cell then gives how many mesh cells (typically choose … per sigma)

# Using DiWaCAT\_FieldSolverUI

* So long as the file in and out names are updated most of the field calculation is done automatically
  + When selecting the folder it’s ok to choose a file within the folder (it’s written to only keep the file path upto the last “/” character)
  + The file path is relative to where the executable is (so can use “.\..\” shortcuts etc.)
* Executable inputs:
  + x0, y0: Transverse positions of the beam within the DLW
    - y0 is always in the direction towards the dielectric even if the plate is oriented vertically
      * 0 is the centre of the DLW
    - x0 is relative to the centre too
  + a: Half-gap within the vacuum (*a* in standard schematics) to the dielectric
  + delta: Dielectric Thickness
  + w: Width of the DLW
  + L: Length of the DLW
  + Epsilon: Relative permittivity of dielectric
  + Mu: Relative permeability
  + nX, nY: Starting number of horizontal and vertical modes
    - If convergence is ticked this is minimum
    - If convergence is unticked this is the number of modes used
      * Only worth unticking if performing quick simulation or you know too many modes have been chosen (I tend to keep the max at 150x150 as this takes ~5 hours)
  + Root Precision: precision of the wavenumbers of each mode
    - I don’t change this
    - Use default setting of 0.01
    - Only called once at the start of the time so setting very high precision doesn’t affect the running of the code
  + Mode Accuracy: Percentage of the field contained in the final 5 modes for convergence
  + No. Steps: number of kicks applied within the DLW
    - Set to 0 if only want the field or will perform beam dynamics externally (i.e. with the Mathematica)
    - 1 if just want a single kick and don’t want detailed tracking within the DLW
      * Ok if short, single DLW with transverse forces not too large
    - >1 not working just yet
  + Max Longitudinal Position
    - If you set this beyond the tail of the bunch more mesh points will be added so the field is calculated upto the max longitudinal position point
* Need to make sure the HDF5 libraries are correctly downloaded
  + Found in
* Main issue is getting the HDF5 libraries to link to where the code is compiled
  + Example of this issue is <https://stackoverflow.com/questions/34050155/symbol-not-found-linking-to-hdf-library>
  + If using Visual Studio:
    - Make sure the right HDF5 libraries are downloaded for the correct VS version (I used VS2015)
    - Project -> (Right Click) Properties
      * At top: Platform: x64 (i.e. 64-bit)
      * C/C++:
        + General

# Simulating Beam Dynamics

# Reading Beam Properties/Diagnostics/Post-DLW