```
> restart:
with(plots):
with(plottools):
```

```
>
Digits:=25:
```

```
> cv := 1:
alv := 1:
```

```
> -(1-al/r)*c^2+dr^2/(1-al/r)+r^2*dte^2:
sqrt(-%):
L:=%;
```

$$L := \sqrt{\left(1-\frac{al}{r}\right)c^2-\frac{dr^2}{1-\frac{al}{r}}-r^2\,dte^2} \tag{1}$$

```
> diff(L, dr):
simplify(%):
radsimp(%) assuming r=3*al, al=1:
pr:=%;

diff(L, dte):
simplify(%):
radsimp(%) assuming r=3*al, al=1:
pte:=%;
```

$$pr := -\frac{r^2\,dr}{\sqrt{\left(al\,dte^2\,r^3-dte^2\,r^4+al^2\,c^2-2\,al\,c^2\,r+c^2\,r^2-dr^2\,r^2\right)r\,(r-al)}}$$

$$pte := \frac{r^3\,(-r+al)\,dte}{\sqrt{\left(al\,dte^2\,r^3-dte^2\,r^4+al^2\,c^2-2\,al\,c^2\,r+c^2\,r^2-dr^2\,r^2\right)r\,(r-al)}} \tag{2}$$

```
> pr*dr+pte*dte-L:
%^2:
simplify(%):
radsimp(%) assuming r=3*al, al=1:
factor(%):
subs([dr^2=dr2, dte^2=dte2], %):
cons1:= %-ep2;
```

$$cons1 := -\frac{(-r+al)^3\,c^4}{\left(al\,dte2\,r^3-dte2\,r^4+al^2\,c^2-2\,al\,c^2\,r+c^2\,r^2-dr2\,r^2\right)r}-ep2 \tag{3}$$

```
> pte^2:
simplify(%):
radsimp(%) assuming r=3*al, al=1:
factor(%):
subs([dr^2=dr2, dte^2=dte2], %):
cons2:=%-k2;
```

$$cons2 := -\frac{dte2\,(-r+al)\,r^5}{al\,dte2\,r^3-dte2\,r^4+al^2\,c^2-2\,al\,c^2\,r+c^2\,r^2-dr2\,r^2}-k2 \tag{4}$$

```
> [cons1, cons2]:
  solve(%, [dr2, dte2]):
  op(%):
  factor(%):
  %;

  subs(%, [dr2, dr2/dte2]):
  wPhi, wPsi := op(%):

  wPhi;
  wPsi;
```

$$\left[dr2 = \frac{c^2 \, (-r + al)^2 \, \left(al \, c^2 \, r^2 - c^2 \, r^3 + al \, c^2 \, k2 - c^2 \, k2 \, r + ep2 \, r^3\right)}{ep2 \, r^5}, dte2 = \frac{c^4 \, k2 \, (-r + al)^2}{ep2 \, r^6}\right]$$

$$\frac{c^2 \, (-r + al)^2 \, \left(al \, c^2 \, r^2 - c^2 \, r^3 + al \, c^2 \, k2 - c^2 \, k2 \, r + ep2 \, r^3\right)}{ep2 \, r^5}$$

$$\frac{\left(al \, c^2 \, r^2 - c^2 \, r^3 + al \, c^2 \, k2 - c^2 \, k2 \, r + ep2 \, r^3\right) r}{c^2 \, k2} \qquad \textbf{(5)}$$

```
> #   Light rays
```

```
> wPhi:
  subs(k2=K2*ep2,%):
  limit(%, ep2=infinity):
  factor(%):
  lwPhi := %;
```

$$lwPhi := \frac{c^2 \, (-r + al)^2 \, \left(K2 \, al \, c^2 - K2 \, c^2 \, r + r^3\right)}{r^5} \qquad \textbf{(6)}$$

```
> wPsi:
  subs(k2=K2*ep2,%):
  limit(%, ep2=infinity):
  factor(%):
  lwPsi := %;
```

$$lwPsi := \frac{\left(K2 \, al \, c^2 - K2 \, c^2 \, r + r^3\right) r}{c^2 \, K2} \qquad \textbf{(7)}$$

```
> lwPhi/lwPsi:
  %;
```

$$\frac{c^4 \, (-r + al)^2 \, K2}{r^6} \qquad \textbf{(8)}$$

```
> Ingoing := -1:
  Outgoing:=  1:
```

```
Clockwise        := -1:
Counterclockwise :=  1:
```

```
> lwPhi:
subs([al=alv, c=cv], %):
sqrt(%):
radsimp(%) assuming r>alv, K2>0:
simplify(%) assuming r>alv, K2>0:
1/%:
subs(r=R, %):
Int(%, R=r0..r):
ltr:=t0+ branch*%;
#
#  branch =  1 : Outgoing
#  branch = -1 : Ingoing
```

$$ltr := \left(\int_{r0}^{r} \frac{R^{5/2}}{(R-1)\sqrt{R^3 - K2\,R + K2}}\,\mathrm{d}R\right) branch + t0 \quad \textbf{(9)}$$

```
> lwPsi:
subs([al=alv, c=cv], %):
sqrt(%):
simplify(%) assuming r>alv, K2>0:
1/%:
subs(r=R, %):
Int(%, R=r0..r):
lter:=te0+ clock * branch* %;

#
#  clock =  1 : Counterclockwise
#  clock = -1 : Clockwise
```

$$lter := \left(\int_{r0}^{r} \frac{\sqrt{K2}}{\sqrt{R}\,\sqrt{R^3 - K2\,R + K2}}\,\mathrm{d}R\right) branch\ clock + te0 \quad \textbf{(10)}$$

```
> # Discussion of different region depends on the sign of R^3 - K2*R
+ K2.
```

```
> lwPsi*K2*c^2/r:
#subs([al=alv, c=cv], %):
%;
diff(%, r):
solve(%, r):
rc:=[%][1];
```

$$K2\,al\,c^2 - K2\,c^2\,r + r^3$$

$$rc := \frac{\sqrt{3}\,\sqrt{K2}\,c}{3} \quad \textbf{(11)}$$

```
> lwPsi*K2*c^2/r:
  #subs([al=alv, c=cv], %):
  subs(r=rc, %):
  subs(K2=K^2, %):
  simplify(%) assuming K>0:
  solve(%/K^2, K):
  K2c:=%^2;
```

