AlpGen_i: An interface between AlpGen and Athena
Version in release 10.4.0 and later

Ian Hinchliffe (I_Hinchliffe@lbl.gov), Georgios Stavropoulos (George.Stavropoulos@cern.ch) and Jon

September 4, 2007

1 Introduction

This package runs AlpGen from within Athena. See the examples in **AlpGen_i/share/jobOptions.AlpgenPythia.py** and **AlpGen_i/share/jobOptions.AlpgenHerwig.py** which show how to read Alpgen events, add parton showers and hadronize them using Pythia or HERWIG. Note that on ATLAS we use the "new" (i.e. Pythia6.4) shower and this does not work with the AlpGen parton shower matching, so HERWIG should be used in general.

Users must first run AlpGen in standalone mode and make a file of events. An athena job then takes these events hadronizes them and passes them down the Athena event chain. The events must be made with a version of AlpGen that is compatible, recent versions that support the Les Houches interface should be acceptable. The current release is compatible with versions 2.13 to 2.11. It is recommended that you use the latest compatible version of the standalone AlpGen.

Within the `GeneratorModules`, `m_ExternalProcess = 4` is used to signify AlpGen to both the HERWIG and Pythia interfaces.

2 AlpGen and Pythia

To hadronize **AlpGen** generated events with Pythia, you need to link the file with the input parameters and the one with the unweighted events produced by AlpGen to the files inparmAlpGen.dat and alpgen.unw_events respectively. Then you only need to run athena with the jobOptions file jobOptions.AlpgenPythia.py by typing in the prompt
athena jobOptions.AlpgenPythia.py

3 AlpGen and HERWIG

To hadronize **AlpGen** generated events with HERWIG, you need to link the file with the input parameters and the one with the unweighted events produced by AlpGen to the files inparmAlpGen.dat and alpgen.unw_events respectively. Then you only need to run athena with the jobOptions file jobOptions.AlpgenHerwig.py by typing in the prompt
athena jobOptions.AlpgenHerwig.py

The program flow in HERWIG is a little complex, especially at the end of a run. The interface Herwig.cxx (in Herwig_i) calls HWHGUP (a HERWIG routine) which calls UPEVNT (modified by ATLAS to make sure it calls the right routine in usealpgen.f, which is in AlpGen_i). When the end of a file is found, UPEVNT calls ALSFIN (in atoher_65.f in Herwig_i), which then calls HWUGUP (HERWIG routine). ATLAS uses a modified version of HWUGUP which instead of abruptly stopping the program (which is bad for Athena!) sets IERROR and GENEV such that events generation is terminated more gracefully. This still leads to some confusing apparent error messages from Athena, but the HERWIG and AlpGen output is at least consistent.

To specify AlpGen running with HERWIG you need to set the line

```
job.Herwig.HerwigCommand = ["iproc alpgen"]
```

in your jobOptions file. It is also possible to specify Higgs decay information for the events by adding a second argument to the `iproc` parameter, which will be interpreted as the ID value (last two digits of IPROC) controlling the Higgs decay. e.g.

```
job.Herwig.HerwigCommand = ["iproc alpgen 12"]
```

would force any Higgs particles present in your input events to decay to two photons. See the HERWIG manual for more information.