Kinematics for J/ψ Photoproduction
RPC Technical Note #22

Keith Griffioen
College of William and Mary

July 17, 2002

Abstract

The following set of formulas define the kinematics for J/ψ photoproduction and are useful for E160 planning, simulation and analysis. These formulas are programmed into the perl script `psi.pl` which can be found in `~griff/bin` on the SLAC computers.

1 Formulas

Fig. 1 shows the kinematics in laboratory and center of momentum frames for J/ψ photoproduction. The following set of formulas is intended to be general. The equations that refer explicitly to elastic production (*i.e.* the nucleon is left in its ground state) are labeled as such. For coherent production on the nucleus, the variable M which is taken to be the nucleon mass, can be replaced by the nuclear mass.

Table 1 defines the variables in the lab and CM frames.

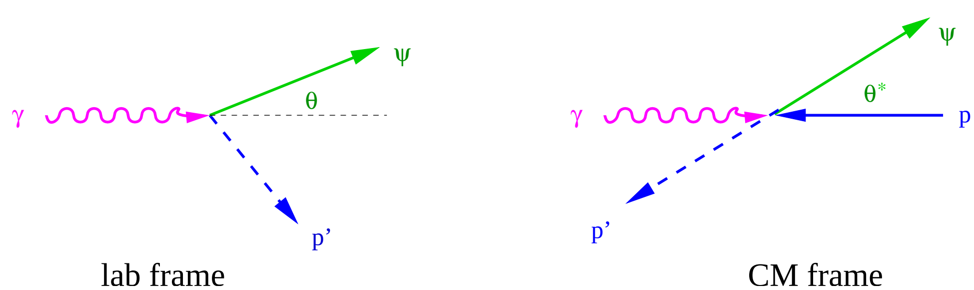

Figure 1: Kinematics of J/ψ photoproduction in lab and center-of-momentum frames.

$$s \equiv (k_\mu + p_\mu)^2 = M^2 + 2Mk \tag{1}$$

$$k^* = \frac{s - M^2}{2\sqrt{s}} \tag{2}$$

lab	CM
$k_\mu = (k, 0, 0, k)$	$k^*_\mu = (k^*, 0, 0, k^*)$
$p_\mu = (M, 0, 0, 0)$	$p^*_\mu = (E^*_p, 0, 0, -k^*)$
$k'_\mu = (E_J, p\sin\theta, 0, p\cos\theta)$	$k'^*_\mu = (E^*_J, k'^*\sin\theta^*, 0, , k'^*\cos\theta^*)$
$p'_\mu = (k + M - E_J, -p\sin\theta, 0, k - p\cos\theta)$	$p'^*_\mu = (E'^*_p, -k'^*\sin\theta^*, 0, -k'^*\cos\theta^*)$
	$E^*_p = \sqrt{M^2 + k^{*2}}$
$E_J = \sqrt{M_J^2 + p^2}$	$E^*_J = \sqrt{M_J^2 + k'^{*2}}$
	$E'^*_p = \sqrt{W^2 + k'^{*2}}$

Table 1: Kinematic variables for J/ψ photoproduction. The stars indicate the CM frame, the primes denote variables for the final reaction states, and the subscript μ indicates a 4-momentum of the form (E, p_x, p_y, p_z). The variable k is the incident photon momentum, p is the outgoing J/ψ momentum, θ is the angle of the J/ψ with respect to the photon direction, M is the nucleon mass, M_J is the J/ψ mass, and W is the invariant mass of the residual nucleon target.

$$k^* = \frac{M}{\sqrt{s}}k \tag{3}$$

$$s = (\sqrt{M^2 + k^{*2}} + k^*)^2 \tag{4}$$

$$u = -2ME_J + M^2 + M_J^2 \tag{5}$$

$$t = M_J^2 - 2k(E_J - p\cos\theta) \tag{6}$$

$$t = -2M(k - E_J) \qquad \text{elastic} \tag{7}$$

$$W^2 = M^2 + t + 2M(k - E_J) \tag{8}$$

$$k'^{*2} = \frac{(s - W^2 - M_J^2)^2 - 4W^2M_J^2}{4s} \tag{9}$$

$$k'^{*2} = k^{*2} + \frac{W^4 - M^4 + M_J^4 - 2s(W^2 - M^2 + M_J^2) - 2W^2M_J^2}{4s} \tag{10}$$

$$p_T \equiv p \sin\theta = k'^* \sin\theta^* \tag{11}$$

$$\theta^* = \arcsin(p_T/k'^*) \tag{12}$$

$$t = M_J^2 - 2k^*(E_J^* - k'^* \cos\theta^*) \tag{13}$$

$$k'^{*2}_{\text{el}} = k^{*2} + \frac{M_J^2(M_J^2 - 2s - 2M^2)}{4s} \qquad \text{elastic} \tag{14}$$