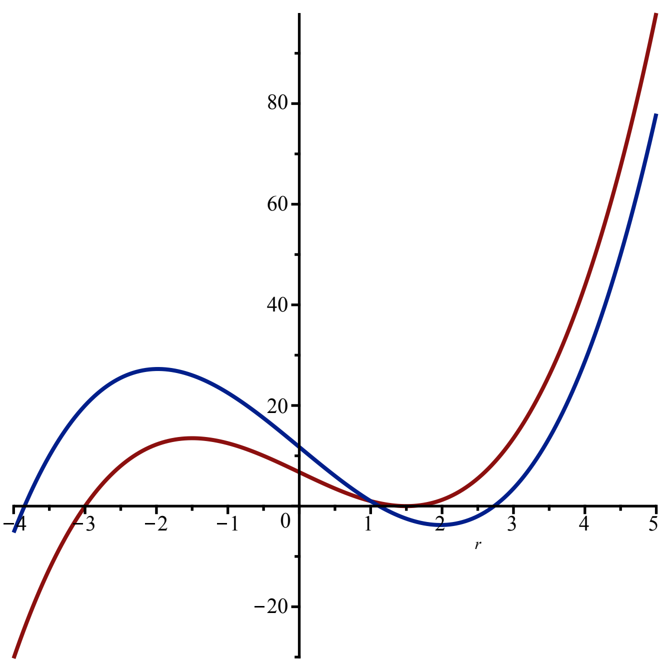
$$K2c := \frac{27\, al^2}{4\, c^2} \tag{12}$$

```
> rc:
  subs(K2=K2c, %):
  simplify(%):
  rc:=%;
```

$$rc := \frac{3\,\sqrt{\frac{al^2}{c^2}}\; c}{2} \tag{13}$$

```
> # per K2 > K2c = 27/4
  # r > rm        #  scattering
  # 1 < r < r0    #  infalling


  lwPsi*K2*c^2/r:
  subs([al=alv, c=cv], %):
  [subs(K2=K2c, %), subs(K2=K2c+5, %)]:
  plot(%, r=-4..5);
```




```
> lwPsi*K2*c^2/r:
  subs([al=alv, c=cv], %):
  %-(r-r1)*(r-rm)*(r+r1+rm):
  collect(%, r):
  [subs(r=0, %), subs(r=0, diff(%, r))]:
  solve(%, [K2, r1]):
  allvalues(%):
```

```
[%][1]:
op(%):
subs(%, [K2, r1]):
K2v, r1v := op(%):

K2v;
r1v;
```

$$\frac{rm^3}{rm-1}$$

$$\frac{\left(-rm+1+\sqrt{rm^2+2\,rm-3}\right)rm}{2\,(rm-1)} \tag{14}$$

```
> r1v:
subs(rm=3/2, %):
simplify(%):
%;
```

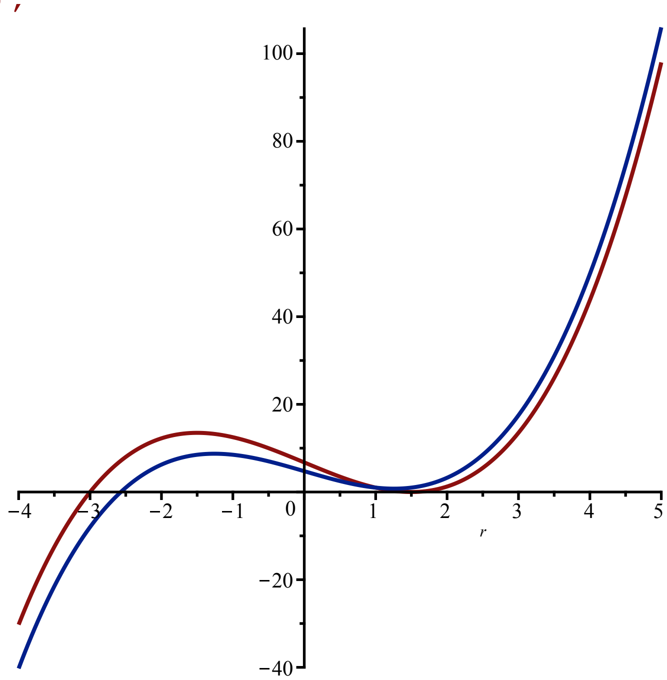
$$\frac{3}{2} \tag{15}$$

```
> # K2 = K2c is singular

# Phi has a double zero which is an asymptotic goal which makes the
intengrals divergent
# it corresponds to 1 (actually 3) single lines, hence we can
complete by continuity and avoid it.
```

```
> # For 0 < K2 < K2c = 27/4
# r > 0    #  infalling


lwPsi*K2*c^2/r:
subs([al=alv, c=cv], %):
[subs(K2=K2c, %), subs(K2=K2c-2, %)]:
plot(%, r=-4..5);
```



```
> WavefrontThrough := proc(r0v, te0v, t0v, branchv, clockv)
    [r*cos(lter), r*sin(lter), ltr]:
    subs([r0=r0v, te0=te0v, t0=t0v], %):
    subs([branch=branchv, clock=clockv], %):
    evalf(%):
  end:
```

```
> # if (0 < K2 < K2c) everything is easy:

  #  if we are on ingoing branch the ray first passes on P0 then it
  falls into the BH,
  #     then we need r > r0
  #  if we are on outgong branch the rays comes from near the BH
  through P0 to infinity
  #     then we need 1 < r < r0
```

```
> # let's check it computes and it is real for some value

  WavefrontThrough(5, Pi/4, 13, Ingoing, Counterclockwise):
  subs(K2=1, %):
  subs(r=12, %):
  evalf(%):
  %;
```

[9.423183589434292165494522, 7.429913259239050461323544,
4.929269597601190917561561] **(16)**

```
> # r < z  so that it is easy to change

  z := 50;

  # The vertex of the wavefront

  Pr := 5;
  Pte := Pi/4;
  Pt := 13;
```

$$z := 50$$

$$Pr := 5$$

$$Pte := \frac{\pi}{4}$$

$$Pt := 13 \quad \textbf{(17)}$$

```
> # let us start from ingoing particles
  #  0 < K2 < K2c  => r0 < r < z
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Counterclockwise):
  subs(K2=0, %):
  simplify(%):
  #%;
  C0:=spacecurve(%, r=Pr..z, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Counterclockwise):
  subs(K2=K2c, %):
  simplify(%):
  Cp:=spacecurve(%, r=Pr..z, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Clockwise):
  subs(K2=K2c, %):
  evalf(%):
  #%;
  Cm:=spacecurve(%, r=Pr..z, color=green, thickness=4):
```

```
> # Surface

  WavefrontThrough(Pr, Pte, Pt, Ingoing, Counterclockwise):
  evalf(%):
  #%;
  W1:=plot3d(%, K2=0..K2c-0.01, r=Pr..z, color=red):
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Clockwise):
  evalf(%):
  #%;
  W2:=plot3d(%, K2=0..K2c, r=Pr..z, color=red):
```

```
> BH := spacecurve([0,0,s, s=-5000..5000],
  axes=box, color = black, linestyle = dash):
```

```
> [R*cos(te), R*sin(te), t]:
  subs(R=1*alv, %):
  plot3d(%, te=0..2*Pi, t=-5000..5000):
  S := %:
```

