

Procedimento para uso do MadGraph_aMC@NLO

Etapa #4: obtendo distribuições cinemáticas

O programa que será usado é escrito em python: `grafico_final_full.py`, onde tu incluirá os arquivos de eventos e gerar os gráficos. Copie o programa para a tua pasta de trabalho:

```
> cp ~/../public/grafico_final_full.py .
```

Esse programa criará pastas contendo figuras em PNG ilustrando as distribuições cinemáticas dos quarks *bottom* individuais e também suas distribuições associadas (*pair*). O programa também criará um arquivo de saída chamado `histos.root`, o qual poderá ser aberto no programa ROOT. Este arquivo lista todos os histogramas produzidos pelo programa e que serão lidos posteriormente para desenhos em conjunto num único histograma.

Dentro do programa tu notarás que existem *flags* chamadas de **#FIXME**, o que significa que as entradas das variáveis nestes pontos devem ser checadas; o restante do programa permanece inalterado. As entradas a serem verificadas são:

FILES	lista de arquivos de entrada
xsec	vetor contendo os valores das seções de choque de cada processo e listadas na ordem dos arquivos em FILES
PDF	nome que aparecerá na legenda dos histogramas
EVTINPUT	label que aparecerá no topo dos histogramas, condizente com Nevt
Nevt	número de eventos a serem lidos; max[eventos gerados no MG5]

O caminho dos arquivos deve ser como mostrado abaixo, seguindo alguns nomes sugeridos para os processos como usado nos exemplos anteriores:

<pre># INPUT FILES: FILES = ['/home/[usuario]/dybbbar.lhe', '/home/[usuario]/ppbbbar.lhe', '/home/[usuario]/pphbbbar.lhe', '/home/[usuario]/ppzz.lhe', '/home/[usuario]/aahhbbbar.lhe'];</pre>	<pre># INPUT FILES: FILES = ['/home/public/dybbbar.lhe', '/home/public/ppbbbar.lhe', '/home/public/pphbbbar.lhe', '/home/public/ppzz.lhe', '/home/public/aahhbbbar.lhe'];</pre>
--	---

Rode o script em python com:

```
> python grafico_final_full.py -b
```

e inclua a opção **-b** para evitar do python abrir a parte gráfica. Todos os histogramas serão produzidos numa pasta chamada **FULL-final-madgraph**.

O programa possui um conjunto de *flags* para construir histogramas de diferentes tipos:

scale	normaliza os histogramas conforme a seção de choque do processo
cuts	aplica cortes cinemáticos para realçar algum processo de interesse
setLog	constrói histogramas em escala logarítmica
filled	pinta os histogramas com cores para facilitar a visualização
stacked	reune os histogramas um sobre o outro para mostrar a soma total
data	inclui dados fictícios para mostrar como seria um gráfico contendo dados e previsões de geradores de eventos

Obtenha os histogramas empregando diferentes combinações destas *flags* para ver o efeito de cada uma delas sobre os eventos geradores no MG5. Evite somente a combinação de **scale** e **stacked** devido a um bug no programa.

O programa criará pastas com nomes diferentes para cada *flag* alterada para **True**.

Boa sorte

