CRAB 3.0α

November 17, 1997

Scott Pratt
National Superconducting Cyclotron Laboratory
Michigan State University, East Lansing, MI 48824
pratt@nscl.msu.edu (517) 333-6438

1 Introduction to CRAB

CRAB stands for "Correlation After Burner". It is a set of codes used to generate correlation codes from semi-classical transport codes. It reads files of phase space points, which describe the final momentum and point of last interaction of generated particles. From this information, along with knowledge of the impact parameters used for the transport-code runs, CRAB will generate correlation functions which account for impact-parameter averaging, experimental acceptances, experimental resolution and kinematic cuts.

CRAB can implement full quantum corrections for Coulomb interactions between the two particles as well as strong interactions for an arbitrary number of partial waves. There is also an option to include interactions of a Breit-Wigner form. CRAB does not account for strong interactions that mix partial waves or for interactions with third bodies, e.g. interacting with the residue of a compound nucleus.

Version 3.0 is completely rewritten in C++, but shares a great deal of philosophy with the previous FORTRAN version. Differences with previous versions are summarized below:

1. Different momentum binnings can be incorporated in the same run. For example, if one wishes to perform cuts in *p_t* and rapidity at the same time you perform calculations for *Q_inv*, it is straight-forward to implement. One can perform an arbitrary number of binnings in the same run.

2. The mesh used for calculating strong-interaction corrections to partial waves can be as fine as one wishes and is not tied to the resolution of the meshes used for binning.

3. Three-dimensional binnings can be incorporated.

4. It is easier to change the interaction between particles as parameters that define the interaction reside in one short file.

5. Phase space pt.s can be read from a file which contains information about all emitted particles, even those that are not considered in the correlation analyses. One can set which kinds of

particles are used for the analyses. For instance, one may wish to include phase space pt.s for π^+, π^- and π^0 for greater statistics, even though only negative pions are truly being considered. The default input format is OSCAR 1997A.

6. Estimates of statistical errors accompany the calculation.

Although previous versions were easy to use, they were not easy to modify. The new version was designed with the intention of making it easy for the user to modify rather than just making it easy to run.

2 Running CRAB

CRAB is run by compiling the file *crab.cpp* with C++. This file consists of a few definitions and several *#include* statements. Many of the choices are set in this file, and one may be able to configure, modify and run CRAB simply from the comments in *crab.cpp*. Other files which are likely to be modified by the user are the filter files, which set the experimental acceptance, the smearing files which set the experimental resolution and the binning files, which define kinematic cuts and conventions for plotting the correlation function.

2.1 Format for Phase Space Points

CRAB requires a file with phase space pt.s that represent a single-particle distribution. Files should be organized such that data for different impact parameter are contained in separate files. For example, supposed that one runs the dynamical code MYMD at five different impact parameters, which generates the following output files:

MYMD_b1.dat, MYMD_b2.dat, MYMD_b3.dat, MYMD_b4.dat, MYMD_b5.dat.

Now, the file MYMDb2.dat should be in the following form:

```
OSCAR1997A
final_id_p_x
MYMD 3.0a 208 82 208 82 com 200 1
event_dummy  NPART  bimp_dummy  phi_dummy
1  IDENT(1)  PX(1)  PY(1)  PZ(1)  E_DUMMY  MASS_DUMMY  X(1)  Y(1)  Z(1)  T(1)
2  IDENT(2)  PX(2)  PY(2)  PZ(2)  E_DUMMY  MASS_DUMMY  X(2)  Y(2)  Z(2)  T(2)
⋮
NPART ···
event_dummy  NPART  bimp_dummy  phi_dummy
⋮
```

The first three lines are the file header as specified by OSCAR standards. These lines are skipped over. The fourth line is the event header. Only the second number, **NPART**, is used. Phase space pt.s 1-**NPART** are then read in. If more than one event is written, another event-header and another set of phase space points would follow. Bold-face quantities are those which will be used by CRAB, while those with dummy suffixes are ignored.

The momenta are assumed to be in GeV/c, the space-time coordinates are assumed to be in fm-fm/c and the appropriate IDENTs can be found in the particle data group: *pdg.lbl.gov/rpp/mcdata/all.mc.*

If one wishes to change the format, it should be straight-forward to edit the file *source_files/crab_prinput.cpp*. If users have other formats they would like to be supported, a message to *pratt@nscl.msu.edu* could result in additional versions of this routine.