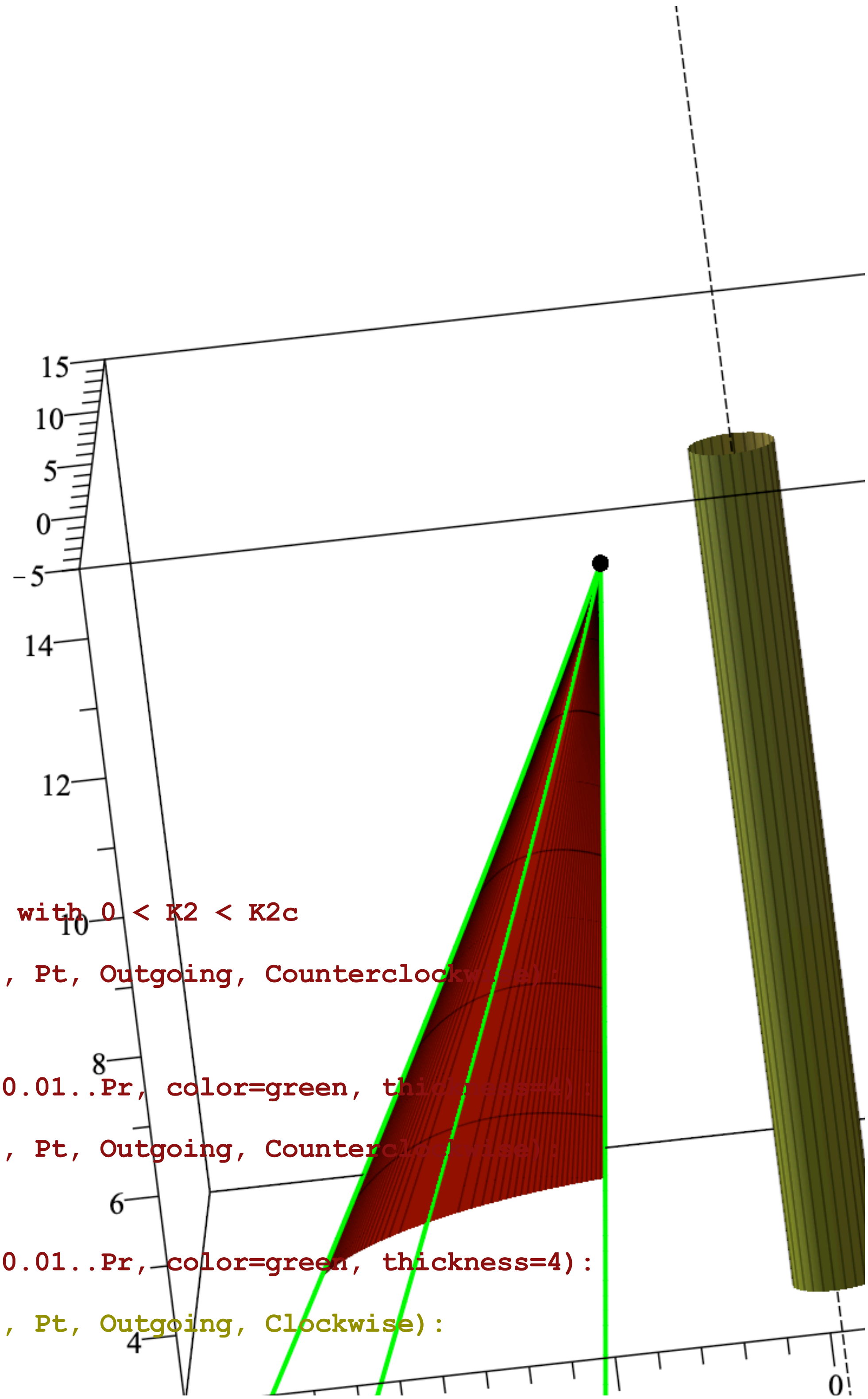
```
> PM := pointplot3d([5*cos(Pi/4), 5*sin(Pi/4), 13],
  color=black, symbol = solidcircle, symbolsize = 10):
```

```
> [z*cos(s), z*sin(s), Pt-z]:
  zLine:=spacecurve(%, s=0..2*Pi, color=green, thickness=1):

  [Pr*cos(s), Pr*sin(s), Pt-z]:
  PLine:=spacecurve(%, s=0..2*Pi, color=green, thickness=1):

  [3/2*cos(s), 3/2*sin(s), Pt-z]:
  zline:=spacecurve(%, s=0..2*Pi, color=green, thickness=1):
```

```
> display(BH, S, PM,
  zLine, PLine, zline,
  C0, Cp,
  W1,
  W2, Cm,
  view=[ -5..15, -5..15, 3..15]);
```




```
>
  # Now Outgoing particles with 0 < K2 < K2c
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  subs(K2=0, %):
  simplify(%):
  #%;
  c0:=spacecurve(%, r=3/2+0.01..Pr, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  subs(K2=K2c, %):
  simplify(%):
  #evalf(%);
  cp:=spacecurve(%, r=3/2+0.01..Pr, color=green, thickness=4):
```

