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

```
> # Let us take note of the initial time, just for efficiency stats

  mTime:= time():
```

```
> # let us set precision to a number of digits.

  Digits:=80;
  digits:=Digits:
```

$$Digits := 80 \tag{1}$$

```
> # let us fix the value of c and alpha. In homogeous units we have


  cv := 299792458:
  MEv:= 597237*10^19:
  Gv:=667430*10^(-16):
  alv:=2*MEv*Gv/cv^2:

  Surface:= 3189000;  # in meter
```

$$Surface := 3189000 \tag{2}$$

```
> # The Lagrangian for material points in a Schwarzschild
  gravitational field (in polar coordinates)

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

$$L := \sqrt{\left(1 - \frac{al}{r}\right) c^2 - \frac{dr^2}{1 - \frac{al}{r}} - r^2\, dte^2} \tag{3}$$

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

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

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

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

```
> # Total energy first integral

  pr*dr+pte*dte-L:
  %^2:
  simplify(%):
  radsimp(%) assuming r>alv, al=alv:
  factor(%):
  subs([dr^2=dr2, dte^2=dte2], %):
  cons1:= %-c^4*ep2;
```

$$cons1 := -c^4\,ep2 - \frac{(-r+al)^3\,c^4}{\left(r^3\,dte2\,al - r^4\,dte2 + al^2\,c^2 - 2\,al\,c^2\,r + c^2\,r^2 - dr2\,r^2\right)\,r} \tag{5}$$

```
> # Angular momentum first integral

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

$$cons2 := -\frac{dte2\,(-r+al)\,r^5}{r^3\,dte2\,al - r^4\,dte2 + al^2\,c^2 - 2\,al\,c^2\,r + c^2\,r^2 - dr2\,r^2} - k2 \tag{6}$$

```
> # Weierstrass equations

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

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

  wPhi;
  wPsi;
```

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

$$\frac{\left(c^2\,ep2\,r^3 + al\,r^2 - r^3 + k2\,al - k2\,r\right)\,r}{k2} \tag{7}$$

```
> # now we need to slow down and put Weierstrass in a smart form
  # so that later we can analytically integrate Weisrstrass
  equations.
  # essentially we want to factorize it in first order polynomials
>
  # Remember that:  -c < ep < 0
```

```
> numer(wPsi)/r:
  collect(%, r):
  P:=%;
```

$$P := \left(c^2\, ep2 - 1\right) r^3 + al\, r^2 - k2\, r + k2\, al \tag{8}$$

```
> P + (ep2*c^2-1)*(rp-r)*(r-rm)*(r-r0):
  collect(%, r):
  [subs(r=0, %), subs(r=0, diff(%, r)), subs(r=0, diff(%, r, r))]:
  simplify(%):
  solve(%, [ep2, k2, r0]):
  op(%):
  Sol1 := %;
```

$$Sol1 := \left[ep2 = -\frac{al^2\, rm + al^2\, rp - al\, rm^2 - 2\, al\, rm\, rp - al\, rp^2 + rm^2\, rp + rm\, rp^2}{c^2\left(al\, rm^2 + al\, rm\, rp + al\, rp^2 - rm^2\, rp - rm\, rp^2\right)}, k2 = -\frac{al\, rm^2\, rp^2}{al\, rm^2 + al\, rm\, rp + al\, rp^2 - rm^2\, rp - rm\, rp^2}, r0 = -\frac{al\, rm\, rp}{al\, rm + al\, rp - rp\, rm}\right] \tag{9}$$

```
> wPhi:
  subs(Sol1, %):
  simplify(%):
  mPhi := %;
```

$$mPhi := \frac{\left(\left((r + rp)\, rm + r\, rp\right) al - r\, rm\, rp\right)(r - al)^2\, c^2\, (r - rp)\, al\, (r - rm)}{r^5\, (rm + rp)\, (al - rp)\, (al - rm)} \tag{10}$$

```
> wPsi:
  subs(Sol1, %):
  simplify(%):
  mPsi := %;
```

$$mPsi := \frac{(r - rp)\, (r - rm)\, r\, \left(\left((al - rp)\, rm + al\, rp\right) r + al\, rm\, rp\right)}{rm^2\, rp^2} \tag{11}$$

```
> (1-al/r)*c^2/mPhi - 1/(1-al/r) - r^2/mPsi:
  simplify(%) assuming c=cv, al=alv, rm=10*alv, rp=30*alv, r=20*alv:
  sqrt(%)/c:
  radsimp(%) assuming c=cv, al=alv, rm=10*alv, rp=30*alv, r=20*alv:
  factor(%):
  dtau := %;
```

$$dtau := \left(r\left(-\left(al\, rm^2 + al\, rm\, rp + al\, rp^2 - rm^2\, rp - rm\, rp^2\right) r\, (r - rp)\, (r - rm)\, (al\, r\, rm + al\, rp\, r + al\, rm\, rp - r\, rm\, rp)\, al\right)^{1/2}\right) \Big/ \left((r - rp)\, (r - rm)\, (al\, r\, rm + al\, rp\, r + al\, rm\, rp - r\, rm\, rp)\, al\, c\right) \tag{12}$$

Utilities

```
> # Utility procedures
```

```
> Real:= proc(x)
  global ErrorFlag:
  local X:
   if type(x, list) then
    map(Real, x):
   else
    X := Im(x):
    if X = 0 then
    elif X < 10^(-Digits+5) and X > -10^(-Digits+5) then
     printf("Imaginary part neglected: %a\n", X);
    elif X > 10^(-Digits+5) or X < -10^(-Digits+5) then
     printf("Imaginary part neglected (Flag): %a\n", X);
     ErrorFlag := true:
    end if:
    Re(x):
   end if:
  end:
```

```
> BranchWith := proc(S, t)
  local TSv:
   TSv:= halfT(S):
   t-tS(S, 0):
   subs(r=minS(S), %):
   evalf(%):
   %/TSv:
   evalf(%):
   floor(%):
  end:
```

```
> # When solving equations we parameterize t and theta in terms of
  r, which is not monotonic
  # thus orbits are broken in branches which need to be glue
  together to get an orbit.
  # That for satellites going along bounded orbits (and similarly
  later for light rays) can be done by the following procdure,
  # best understood by checking laler usage.
```

```
> Branch := proc(fr, branch, F, f0)
    if(floor(branch) mod 2 = 0) then
     fr:
     f0 + % + 2*F*floor(branch/2):
    else
     fr:
     f0 + 2*F*(floor(branch/2)+1) - %:
    end if:
    evalf(%):
  end:
```

```
> isOutgoing:=proc(b)
   if b mod 2 = 0 then
    return true:
   end if:
   return false:
  end:
```

```
> myRange:= proc(x1, x2)
   if x1<x2 then
    return x1..x2;
   else
    return x2..x1;
   end if:
  end:
```

```
> # Utility ciclic permutations

  Cycle:= proc(list)
   local x, n, N, r:
   x:= list[1]:
   N:= nops(list):
   r:= list:
   for n from 1 to N-1 do
    r[n] := list[n+1]:
   end do:
   r[N]:= x:
   r;
  end:
```

```
> CycleUntil:=proc(list, n)
   local r:
   if not(n in list) then
     printf("code %d is not in the list %a\n", n, list);
     return list:
   end if:
   r:= list;
   do
    r:= Cycle(r):
   until r[1]=n:
   r:
  end:
```

```
> CreateList := proc(n, v := none)
    local k:
    [seq(v,  1..n)]:
  end:
```

```
> NextEvent:=proc(Sat)
    global NextSignalAvailable:
    3*NextSignalAvailable[Sat+1] +Sat +1:
  end:
```

```
> Q:= proc(x)
   round(x*10^(Digits-5))/10^(Digits-5):
  end:
```

```
> #
  # Now let us formalize satellietes
  #
> # Global variables

  MaxSatellites := 3;

  MaxGenerations:= 5;


  MaxSignals := 2^MaxGenerations-1;
```

$$MaxSatellites := 3$$

$$MaxGenerations := 5$$

$$MaxSignals := 31 \quad \textbf{(13)}$$

```
> MaxT:= 1/2; # Indietro
  ErrorFlag := false:
```

$$MaxT := \frac{1}{2} \quad \textbf{(14)}$$

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

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

$$Ingoing := -1$$

$$Outgoing := 1$$

$$Clockwise := -1$$

$$Counterclockwise := 1 \quad \textbf{(15)}$$

```
> #isScattering := true;
```

```
> Infalling:= 1;
  SameScattering:= 2;
  OtherScattering := 3;
```

$$Infalling := 1$$

$$SameScattering := 2$$

$$OtherScattering := 3 \quad \textbf{(16)}$$

```
> Side:= 1;
  Front:= 2;
  Behind:= 3;
```

$$Side := 1$$

$$Front := 2$$

$$Behind := 3 \quad \textbf{(17)}$$

```
# Verbose Flags
```

```
> ProducePlots:= true;
  DebugOn := true;
  DebugTimeOn := true;
  DebugFlowControlOn := true;
  DebugSolutionsOn := true;
```

$$ProducePlots := true$$

$$DebugOn := true$$

$$DebugTimeOn := true$$

$$DebugFlowControlOn := true$$

$$DebugSolutionsOn := true \quad (18)$$

```
> BestKnownSolution:= none:
```

```
> ClearTypeToDo:= proc()
   global TypeToDo:
   TypeToDo := [
        true, true, true, true,
        true, true, true, true,
        true, true, true, true
         ]:
  end:

  ClearTypeToDo():
```

```
> TypeToDo ;
```

$$[true, true, true, true, true, true, true, true, true, true, true, true] \quad (19)$$