2.2 Setting up the file *phasedescription.dat*

CRAB needs to know where phase space points are stored, and what impact parameters are being used. This information must be supplied in the file *phasedescription.dat.* Consider the example above where points for five impact parameters are generated. The file *phasedescription.dat* might contain the following lines:

```
5
MYMDb1.dat 0.5 0.0 1.0 1 1 1.0
MYMDb2.dat 1.5 1.0 2.0 1 1 0.9
MYMDb3.dat 2.5 2.0 3.0 1 1 0.8
MYMDb4.dat 3.5 3.0 4.0 1 1 0.5
MYMDb5.dat 4.5 4.0 5.0 1 1 0.2
```

The first line gives the number of impact parameters. The following lines are of the following form:

filename b bmin bmax nevents ntestparts bcut

The file *filename* should obtain results from several events run at impact parameter b. The range of impact parameters described by this one impact parameter is *bmin* to *bmax.* Since pairs from different events at the same impact parameter are mixed in order to increase statistics, it is best to calculate for not much more than a half dozen impact parameters. The number of events at each impact parameter is *nevents* and the number of test particles(unity for molecular dynamics but large for Boltzmann approaches) are also given. The variable *bcut* gives the probability that an event at such an impact parameter would pass the centrality filter, which is assumed to be external to information used to generate the correlation.

2.3 Defining the Interaction

Interactions are defined in routines stored in the *interactions* subdirectory. If one wishes to use an interaction not defined or if one wishes to modify an existing interaction, copy one of the interactions already defined to a new file, edit it, then include from *crab.cpp*.

Even when using a predefined interaction, e.g. proton-proton, one may wish to modify the interaction file. For the following discussion one should view *crab_interaction_pp.cpp*. For instance, if one is modeling proton-proton correlations, the user may wish to make use of phase space points for both neutrons and protons.

The first four lines of *crab_interaction_pp.cpp* would then be:

```
#define N1TYPES 2
#define N2TYPES 2
const int IDENT1[N1TYPES]={2212,2112};
const int IDENT2[N2TYPES]={2212,2112};
```

If the two particles for which the correlation is calculated have the same mass, e.g. K^+K^-, and use the same phase space pt.s as mentioned above, the parameter IDENTICAL should be defined. This prevents CRAB from saving two sets of phase space points.

If the Coulomb and strong interactions are to be used the parameters COULOMB and STRONG_INTERACTION should be defined respectively. The momentum mesh used for the interaction is defined by INTERACTION_DELK and INTERACTION_NKMAX.

The fractions of spin channels that require a symmetric/anti-symmetric wave function is INTERACTION_WSYM and INTERACTION_ANTI. For non-identical particles, both are one half. For the pp case, there is an S=0 singlet (must be symmetric in coordinate space) and an S=1 triplet, which requires the spatial part of the wave function to be anti-symmetric.

If the STRONG_INTERACTION is defined, one must set POTENTIAL to the name of the function used to calculate the potential, *v(r,ipart)*. The *pp* example illustrates such a case. The variable *ipart* denotes the partial wave to be calculated, which runs from zero to STRONG_NETPARTIALS-1. The parameter STRONG_NSPINS denotes the number of spin channels for which there is a stong-interaction correction. The array STRONG_NPARTIALS gives the number of partial waves used for each spin channel. The sum of the values in STRONG_NPARTIALS-[STRONG_NSPINS] should equal STRONG_NETPARTIALS. The array STRONG_SYMM[STRONG_NSPINS] describes the symmetrization of the spatial wavefunction for a given spin channel, STRONG_WEIGHT[STRONG_NSPINS] gives the fraction each channel comprises. The angular momentum of each partial wave is denoted by STRONG_L[STRONG_NETPARTIALS].

As an example if three spin channels (S=0,S=1,S=2) were defined, and two partial waves were defined for the S=0 and S=2 cases but only one for the S=1 case, the array STRONG_NPARTIALS-

[STRONG_NSPINS] would be set to 2,1,2, and the partial waves would be numbered from zero to 4. Experienced cryptologists might try to decipher the logic by examining the file *source_files/crab_corrcalc.cpp*.