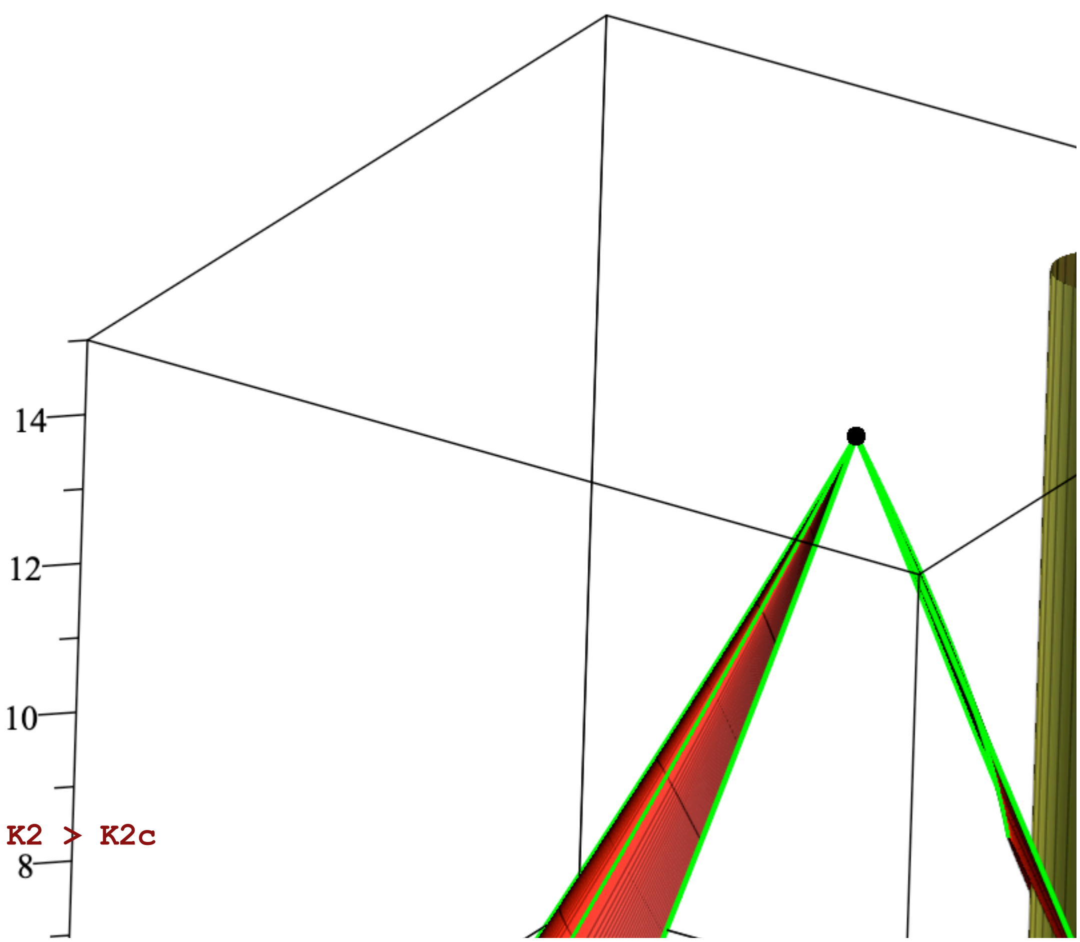
```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
  subs(K2=K2c, %):
  simplify(%):
```

```
#evalf(%);
cm:=spacecurve(%, r=3/2+0.01..Pr, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
simplify(%):
W3:=plot3d(%, K2=0..K2c, r=3/2+0.01..Pr, color=red):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
simplify(%):
W4:=plot3d(%, K2=0..K2c, r=3/2+0.01..Pr, color=red):
```

```
> display(BH, S, PM,
zLine, PLine, zline,
C0, Cp, W1,          W2, Cm,
#
c0, cp,
W3,
W4, cm,
view=[ -5..15, -5..15, 3..15]);
```




```
> # Then we focus on K2 > K2c
# 3 solutions
```

```
> # We split  rm < r      Ingoing  (now)
  #           1 < r < r1 Outgoing (later)
```

```
> # for K2 > K2c each ray has an incoming and outgoing branch.
  # the point P0 is on one branch, the other branch is called the
  Other.

  # Thus we can consider an ingoing ray (meaning that P0 is on the
  incoming branch)
  # and we call Other branch the outgoing branch.

  # this time we need to draw both branches of each ray.
```

```
> # Let us start from P0 branch, setting initial conditions in P0
```

```
> lwPhi:
  subs([al=alv, c=cv], %):
  subs(r^3=(r-r1)*(r-rm)*(r+r1+rm)+K2*r-K2, %):
  subs(K2=K2v, %):
  radsimp(%) assuming r>rm, rm>r1, r1>3/2:
  simplify(%) assuming r>rm, rm>r1, r1>3/2:
  sqrt(%):
  radsimp(%) assuming r>rm, rm>r1, r1>3/2:
  simplify(%) assuming r>rm, rm>r1, r1>3/2:
  1/%:
  subs(r=R, %):
  int(%, R=r0..r) assuming r>r0, r0>rm, rm>r1, r1>3/2:
  radsimp(%) assuming r>rm, rm>r1, r1>3/2, r0>rm:
  simplify(%) assuming r>rm, rm>r1, r1>3/2, r0>rm:
  ltr:=t0+ branch*%:
  #%;

  #
  #  branch =  1 : Outgoing
  #  branch = -1 : Ingoing
```

```
> lwPsi:
  subs([al=alv, c=cv], %):
  subs(r^3=(r-r1)*(r-rm)*(r+r1+rm)+K2*r-K2, %):
  subs(K2=K2v, %):
  sqrt(%):
  radsimp(%) assuming r>rm, rm>r1, r1>3/2:
  simplify(%) assuming r>rm, rm>r1, r1>3/2:
  1/%:
  subs(r=R, %):
  int(%, R=r0..r) assuming r>r0, r0>rm, rm>r1, r1>3/2:
  radsimp(%) assuming r>rm, rm>r1, r1>3/2, r0>rm:
  simplify(%) assuming r>rm, rm>r1, r1>3/2, r0>rm:
  lter:=te0+ clock * branch* %:
  #%;

  #
  #  clock =  1 : Counterclockwise
  #  clock = -1 : Clockwise
```

```
> #
  #  Clockwise          Outgoing    red  small
  #  Clockwise          Outgoing    blue small   rm =[1.5 .. 5], r =
  [rm .. 5]  W5
  #
  #  Clockwise          Ingoing     red  big
  #  Clockwise          Ingoing     blue big     rm =[1.5 .. 5], r =
  [5 .. 50]  W8

  #  Counterclockwise   Outgoing
  #  Counterclockwise   Ingoing
  #
```

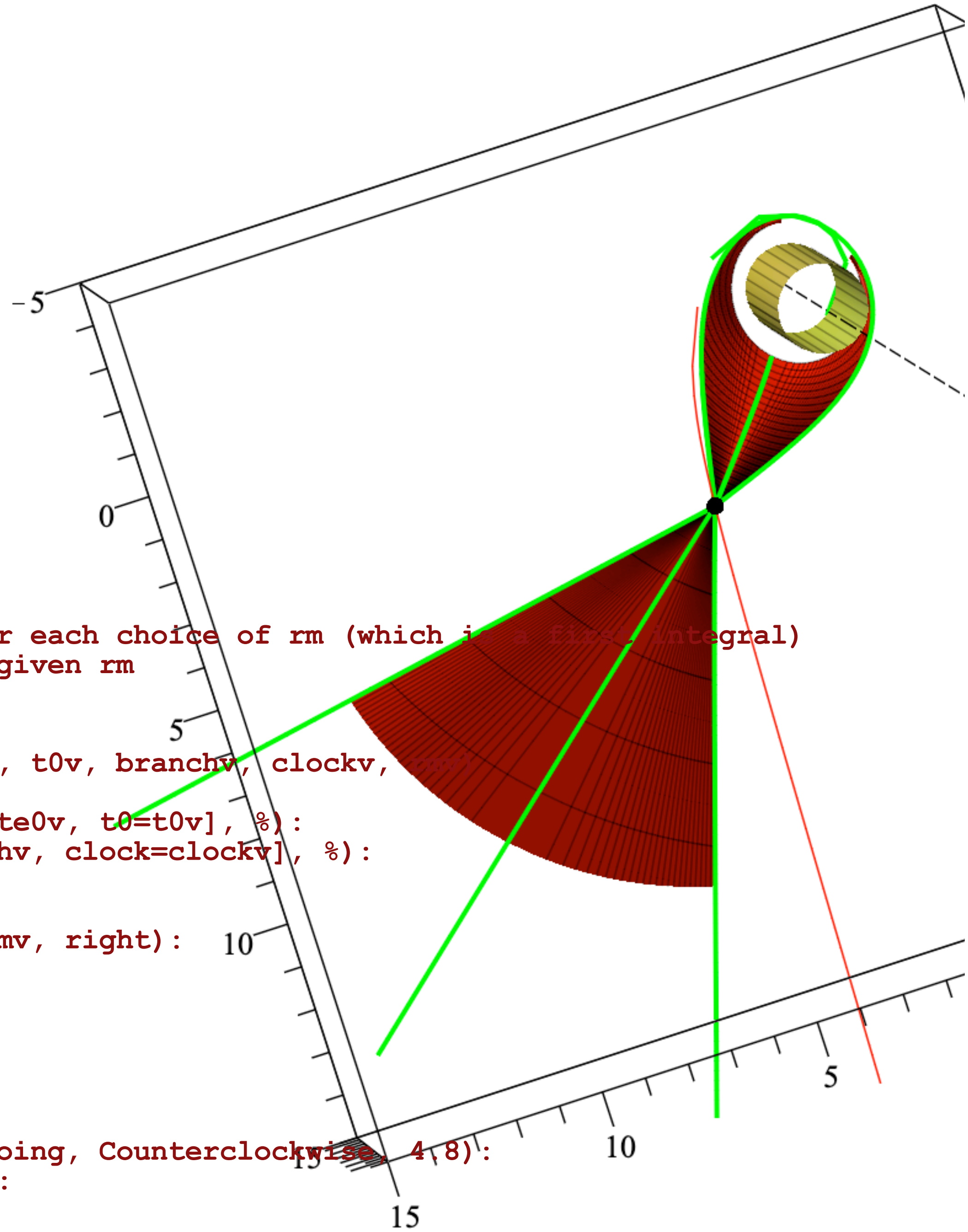
```
> WavefrontThrough := proc(r0v, te0v, t0v, branchv, clockv)
    [r*cos(lter), r*sin(lter), ltr]:
    subs([r0=r0v, te0=te0v, t0=t0v], %):
    subs([branch=branchv, clock=clockv], %):
  end:
```

```
>
  # let us just check that Ingoing/Outgoing and
  Clockwise/Counterclockwise
  # are correctly implemented
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Clockwise):
  #WavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
  #WavefrontThrough(Pr, Pte, Pt, Ingoing, Counterclockwise):
  #WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):


  simplify(%):
  subs(r1=r1v, %):
  subs(rm=3, %):
  radsimp(%):
  simplify(%):
  evalf(%):
  pr1:= spacecurve(%, r=3.01..z, color=red, thickness=1):
```

```
> display(BH, S, PM,
  zLine, PLine, zline,
  C0, Cp, W1,        W2, Cm,
  #
  c0, cp, W3,        W4, cm,
  pr1,
  view=[ -5..15, -5..15, 3..15]);
```




```
> # we have one ray for each choice of rm (which is a first integral)
  # let is compute Pm given rm
```

```
> Pm := proc(r0v, te0v, t0v, branchv, clockv, r1v)
     [r, lter, ltr]:
     subs([r0=r0v, te0=te0v, t0=t0v], %):
     subs([branch=branchv, clock=clockv], %):
     subs(r1=r1v, %):
     subs(rm = rmv, %):
     map(limit,%, r = rmv, right):
     evalf(%):
     #evalf(%):
     #map(Re, %):
  end:
```

```
> Pm(Pr, Pte, Pt, Outgoing, Counterclockwise, 4.8):
  Rm, Tem, Tm := op(%):

  Rm;
  Tem;
  Tm;
```

