

Significant updates for hps-mc

T. Cao & J. McCormick
July 1st, 2020

Overview

- Software updates: According to software packages at JLab farm, which I used to produce MC samples for 2016, 2019, future 2.3 and 3.7 GeV, hps-mc software packages were updated.
- Job system updates: The job system has been updated a lot with Jeremy's efforts. Plenty of tests have been taken. Although some further updates are ongoing, the system is available now. Jeremy will give a talk in details next time. Today, I will briefly introduce the system.
- Based on the system, some scripts for MC production have been developed and tested.

Software Updates: MadGraph

- We have 2 MagGraph versions: MG4 and MG5
 - MG4 for generation of ap and wab
 - MG5 for generation of rad and tritrig
 - History for MG application from Takashi: “The full diagram trident events are the events we detect in our experiment. However, the full diagram tritrig events generated by MG4 did not agree with the 1.05 GeV data in the cross section and the Esum distribution. The tritrig events generated by MG5 gave a better agreement. So we wanted to switch to MG5 for all the processes. It turned out that MG5 did not converge when the e^+e^- mass (A' mass) got small in the ap process. WAB also did not work with MG5.”
- For each package, run_cards are updated. In run cards, beam energy and kinematic limits are set. For each beam energy, we have a corresponding run card, like run_card_2pt3.dat, run_card_4pt55.dat, run_card_3pt7.dat, etc.

Software Updates: EGS5 programs

- Programs for ap:
 - prompt: lhe_prompt
 - displaced: lhe_exponential; lhe_uniform (starting from 2019 MC)
- program for rad: lhe_rad
- program for tritrig and wab: lhe_v1
- Programs for beam generation: v6 (for 2016 MC), and v7 (for 2019 MC and later) were added into hps-mc. For each beam energy, a corresponding program was built since kinematic limits depend on beam energy, such as beam_v7_2pt3, beam_v7_4pt55, beam_v7_3pt7, etc. For different programs, kinematic limits are different.

Software Updates:

kinematic limits in event generators

Kinematic limits in run cards of MadGraph and EGS5 programs are set with consideration of MC efficiency, acceptance, cuts in trigger, cuts in event selection and divergence of MadGraph

- ap/MG4: no limits

- rad/MG5:

Limits	2016	2019	3.7 GeV
Minimum for energy of e ⁺ or e ⁻ from pair	50 MeV	100 MeV	80 MeV
Minimum for y direction (p _y /p) of e ⁺ or e ⁻ from	0.005	0.005	0.005
Minimum for total energy of e ⁺ e ⁻ pair	500 MeV	1000 MeV	800 MeV
Minimum for invariant mass of e ⁺ e ⁻ pair	10 MeV/c ²	10 MeV/c ²	10 MeV/c ²

- tritrig/MG5:

Limits	2016	2019	3.7 GeV
Minimum for energy of e ⁺	100 MeV	200 MeV	160 MeV
Minimum for y direction (p _y /p) of e ⁺ or e ⁻ from	0.005	0.005	0.005
Minimum for total energy for at least one pair	1000 MeV	2000 MeV	1600 MeV
Minimum for invariant mass for at least one	10 MeV/c ²	10 MeV/c ²	10 MeV/c ²

- wab/MG4:

Limits	2016	2019	3.7 GeV
Minimum for energy of photon	400 MeV	800 MeV	640 MeV
Minimum for y direction (p _y /p) of photon	0.005	0.005	0.005

- beam/EGS5:

Limits	2016/2019/3.7
Minimum for energy of e ⁻	0.005*E _{beam}
Minimum for trans. (2016) / y (2019, future 2.3 and 3.7 GeV) direction if energy is larger than 0.6*E _{beam} for e ⁻	0.005
Minimum for y direction of photon	0.004
Max. for y dir. if energy > 400 MeV for 2016 (800 MeV for 4.55 GeV; 640 MeV for 3.7 GeV) for photon	0.005

Software Updates:

stdhep tools — add_mother_full_truth

- To obtain full truth of ap and rad samples, we need both MG and EGS5 output information.
- I ever developed a tool, called as add_mother_update2, in the stdhep tool package at JLab farm. For the tool, MG input is a root file, which is obtained from a lhe file by a converter.
- However, currently we do not want to include ROOT in hps-mc. So a new tool, called as, add_mother_full_truth, is developed.
 - For the tool, input for MG information is directly a lhe file, i.e., codes for reading and parsing lhe file are included in the tool.
 - Command: `add_mother_full_truth [input1.stdhep] [input2.lhe] [output.stdhep]`
- Note: The new tool is applicable for ap and rad. For tritrig and wab, we still use the tool add_mother.