SearchSignal procedures

Searching for a signal is a complicated issue with many attempts and tries.
Never look for two signals at the same time.
Let us describe a search by a structure on which we can do many operations
and then eventually save the result out of the structure and then

```
> TargetSat      := 1:        #  -1, 0, 1, 2
  TargetP        := 2:        #  [r, te, t]
  TargetBranch   := 3:        #  none is no sat
  TargetCol      := 4:        #  black, red, blue, green

  SourceSat      := 5:        #  0, 1, 2
  SourceBranch   := 6:        #  ..., -1, 0, 1, ...
  SourceCol      := 7:        #  black, red, blue, green
  PreviousSourceBranch:= 8:   # true | false(D)

  Gen            := 9:        # 0(D), 1, 2, ...

  IndrGuessMax   :=10:
  IndrGuessMin   :=11:
```

```
SourceCrossing:=12:

IndrmGuess    :=13:
IndK2Guess    :=14:
Indsv         :=15:
Indkv         :=16:
Indscos       :=17:

Intervalr1    :=18:
Intervalr2    :=19:
Intervalrm    :=20:
IntervalK2    :=21:

RisP          :=22:
Risr          :=23:
RisBranch     :=24:
Risrm         :=25:
RisK2         :=26:

RayType        :=27:       # Infalling | SameScattering |
OtherScattering
RayBranch      :=28:       # Ingoing | Outgoing  (at Target)
RayClock       :=29:       # Clockwise | Counterclockwise
SolveType      :=30:       # Side | Front | Behind

> SearchSignalClear := [

  none,          #x TargetSat      := 1:       #  -1, 0, 1, 2
  none,          #x TargetP        := 2:       #  [r, te, t]
  none,          #x TargetBranch   := 3:       #  none is no sat
  none,          #x TargetCol      := 4:       #  black, red, blue,
green

  none,          #x SourceSat      := 5:       #  0, 1, 2
  none,          #x SourceBranch   := 6:       #  ..., -1, 0, 1, ..
.
  none,          #x SourceCol      := 7:       #  black, red, blue,
green
  false,         # PreviousSourceBranch:= 8: # true | false(D)

  0,             #x Gen            := 9:       # 0(D), 1, 2, ...

  none,          # IndrGuessMax  :=10:
  none,          # IndrGuessMin  :=11:
  none,          # SourceCrossing:=12:

  none,          # IndrmGuess    :=13:
  none,          # IndK2Guess    :=14:
  none,          # Indsv         :=15:
  none,          # Indkv         :=16:
  none,          # Indscos       :=17:

  none,          # Intervalr1    :=18:
  none,          # Intervalr2    :=19:
  none,          # Intervalrm    :=20:
  none,          # IntervalK2    :=21:
```

```
    none,          # RisP          :=22:
    none,          # Risr          :=23:
    none,          # RisBranch     :=24:
    none,          # Risrm         :=25:
    none,          # RisK2         :=26:

    none,          # RayType       :=27:       # Infalling |
  SameScattering | OtherScattering
    none,          # RayBranch     :=28:       # Ingoing | Outgoing
  (at Target)
    none,          # RayClock      :=29:       # Clockwise |
  Counterclockwise
    none           # SolveType     :=30:       # Side | Front |
  Behind
  ]:

  SearchSignal := SearchSignalClear:



> ClearSearchSignal := proc()
    global SearchSignal;
    SearchSignal := SearchSignalClear:
  end:

> SaveSearchSignal := proc()
    return SearchSignal:
  end:

> RestoreSearchSignal := proc(saved)
    global SearchSignal;
    SearchSignal := saved:
  end:

> Get := proc(field)
  global SearchSignalFields:
    SearchSignal[field];
  end:

> Set := proc(field, val)
    global SearchSignal;
    #print("Set %d = %a", field, val);
    SearchSignal[field] := val;
  end:

> NewSearch:= proc()
    Set(RisP, none):
    Set(Risr, none):
    Set(RisBranch, none):
    Set(Risrm, none):
    Set(RisK2, none):


    #Set(RayType, none):       #check they are not set in
  LinearGuess
    #Set(RayBranch, none):
```

```
    #Set(RayClock, none):
    #Set(SolveType, none):
  end:

> SetFreeTarget := proc(P)
    Set(TargetSat, -1):
    Set(TargetP, P):
    Set(TargetCol, black):
  end:

> SetTargetPoint := proc(Sat, Pv)
    Set(TargetSat, Sat):
    Set(TargetP, Pv):
    Set(TargetBranch, BranchWith(Sat, Pv[3])):
    if Sat = 1 then
      Set(TargetCol, red):
    elif Sat = 2 then
      Set(TargetCol, blue):
    elif Sat = 3 then
      Set(TargetCol, green):
    else
      Set(TargetCol, black):
    end if:
  end:

> SetSourceSat := proc(Sat)
    Set(SourceSat, Sat):
    Set(SourceBranch, BranchWith(Sat, Get(TargetP)[3])):
    if Sat = 1 then
      Set(SourceCol, red):
    elif Sat = 2 then
      Set(SourceCol, blue):
    elif Sat = 3 then
      Set(SourceCol, green):
    else
      Set(SourceCol, black):
    end if:
  end:

> SetGeneration := proc(G)
      Set(Gen, G):
  end:

>  GetGeneration := proc()
      Get(Gen):
  end:

# if Pr<rSat.min          |     P   <--   S
#
#           Scattering                        Infalling
# sv<0       Nope                               Nope
# 0<sv<1      Out Other                          Nope
# sv>1       In Same                            In (single branch)
#
```

```
#
# if Pr>rSat.max          |     S   -->   P
#
#           Scattering                          Infalling
# sv<0       Out Same                              Out (single branch)
# 0<sv<1      Out Other                             Nope
# sv>1        Nope                                 Nope
#
```

```
> LinearGuess:= proc()
   local targetR, targetTe, targetT:
   local sourceSat, sourceBranch:
   local rGuessMax, rGuessMin, d2, sv, rmGuess, kv, scos:
   local mmin, mmax, tm, tM, rguess:

   targetR, targetTe, targetT := op(Get(TargetP)):
   sourceSat := Get(SourceSat):
   sourceBranch := Get(SourceBranch):
   mmin := minS(sourceSat):
   mmax := maxS(sourceSat):

   targetT := Real(targetT):
   Re(tS(sourceSat, sourceBranch)-targetT):
   evalf(%):
   fsolve(%, r=mmin..mmax, fulldigits):
   rGuessMax:=Q(%):
   Set(IndrGuessMax, rGuessMax):

   BranchWith(sourceSat, targetT- MaxT):
   Re(tS(sourceSat, %)-(targetT - MaxT)):
   evalf(%):
   fsolve(%, r=mmin..mmax, fulldigits):
   rGuessMin:= %:
   Set(IndrGuessMin, rGuessMin):
   # notice: it is guaranteed that rGuessMin < rGuessMax (it
  depends on the satallite branch)
   # what is guaranteed is that t at rGuessMin is less than t at
  rGuessMax
   # Therefore be careful when writing Ranges.

   tS(sourceSat, sourceBranch):
   subs(r=mmin, %):
   Real(%):
   simplify(%):
   evalf(%):
   tm:=%:

  tS(sourceSat, sourceBranch):
  subs(r=mmax, %):

  #printf("[attempt Real to %a]  ", %):

   tS(sourceSat, sourceBranch):
   subs(r=mmax, %):
```

```
 Real(%):
 simplify(%):
 evalf(%):
 tM:=%:
#printf(".1");
#printf("targetT-MaxT= %a ", targetT-MaxT);
#printf("tm= %a ", tm);
#printf("tM= %a ", tM);
#printf("targetT= %a \n", targetT);
#printf("mmin= %a \n", mmin);
#printf("mmax= %a \n", mmax);
#printf("rGuessMax= %a \n", rGuessMax);
#printf("rGuessMin= %a \n", rGuessMin);
 #    tm< targetT - MaxT (< targetT < tM ) ?
 if  targetT-MaxT < tm  and tm < targetT then
     Set(PreviousSourceBranch,false):
     Set(SourceCrossing, true):
     Set(Intervalr1, myRange(mmin, rGuessMax)):
     Set(Intervalr2, myRange(mmin, rGuessMin)):
     if DebugOn then
       printf("two intervals r = %a or r = %a\n", myRange(mmin,
rGuessMax), myRange(mmin, rGuessMin)):
     end if:
 elif  targetT-MaxT < tM  and tM < targetT then
     Set(PreviousSourceBranch,false):
     Set(SourceCrossing, true):
     Set(Intervalr1, myRange(rGuessMax, mmax)):
     Set(Intervalr2, myRange(rGuessMin, mmax)):
     if DebugOn then
       printf("two intervals r = %a or r = %a\n", myRange
(rGuessMax, mmax), myRange(rGuessMin, mmax)):
     end if:
 else
   Set(PreviousSourceBranch,false):
   Set(SourceCrossing, false):
   Set(Intervalr1, myRange(rGuessMax, rGuessMin)):
   Set(Intervalr2, none):
   if DebugOn then
     printf("one interval r = %a\n", myRange(rGuessMax,
rGuessMin)):
   end if:
 end if:
#printf(".2");

 (s*targetR*cos(targetTe)+(1-s)*xS(sourceSat, sourceBranch))^2
  +(s*targetR*sin(targetTe) + (1-s)*yS(sourceSat, sourceBranch))
^2:
 subs(r=rGuessMax, %):
 d2:= evalf(%):
 diff(%, s):
 solve(%, s):
 sv:=%;
 Set(Indsv, sv):
#printf(".3");

 subs(s=sv, d2):
 sqrt(%):
```

```
 evalf(%):
 rmGuess:=%;                                          # many times
the linear approximation is a good starting point to locate rm.
 Set(IndrmGuess, rmGuess):
#printf(".4");


# printf("[xS, yS]=%a\n", [xS(sourceSat, sourceBranch), yS
(sourceSat, sourceBranch)]);
# printf("Target=%a\n", [targetR*cos(targetTe), targetR*sin
(targetTe)]);
# printf("Target=%a\n", [targetR*cos(targetTe), targetR*sin
(targetTe)]);
 xS(sourceSat, sourceBranch)*targetR*sin(targetTe)-yS(sourceSat,
sourceBranch)*targetR*cos(targetTe):
 subs(r=rGuessMax, %):
 evalf(%):
 kv:=%:
 Set(Indkv, kv):
#printf(".5");

 targetR*cos(targetTe)*xS(sourceSat, sourceBranch)+targetR*sin
(targetTe)*yS(sourceSat, sourceBranch):
 subs(r=rGuessMax, %):
 scos:= %:                                            # >0 target
and source are on the same side of BH
 Set(Indscos, scos):
#printf(".6");

 if kv > 0 then
   Set(RayClock, Counterclockwise):
 else
   Set(RayClock, Clockwise):
 end if:
 Set(Intervalrm, rcrit..min(targetR, mmax)):
 Set(IntervalK2, 0..K2crit):
#printf(".7");

# forse è più importante stare larghi così da sapere che in sides
non ci possono essere infalling
# e lasciare la possibilità tutta in front e behind

 if rmGuess > 4*alv then
    Set(SolveType, Side):
    if targetR > mmax then                        #           |
S  ---> P
      if sv<0 then
         Set(RayBranch, Outgoing):
         Set(RayType, SameScattering):
      elif sv>0 and sv<1 then
         Set(RayBranch, Outgoing):
         Set(RayType, OtherScattering):
      else       # sv>1
         printf("It cannot be s>1 and |      S  ---> P\n");
      end if:
    elif targetR < mmin then                      #           |
P  <--- S
```

```
      if sv<0 then
         printf("It cannot be s<0 and |      P  <--- S\n");
      elif sv>0 and sv<1 then
         Set(RayBranch, Outgoing):
         Set(RayType, OtherScattering):
      else       # sv>1
         Set(RayBranch, Ingoing):
         Set(RayType, SameScattering):
      end if:
    else
      printf("The user is in the satellite ring. Too messy to be
considered. At least now.\n");
    end if:
 else
    if scos > 0 then                        # same side of BH
     Set(SolveType, Front):
     if rmGuess > rcrit then
       if targetR > mmax then                          #  |      S  --
-> P
         Set(RayBranch, Outgoing):
         Set(RayType, SameScattering ):
       elif targetR < mmin then                        #  |      P  <-
-- S
         Set(RayBranch, Ingoing):
         Set(RayType, SameScattering ):
       end if:
     else
       if targetR > mmax then                          #  |      S  --
-> P
         Set(RayBranch, Outgoing):
         Set(RayType, Infalling):
       elif targetR < mmin then                        #  |      P  <-
-- S
         Set(RayBranch, Ingoing):
         Set(RayType, Infalling):
       end if:
     end if:
    else                                    # opposite sides of BH
      Set(SolveType, Behind):
      if targetR > mmax then                         #  |      S  --
-> P
         Set(RayBranch, Outgoing):
         Set(RayType, OtherScattering ):
      elif targetR < mmin then                       #  |      P  <--
- S
         Set(RayBranch, Outgoing):
         Set(RayType, OtherScattering ):
      end if:
    end if:
 end if:
#printf(".end");
end:

> SetPreviousSourceBranch := proc()
    local Crossing, Int2, sourceBranch, Sat:
    Crossing := Get(SourceCrossing):
```

```
    Set(SourceCrossing, false):
    Sat := Get(SourceSat):
    if Crossing then
      Int2 := Get(Intervalr2):
      Set(Intervalr1, Int2):
      Set(Intervalr2, none):
      sourceBranch := Get(SourceBranch):
      Set(SourceBranch, sourceBranch-1):
      if (sourceBranch mod 2) = 0 then
         # even then Outgoing, I crossed rm
         Set(rGuessMax, minS(Sat))
      else
         # odd then Ingoing, I crossed rp
         Set(rGuessMax, maxS(Sat))
      end if:
    end if:
  end:

> OtherClock := proc()
   local rayClock:
   rayClock := Get(RayClock):
   Set(RayClock, -rayClock):
  end:

>

> CreateHint:= proc()
   local h1, h2, h3, h4, h5, h6, h7, h8:
   h1:= Get(Risr):
   h6:= Get(Intervalr1);
   if DebugOn then
     printf("r=%g in [%a]\n",h1, h6):
   end if:
   h2:= Get(RayType):
   if h2 = Infalling then
     h7:= Get(IntervalK2);
     h5:= Get(RisK2);
     if DebugOn then
       printf("Infalling ray (K2=%g) in [%a].\n", h5, h7);
     end if:
   elif h2 = SameScattering then
     h7:= Get(Intervalrm);
     h5:= Get(Risrm);
     if DebugOn then
       printf("Scattering ray (rm=%g) in [%a]: target and source on
  the same branch.\n", h5, h7);
     end if:
   elif h2 = OtherScattering then
     h7:= Get(Intervalrm);
     h5:= Get(Risrm);
     if DebugOn then
       printf("Scattering ray (rm=%g) in [%a]: target and source on
  the different branches.\n", h5, h7);
     end if:
   else
     printf("Type of ray unknown.\n");
```

```
    end if:
    h3:= Get(RayClock):
    if DebugOn then
      if h3 = Clockwise then
        printf("Clockwise ray.\n");
      else
        printf("Counterclockwise ray.\n");
      end if:
    end if:
    h4:= Get(RayBranch):
    if DebugOn then
      if h4 = Outgoing then
        printf("Ray outgoing at target.\n");
      else
        printf("Ray outgoing at target.\n");
      end if:
    end if:
    h8 := Get(SolveType):
    if DebugOn then
      if h8 = Side then
        printf("Solve Side.\n");
      elif h8 = Front then
        printf("Solve Front.\n");
      elif h8 = Behind then
        printf("Solve Behind.\n");
      else
        printf("Solve type unknown.\n");
      end if:
    end if:
    [h1, h2, h3, h4, h5, h6, h7, h8]:
  end:

> UseHint:= proc(h)
    if DebugOn then
      printf("hint used Hint := %a\n", h);
    end if:
    if type(h, list) and nops(h)=8 then
     Set(Risr, h[1]):
     Set(IndrGuessMax, h[1]):
     Set(RayType, h[2]):
     if h[2] = Infalling then
       Set(RisK2, h[5]):
       Set(IndK2Guess, h[5]):
       Set(IntervalK2, h[7]):
     elif h[2] = SameScattering then
       Set(Risrm, h[5]):
       Set(IndrmGuess, h[5]):
       Set(Intervalrm, h[7]):
     elif h[2] = OtherScattering then
       Set(Risrm, h[5]):
       Set(IndrmGuess, h[5]):
       Set(Intervalrm, h[7]):
     else
       printf("Type of ray unknown.\n");
     end if:
     Set(RayClock, h[3]):
```

```
      Set(RayBranch, h[4]):
      Set(Intervalr1, h[6]):
      Set(SolveType, h[8]):
     else
      printf("No hint used.\n");
      return:
     end if:
   end:
```

```
> ComputeSat:= proc(tein, tin, tauin, rmv, rpv)
   local tr, ter, taur, Te, T, Tau;

   mPhi:
   1/%:
   simplify(%)assuming c=cv, al=alv, rm=rmv, rp=rpv, r=(rmv+rpv)/2:
   sqrt(%):
   radsimp(%) assuming c=cv, al=alv, rm=rmv, rp=rpv, r=(rmv+rpv)/2:
   #%;
   subs(r=R, %):
   subs([c=cv, al=alv, rm=rmv, rp=rpv], %):
   int(%, R=rmv..r) assuming  r>rmv, r<rpv:
   simplify(%) assuming r>rmv, r<rpv:
   radsimp(%) assuming r>rmv, r<rpv:
   tr:= %:

   mPsi:
   1/%:
   simplify(%) assuming c=cv, al=alv, rm=rmv, rp=rpv, r=(rmv+rpv)/2:
   sqrt(%):
   radsimp(%) assuming c=cv, al=alv, rm=rmv, rp=rpv, r=(rmv+rpv)/2:
   #%;
   subs(r=R, %):
   subs([c=cv, al=alv, rm=rmv, rp=rpv], %):
   int(%, R=rmv..r) assuming  r>rmv, r<rpv :
   simplify(%) assuming r>rmv, r<rpv:
   radsimp(%) assuming r>rmv, r<rpv:
   ter := %:

   dtau :
   subs(r=R, %):
   subs([c=cv, al=alv, rm=rmv, rp=rpv], %):
   int(%, R=rmv..r) assuming  r>rmv, r<rpv :
   simplify(%) assuming r>rmv, r<rpv:
   radsimp(%) assuming r>rmv, r<rpv:
   taur := %:

   ter:
   subs(r=rpv, %):
   simplify(%):
   Te:=%:
```

```
    tr:
    subs(r=rpv, %):
    simplify(%):
    T:=%:

    taur:
    subs(r=rpv, %):
    simplify(%):
    Tau:=%:
    [tr, ter, taur, Te, T, Tau]:
   end:
```

```
> ComputeSat0:= proc()
    global te0in, t0in, tau0in, rm0v, rp0v;
    global t0r, te0r, tau0r, Te0, T0, Tau0;
    ComputeSat(te0in, t0in, tau0in, rm0v, rp0v):
    t0r, te0r, tau0r, Te0, T0, Tau0 := op(%):
   end:
```

```
> ComputeSat1:= proc()
    global te1in, t1in, tau1in, rm1v, rp1v;
    global t1r, te1r, tau1r, Te1, T1, Tau1;
    ComputeSat(te1in, t1in, tau1in, rm1v, rp1v):
    t1r, te1r, tau1r, Te1, T1, Tau1 := op(%):
   end:
```

```
> ComputeSat2:= proc()
    global te2in, t2in, tau2in, rm2v, rp2v;
    global t2r, te2r, tau2r, Te2, T2, Tau2;
    ComputeSat(te2in, t2in, tau2in, rm2v, rp2v):
    t2r, te2r, tau2r, Te2, T2, Tau2 := op(%):
   end:
```

```
> BaseSatOriginal:= 0:
  BaseROriginal:= 23889000:
  BaseBranchOriginal:=0:
```

```
> # First satellite
```

```
> rp0v := Surface+20900000;
  rm0v := Surface+20100000;

  t0in := -1500:
  te0in := Pi/6:
  tau0in := -1500:
```