$$4.8$$

$$0.4436453641746036429047941$$

$$11.112013418788173832500006 \tag{18}$$

```
> OtherRayThrough := proc(r0v, te0v, t0v, branchv, clockv, rmv)
    local Rm, Tem, Tm:
    Pm(r0v, te0v, t0v, branchv, clockv, rmv):
    Rm, Tem, Tm := op(%):
    WavefrontThrough(Rm, Tem, Tm, -branchv, clockv):
    simplify(%):
    subs(r1=r1v, %):
    subs(rm=rmv, %):
  end:
```

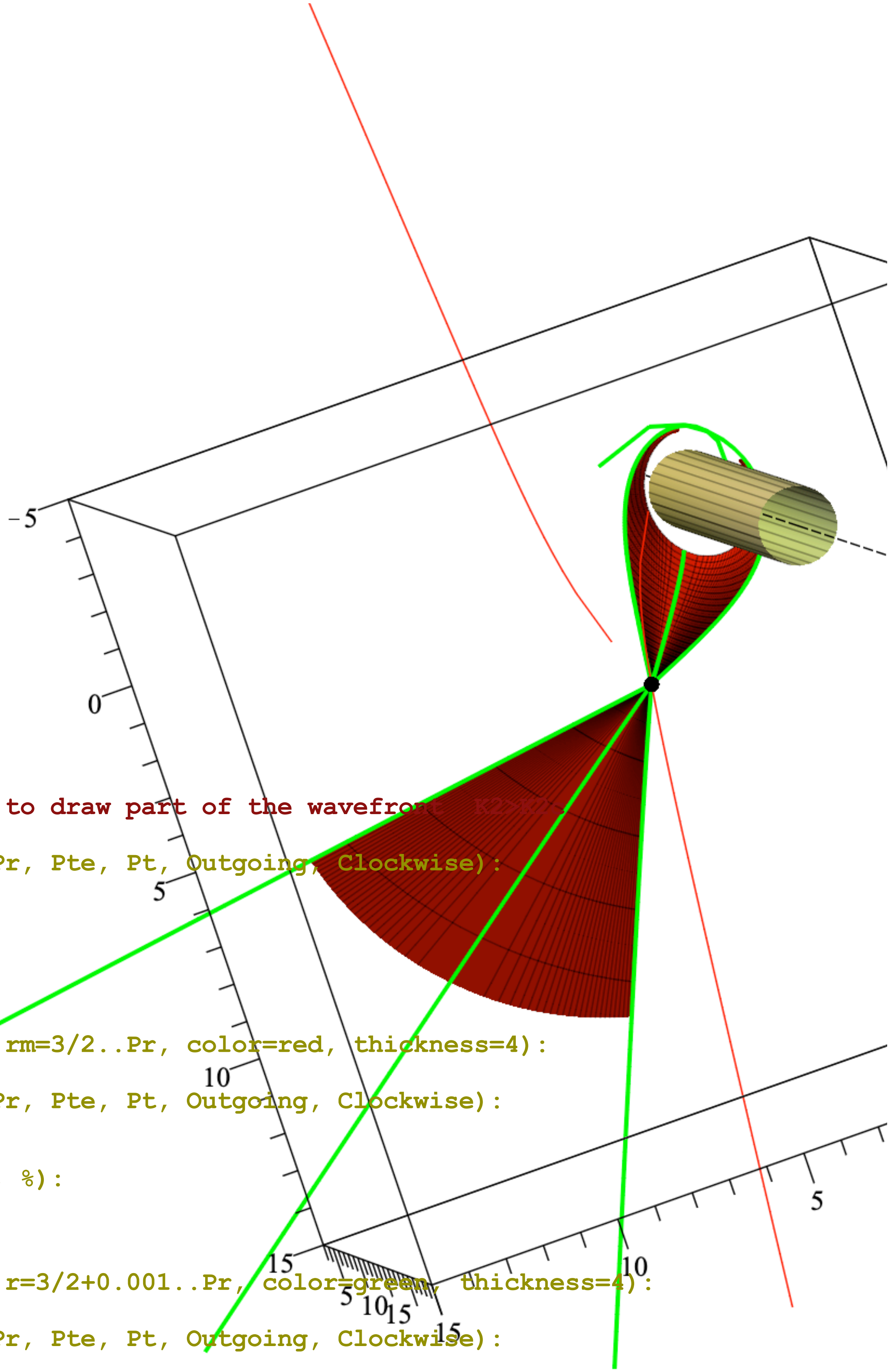
```
> OtherRayThrough(Pr, Pte, Pt, Ingoing, Clockwise, 3):

  subs(r=3, %):
  evalf(%):
  %;
```