There is also an option to add a Breit-Wigner form to the calculation of the squared wave function. One can look at the file *crab_interaction_k+k-* to view the example where the $K^+K^- \rightarrow \phi$ case is considered. The parameters are self-explanatory. This method incorporates a simple gaussian form for $|\phi|^2$ that has the appropriate weight to give the correction associated with the density of states for a Breit-Wigner form. The range of the wave function is a sort of coalescence radius which is a variable that is set in the file above. The result should not depend on the coalescence radius as long as it is much shorter than characteristic sizes of the emission region. Of course, one could develop a better form for such cases, but this is good for quick estimates of effects.

2.4 Defining Momentum Bins

CRAB can calculate multiple correlation functions in a single run. One can view the file *binnings/crab_bindefs_example.cpp* to see how to define binnings. Four examples are given in that file:

1. Bin according to Q_{inv}.
2. Plot vs. Q_{inv}, but with cuts on the pair's transverse momentum.
3. Perform 3 directional cuts, making three correlation functions as a function of —**Q**— with constraints made on the two components of **Q** in the unwanted directions.
4. Bin the correlation in three dimensions.

Binnings are controlled by information in the binning file which is included via an `#include` command in *crab.cpp*. Binnings and the printing of correlations are performed by objects(C++ lingo). An object is created by each binning. For instance, if 10 p_t cuts are made, 10 separate one-dimensional binning objects are created. Three objects are created to perform the three directional projections. To print correlation functions as an actual three-dimensional quantity, a different class is used. The actual binning of the correlation function and printing of the results are member functions of each object. The two classes of objects are *onedbin_class* and *threedbin_class*.

Before being sent to the objects binning function, the momenta *p1* and *p2* can be checked for kinematic cuts. The function *decompose* can help in this regard by providing three projections of the relative momentum. Several options can be set in the call to *decompose* which set the reference frame and directional convention. This function is described in the file *binnings/crab_bindefs_example.cpp*. There are a variety of conventions for plotting directional cuts, and for Lorentz frames. The parameters fed to decompose will set the convention. If these are insufficient, it should be straight forward to add a few lines of code to produce any projection the user might wish.

The binning file also includes a definition of ABSOLUTE_MAXMOM which informs CRAB to skip calculation of pairs for relative momenta greater than this value. This is ignored when event mixing is performed, since the particles will be needed to generate the denominator.

2.5 Setting Filters

The experimental acceptance is defined by two functions in the filter file included into *crab.cpp*. An example file in the filters directory is *crab_filters_rapiditycut.cpp*. The two functions are *onepart_filter* and *twopart_filter*. The first filter is used to screen phase space points before they are stored for later use. Since they will be rotated around the beam axis, this filter should only depend on θ and $|p|$, not on ϕ. The two-particle filter includes the ϕ dependence as well. These functions depend on *p1* and *p2* and return *ifilter* equal to unity if the particles pass the filter.

2.6 Creating the Denominator with Mixed Pairs

The usual definition of the correlation function has a denominator which is a product of single-particle probabilities. If however, the denominator is created from events where only two particles were observed in a narrow acceptance, the single-particle spectra is distorted. For experiments with a narrow acceptance, e.g. NA44, where only two-particle events are used to construct the denominator, the parameter MIXED_PAIRS_FOR_DENOM should be set in *crab.cpp*. When this parameter is set, particles are stored as into an array as they are chosen for the denominator, along with the correlation weight. The denominator is then calculated by choosing from these particles at random and including a weight $w_1 \cdot w_2$.

2.7 Smearing the Momentum

Experiments measure momenta with limited resolution. To account for that momenta can be smeared after the correlation weight is calculated, but before they are bin. The smearing is handled by calling the function *momentum_smear*. The parameter SMEAR_MOMENTA must be set in *crab.cpp* in order for the smearing to take effect. The *smearings* directory includes some sample files used for smearing. The actual file incorporated into CRAB is set by an `#include` statement in *crab.cpp*.

2.8 Compiling and Running

CRAB can be compiled by typing:

```
c++ crab.cpp
```

When run, CRAB prompts the user for the number of pairs to sample in the numerator and denominator. This number is the number of pairs that

A. Pass the two-particle filter

B. Have an invariant relative momentum less than ABSOLUTE_MAXMOM which is set in the binnings file. (Neglected when event mixing is performed for the denominator).