$$rp0v := 24089000$$

$$rm0v := 23289000 \qquad \textbf{(20)}$$

```
> Sat0 := [te0in, t0in, tau0in, rm0v, rp0v];
```

$$Sat0 := \left[\frac{\pi}{6}, -1500, -1500, 23289000, 24089000\right] \quad \textbf{(21)}$$

```
> # for each satellite we can analytically integrate Weierstrass
  equations
  # This is painfully detailed and sensitive.
  # Small changes end up with Maple find worse representations of the
  integrals.
```

```
> te0in:
  evalf(%);

  t0in:
  evalf(%);

  tau0in:
  evalf(%);


  rp0v:
  evalf(%);

  rm0v:
  evalf(%);
```

0.523598775598298873077107230546583814032861566562517636829157432051302734381035\
484

$$\begin{bmatrix} -1500. \\ -1500. \\ 2.4089000 \times 10^{7} \\ 2.3289000 \times 10^{7} \end{bmatrix} \quad \textbf{(22)}$$

```
> ComputeSat0():

  t0r:
  evalf(%):
  %:

  te0r:
  evalf(%):
  %:

  tau0r:
  evalf(%):
  %:

  Te0:
  evalf(%):
  %;

  # 3.14159265535486
```

```
# 3.1415926553548523984
# 3.14159265535485239825426539013409419856260322061411445326054936278
#
3.14159265535485239825426539013409419856260322061411445326054936278
37502485449557
#
3.14159265535485239825426539013409419856260322061411445326054936278
375024854495562322064808510137802951136526164386486406

T0:
evalf(%):
%;


# 18142.3587447772
# 18142.358744777252186
# 18142.3587447772521860591366983412221786668982957166911430921766664
#
18142.3587447772521860591366983412221786668982957166911430921766664
28555043660612
#
18142.3587447772521860591366983412221786668982957166911430921766664
285550436606122065933274111800343050009373669231962657

Tau0:
evalf(%):
%;


# 18142.3587396824
# 18142.358739682192163
# 18142.3587396821921630244585203023007633
#
18142.3587396821921630244585203023000763226421341640380742732892052
57465944332626
#
18142.3587396821921630244585203023000763226421341640380742732892052
57465944332625415644937375867083132267520491886176819 5
```