Software Updates: stdhep tools — beam_coords

- There are several version tools for beam and target adjustments. These tools have similar features.
- In the updated hps-mc, only the most powerful tool is kept and updated to meet all of our requirements: rotation around x and y axes for momentum and vertex, rotation around z axes for vertex, shift in x, y and z directions for vertex, as well as beam-size setup.

- Options and their default:

```
printf("-x: beam sigma_x in mm; default = 0\n");  
printf("-y: beam sigma_y in mm; default = 0\n");  
printf("-u: beam x rotation in radians; default = 0\n");  
printf("-v: beam y rotation in radians; default = 0.0305 rad\n");  
printf("-w: beam z rotation in radians; default = 0\n");  
printf("-X: target X in mm; default = 0\n");  
printf("-Y: target Y in mm; default = 0\n");  
printf("-Z: target Z in mm; default = 0\n");
```

- Setup for 2016 MC: -x 0.125 -y 0.030 -u -0.00033 -v 0.03017 -w 0.2618 -X -0.224 -Y -0.08 -Z -4.3
- Setup for pass0 of 2019 MC: -Z -7.5; I am working on extraction of parameters from 2019 data

Software Updates: application of stdhep tools

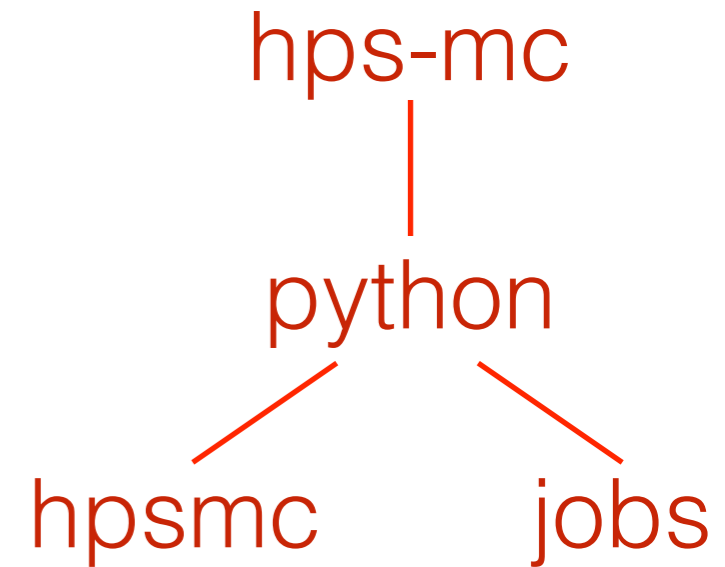
- for beam:
 - beam_coords
 - random_sample: build beam bunches
- for ap and rad:
 - add_mother_full_truth
 - beam_coords
- for tritrig and wab
 - add_mother
 - beam_coords

Job System

- HPS MC production is a complicated proceed. The job system is developed to let users conveniently take MC production by implementation of a few commands.
- The system can work for
 - single job
 - batch jobs with 4 different modes:
 - local and pool modes (run a set of jobs using Python's multiprocessing module) for local machine
 - auger mode for JLab batch machine
 - Isf mode for SLAC LSF machine

Job System: Packages

- Packages for the job system are in hps-mc/python, including
 - **package hpsmc**:
 - run_param.py: parameter setup for beam, target and ap mass
 - component.py: a base class for components
 - generators.py: son classes for generator components
 - tools.py: son classes for tool components
 - job.py: a class for running single job
 - batch.py: a class for running batch jobs, and four son classes for four different modes
 - job_store.py: a class for producing a json-format parameter file with a job list
 - other support scripts
 - **package jobs**: various job scripts; each job script sets a MC chain with application of generator and tool components

