

Summer Student Update 2

Runtime Comparison for the Macros

Konrad Helms

FTX Software Meeting

18.08.22

What happened so far:

- wrote four macros (with different sub-versions) for Higgs recoil mass in $e^+e^- \rightarrow Z^* \rightarrow ZH \rightarrow H\ell^+\ell^-$:

without bkg	with bkg
higgs_recoil_edm4hep.C	higgs_recoil_with_bkg_edm4hep.C
	higgs_recoil_with_bkg_edm4hep_python.py
	higgs_recoil_with_bkg_edm4hep_uproot.py
	(higgs_recoil_with_bkg_edm4hep_RDataframe.py)

- macros are on GitLab

Plot Comparison

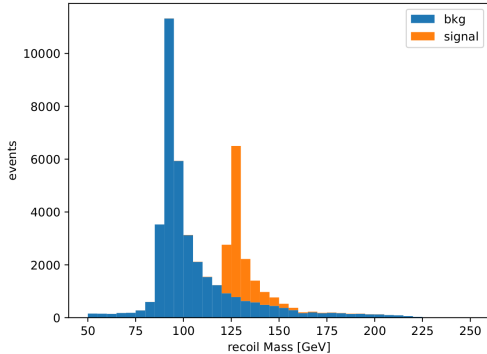


Abbildung: Python using uproot and
Matplotlib

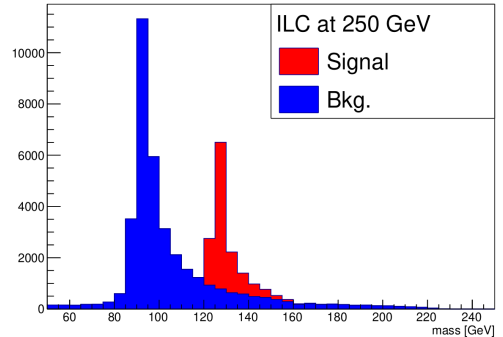


Abbildung: ROOT/ Python using ROOT
plotting

Number of Events to handle

- .root files generated using
`DelphesSTDHEP_EDM4HEP $DELPHES/cards/delphes_card_ILD.tcl`
`$K4SIMDELPHES/edmhep_output_config.tcl <filename.root>`
`<filename>.stdhep`

Signal	events: 17,143
E250-TDR_ws.Pe2e2h.Gwhizard-1_95.eL.pR.I106479.001.root	
Background	events per file: ~36,500
	→ total number of events: 292,806
E250-TDR_ws.P4f_zz_sl.Gwhizard-1_95.eL.pR.I106575.00(1-4).root	
E250-TDR_ws.P4f_zz_sl.Gwhizard-1_95.eR.pL.I106576.00(1-4).root	

Runtime Comparison

- averaged over 5 executions
- `time taskset -c <CPU Number> <run macro>`
- table shows total runtimes

Language	Macro	Avg. runtime	Min. runtime	Max. runtime
ROOT	<code>higgs_recoil_with_bkg_edm4hep.C</code>	17.97 s	17.54 s	18.08 s
Python	<code>higgs_recoil_with_bkg_edm4hep_python.py</code>	20.20 s	19.86 s	20.56 s
	<code>higgs_recoil_with_bkg_edm4hep_uproot.py</code>	12.44 s	11.91 s	12.95 s
	<code>higgs_recoil_with_bkg_edm4hep_uproot_conc.py</code>	59.49 s	59.42 s	59.57 s
ROOT	<code>higgs_recoil_edm4hep.C</code>	0.86 s	0.82 s	1.01 s

Uproot: When to Apply Cuts

avg. runtime: 24.33 s

```
def doEvtLoop(events):
    muons = events["Muon#0"]
    muons_index = muons["Muon#0.index"]

    number_of_events = len(muons_index)
    print(f"read {number_of_events} events")

    reco = events["ReconstructedParticles"]
    px = reco["ReconstructedParticles.momentum.x"][muons_index]
    py = reco["ReconstructedParticles.momentum.y"][muons_index]
    pz = reco["ReconstructedParticles.momentum.z"][muons_index]
    e = reco["ReconstructedParticles.energy"][muons_index]

    cut = (ak.count(muons_index,axis=-1)==2)

    mu1 = muons_index[cut,0]
    mu2 = muons_index[cut,1]

    mu1_px = px[cut,0]
    mu1_py = py[cut,0]
    mu1_pz = pz[cut,0]
    mu1_e = e[cut,0]

    mu2_px = px[cut,1]
    mu2_py = py[cut,1]
    mu2_pz = pz[cut,1]
    mu2_e = e[cut,1]
```

avg. runtime: 12.44 s

```
def makeLorentzVector(events,muons,index): # leading muon: index = 0, sub-leading muon: index = 1
    E = events["ReconstructedParticles.energy"][muons][:,index]
    px = events["ReconstructedParticles.momentum.x"][muons][:,index]
    py = events["ReconstructedParticles.momentum.y"][muons][:,index]
    pz = events["ReconstructedParticles.momentum.z"][muons][:,index]
    return [E,px,py,pz]

def applyCut(events,nmuons):
    muons_index = events["Muon#0"]["Muon#0.index"]
    number_of_events = len(muons_index)
    print(f"read {number_of_events} events")
    cut = (ak.count(muons_index,axis=-1)==nmuons)
    muons = muons_index[cut]
    return events[cut],muons

def doEvtLoop(events_precut):
    events,muons = applyCut(events_precut,2)

    mu1_p4 = makeLorentzVector(events,muons,0)
    mu2_p4 = makeLorentzVector(events,muons,1)
```

Improve Runtime for ROOT Macro

- wrote cmake file to build make file and compile ROOT macro
 - used optimisation flags $\implies$ avg. total runtime improvement -1.06 s
- $\implies$ not a significant increase because
- ROOT already uses O2 for JIST
 - only event loop affected by O2 flag but everything that is used in event loop was compiled with O2 previously
- problem: **next slide**

Potential in Speedup for ROOT Macro

```
for(int i = 0; i < nfiles; i++){
    const auto histname_tmp = histtag + std::to_string(i);
    hist.push_back(fillhisto(filename[i], histname_tmp, maxEvt, nbin, xmin, xmax, title));
}

TH1F* fillhisto(const std::string& file, const std::string& histtag, int maxEvt, int nbin, float xmin, float xmax, const std::string& title){
    TH1F* hist = new TH1F(histtag.c_str(), title.c_str(), nbin, xmin, xmax);

    // -- Collection -- //
    const std::string muonColName = "Muon";

    auto reader = podio::ROOTReader();
    reader.openFile(file);
    auto store = podio::EventStore();
    store.setReader(&reader);
    const auto nEntries = reader.getEntries();

    if(maxEvt == -1){
        maxEvt = nEntries; // if maxEvt is set to -1, all events are taken, change as needed
    }
    for(int entry = 0; entry < maxEvt; ++entry){
        auto& muons = store.get<edm4hep::ReconstructedParticleCollection>(muonColName);

        // only want di-muon events
        if(muons.size() != 2){
            store.clear();
            reader.endOfEvent();
            continue;
        }

        // construct system of muons
        const auto mu1 = edm4hep::utils::p4(muons[0], edm4hep::utils::UseEnergy);
        const auto mu2 = edm4hep::utils::p4(muons[1], edm4hep::utils::UseEnergy);
    }
}
```