3.14159265535485239825426539013409419856260322061411445326054936278375024854495\
57
18142.3587447772521860591366983412221786668982957166911430921766664285550436606\
12
18142.3587396821921630244585203023000763226421341640380742732892052574659443326\ **(23)**
26

```
>
# (half-)periods
```

```
evalf(2*T0);
%/60/60;
#  36284.71748955450437211827339668244435732
#
36284.71748955450437211827339668244435733379659143338228618435333285711008732
57110087321224
```

$$36284.71748955450437211827339668244435733379659143338228618435333285711008732124$$

$$10.07908819154291788114396483241179009925938794206482841282898703690475280203367373$$ **(24)**

```
> evalf(2*(Te0-Pi)); # Precession (in radiant)

  #   3.530118319583244013709182628732*10^(-9)
  #
  3.530118319583244013709182628730867642478017264571209540951867684517493
  74933*10^(-9)
```

$$3.530118319583244013709182628730867642478017264571209540951867684517493 \times 10^{-9}$$ **(25)**

```
> evalf(2*Tau0);

  #    36284.717479364384326048917040604600152645284268328076148546578410514
  #
  36284.717479364384326048917040604600152645284268328076148546578410514931888665252

  evalf(Tau0/T0); # Slowing time factor

  #    0.99999999971916220516246691698241163504326993418205012998397463349
  #
  0.99999999971916220516246691698241163504326993418205012998397463349053450494301666
```

$$36284.71747936438432604891704060460015264528426832807614854657841051493188866522$$

$$0.99999999971916220516246691698241163504326993418205012998397463349053450494301666$$ **(26)**

```
>
  # Second satellite

> rp1v := Surface+21900000;
  rm1v := Surface+21100000;

  t1in := -1600:
  te1in := -Pi/6:
  tau1in := -1600:
```

$$rp1v := 25089000$$ **(27)**

$$rm1v := 24289000 \quad (27)$$

```
> Sat1 := [te1in, t1in, tau1in, rm1v, rp1v];
```

$$Sat1 := \left[-\frac{\pi}{6}, -1600, -1600, 24289000, 25089000\right] \quad (28)$$

```
> ComputeSat1():

  t1r:
  evalf(%):
  %:

  te1r:
  evalf(%):
  %:

  tau1r:
  evalf(%):
  %:

  Te1:
  evalf(%):
  %;


  T1:
  evalf(%):
  %;

  Tau1:
  evalf(%):
  %;
```

3.14159265283322340782345120016275877553771337644222556155683918825805413694777\
85
19303.18211302455233332661930535341676357009344191493432381882008428056652298065\
35
19303.18210782306326017425688729853980400347257670621621882403771537355770671025\ **(29)**
44

```
> evalf(2*T1);
  %/60/60;
  #    38606.364226049104666653238610706833527140186883829868647637640168561133045961271
```

38606.364226049104666653238610706833527140186883829868647637640168561133045961271\
71

(30)

10.72399006279141796295923294741856486865005191217496351323267782460031473498921\
42 (30)

```
> evalf(2*(Te1-Pi)); # Precession (in radiant)

  #    3.3870582046394034734735459867132038765382334703614786530359780148171390*10^(-9)
```

$$3.387058204639403473473545986713203876538233470361478653035978014817139 0 \times 10^{-9} \quad (31)$$

```
> evalf(2*Tau1);

  #    38606.364215646126520348513774597079608006945153412432437648075430747115413420488\

  evalf(Tau1/T1); # Slowing time factor

  #    0.999999999730537222169046917590231686974595563925374580848649270148857211389738\
```

(32)

```
> # Terzo satellite

> rp2v := Surface+22900000;
  rm2v := Surface+22100000;

  t2in := -1400:
  te2in := 0:
  tau2in := -1400:
```

$$rp2v := 26089000$$
$$rm2v := 25289000 \quad (33)$$

```
> Sat2 := [te2in, t2in, tau2in, rm2v, rp2v];
```

$$Sat2 := [0, -1400, -1400, 25289000, 26089000] \quad (34)$$

```
> OrbitalParameters := [Sat0, Sat1, Sat2];
```

$$OrbitalParameters := \left[\left[\frac{\pi}{6}, -1500, -1500, 23289000, 24089000\right], \left[-\frac{\pi}{6}, -1600, -1600, 24289000, 25089000\right], [0, -1400, -1400, 25289000, 26089000]\right] \quad (35)$$

```
> ComputeSat2():

  t2r:
  evalf(%):
  %:

  te2r:
  evalf(%):
  %:
```

```
tau2r:
evalf(%):
%:

Te2:
evalf(%):
%;


T2:
evalf(%):
%;

Tau2:
evalf(%):
%;
```

3.14159265521736542784820294411682633696760460533057738293547333639087419014453\
16
20487.75898732513113742426435899419444117577212742708016664702436099071277410 11\
70
20487.75898201934745186745874168545755717309714658676728269363563055743275420 20\ **(36)**
48

```
> evalf(2*T2);
%/60/60;
#    40975.5179746502622748485287179883888234
```

40975.51797465026227484852871798838888235154425485416033329404872198142554820 23\
40
11.38208832629173952079125797721899691176431784857060009258168020055039598561 17\ **(37)**
61

```
> evalf(2*(Te2-Pi)); # Precession (in radiant)

#    3.25514437877111912167464690554087041191094312392105748816611556771664 49*10^(-9)
```

$3.25514437877111912167464690554087041191094312392105748816611556771664 49 \times 10^{-9}$ **(38)**

```
> evalf(2*Tau2);

#    40975.51796403869490373491748337091511433

evalf(Tau2/T2); # Slowing time factor

#    0.99999999974102664479608243471530305079 72
```

40975.51796403869490373491748337091511434619429317353456538727126111486550840 40\
97

(39)

0.99999999974102664479608243471530305079700106987749669991784714719904054784882\ **(39)**
590

Satellites

Satellite 0

```
> # Satellite 0
> xS0 := proc(branchv)
   r*cos(Branch(te0r,branchv, Te0, te0in)):
  end:

> yS0 := proc(branchv)
   r*sin(Branch(te0r,branchv, Te0, te0in)):
  end:

> tS0 := proc(branchv)
   Branch(t0r, branchv, T0, t0in):
  end:

> minS0:= proc()
    rm0v;
  end:

> maxS0:= proc()
    rp0v;
  end:

> halfT0:= proc()
    T0;
  end:

> halfTe0:= proc()
    Te0;
  end:

> halfTau0:= proc()
    Tau0;
  end:
```

Satellite 1

```
> xS1 := proc(branchv)
   r*cos(Branch(te1r,branchv, Te1, te1in)):
```

```
end:
```

```
> yS1 := proc(branchv)
   r*sin(Branch(te1r,branchv, Te1, te1in)):
  end:
```

```
> tS1 := proc(branchv)
   Branch(t1r, branchv, T1, t1in):
  end:
```

```
> minS1:= proc()
    rm1v;
  end:
```

```
> maxS1:= proc()
    rp1v;
  end:
```

```
> halfT1:= proc()
    T1;
  end:
```

```
> halfTe1:= proc()
    Te1;
  end:
```

```
> halfTau1:= proc()
    Tau1;
  end:
```

Satellite 2

```
> xS2 := proc(branchv)
   r*cos(Branch(te2r,branchv, Te2, te2in)):
  end:
```

```
> yS2 := proc(branchv)
   r*sin(Branch(te2r,branchv, Te2, te2in)):
  end:
```

```
> tS2 := proc(branchv)
   Branch(t2r, branchv, T2, t2in):
  end:
```

```
> minS2:= proc()
    rm2v;
  end:
```

```
> maxS2:= proc()
    rp2v;
  end:
```

```
> halfT2:= proc()
    T2;
  end:

> halfTe2:= proc()
    Te2;
  end:

> halfTau2:= proc()
    Tau2;
  end:
```

```
> rSat:= proc(S, rv, branchv)
   rv:
  end:

> teSat:= proc(S, rv, branchv)
   if S=0 then
     Branch(te0r,branchv, Te0, te0in):
   elif S=1 then
     Branch(te1r,branchv, Te1, te1in):
   elif S=2 then
     Branch(te2r,branchv, Te2, te2in):
   end if:
   subs(r=rv, %):
   evalf(%):
  end:

> tSat:= proc(S, rv, branchv)
   if S=0 then
     Branch(t0r, branchv, T0, t0in):
   elif S=1 then
     Branch(t1r, branchv, T1, t1in):
   elif S=2 then
     Branch(t2r, branchv, T2, t2in):
   end if:
   subs(r=rv, %):
   evalf(%):
  end:

> tauSat:= proc(S, rv, branchv)
   if S=0 then
     Branch(tau0r,  branchv, Tau0, tau0in):
   elif S=1 then
     Branch(tau1r,  branchv, Tau1, tau1in):
   elif S=2 then
     Branch(tau2r,  branchv, Tau2, tau2in):
   end if:
   subs(r=rv, %):
   evalf(%):
  end:
```

```
> xS := proc(S, b)
   if S = 0 then
    xS0(b):
   elif S = 1 then
    xS1(b):
   elif S = 2 then
    xS2(b):
   end if:
  end:

> yS := proc(S, b)
   if S = 0 then
    yS0(b):
   elif S = 1 then
    yS1(b):
   elif S = 2 then
    yS2(b):
   end if:
  end:

> tS := proc(S, b)
   if S = 0 then
    tS0(b):
   elif S = 1 then
    tS1(b):
   elif S = 2 then
    tS2(b):
   end if:
  end:

> maxS := proc(S)
   if S = 0 then
    maxS0():
   elif S = 1 then
    maxS1():
   elif S = 2 then
    maxS2():
   end if:
  end:

> minS := proc(S)
   if S = 0 then
    minS0():
   elif S = 1 then
    minS1():
   elif S = 2 then
    minS2():
   end if:
  end:

> halfT:= proc(S)
   if S = 0 then
    halfT0():
   elif S = 1 then
    halfT1():
   elif S = 2 then
```

```
    halfT2():
   end if:
  end:
```

```
> halfTe:= proc(S)
   if S = 0 then
    halfTe0():
   elif S = 1 then
    halfTe1():
   elif S = 2 then
    halfTe2():
   end if:
  end:
```

```
> halfTau:= proc(S)
   if S = 0 then
    halfTau0():
   elif S = 1 then
    halfTau1():
   elif S = 2 then
    halfTau2():
   end if:
  end:
```

```
> tSat(BaseSatOriginal, BaseROriginal, BaseBranchOriginal):
  simplify(%):
  #evalf(%):
  TOffset:=%;
```

$$TOffset := \text{(40)}$$

10510.458140193974128106990086403103436416267811869852971885475213316854268817156

```
> teSat(BaseSatOriginal, BaseROriginal, BaseBranchOriginal):
  simplify(%):
  #evalf(%):
  TeOffset:=%;
```

$$TeOffset := \text{(41)}$$

2.6325564346011069999679513735084001479836933412413039069725878804575240036514658

```
> tauSat(BaseSatOriginal, BaseROriginal, BaseBranchOriginal):
  simplify(%):
  #evalf(%):
  TauOffset:=%;
```

$$TauOffset := \text{(42)}$$

10510.458136789362078420147641823245967889085957114064057368661860285398060678877

```
#   Light rays
```

```
> wPhi:
  subs(k2=K2*ep2,%):
  limit(%, ep2=infinity):
  factor(%):
  subs([al=alv, c=cv], %):
  lwPhi := %;
```

$$lwPhi := \frac{1}{r^5}\left(\left(-r + \frac{199306945455000}{22468879468420441}\right)^2 \left(89875517873681764\, r^3 - K2\, r + \frac{199306945455000}{22468879468420441}\, K2\right)\right) \tag{43}$$

```
> wPsi:
  subs(k2=K2*ep2,%):
  limit(%, ep2=infinity):
  factor(%):
  subs([al=alv, c=cv], %):
  lwPsi := %;
```

$$lwPsi := \frac{\left(89875517873681764\, r^3 - K2\, r + \frac{199306945455000}{22468879468420441}\, K2\right) r}{K2} \tag{44}$$

```
> # Again we need to put Weierstrass in a form suitable for later
  integration
  # even though in this case we have both scattering and infalling
  rays.

  wPsi:
  subs(k2=K2*ep2,%):
  limit(%, ep2=infinity):
  factor(%):
  numer(%)/r:
  lp:=%;

  subs(r=al, %):
  #subs([al=alv, c=cv], %):
  evalf(%);
```