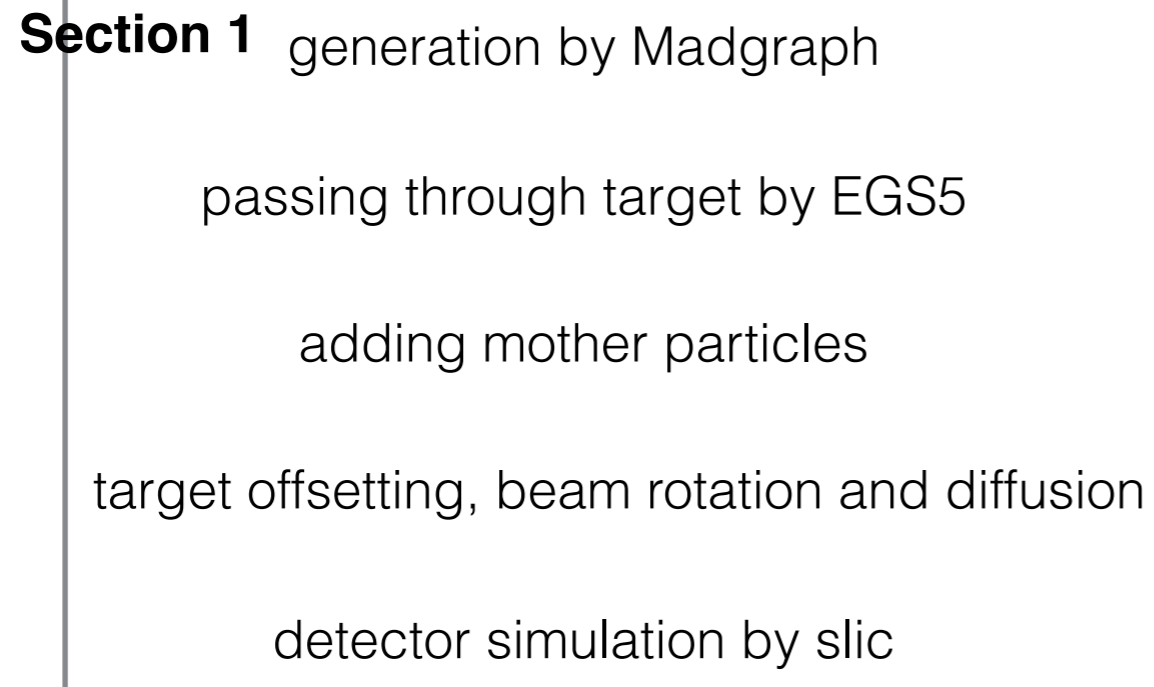


Job System: How to run

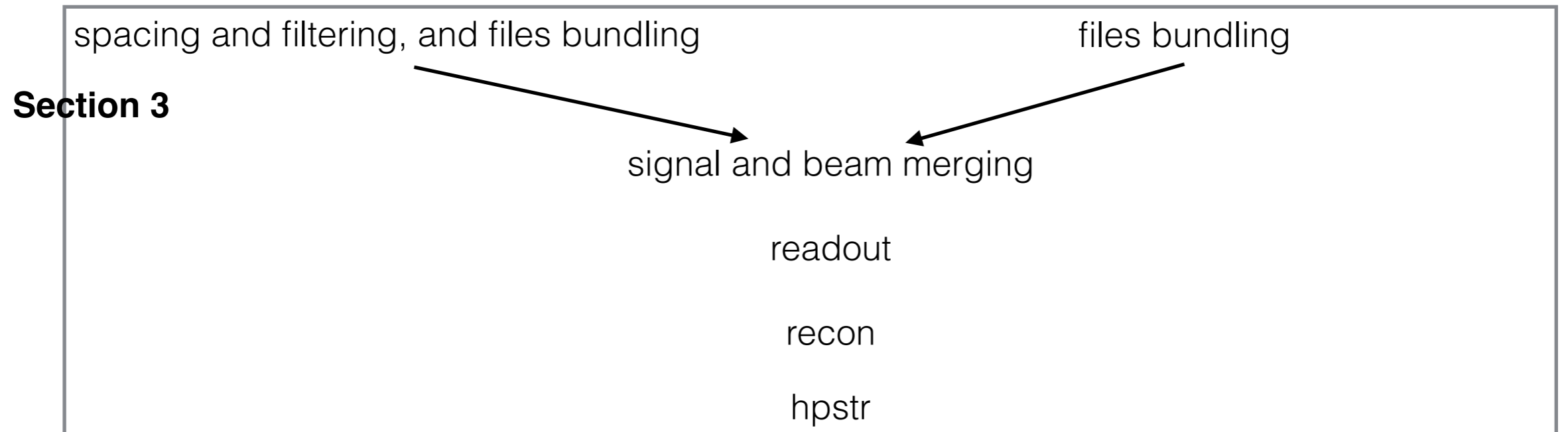
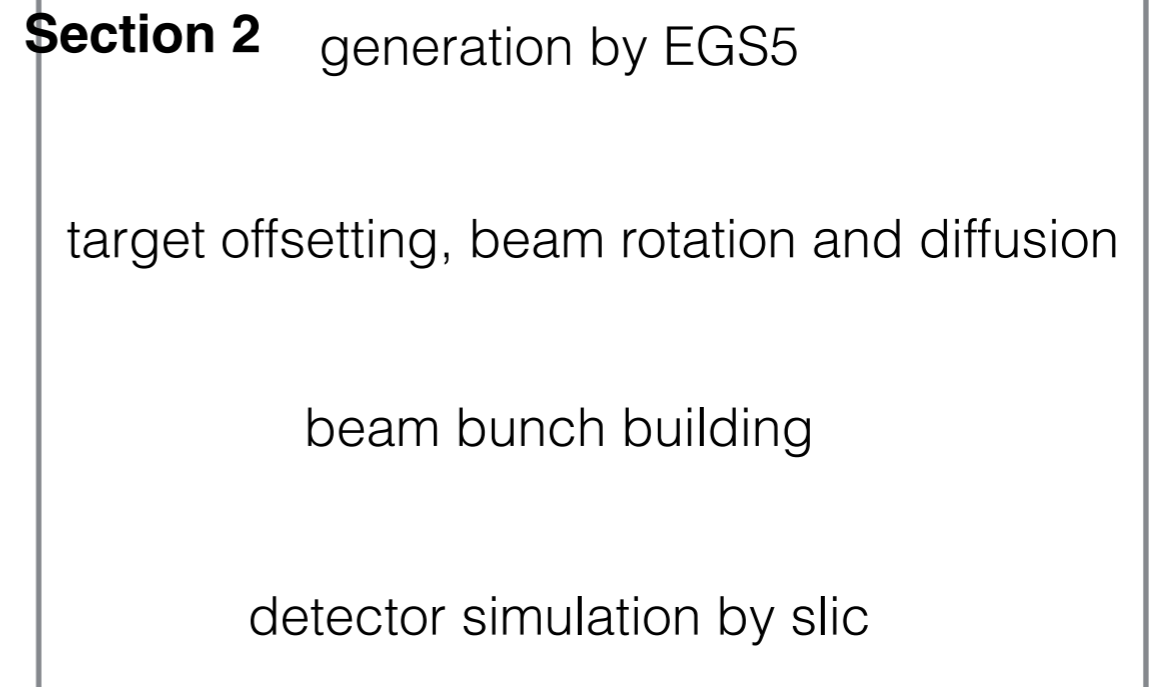
- Environment setup: `path-of-dir/hps-mc/install/bin/hps-mc-env.sh` (`hps-mc-env.csh`)
- Commands: command scripts are in `hps-mc/scripts`
 - single job: `hps-mc-job run [other-args] [-c configuration-file] [job-script] [parameter-file]`
 - batch jobs: `hps-mc-batch [mode] [other-args] [-c configuration-file] [job-script] [parameter-file]`
- **Configuration-file:**
 - Currently parsed by ConfigParser (Python2)
 - Includes setup for the Job class and configurations for packages of generators and tools
 - Software installation directories in configuration file need to be specified by users
 - Multiple configuration files are available in command, such as `-c configuration-file-1 -c configuration-file-2`
 - Suggest to put common configuration setup (i.e., software installation directories) in a configuration file, which is shared by all jobs
- **job-script:** a job script name in `hps-mc/python/jobs`
- **parameter-file:**
 - json format
 - Set job id, input files, output files, directories of input and output files, and parameters for generators and tools;
 - Includes one job for single job, while includes a job list for batch jobs.
