pandora for BSM processes

in collaboration with
Michael Davenport and
Joshua Ruderman

M. E. Peskin
MC4BSM   March 2006

The goal of this meeting is to discuss methods by which theorists can create event generators for new, even arbitrary, models of physics beyond the Standard Model.

I thought about this problem in connection with the Linear Collider studies of '98-'02 and began developing a solution:

pandora

pandora has been dormant since 2002. But, hopefully, this solution is relevant to LHC physics. Some of my students and I are trying now to revive it.

The goals of pandora are to provide

a toolbox for creating event generators for general physics processes

extensible code that makes it easy to change particle couplings and decay chains

full accounting of (longitudinal) polarization

pandora is not intended to achieve precision.   It seems pointless to compete with the huge efforts devoted to precision SUSY and Higgs tools.   We hope that pandora will help people fill in the gaps in the major tools.

- workflow of pandora event generation
- process class
- inheritance
- application: $e^+e^- \longrightarrow f\overline{f}$
- decay class
- application: $\widetilde{\tau}_a^- \longrightarrow N_j^0 \tau^-$
- complexdecay class
- to-do list

pandora proper is a parton-level event generator written in C++, with an interface to PYTHIA hadronization.

The underlying structure is a Monte Carlo integrator:

$$\sigma = \int d^N x \; f(\{x_i\}) \; \frac{d\sigma}{d\{x_j\}}$$

In principle, any algorithm could be used to perform the integral; currently, we use VEGAS.

A pandora class P has methods

```
LE = P.getEvent();          LE = P.getEvent(weight);
```

where LE is constructed as an instance of an LEvent class that includes parton identities, 4-vectors, and color connections.

A separate program pandora_pythia, by Masako Iwasaki, writes LEvents to an external file and then stuffs them into PYTHIA for parton shower generation and hadronization.

pandora_pythia calls TAUOLA to generate longitudinally-polarized τ decays.

pandora does not yet implement the Les Houches accord. Because pandora was produced for e+e- physics, not much thought was given to complicated color final states. Also, initial-state protons were not included. We are working now to correct this.

Polarization of τ, b was left out of the Les Houches accord, but this was an oversight.   Can future releases of PYTHIA allow and even treat polarization effects ?

Parton reactions are described in pandora by a process class that is designed to compute cross sections according to

$$\frac{d\sigma}{d\{x_j\}} = g(s)\Big|\mathcal{M}(e^-(h_1)e^+(h_2) \to A(h_A)B(h_B)\cdots)$$

$$\cdot\mathcal{D}(A(h_A) \to F_A)\mathcal{D}(B(h_B) \to F_B)\cdots\Big|^2$$

and to create the final state $F_A F_B \cdots$ as 4-vectors in an LEvent.

$g(s)$	=	canonical prefactor
$\mathcal{M}$	=	helicity-dependent production amplitude
$\mathcal{D}$	=	helicity-dependent decay amplitude

At present, these amplitudes are not computed automatically. You compute them by hand; pandora does the rest.

C++ contains a mechanism for generalization called inheritance. Pandora uses this mechanism extensively.

C++ classes can be set up in a hierarchy of parent and daughter classes. A daughter class inherits the functionality of the parent class. New functions can be added. Specific functions, called virtual functions, can be modified.

e.g., in pandora,

eetottbar   is a daughter class of   twototwomm

The parent class handles the kinematics, the production of the cross section integrand, and the setup of 4-vectors and labels in the generated event.

To build eetottbar, we need to compute the correct production amplitudes and to call out the final-state color connections.

The production amplitude is computed by a virtual function of the parent class. So it is possible to modify or overwrite this function to change the physics without modifying the setup of the final state.

e.g., in pandora, include Z-primes in $e^+e^- \to q\bar{q}$ with

```
    class eetoqqbarwschannel  :   public  eetoqqbar
```

for which the constructor is `eetoqqbarwschannel(schannel & SC);`

The only method modified is the one that computes the production amplitudes.

An schannel is a data structure that describes γ, Z, arbitrarily many Z-primes, and a contact interaction.

Some specific predefined schannel classes are:

sequentialZprime, EsixZprime

For supersymmetry, pandora defines a separate class called SUSYspectrum that contains the basic data for the SUSY parameters:

values of 24 weak-scale MSSM parameters

a function fill() to compute all masses and couplings from these parameters (at the tree level)

functions to input the MSSM parameters from various small parameter sets (e.g. mSUGRA)

a function to input the spectrum parameters from ISAJET output

gaugino masses, mu, and A parameters can be complex, incorporating CP violation.

The process classes that creates a massive particle contain a pointer to a decay class for that particle.

pandora provides parent classes for decays that compute the decay kinematics and set up the 4-vectors of the final state.

pandora also supplies a default nodecay class.

To create a daughter decay class, one supplies the decay amplitude for the particle at rest, as a function of the spin orientation S^3.

The process classes combine these polarized decay amplitudes with helicity-dependent production amplitudes. To create events, they accept the 4-vectors from the decay classes and boost and rotate these to the appropriate frame and orientation.

In using helicity amplitudes, it is necessary to carefully match the conventions used to define production and decay amplitudes with definite polarizations states.

pandora uses an ideosyncratic but definite convention (available upon request).

We intend to make available a library of scalar products from which 2->2 production processes and 1->2 and 1->3 decays can be assembled.

pandora also contains a library of integrals over phase space times Breit-Wigner functions, to be used in computing total widths to populate decay tables.

The decay classes that create a massive particle contain a pointer to a decay class for that particle.

.. and so on (using the narrow resonance approximation for each intermediate state)

For a decay chain of arbitrary length, the full amplitude and the 4-vectors in the final state are built up recursively.

Some examples of decay classes for SUSY are:

general decay for 1st and 2nd generation,

$$\widetilde{f}(a) \to \widetilde{N}_j^0 f_L$$

```
sftoNfdecay(ID, a, j, finalID, particle, color, S, sfmass, GNfj);
```

depending on quantum numbers, a SUSY spectrum data class, and the scalar decay amplitude.

specific decays for 3rd generation,

$$\widetilde{\tau}(a) \to \widetilde{N}_j^0 \tau_{L,R}^-$$

```
stautoNtauLdecay(a, j, particle, S);
stautoNtauRdecay(a, j, particle, S);
```

To handle particles with multiple decay channels, pandora contains a parent class complexdecay

This contains a list of pointers to decay classes, a mechanism to query these classes and request their partial widths, and a mechanism to choose a given decay randomly according to the decay table constructed from these partial widths.

For example, for the b quark superpartners,

```
sbdecay(a, particle, S)
```

is a complexdecay that controls a list of 15 decay classes.

Current pandora to-do list:

inclusion of e+e- -> SUSY proceses and decays,
and testing of the underlying structure

M. Davenport has almost completed this

inclusion of UED by generalization of the SUSY classes

J. Ruderman has constructed a UED spectrum data class,
and some simple e+e- processes and decays

construction of beam (parton density) classes for pp collisions

implementation of Les Houches accord, full color connection structure in LEvent and pandora_pythia