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

$$al^3\, c^2 \tag{45}$$

```
> lp:
  diff(%, r):
  solve(%, r):
  [%][1]:
  subs([al=alv, c=cv], %):
  rcrit:=%;
```

$$rcrit := \frac{\sqrt{3}\,\sqrt{K2}}{899377374} \tag{46}$$

```
> lp:
  subs(r=rcrit, %):
  subs(K2=K^2, %):
  radsimp(%):
  %/K^2:
  simplify(%):
  solve(%, K):
  %^2:
  subs([al=alv, c=cv], %):
  K2crit:=%;
  evalf(%);
```

$$K2crit := \frac{10725279796782633192396750000000}{22468879468420441}$$

$$4.773393266832375251471851872452971366257453291608860842931941751352531287614961 \times 10^{13} \tag{47}$$

```
> rcrit:
  subs(K2=K2crit, %):
  simplify(%):
  rcrit:= %;
  %-alv*3/2;
```

$$rcrit := \frac{298960418182500}{22468879468420441}$$

$$0 \tag{48}$$

```
> subs(K2=K2crit, lp):
  subs([al=alv, c=cv], %):
  subs(r=rcrit, %):
  #evalf(%):
  %;
```

$$0 \tag{49}$$

```
> #
  # One real solution
  # 0 < K2 < K2crit

  #
  # To be honest we should work better for the integration of
  infalling branches
  # (to be improved, being currently numerical)
  #

> lwPhi:
  #subs([al=alv, c=cv], %):
  sqrt(%):
  radsimp(%) assuming r>0:
  simplify(%) assuming r>0:
  1/%:
```

```
subs(r=R, %):
Int(%, R=r0..r) assuming K2>0:
inltr:=t0+ branch*%;

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

$$inltr := \left(\int_{r0}^{r} \left(33680003021717486924169 89\, R^{5/2} \right) \Big/ \left((22468879468420441\, R - 199306945455000) \left((-22468879468420441\, R + 199306945455000)\, K2 + 20194021782656225573159501865379 24\, R^{3} \right)^{1/2} \right) dR \right) branch + t0 \tag{50}$$

```
> lwPsi:
  sqrt(%):
  radsimp(%) assuming r>0, K2>0:
  simplify(%) assuming r>0:
  1/%:
  subs(r=R, %):
  Int(%, R=r0..r):
  #%;
  inlter:=te0+ clock*branch*%;
```

$$inlter := \left(\int_{r0}^{r} (149896229\, K2) \Big/ \left(\sqrt{R} \left(-22468879468420441\, K2 \left(\left(R - \frac{199306945455000}{22468879468420441} \right) K2 - 89875517873681764\, R^{3} \right) \right)^{1/2} \right) dR \right) branch\ clock + te0 \tag{51}$$

```
> # Test

  inltr:
  subs([t0=13, r0=5*10^6, te0=Pi/4], %):
  subs([branch=Ingoing], %):
  subs([K2=1/3*K2crit], %):
  subs([r=4*10^6], %):
  evalf(%):
  %;
```

```
# 13.0033356409586
# 13.003335640958583963
# 13.00333564095858396346014332501596545067
#
13.00333564095858396346014332501596545067457165152471577098839118522
71161021669 49
#
13.00333564095858396346014332501596545067457165152471577098839118522
7116102166948870444872530460199816962043315625036602813.00333564095
85839634601433250159654506745716515247157709883911852271161021669\
```

(52)

```
> # Test

  inlter:
  subs([t0=13, r0=5*10^6, te0=Pi/4], %):
  subs([branch=Ingoing, clock=Clockwise], %):
  subs([branch=Ingoing], %):
  subs([K2=1/3*K2crit], %):
  subs([r=4*10^6], %):
  evalf(%):
  %;


  # 0.785398162732171
  # 0.78539816273217165752
  # 0.7853981627321716575185171785762213111637
  #
  0.785398162732171657518517178576221311163793270214455722405161285
  27790383719947160
  #
  0.785398162732171657518517178576221311163793270214455722405161285
  27903837199471595095447571793144393708942829965589722622

0.785398162732171657518517178576221311163793270214455722405161285277903837199471\
60
```

(53)

```
>
  # Three real solutions
  # K2crit < K2

> lp-cv^2*(r+rh+rm)*(r-rh)*(r-rm):
  expand(%):
  collect(%, r):
  [subs(r=0, %), subs(r=0, diff(%, r))]:
  solve(%, [K2, rh]):
  op(%):
  allvalues(%):
  [%][2]:
  subs([al=alv, c=cv], %):
  simplify(%):
  lSol:=%;
```

$$lSol := \left[K2 = \frac{2019402178265622557315950186537924\, rm^3}{-199306945455000 + 22468879468420441\, rm}, rh\right] \quad (54)$$

$$= \frac{1}{-398613890910000 + 44937758936840882\, rm}\left(\left(199306945455000\right.\right.$$

$$- 22468879468420441\, rm$$

$$+ \left(5048505445664056393289875466344 81\, rm^2 + 895640746929488445878809131 0000\, rm\right.$$

$$\left.\left.\left. - 11916977551980703547107 5000000\right)^{1/2}\right) rm\right)\Big]$$

```
> subs(lSol, rh):
  radsimp(%):
  rhv:=%;
```

$$rhv := -\left(\left(-199306945455000 + 22468879468420441\, rm\right.\right. \tag{55}$$

$$- \left(5048505445664056393289875466344 81\, rm^2 + 895640746929488445878809131 0000\, rm\right.$$

$$\left.\left.\left. - 11916977551980703547107 5000000\right)^{1/2}\right) rm\right) \Big/ \left(2\,\left(-199306945455000\right.\right.$$

$$\left.\left. + 22468879468420441\, rm\right)\right)$$

```
> # I could also try to use the numerical integration with K2(rm)
  # which seems pretty stable.

  subs(lSol, K2):
  radsimp(%):
  K2v:=%;
```

$$K2v := \frac{20194021782656225573159501865379 24\, rm^3}{-199306945455000 + 22468879468420441\, rm} \tag{56}$$

```
> # verify r0 is geater than al nd less that 3/2*al (result must be
  true)

  [alv, rhv, 3/2*alv]:
  subs([rm=2*rcrit], %):
  evalf(%):
  evalb( %[1]<%[2] and %[2] < %[3] ):
  %;
```

$$true \tag{57}$$

```
> lwPhi;
  cv^2;
```

$$\frac{1}{r^5}\left(\left(-r + \frac{199306945455000}{22468879468420441}\right)^2\left(89875517873681764\, r^3 - K2\, r\right.\right.$$

$$\left.\left. + \frac{199306945455000}{22468879468420441}\, K2\right)\right)$$

$$89875517873681764 \tag{58}$$

```
> subs([al=alv, c=cv],lp)=cv^2*(r+rh+rm)*(r-rh)*(r-rm):
  %;
```

$$89875517873681764\, r^3 - K2\, r + \frac{199306945455000}{22468879468420441}\, K2 = 89875517873681764\,(r + rh \tag{59}$$

$$+ rm)\ (r - rh)\ (r - rm)$$

```
> lwPhi:
  subs([subs([al=alv, c=cv],-lp)=-cv^2*(r+rh+rm)*(r-rh)*(r-rm)], %):
  subs([subs([al=alv, c=cv],lp)=cv^2*(r+rh+rm)*(r-rh)*(r-rm)], %):
  #subs(rh=rhv, %):
  #subs([al=alv, c=cv], %):
  simplify(%):
  lPhi:=%;
```

$$lPhi := \frac{4\ (22468879468420441\ r - 199306945455000)^2\ (r + rh + rm)\ (r - rh)\ (r - rm)}{22468879468420441\ r^5} \quad \textbf{(60)}$$

```
>
  # It is convenient to integrate from rm 'cos it removes divergences


  #                      r                    r
  #    ltr = t0 + b*int        = tm + b*int                    (t0,
  r0, branch; r, rm)
  #                      r0                   rm


  # for r=r0 ltr=t0
  # it is regular in r=rm
```

```
> lPhi:
  simplify(%):
  sqrt(%):
  radsimp(%)  assuming  r>Surface, r>rm, rm>rh, rh<3/2*alv, rh>alv:
  simplify(%) assuming  r>Surface, r>rm, rm>rh, rh<3/2*alv, rh>alv:
  1/%:
  #%;
  subs(r=R, %):
  int(%, R=rm..r) assuming r>Surface, r>rm, rm>3/2*alv, rh<3/2*alv,
  rh>alv:
  #subs(rh=rhv, %):
  simplify(%) assuming r>Surface, r>rm, rm>3/2*alv:
  radsimp(%)  assuming r>Surface, r>rm, rm>3/2*alv:
  simplify(%) assuming r>Surface, r>rm, rm>3/2*alv:
  t0 + branch*(%- subs([r=r0], %)):
  #subs(rm=Q(rm), %):
  #subs(r=Q(r), %):
  radsimp(%)  assuming r>Surface, r0>Surface, r>rm, r0>rm, rm>3/2*
  alv:
  simplify(%) assuming r>Surface, r0>Surface, r>rm, r0>rm, rm>3/2*
  alv:
  #factor(%):
  ltr:= %:
```

```
> lwPsi:
  subs([subs([al=alv, c=cv],-lp)=-cv^2*(r+rh+rm)*(r-rh)*(r-rm)], %):
```

```
subs([subs([al=alv, c=cv],lp)=cv^2*(r+rh+rm)*(r-rh)*(r-rm)], %):
subs(K2=K2v, %):
#subs(rh=rhv, %):
#subs([al=alv, c=cv], %):
lPsi:=%;
```

$$lPsi := \frac{(r+rh+rm)\,(r-rh)\,(r-rm)\,r\,(-199306945455000+22468879468420441\,rm)}{22468879468420441\,rm^3} \quad (61)$$

