

0.1 Coulomb Fitter

When fitting the $\Xi^-(\bar{\Xi}^+)K^\pm$ results, it is necessary to include both strong and Coulomb effects. In this case, Equation ?? is no longer valid, and, in fact, there is no analytical form with which to fit. We therefore must take a more basic approach, and integrate out Eq.?? by hand. To achieve this, one has two options. The first option is to numerically integrate Eq.?. The second option is to simulate a large sample of particle pairs, calculate the wave function describing the interaction, and average to obtain the integral. Having no experience with either of these options, we elected the latter of simulating pairs. The code developed to achieve this functionality is called ‘‘CoulombFitter’’. Currently, in order to generate the statistics needed for a stable fit, we find that $\sim 10^4$ simulated pairs per 10 MeV bin are necessary. The nature of this process means that the ‘‘CoulombFitter’’ takes much longer to run than the ‘‘LednickFitter’’ of Section ??.

Unfortunately, with this analysis, we are not sensitive to, and therefore not able to distinguish between, the iso-spin singlet and triplet states. We proceed with our analysis, but the results must be interpreted as iso-spin averaged scattering parameters.

As stated before, to generate a fit correlation function, we must simulate a large number of pairs, calculate the wave-function, and average Ψ^2 over all pairs in a given k^* bin. Essentially, we calculate Equation ?? by hand:

$$\begin{aligned}
 C(\mathbf{k}^*) &= \sum_S \rho_S \int S(\mathbf{r}^*) |\Psi_{\mathbf{k}^*}^S(\mathbf{r}^*)|^2 d^3\mathbf{r}^* \\
 \longrightarrow C(|\mathbf{k}^*|) &\equiv C(k^*) = \sum_S \rho_S \langle |\Psi^S(\mathbf{k}_i^*, \mathbf{r}_i^*)|^2 \rangle_i \\
 \longrightarrow C(k^*) &= \lambda \sum_S \rho_S \langle |\Psi^S(\mathbf{k}_i^*, \mathbf{r}_i^*)|^2 \rangle_i + (1 - \lambda)
 \end{aligned} \tag{1}$$

where $\langle \rangle_i$ represents an average over all pairs in a given k^* bin.

In summary, for a given k^* bin, we must draw $N_{pairs} \sim 10^4$ pairs, and for each pair:

1. Draw a random \mathbf{r}^* vector according to our Gaussian source distribution $S(\mathbf{r}^*)$
2. Draw a random \mathbf{k}^* vector satisfying the $|\mathbf{k}^*|$ restriction of the bin
 - We draw from real k^* vectors obtained from the data
 - However, we find that drawing from a distribution flat in k^* gives similar results
3. Construct the wave-function Ψ

After all pairs for a given k^* bin are simulated and wave-functions obtained, the results are averaged to give the fit result.

Construction of the wave-functions, Equation ??, involves a number of complex functions not included in standard C++ or ROOT libraries (namely, $h(\eta)$, $\tilde{G}(\rho, \eta)$, and $F(-i\eta, 1, i\xi)$). These functions were even difficult to find and implement from elsewhere. Our solution was to embed a Mathematica kernel into our C++ code to evaluate these functions. However, having Mathematica work on-the-fly with the fitter was far too time consuming (fitter would have taken days, maybe weeks to finish). Our solution was to use Mathematica to create matrices representing these functions for different parameter values. During fitting, these matrices were then interpolated and the results used to build the wave-functions. This method decreased the running time dramatically, and we are now able to generate results in under ~ 1 hour. This process will be explained in more detail in future versions of the note.