$$t_{\min} = M_J^2 - 2k^*(\sqrt{M_J^2 + k'^{*2}_{\text{el}}} - k'^*_{\text{el}}) \tag{15}$$

$$-(t - t_{\min}) = 4k^* k'^* \sin^2\frac{\theta^*}{2} \qquad \text{elastic} \tag{16}$$

$$k_{\min} = \frac{2MM_J + M_J^2}{2M} \tag{17}$$

$$k^*_{\text{muon}} = \sqrt{\frac{M_J^2}{4} - M_\mu^2} \tag{18}$$

$$\gamma = \sqrt{1 - \frac{k^{*2}}{M^2}} \qquad \text{Lorentz boost from CM to lab} \tag{19}$$

$$\beta = \sqrt{1 - 1/\gamma^2} \tag{20}$$

$$E_J = \gamma(E_J^* + \beta k'^* \cos\theta^*) \tag{21}$$

$$p_{||} \equiv p\cos\theta = \gamma(k'^* \cos\theta^* + \beta E_J^*) \tag{22}$$

$$k = \gamma(1+\beta)k^* \tag{23}$$

$$k^* = \gamma(1-\beta)k \tag{24}$$

$$\gamma(1+\beta) = \frac{\sqrt{s}}{M} \tag{25}$$

$$\gamma(1-\beta) = \frac{M}{\sqrt{s}} \tag{26}$$

$$k'^{*2}_{\max} = \frac{(s - M^s - M_J^2)^2 - 4M^2M_J^2}{4s} \tag{27}$$

$$E^*_{J\,\max} = \frac{s - M^2 + M_J^2}{2\sqrt{s}} \tag{28}$$

$$E^*_{J\,\max} = k^* + \frac{M_J^2}{2\sqrt{s}} \tag{29}$$

$$p_{\max} = \gamma(k'^*_{\max} + \beta E^*_{J\,\max}) \tag{30}$$

$$E_{J\,\max} = \gamma(E^*_{J\max} + \beta k'^*_{\max}) \tag{31}$$

$$x_F = \frac{k'^* \cos\theta^*}{k'^*_{\max}} \tag{32}$$

$$s + t + u = M^2 + W^2 + M_J^2 \tag{33}$$

2 Perl Script

The latest version of the Perl script `psi.pl` can be found at `~griff/bin` on the SLAC Unix cluster. The output gives numbers that correspond to the equation numbers in this document as the reference to the way a quantity has been calculated. The redundancy in the output shows the consistency of the different formulae. `psi.pl` expects k, p, and *theta* as input. It does not exit nicely for forbidden kinematics—either it tries to take the square root of a negative number, or it simply gives absurd results such as $W < M$.

```
#!/usr/bin/perl

# Psi photoproduction kinematics

if( $#ARGV != 2) {
  die "usage: psi.pl k (GeV) p (GeV) theta (degrees)\n";
}

$M = 0.938; # nucleon mass
$MJ = 3.09688; # J/Psi mass
$MU = 0.105658389; # muon mass
$torad = 3.1415926535/180.0;

#print "k (GeV) p (GeV) theta (deg) ";
#$_ = <STDIN>;
#@array = split;
#$kk = $array[0];
#$pp = $array[1];
#$theta = $array[2];

$kk = shift;
$pp = shift;
$theta = shift;


print "     k = $kk\n";
print "     p = $pp\n";
print "     theta = $theta\n";
$theta *= $torad;

$ss = $M*$M + 2*$M*$kk;
print "(1)  s = $ss\n";

$kstar = ($ss - $M*$M)/2/sqrt($ss);
print "(2)  kstar = $kstar\n";

$kstar = $M * $kk / sqrt($ss);
print "(3)  kstar = $kstar\n";

$tmp = sqrt($M*$M+$kstar*$kstar) + $kstar;
$ss = $tmp * $tmp;
print "(4)  s = $ss\n";

$EJ = sqrt($pp*$pp + $MJ*$MJ);
```

```
print "     EJ = $EJ\n";
$uu = -2*$M*$EJ + $M*$M + $MJ*$MJ;
print "(5)  u = $uu\n";

$tt = $MJ*$MJ - 2*$kk*($EJ-$pp*cos($theta));
print "(6)  t = $tt\n";

$tt_el = -2*$M*($kk-$EJ);
print "(7)  t_el = $tt_el\n";

$WW = sqrt( $M*$M + $tt + 2*$M*($kk-$EJ) );
print "(8)  W = $WW\n";

$tmp = ($ss - $WW*$WW - $MJ*$MJ)*($ss - $WW*$WW - $MJ*$MJ);
$tmp -= 4*$WW*$WW*$MJ*$MJ;
$tmp /= 4*$ss;
$kprimestar = sqrt($tmp);
print "(9)  kprimestar = $kprimestar\n";