```
> # It is convenient to integrate from rm 'cos it removes divergences


  #                          r                          r
  #    lter = te0 + b*int          = tem + b*int
  (te0, r0, branch, clock; r, rm)
  #                          r0                         rm


  # for r=r0 lter=te0
  # it is regular in r=rm
```

```
> lPsi:
  simplify(%):
  sqrt(%):
  radsimp(%)  assuming  r>rm, rm>rh, rh<3/2*alv, rh>alv:
  simplify(%) assuming  r>rm, rm>rh, rh<3/2*alv, rh>alv:
  1/%:
  #%;
  subs(r=R, %):
  int(%, R=rm..r) assuming r>rm, rm>3/2*alv, rh<3/2*alv, rh>alv:
  #subs(rh=rhv, %):
  simplify(%) assuming r>rm, rm>3/2*alv:
  radsimp(%)  assuming r>rm, rm>3/2*alv:
  simplify(%) assuming r>rm, rm>3/2*alv:
  te0+ clock*branch*(% -subs(r=r0, %)):
  radsimp(%)  assuming r>rm, r0>rm, rm>3/2*alv:
  simplify(%) assuming r>rm, r0>rm, rm>3/2*alv:
  lter:=%:
```

```
> evalf(subs(rm=80, rhv), 60);
  R;
  rh=rhv;
```

$$0.00887035547033753326125340071062278747928439084598552211 44$$

$$R$$

$$rh = -\left(\left(-199306945455000+22468879468420441\,rm - \left(5048505445664056393289875466344 81\,rm^2+8956407469294884458788091310000\,rm - 11916977551980703547107500000 0\right)^{1/2}\right)rm\right)\Big/\left(2\,(-199306945455000+22468879468420441\,rm)\right) \quad (62)$$

```
> Evalf:= proc(x)
   evalf(x, Digits+20):
   evalf(%):
  end:
```

```
> # I could also try to use the numerical integration with K2(rm)
  # which seems pretty stable.


  K2v:
  subs(rm=4000, %):
  evalf(%, Digits+20):
  K2vrm:=%;

  evalf(K2crit);
```

$$K2vrm := 1.43801147489710721341078442729653394805549952864509998155930046906887147301\backslash 04110962999485210202384854.773393266832375251471851872452971366257453291608860842931941751352531287614961\backslash 95 \times 10^{13}$$ **(63)**

```
> inltr:
  subs([t0=13, r0=5000], %):
  subs([branch = Ingoing, clock=Clockwise], %):
  subs(K2=K2vrm, %):
  subs(r=5000.0, %):
  value(%):
  %;

  # fa casino a r=rm, per il resto è preciso
  #  per il plot non è un problema, posso vivere senza andare vicino
  a r~rm
  #  per fsolve il casino è se sourceR è ~rm (sv~0) che mi sa prima o
  poi, ma alcune le dovrebbe trovare)
  #            Mi sa che fa casino pure per other branch che con
  infalling non c'è.

  inltr:
  subs([t0=13, r0=5000], %):
  subs([branch = Ingoing, clock=Clockwise], %):
  subs(K2=K2vrm, %):
  subs(r=4000.1, %):
  evalf(%, Digits+20):
  evalf(%):
  #value(%):
  %;

  # 13.000009912601271801366487974800399054443724614698408427799104136133027523466637687227490988335000136644102403220260506 3
```

13.

13.000009912601271801366487974800399054443724614698408427799104136133027523466 6\ **(64)**

38

```
> #Test

  ltr:
  subs(rh=rhv, %):
  subs([t0=13, r0=5000], %):
  subs([branch = Ingoing, clock=Clockwise], %):  # even though clock
  is not used
  subs([r=5000., rm=4000.], %):
  #factor(%):
  evalf(%, Digits+20):
  evalf(%):
  %;


  # 13.0000000000000
  # 13.000000000000000000
  # 13.000000000000000000000000000000000000
  #
  13.000000000000000000000000000000000000000000000000000000000000000000000
  00000000000000
  #
  13.000000000000000000000000000000000000000000000000000000000000000000000
  000000000000000000000000000000000000000000000000000
```

12.999999999999999999999999999999999999999999999999999999999999999999999575163041\ **(65)**

47

```
> ltr:
  subs(rh=rhv, %):
  subs([t0=13, r0=5000], %):
  subs(branch = Ingoing, %):
  subs(rm=4000., %):
  subs(r=4000., %):
  evalf(%, Digits+20):
  evalf(%):
  %;

  # 13.0000100069483
  # 13.000010006948296459
  # 13.000010006948296459348780472629250456
  #
  13.000010006948296459348780472629250456346925908083609756006002180
  48670229387471
  #
  13.000010006948296459348780472629250456346925908083609756006002180
  486702293874707942541871684323815155008409344186458339
```

13.000010006948296459348780472629250456346925908083609756006002180848627745658 9\ **(66)**

15

```
> # Test
  lter:
  subs(rh=rhv, %):
  subs([te0=Pi/4, r0=5000], %):
  subs([branch = Ingoing, clock=Clockwise], %):
  subs([rm=4000], %):
  subs(r=5000, %):
  evalf(%):
  %;

  # 0.785398163397448
  # 0.78539816339744830962
  # 0.78539816339744830961566084581987572104929234984377645524373614807
  #
  0.78539816339744830961566084581987572104929234984377645524373614807
  695410157155225
  #
  0.78539816339744830961566084581987572104929234984377645524373614807
  69541015715522496570087063355292669955370216283205766662
  #

  lter:
  subs(rh=rhv, %):
  subs([te0=Pi/4, r0=5000], %):
  subs([branch = Ingoing, clock=Clockwise], %):
  subs([rm=4000], %):
  subs(r=4000, %):
  evalf(%):
  %;

  # 0.141896019726870
  # 0.14189601972687309574
  # 0.14189601972687309572690449533406877149
  #
  0.14189601972687309572690449533406877149804469972737078694229940889
  431121598109676
  #
  0.14189601972687309572690449533406877149804469972737078694229940889
  43112159810967485413284875394414162778895669451520318 24
  #
```

$$0.78539816339744830961566084581987572104929234984377645524373614807695410157155\ 221$$

$$0.14189601972687309572690449533406877149804469972737078694229940889431121598109\ 673 \quad \textbf{(67)}$$

Light rays

```
> # Procedures for rays
>
  #  Infalling rays are made of one branch only.

> InFallingRayThrough := proc(r0v, te0v, t0v, branchv, clockv)
    [r*cos(inlter), r*sin(inlter), inltr]:
    subs([r0=r0v, te0=te0v, t0=t0v], %):
    subs([branch=branchv, clock=clockv], %):
    evalf(%, Digits+20):
    #evalf(%):
  end:


> ScatteringBranchThrough := 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(%, Digits+20):
    #evalf(%):
  end:


> ScatteringOtherBranchThrough := proc(r0v, te0v, t0v, branchv,
  clockv) local Tm, Tem, Rm:
   [r, lter, ltr]:
   subs([r0=r0v, te0=te0v, t0=t0v], %):
   subs([branch=branchv, clock=clockv], %):
   subs(r=rm, %):
   evalf(%, Digits+20):
   Rm, Tem, Tm := op(%):
   ScatteringBranchThrough(Rm, Tem, Tm, -branchv, clockv)
  end:

> ScatteringRayThroughEm := proc(rmv, tem, tm, clockv := Clockwise)
  :
  local crm0, crm1, eq1, eq2, dt:

   ScatteringBranchThrough(rmv, tem, tm, Ingoing, clockv):
   subs(rm=rmv, %):
   eq1:=%:
   subs(r= maxS(2), %):
   dt:= evalf(%[3]):

   ScatteringOtherBranchThrough(rmv, tem, tm, Ingoing, clockv):
   subs(rm=rmv, %):
   eq2:=%:
   subs(r= maxS(2), %)[3]- dt:
   dt:= evalf(%):
   printf("dt= %g\n", dt);

   [op(eq1), r=rmv..5*10^6]:
   crm0:= spacecurve(%, axes=box, color = black, linestyle = solid)
  :
   [op(eq2), r=rmv..5*10^6]:
   crm1:= spacecurve(%, axes=box, color = black, linestyle = solid)
  :
```

```
    [crm0, crm1], dt:
  end:

> ScatteringRayThrough := proc(rmv, clockv := Clockwise):
    ScatteringRayThroughEm(rmv, 0, 0, clockv):
  end:

> rayMin:= proc()
    local sourceR, targetR, t:
    global SearchSignal:


    sourceR := SearchSignal[Risr]:
    targetR := SearchSignal[TargetP][1]:
    t := SearchSignal[RayType]:

    if t = Infalling or t = SameScattering then
      return min(sourceR, targetR);
    elif t = OtherScattering then
      return SearchSignal[Risrm];
    else
      printf("Ray type unknown %d. \n", t);
    end if:
  end:
```

Procedure for plotting

```
> # Try to get a segment from Sat 0 (r=13 on branch 1) to Sat 2

> AddPlotPointCartesian:= proc(P, PSat, G)
    local col:
    global Plots:
    if PSat = -1 then
     col := black:
    elif PSat = 0 then
     col := red:
    elif PSat = 1 then
     col := blue:
    elif PSat = 2 then
     col := green:
    end if:
    P:
    #pointplot3d(%, color=col, symbol = solidcircle, symbolsize =
  6+3*G):
    pointplot3d(%, color=col, symbol = solidcircle, symbolsize = 6):
    Plots := [op(Plots), %]:
  end:


> AddPlotPointPolar:= proc(P, PSat, G)  # Polar
    local R, Te, T, col:
    global Plots:
```

```
   R, Te, T := op(P):
   [R*cos(Te), R*sin(Te), T]:
   AddPlotPointCartesian(%, PSat, G):
  end:

> AddPlotRayScatteringSame := proc(Through, r1v, S1, G1, rmv,
  branchv, clockv)
   local eq:
   global Plots:
   ScatteringBranchThrough(Through[1], Through[2], Through[3],
  branchv, clockv):
   subs(rh=rhv, %):
   subs(rm=rmv, %):
   eq:= %:
   subs(r=r1v, eq):
   AddPlotPointCartesian(%, S1, G1+1):
   if r1v > Through[1] then
    eq:=[op(eq), r=Through[1]..r1v]:
   else
    eq:=[op(eq), r=r1v..Through[1]]:
   end if:
   spacecurve(eq, axes=box, color = black, linestyle = solid,
  thickness = 1):
   Plots := [op(Plots), %];
  end:

> AddPlotRayScatteringOther := proc(Through, r1v, S1, G1, rmv,
  branchv, clockv)
   local eq:
   global Plots:
    ScatteringBranchThrough(Through[1], Through[2], Through[3],
  branchv, clockv):
   subs(rh=rhv, %):
   subs(rm=rmv, %):
   eq:= %:
   [op(eq), r=Through[1]..rmv]:
   spacecurve(%, axes=box, color = black, linestyle = solid,
  thickness = 1):
   Plots := [op(Plots), %];