 - For batch jobs, use command `hps-mc-jobstore` to apply the class `job_store.py` for production of a job list using a json template, variable substitutions, input file list(s)
- Examples for application: `hps-mc/examples` includes examples for application of the job system

MC Production: Scheme

Signal (ap, rad, tritrig, wab)



Beam



MC Production: Job Scripts

- Section 1: from generation to slic for signals
 - rad: rad_gen_to_slic_job.py
 - tritrig: tritrig_gen_to_slic_job.py
 - wab: wab_gen_to_slic_job.py
 - ap: update for ap generator component is in progress; job scripts for ap will be developed later
- Section 2: from generation to slic for beam
 - beam_gen_to_slic_job.py
- Section 3: from merging to recon for signal-beam
 - signal_beam_merge_to_recon_job.py

MC Production: Examples

- Examples:
 - Section 1:
 - rad: `hps-mc/examples/rad_gen_to_slic`
 - tritrig: `hps-mc/examples/tritrig_gen_to_slic`
 - wab: `hps-mc/examples/wab_gen_to_slic`
 - Section 2:
 - beam: `hps-mc/examples/beam_gen_to_slic`
 - Section 3:
 - signal-beam: `hps-mc/examples/tritrig_beam_merge_to_recon`
- When practice, please take care of configuration files and input-files setup for section3!!!
- PR for job scripts and examples are under review (branch: job-scripts-for-full-chain)