$$[2.6520448299475280139475 56, -1.4023759196266125342904 70, 19.156077203369455241899 19] \tag{19}$$

```
> OtherRayThrough(Pr, Pte, Pt, Ingoing, Clockwise, 4.8):
  radsimp(%):
  simplify(%):
  evalf(%):
  Otherpr1:= spacecurve(%, r=4.8001..z, color=red, thickness=1):
```

```
> display(BH, S, PM,
  zLine, PLine, zline,
  C0, Cp, W1,        W2, Cm,
  #
  c0, cp, W3,        W4, cm,
  pr1, Otherpr1,
  view=[ -5..15, -5..15, 3..25]);
```




```
> # Ok we are ready to draw part of the wavefront  K2>K2.
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(r=rm, %):
  evalf(%):
  #map(Re,%):
  #%;
  kp:=spacecurve(%, rm=3/2..Pr, color=red, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(rm=3/2+0.001, %):
  evalf(%):
  #map(Re,%):
  #%;
  km:=spacecurve(%, r=3/2+0.001..Pr, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
  simplify(%):
  subs(r1=r1v, %):
  evalf(%):
  #%;
  W5:=plot3d(%, rm=3/2..Pr, r=rm..Pr, color=blue):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(r=rm, %):
  evalf(%):
  #map(Re,%):
  #%;
  Kp:=spacecurve(%, rm=3/2..Pr, color=red, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(rm=3/2+0.001, %):
  evalf(%):
  #map(Re,%):
  #%;
  Km:=spacecurve(%, r=3/2+0.001..Pr, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  simplify(%):
  subs(r1=r1v, %):
  evalf(%):
  #%;
  W6:=plot3d(%, rm=3/2..Pr, r=rm..Pr, color=blue):
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Counterclockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(rm=Pr, %):
  evalf(%):
  #map(Re,%):
  #%;
  K:=spacecurve(%, r=Pr..z, color=green, thickness=4):
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Counterclockwise):
  simplify(%):
  subs(r1=r1v, %):
  evalf(%):
  #subs(rm=3/2, %):
  #subs(r=3/2, %):
  #evalf(%):
  #%;
  W7:=plot3d(%, rm=3/2..Pr, r=Pr..z, color=blue):
```

```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Clockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(rm=Pr, %):
  evalf(%):
  #map(Re,%):
  #%;
  k:=spacecurve(%, r=Pr..z, color=green, thickness=4):
```

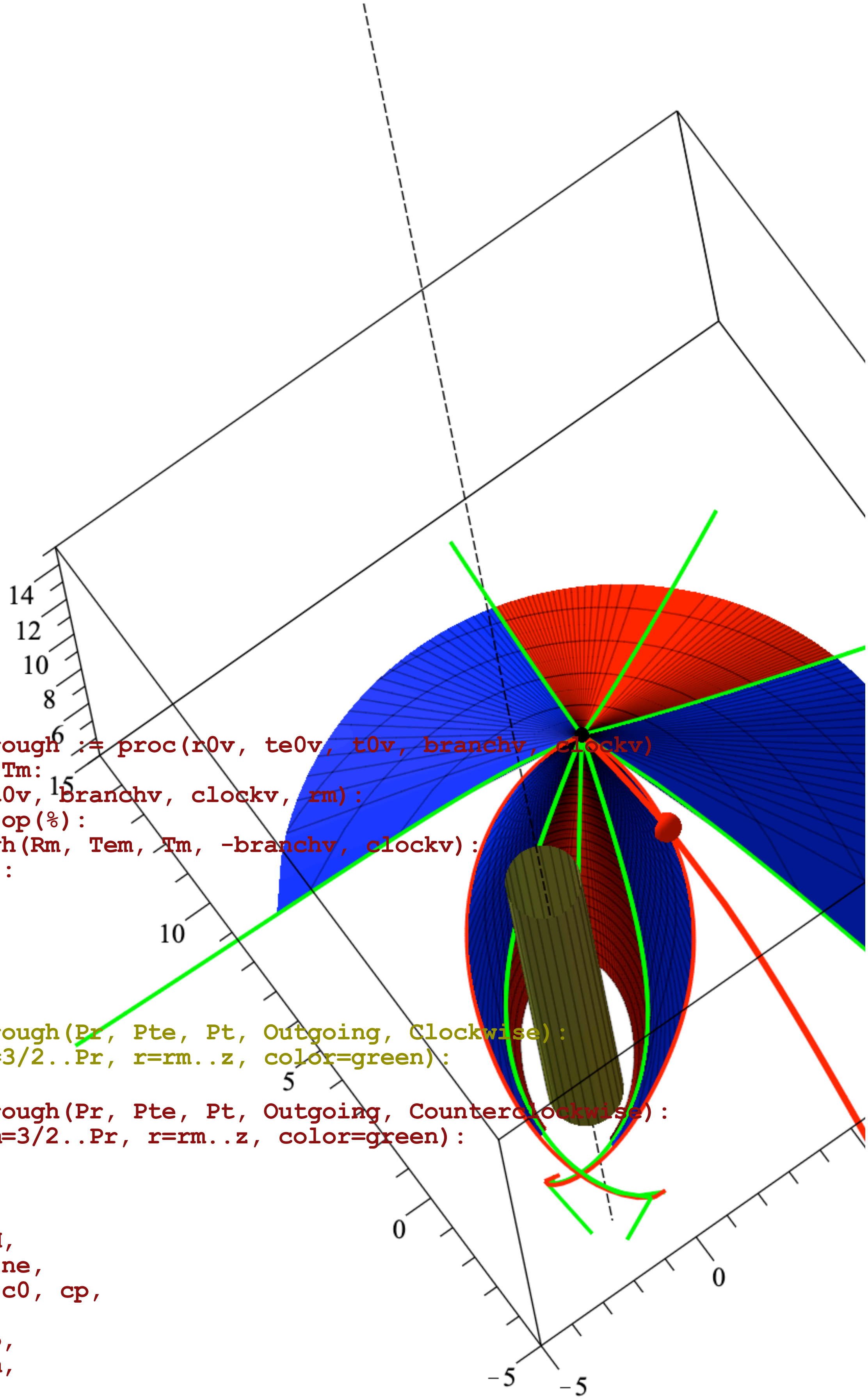
```
> WavefrontThrough(Pr, Pte, Pt, Ingoing, Clockwise):
  simplify(%):
  subs(r1=r1v, %):
  evalf(%):
  #%;
  W8:=plot3d(%, rm=3/2..Pr, r=Pr..z, color=blue):
```

```
> WavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  simplify(%):
  subs(r1=r1v, %):
  subs(rm=4.6, %):
  evalf(%):
  #%;
  OneLine:=spacecurve(%, r=4.6..Pr, color=red, thickness=14):
```

```
> Pm(Pr, Pte, Pt, Outgoing, Counterclockwise, 4.6):
  Rm, Tem, Tm := op(%):
  [Rm*cos(Tem), Rm*sin(Tem), Tm]:
  M := pointplot3d(%,
  color=red, symbol = solidcircle, symbolsize = 20):
```

```
> OtherRayThrough(Pr, Pte, Pt, Outgoing, Counterclockwise, 4.6):
  evalf(%):
  #%;
  OtherLine:=spacecurve(%, r=4.6..z, color=red, thickness=14):
```

```
> display(BH, S, PM,
  zLine, PLine, zline,
  W1,  W3, C0, Cp, c0, cp,
  W2, W4,  Cm,  cm,
  W6, W8, Km, K, Kp,
  W5, W7, k, kp, km,
  OneLine, OtherLine, M,
  view=[ -5..15, -5..15, 3..15]);
```