   ScatteringOtherBranchThrough(Through[1], Through[2], Through[3],
  branchv, clockv):
   subs(rh=rhv, %):
   subs(rm=rmv, %):
   eq:= %:
   subs(r=r1v, eq):
   AddPlotPointCartesian(%, S1, G1+1):
   [op(eq), r=rmv..r1v]:
   spacecurve(%, axes=box, color = black, linestyle = solid,
  thickness = 1):
   Plots := [op(Plots), %];
  end:

> AddPlotRayInfalling := proc(Through, r1v, S1, G1, K2v, branchv,
  clockv)
   local eq:
```

```
 global Plots:
 #printf("Step 1\n");
 InFallingRayThrough(Through[1], Through[2], Through[3], branchv,
clockv):
 subs(K2=K2v, %):
 eq:= %:
 subs(r=r1v, eq):
 AddPlotPointCartesian(%, S1, G1):
 if r1v > Through[1] then
  eq:=[op(eq), r=Through[1]..r1v]:
 else
  eq:=[op(eq), r=r1v..Through[1]]:
 end if:
 spacecurve(eq, axes=box, color = black, linestyle = solid,
thickness = 1):
 Plots := [op(Plots), %];
 #printf("Step out\n");
end:

> PlotLinear := proc()
  local sourceSat, sourceBranch, rGuessMax, rGuessMin, Previous,
targetP, St:
  global OtherPlots:
   St := time():
   sourceSat := Get(SourceSat):
   sourceBranch := Get(SourceBranch):
   rGuessMax := Get(IndrGuessMax):
   rGuessMin := Get(IndrGuessMin):
   Previous:= Get(PreviousSourceBranch):
   targetP := Get(TargetP):

   [xS(sourceSat, sourceBranch), yS(sourceSat, sourceBranch), tS
(sourceSat, sourceBranch)]:
   subs([r=rGuessMax],%):
   pointplot3d(%, color=black, symbol = solidcircle, symbolsize =
9):
   OtherPlots := [op(OtherPlots), %]:
   if Previous then
     [xS(sourceSat, sourceBranch-1), yS(sourceSat,
sourceBranch-1), tS(sourceSat, sourceBranch-1)]:
     subs([r=rGuessMin],%):
     pointplot3d(%, color=black, symbol = solidcircle, symbolsize
= 9):
     OtherPlots := [op(OtherPlots), %]:
   else
     [xS(sourceSat, sourceBranch), yS(sourceSat, sourceBranch),
tS(sourceSat, sourceBranch)]:
     subs([r=rGuessMin],%):
     pointplot3d(%, color=black, symbol = solidcircle, symbolsize
= 9):
     OtherPlots := [op(OtherPlots), %]:
   end if:
   [s*targetP[1]*cos(targetP[2]) + (1-s)*xS(sourceSat,
sourceBranch),
    s*targetP[1]*sin(targetP[2]) + (1-s)*yS(sourceSat,
sourceBranch),
```

```
      targetP[3],
      s=0..1
     ]:
     subs(r=rGuessMax, %):
     spacecurve(%, axes=box, color = black, linestyle = solid,
  thickness = 1):
     OtherPlots := [op(OtherPlots), %]:
     if DebugTimeOn and ProducePlots then
       printf("Time Approximations %g.\n\n", time()-St);
     end if:
  end:
```

```
> AddPlotRay:= proc()
    local sourceSat, targetP, risr,risrm, risK2, Gen, rayBranchv,
  rayClockv, rayTypev, St:

    St:=time():
    if Get(Risr) = none then
      printf("No solution found\n");
    else
      sourceSat:= Get(SourceSat):
      targetP:= Get(TargetP):
      risr:= Get(Risr):
      risrm:= Get(Risrm):
      risK2:= Get(RisK2):
      Gen:= GetGeneration():
      rayBranchv:= Get(RayBranch):
      rayClockv:= Get(RayClock):
      rayTypev:= Get(RayType):

      if ProducePlots then
             if rayTypev = Infalling then
                AddPlotRayInfalling(targetP, risr, sourceSat,
  GetGeneration(), risK2, rayBranchv, rayClockv):
             elif rayTypev = SameScattering then
                AddPlotRayScatteringSame(targetP, risr, sourceSat,
  GetGeneration(), risrm, rayBranchv, rayClockv):
             elif rayTypev = OtherScattering then
                AddPlotRayScatteringOther(targetP, risr, sourceSat,
  GetGeneration(), risrm, rayBranchv, rayClockv):
             else
                  printf("Ray type unknown\n"):
             end if:
       end if:
    end if:
    if DebugTimeOn and ProducePlots then
      printf("Time Plot %g s.\n", time()-St);
    end if:
  end:
```

```
> MaxT:= 1/2;
```

$$MaxT := \frac{1}{2} \tag{68}$$

```
> # Then for any exchange, we should not need to go back in time
  further than 90 and closer to BH than 3.5
  #
  # I'm not saying that being stricter on limits gets us faster
  though.
  # I would be much more first test than decide about it.
```

Procedures for solving signals

```
> Equations:= proc()
    local rayTypev, rayBranchv, rayClockv, sourceSat, sourceBranch,
  targetR, targetTe, targetT:
    local eqs, xx1, yy1, tt1, xx2, yy2, tt2, rs, tes, ts:

    rayTypev := Get(RayType):
    rayBranchv := Get(RayBranch):
    rayClockv := Get(RayClock):
    sourceSat := Get(SourceSat):
    sourceBranch := Get(SourceBranch):
    targetR, targetTe, targetT := op(Get(TargetP)):

    rs:= rSat(sourceSat, s, sourceBranch):
    tes:= teSat(sourceSat, s, sourceBranch):
    ts:= tSat(sourceSat, s, sourceBranch):

    if rayTypev = Infalling then
      InFallingRayThrough(targetR, targetTe, targetT, rayBranchv,
  rayClockv):
      xx1, yy1, tt1 := op(%):   # (r, K2)

      InFallingRayThrough(rs, tes, ts, rayBranchv, rayClockv):
      subs(s=r, %): # (r, rm)
      xx2, yy2, tt2 := op(%):   # (r, s, K2)
      [xx1-xx2, yy1-yy2, tt1-tt2]:  # (r, K2)
      eqs:=%:
    elif rayTypev = SameScattering then
      ScatteringBranchThrough(targetR, targetTe, targetT,
  rayBranchv, rayClockv):
      xx1, yy1, tt1 := op(%): # (rh, rm, r)

      ScatteringBranchThrough(rs, tes, ts, rayBranchv, rayClockv):
      xx2, yy2, tt2 := op(%):
      [xx1-xx2, yy1-yy2, tt1-tt2]:  # (s, rm)
      subs(rh=rhv, %):
      subs(r=rm, %):
      subs(s=r, %): # (r, rm)
      eqs:=%:
    elif rayTypev = OtherScattering then
      ScatteringBranchThrough(targetR, targetTe, targetT,
  rayBranchv, rayClockv):
      xx1, yy1, tt1 := op(%):

      ScatteringBranchThrough(rs, tes, ts, -rayBranchv, rayClockv):
```

```
      xx2, yy2, tt2 := op(%):
      [xx1-xx2, yy1-yy2, tt1-tt2]:  # (s, rm)
      subs(rh=rhv, %):
      subs(r=rm, %):
      subs(s=r, %): # (r, rm)
      eqs:=%:
    else
    end if:
    eqs:
end:

> SolveOne:= proc()
  local sourceSat, sourceBranch:
  local targetR, targetTe, targetT:
  local sv, rGuessMax, rGuessMin, rGuess, rmGuess, kv, scos,
Previous, Crossing:
  local rayBranchv, rayClockv, rayTypev, SolveTypev:
  local eqs, xx, yy, tt, avoidSols, pstring, ris, interval, Delta,
St:
  local SignalSave:


  St:= time():
  sourceSat := Get(SourceSat):
  sourceBranch := Get(SourceBranch):
  sv := Get(Indsv):
  rGuessMax := Get(IndrGuessMax):
  rGuessMin := Get(IndrGuessMin):
  Previous := Get(PreviousSourceBranch):
  Crossing := Get(SourceCrossing):
  rmGuess := Get(IndrmGuess):
  kv := Get(Indkv):
  scos := Get(Indscos):
  rayTypev := Get(RayType):
  SolveTypev := Get(SolveType):
  targetR, targetTe, targetT := op(Get(TargetP)):
  rayBranchv := Get(RayBranch):
  rayClockv := Get(RayClock):

  if DebugFlowControlOn then
    pstring:= "I search for ":
    if rayTypev = Infalling then
      pstring := cat(pstring,"an infalling ray "):
    elif rayTypev = SameScattering then
      pstring := cat(pstring,"an scattering ray on same branch "):
    elif rayTypev = OtherScattering then
      pstring := cat(pstring,"an scattering ray on opposite
branches "):
    else
      pstring := cat(pstring,"an unknown ray "):
    end if:
    if sv<0 then
      pstring := cat(pstring,"with sv<0 (%g) "):
    elif sv>0 and sv<1 then
      pstring := cat(pstring,"with 0<sv<1 (%g) "):
    else # sv>1
```

```
      pstring := cat(pstring,"with sv>1 (%g) "):
    end if:

    if targetR < minS(sourceSat) then
      pstring := cat(pstring,"|   P  <--- S \n"):
    elif targetR > minS(sourceSat) then
      pstring := cat(pstring,"|   S  ---> P \n"):
    else
      pstring := cat(pstring,"mixing source and target.\n"):
    end if:
    printf(pstring, sv);
    printf("rGuessMin=%g   rGuessMax=%g   rmGuess=%g  k=%g   scos=
%g\n",  rGuessMin, rGuessMax, rmGuess, kv, scos);
    pstring := "branch ":
    if rayBranchv = Outgoing then
      pstring := cat(pstring," outgoing at target, "):
    else
      pstring := cat(pstring," ingoing at target, "):
    end if:
    if rayClockv = Clockwise then
      pstring := cat(pstring,"Clockwise\n"):
    else
      pstring := cat(pstring,"Counterclockwise\n"):
    end if:
    printf(pstring);
  end if:

  if rayTypev = Infalling then
    interval:={r= Get(Intervalr1), K2 = Get(IntervalK2)}:
  elif rayTypev = SameScattering then
    interval:={r= Get(Intervalr1), rm = Get(Intervalrm)}:
  elif rayTypev = OtherScattering then
    interval:={r= Get(Intervalr1), rm = Get(Intervalrm)}:
  else
    interval:={}:
  end if:
  Equations():
  subs(rh=rhv, %):
  #evalf(%, Digits+20):
  evalf(%):
  eqs:= %:
  avoidSols := {}:
  ris := "searching":
  do
   if rayTypev = Infalling then
     if DebugSolutionsOn then
       printf("(infalling) fsolve({eqs1, eqs3}, {r=%a, K2=0}, %a,
avoid=%a);\n", rGuessMax, interval, avoidSols);
     end if:
     fsolve({eqs[1], eqs[3]}, {r=rGuessMax, K2=0}, interval,
avoid=avoidSols, fulldigits):
     ris:= %:
   else
     # il lato alto di Intervalr1
     #rGuess:= op(2, Get(Intervalr1)):
     #printf("Step1 (%d)\n", Digits);
     rGuess:= Get(IndrGuessMax):
```