After all wave functions are calculated and phasespace points are read, CRAB prints the statement:

```
Initialization Completed
```

Correlation functions are sent to the *results* directory. In the case that a strong-interaction is implemented, phase shifts are also printed in the *results* directory.

3 How CRAB works

3.1 Basic Theory

CRAB is based on the formula:

$$C(\mathbf{P}_{tot},\mathbf{q}) = 1 + \frac{\int d^4x_1 d^4x_2 S_1(x_1,\mathbf{p}_1)S(x_2,\mathbf{p}_2)|\phi_{rel}(x_2' - x_1')|^2}{\int d^4x_1 d^4x_2 S_1(x_1,\mathbf{p}_1)S(x_2,\mathbf{p}_2)} \tag{1}$$

The primes refer to the fact that coordinates are determined in the pair's rest frame.

This formula is based on a variety of subtle formulas. Since it is semi-classical in nature it is never valid for small sources or whenever the product of characteristic momenta and characteristic distances is not many times larger than $\hbar$. It is also questionable whenever an interaction with a third body greatly distorts the relative trajectory. This may be the case for two charged particles which barely leak over the Coulomb barrier, or when two particles have small relative momenta, but leave on opposite sides of a large residue. Symmetrization with other identical particles can also distort the correlation function when final phase space densities approach unity.

The correlation function can then be thought of as an average weight of the squared wave function which is unity for non-interacting particles.

3.2 How Pairs are Chosen

CRAB first calculates the numerator by choosing pairs of particles at random from the same impact parameter. The impact parameter is chosen according to the cross-section represented by the particular impact parameter. For instance, if one impact parameter is run at .5 fm which represents collisions between 0 and 1.0 fm and a second file contains results from a calculation at $b = 1.5$ fm, which represents the range $1.0 < b < 2.0 fm$, the weighting of the second impact parameter will be 3 times larger, since the cross-sectional area is 3 times larger.

CRAB treats particles from different events with the same impact parameter as if they came from the same event. This is crucial for generating sufficient statistics. This is only invalid if there are strong fluctuations present in the emission, e.g. droplet formation and decay.

The weighting for the numerator also includes the product of the number of particles of each type and the probability that an event with such an impact parameter would pass the impact-parameter filter. The weighting includes a division by the square of the number of events run at that impact parameter and the square of the number of test particles (used in BUU codes) used for that impact parameter.

When calculating the denominator, a different weighting is used. In this case the two particles are chosen from different impact parameters. Similar factors as those described above are used in determining the weighting of pairs in the denominator. This weighting is performed in the file *source_files/crab_prinput.cpp*.

Uncertainties for all correlation bins are estimated from the root-mean-square fluctuation of the wave-function weights in each bin. An additional uncertainty of $1/N^{1/2}$ is added to account for the fluctuation of the number of pairs in both the numerator and denominator.

When all phase space pt.s originate from a zero-impact-parameter calculation, (CRAB senses this by noticing there is only one file and that $b = 0$) the denominator is not calculated, and the number of pairs in the numerator is used for the denominator. The uncertainty estimate then neglects the $1/N^{1/2}$ contribution mentioned above.

4 Testing CRAB

CRAB may be tested by generating Gaussian phasespace points with the code *phasemaker.c*. The user is prompted to input a temperature and Gaussian sizes R_x, R_y, R_z and τ. The phase space points are written to files *thermal_gauss01.dat*, *thermal_gauss02.dat*, $\cdots$. Each file represents a different impact parameter. The number of impact parameters can be defined at the top of *phasemaker.c*. One can also set the types of particles generated, which will all be created with equal probability. By choosing a small temperature, e.g. 5 MeV, the phase space points will be compact in momentum space and will allow CRAB to find useful pairs quickly. A small temperature also allows one to predict correlation functions for Gaussian sources without distortions due to Lorentz contraction.

5 Tested Systems

CRAB has been tested successfully for the following:

1. Windows NT, Pentium Pro
   Borland C++ (using BCC32) and g++.
2. UNIX, alpha
   g++
3. Linux, Pentium
   g++
4. VMS, alpha
   DIGITAL cxx

CRAB fails to work properly for:

1. Windows NT, Pentium Pro
   Borland C++, (when BCC32i is invoked)