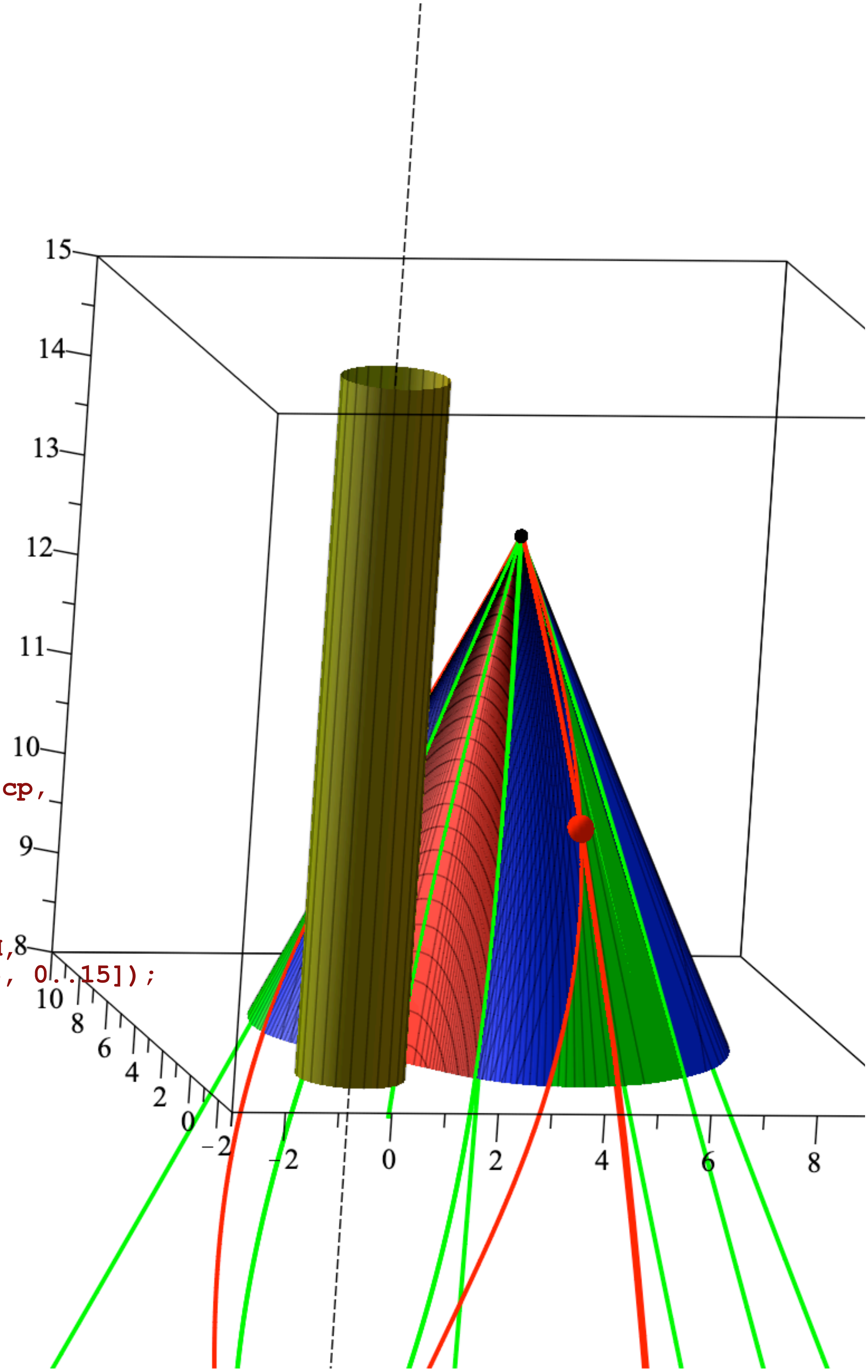

```
> OtherWavefrontThrough := proc(r0v, te0v, t0v, branchv, clockv)
    local Rm, Tem, Tm:
    Pm(r0v, te0v, t0v, branchv, clockv, rm):
    Rm, Tem, Tm := op(%):
    WavefrontThrough(Rm, Tem, Tm, -branchv, clockv):
    subs(r1=r1v, %):
  end:
```

```
> OtherWavefrontThrough(Pr, Pte, Pt, Outgoing, Clockwise):
  w9:=plot3d(%, rm=3/2..Pr, r=rm..z, color=green):
```