$tmp = $WW*$WW*$WW*$WW - $M*$M*$M*$M + $MJ*$MJ*$MJ*$MJ;
$tmp -= 2*$ss*($WW*$WW-$M*$M + $MJ*$MJ);
$tmp -= 2*$WW*$WW*$MJ*$MJ;
$tmp /= 4*$ss;
$kprimestar = sqrt($kstar*$kstar + $tmp);
print "(10) kprimestar = $kprimestar\n";

$pt = $pp * sin($theta);
print "(11) pt = $pt\n";
$pt2 = $pt * $pt;
print "     pt2 = $pt2\n";

$thetastar = asin($pt/$kprimestar);
$thetastar /= $torad;
print "(12) thetastar = $thetastar\n";
$thetastar *= $torad;

$EJstar = sqrt($MJ*$MJ + $kprimestar*$kprimestar);
print "     EJstar = $EJstar\n";
$tt = $MJ*$MJ - 2*$kstar*($EJstar - $kprimestar*cos($thetastar) );
print "(13) t = $tt\n";

$tmp = $MJ*$MJ*($MJ*$MJ - 2*$ss - 2*$M*$M);
$tmp /= 4*$ss;
$kprimestar_el = sqrt($kstar*$kstar + $tmp);
print "(14) kprimestar_el = $kprimestar_el\n";
$EJstar_el = sqrt($kprimestar_el*$kprimestar_el+$MJ*$MJ);
```

```
print "     EJstar_el = $EJstar_el\n";

$tmin = $MJ*$MJ - 2*$kstar*($EJstar_el - $kprimestar_el);
print "(14) tmin = $tmin\n";

$tmtmin = $tt - $tmin;
print "     t-t_min = $tmtmin\n";
# only for elastic:
#$tmtmin = -4*$kstar*$kprimestar*sin($thetastar/2)*sin($thetastar/2);
#print "(16) t-t_min = $tmtmin\n";

$kmin = (2*$M*$MJ + $MJ*$MJ)/2/$M;
print "(17) k_min = $kmin\n";

$pt2 = $kprimestar*$kprimestar*sin($thetastar)*sin($thetastar);
print "     pt2 = $pt2\n";

$kmustar = sqrt($MJ*$MJ/4.0 - $MU*$MU);
print "(18) k_mu^star = $kmustar\n";

$gamma = sqrt($M*$M+$kstar*$kstar)/$M;
print "(19) gamma = $gamma\n";
$beta = sqrt(1-1/$gamma/$gamma);
print "(20) beta = $beta\n";

$EJ = $gamma * ($EJstar + $beta*$kprimestar*cos($thetastar));
print "(21) EJ = $EJ\n";
$ppara = $pp * cos($theta);
print "     p_para = $ppara\n";
$ppara = $gamma * ($kprimestar*cos($thetastar) + $beta*$EJstar);
print "(22) p_para = $ppara\n";

$kk = $gamma*(1+$beta)*$kstar;
print "(23) k = $kk\n";
$kstar = $gamma*(1-$beta)*$kk;
print "(24) kstar = $kstar\n";

$ss = ($gamma * (1+$beta) * $M ) * ($gamma * (1+$beta) * $M );
print "(25) s = $ss\n";
$ss = ($M/$gamma/(1-$beta)) * ($M/$gamma/(1-$beta));
print "(26) s = $ss\n";

$tmp = ($ss - $M*$M - $MJ*$MJ) * ($ss - $M*$M - $MJ*$MJ);
$tmp -= 4*$M*$M*$MJ*$MJ;
$tmp /= 4*$ss;
$kprimestar_max = sqrt($tmp);
```

```
print "(27) kprimestar_max = $kprimestar_max\n";

$EJstar_max = ($ss - $M*$M + $MJ*$MJ)/2/sqrt($ss);
print "(28) EJstar_max = $EJstar_max\n";
$EJstar_max = sqrt($MJ*$MJ+$kprimestar_max*$kprimestar_max);
print "     EJstar_max = $EJstar_max\n";
$EJstar_max = $kstar + $MJ*$MJ/2/sqrt($ss);
print "(29) EJstar_max = $EJstar_max\n";

$pmax = $gamma * ($kprimestar_max + $beta * $EJstar_max);
print "(30) pmax = $pmax\n";
$EJ_max = $gamma * ($EJstar_max + $beta * $kprimestar_max);
print "(31) EJ_max = $EJ_max\n";
$EJ_max = sqrt($MJ*$MJ + $pmax * $pmax);
print "     EJ_max = $EJ_max\n";

$xf = $kprimestar * cos($thetastar) / $kprimestar_max;
print "(32) x_F = $xf\n";

$tmp = $ss + $tt + $uu;
print "     s+t+u = $tmp\n";
$tmp = $M*$M + $WW*$WW + $MJ*$MJ;
print "(33) s+t+u = $tmp\n";

sub asin { atan2($_[0], sqrt(1 - $_[0] * $_[0])) }
```