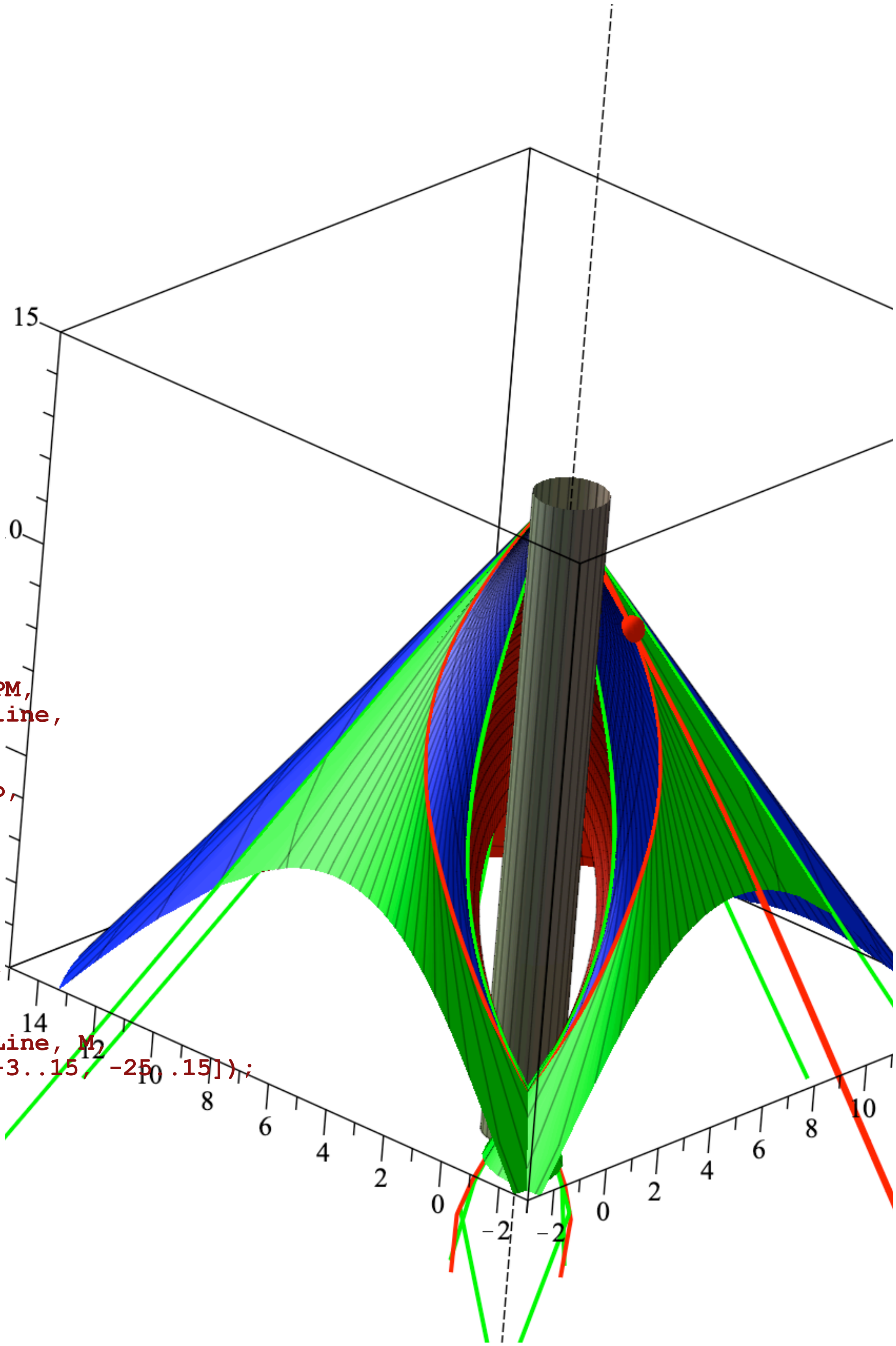
```
> OtherWavefrontThrough(Pr, Pte, Pt, Outgoing, Counterclockwise):
  w10:=plot3d(%, rm=3/2..Pr, r=rm..z, color=green):
```

```
> display(BH, S, PM,
  zLine, PLine, zline,
  W1,  W3, C0, Cp, c0, cp,
  W2, W4,  Cm,  cm,
  W6, W8, Km, K, Kp,
  W5, W7, k, kp, km,
  w9,
```

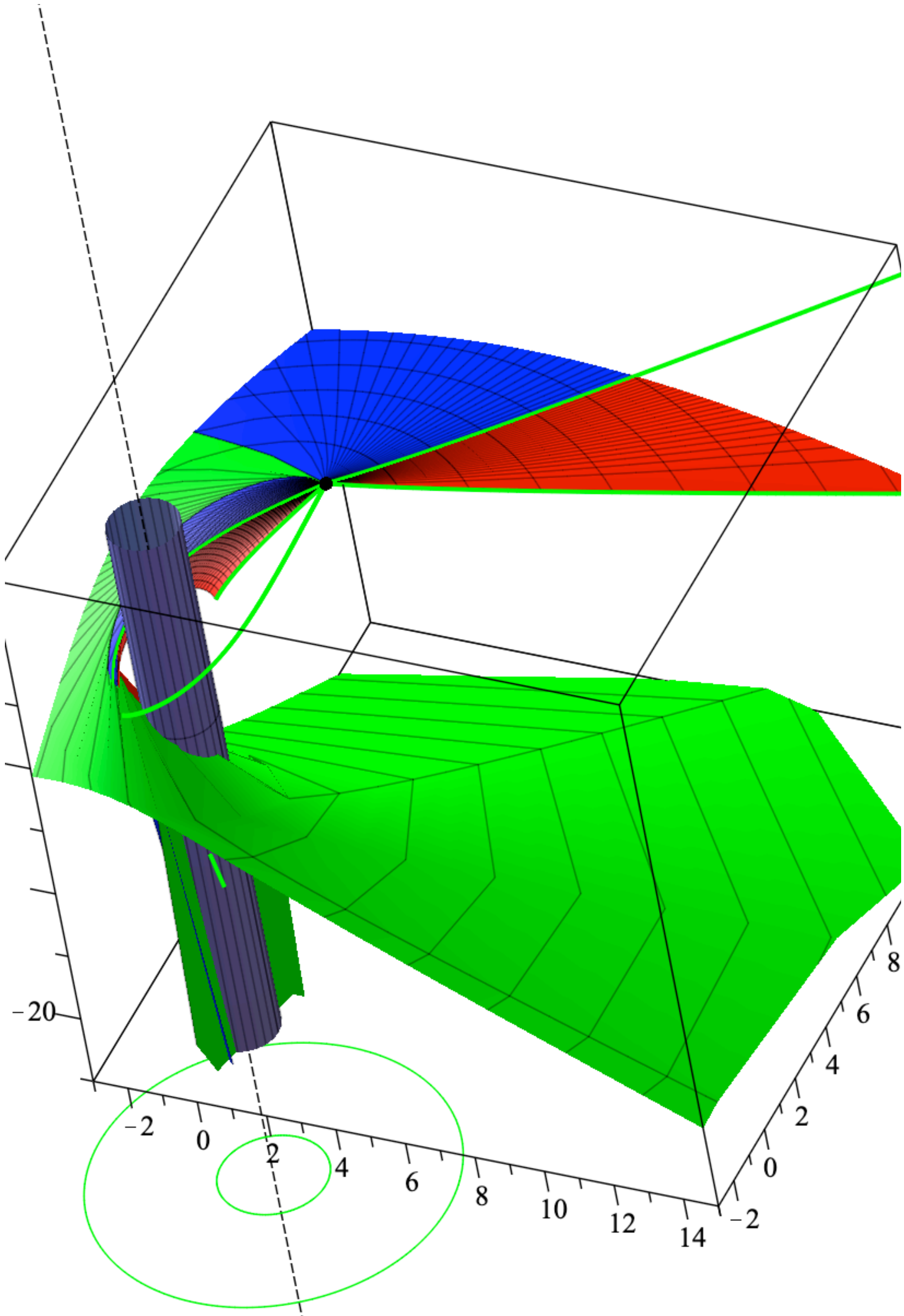
```
w10,
OneLine, OtherLine, M,
view=[ -3..10, -3..10, 8..15]);
```




```
> display(BH, S, PM,
zLine, PLine, zline,
W1,  W3, C0, Cp, c0, cp,
W2, W4,  Cm,  cm,
W6, W8, Km, K, Kp,
W5, W7, k, kp, km,
w9,
w10,
OneLine, OtherLine, M,
view=[ -3..15, -3..15, 0..15]);
```




```
> display(BH, S, PM,
  zLine, PLine, zline,
  #W1,
  #W3,
  C0,  c0, cp, #Cp,
  W2,
  W4,    cm, Cm,
  #W6,
  W8,
   Km, #Kp, #K,
  #W7,
  W5, k, #kp, #km,
  w9,
  #w10,
  #OneLine, OtherLine, M,
  view=[ -3..15, -3..15, -25..15]);
```



−20
−2
0
2
4
6
8
10
12
14
−2
0
2
4
6
8