```
    #interval:= {r=myRange(Get(IndrGuessMin), rGuess), rm=
rmGuess-500000..rmGuess+500000}:

    if DebugSolutionsOn then
      printf("(Scattering) fsolve(eqs, {r=%a, rm=%a}, %a, avoid=
%a); \n", evalf(rGuess), evalf(rmGuess), evalf(interval),
avoidSols);
    end if:
    fsolve({eqs[1], eqs[3]}, {r=rGuess, rm=rmGuess}, interval,
avoid=avoidSols, fulldigits):
    #fsolve({eqs[1], eqs[3]}, {r=rGuess, rm=rmGuess}, interval,
avoid=avoidSols):
    #fsolve({eqs[1], eqs[3]}, {r=rGuess, rm=rmGuess}, avoid=
avoidSols):
    ris:= %:
  end if:
  if type(ris, 'set') then
     eqs[2]:
     subs(ris, %):
     Delta := abs(evalf(%)):
     if Delta > 10^(-Digits+20) then
       avoidSols:= {op(avoidSols), ris}:        # forse posso
restringere l'intervallo?
       if DebugSolutionsOn then
           printf("Rejected {r=%g, rm=%g} for Delta=%g\n", subs
(ris, r), subs(ris, rm), Delta);
           if DebugTimeOn then
              printf("in partial time = %g s \n", time()-St);
           end if:
       end if:
       ris:="rejected":
     else
       if DebugSolutionsOn then
         if rayTypev = Infalling then
              print(ris);
              printf("Accepted {r=%g, K2=%g} with Delta=%g\n",
subs(ris, r), subs(ris, K2), Delta);
         else
             printf("Accepted {r=%g, rm=%g} with Delta=%g\n",
subs(ris, r), subs(ris, rm), Delta);
         end if:
         subs(ris, eqs):
         printf("Equations at solution: %a\n", %):
       end if:
     end if:
  else
   ris:= {}:
  end if:
 until type(ris, set):
 if nops(ris)> 0 then
   [r, teSat(sourceSat, r, sourceBranch), tSat(sourceSat, r,
sourceBranch)]:
   subs(r=subs(ris, r), %):
   Set(RisP, %):
   Set(Risr, subs(ris, r)):
   Set(RisBranch, sourceBranch):
```

```
    if rayTypev = Infalling then
      Set(RisK2, subs(ris, K2)):
      ris:= [sourceSat, subs(ris, r), subs(ris, K2), sourceBranch,
rayTypev];
    else
      Set(Risrm, subs(ris, rm)):
      ris:= [sourceSat, subs(ris, r), subs(ris, rm), sourceBranch,
rayTypev];
    end if:
  else
    #  if no solution found
    #     Se crossing vando a cercare sull'altro Ramo
    #     Per -clock e infalling/scattering invece lascio fare al
chiamante
    #
    if Crossing then
      if DebugFlowControlOn and DebugTimeOn then
        printf("Turn to the other branch [nested SolveOne()]\n");
      end if:
      SignalSave := SaveSearchSignal():
      SetPreviousSourceBranch():
      ris := SolveOne():
      RestoreSearchSignal(SignalSave):
    else
      # ris:= {}:
    end if:
  end if:
  if DebugFlowControlOn and DebugTimeOn then
    printf("Solution in %gs\n\n", time()-St);
  end if:
  ris;
end:
```

```
> CompareSolution:= proc()
    local Newrm, Bestrm:
    global SearchSignal, BestKnownSolution:

    Newrm := SearchSignal[Risrm]:
    if not(Newrm = none) and BestKnownSolution = none then
      BestKnownSolution := SearchSignal:
    elif not(Newrm = none) then
       Bestrm := BestKnownSolution[Risrm]:
       if Bestrm < Newrm then
         BestKnownSolution := SearchSignal:
       end if:
    end if:
  end:
```

```
> ValidateSolution:= proc()  #  -> Bool
   local sourceR, targetR, t:
   global SearchSignal, BestKnownSolution:

   sourceR := SearchSignal[Risr]:
   targetR := SearchSignal[TargetP][1]:
   if sourceR = none then
```

```
    return false:
   end if:
   if rayMin() > 4*alv then
     BestKnownSolution := SearchSignal:
     return true:
   end if:
   CompareSolution():
   return false:
  end:

> AcceptBestSolution := proc()
   global SearchSignal, BestKnownSolution:
   SearchSignal := BestKnownSolution:
  end:

> #(Infalling0 | SameScattering1 | OtherScattering2) + (Ingoing0 +
  Outgoing1) + (Orario0 | Antitorario1)


  Sol2Code:=proc()
   local r:
   global SearchSignal:

   if Get(RayClock) = Clockwise  then
     r:= 0*1:
   else
     r:= 1*1:
   end if:
   if Get(RayBranch) = Outgoing then
     r:= r+1*2:
   else
     r:= r+0*2:
   end if:
   if Get(RayType) = Infalling  then
     r:= r+ 0*4:
   elif Get(RayType) = SameScattering  then
     r:= r+ 1*4:
   elif Get(RayType) = OtherScattering  then
     r:= r+ 2*4:
   else
     printf("RayType unknown %d\n", Get(RayType)):
   end if:
   return r+1;
  end:

> DoneCode:=proc()
   local r:
   global TypeToDo:
   r:= Sol2Code():
   TypeToDo[r]:= false:
  end:


> #(Infalling0 | SameScattering1 | OtherScattering2) + (Ingoing0 +
  Outgoing1) + (Orario0 | Antitorario1)
```

```
Code2Sol := proc(n)
 global SearchSignal;

 if n = 1 then                    #  Infalling       Ingoing
Clockwise
  Set(RayType, Infalling);
  Set(RayClock, Clockwise);
  Set(RayBranch, Ingoing);
 elif n = 2 then                  #  Infalling       Ingoing
Counterclockwise
  Set(RayType, Infalling);
  Set(RayClock, Counterclockwise);
  Set(RayBranch, Ingoing);
 elif n = 3 then                  #  Infalling       Outgoing
Clockwise
  Set(RayType, Infalling);
  Set(RayClock, Clockwise);
  Set(RayBranch, Outgoing);
 elif n = 4 then                  #  Infalling       Outgoing
 Counterclockwise
  Set(RayType, Infalling);
  Set(RayClock, Counterclockwise);
  Set(RayBranch, Outgoing);

 elif n = 5 then                  #  SameScattering  Ingoing
 Clockwise
  Set(RayType, SameScattering);
  Set(RayClock, Clockwise);
  Set(RayBranch, Ingoing);
 elif n = 6 then                  #  SameScattering  Ingoing
 Counterclockwise
  Set(RayType, SameScattering);
  Set(RayClock, Counterclockwise);
  Set(RayBranch, Ingoing);
 elif n = 7 then                  #  SameScattering  Outgoing
 Clockwise
  Set(RayType, SameScattering);
  Set(RayClock, Clockwise);
  Set(RayBranch, Outgoing);
 elif n = 8 then                  #  SameScattering  Outgoing
 Counterclockwise
  Set(RayType, SameScattering);
  Set(RayClock, Counterclockwise);
  Set(RayBranch, Outgoing);

 elif n = 9  then                 #  OtherScattering Ingoing
 Clockwise
  Set(RayType, OtherScattering);
  Set(RayClock, Clockwise);
  Set(RayBranch, Ingoing);
 elif n = 10 then                 #  OtherScattering Ingoing
 Counterclockwise
  Set(RayType, OtherScattering);
  Set(RayClock, Counterclockwise);
  Set(RayBranch, Ingoing);
 elif n =11 then                  #  OtherScattering Outgoing
 Clockwise
```

```
    Set(RayType, OtherScattering);
    Set(RayClock, Clockwise);
    Set(RayBranch, Outgoing);
   elif n = 12 then                #  OtherScattering Outgoing
   Counterclockwise
    Set(RayType, OtherScattering);
    Set(RayClock, Counterclockwise);
    Set(RayBranch, Outgoing);
   else
    printf("Type Code unknown %d\n", n);
   end if:
  end:

> SolveHard:= proc(TargetSat, TargetPoint, SourceSat, Gen, hint:=
  none)
   global Plots, OtherPlots, TypeToDo, BestKnownSolution,
  ErrorFlag:
   local sol, OriginalSS,  loop, n, targetR, sourceMin, sourceMax,
  recCodes, St:
    St:= time():
    if DebugFlowControlOn then
      printf("%d --> %d  target = %a\n", SourceSat, TargetSat,
  TargetPoint);
    end if:
    sol:=[]:

    Plots:=[];
    OtherPlots:=[];
    ClearSearchSignal():
    ClearTypeToDo();

    targetR:= TargetPoint[1]:
    sourceMin:= minS(SourceSat):
    sourceMax:= maxS(SourceSat):

    SetTargetPoint(TargetSat, TargetPoint):
    SetSourceSat(SourceSat):
    SetGeneration(Gen):
  #printf("Linear ..");
    LinearGuess():
  #printf(". ok!   ");
    if ProducePlots then
      PlotLinear():
    end if:
    if ErrorFlag then
       if DebugTimeOn then
        printf("Solution not found\n");
        printf("Exiting SolveHard() after %g\n", time()-St);
       end if:
       return [];
    end if:
    # Plot the ray through Target at rmGuess
    if ProducePlots then
     [TargetPoint[1]*cos(TargetPoint[2]), TargetPoint[1]*sin
  (TargetPoint[2]), TargetPoint[3]]:
     pointplot3d(%, color=orange, symbol = solidcircle, symbolsize
```

```
= 9):
   OtherPlots:=[op(OtherPlots), %]:
  end if:
  OriginalSS := SaveSearchSignal():
  if type(hint, list) then
    # hint ha anche informazioni su guess e forse intervals
    UseHint(hint):
    sol := SolveOne():
    DoneCode():
    if ValidateSolution() then
      AddPlotRay():
      if DebugTimeOn then
        printf("Exiting SolveHard() after %g\n", time()-St);
      end if:
      return sol:
    end if:
    if type(BestKnownSolution, list) then
      AcceptBestSolution():
      if DebugTimeOn then
        printf("Exiting SolveHard() after %g\n", time()-St);
      end if:
      return sol:
    end if:
  end if:
  RestoreSearchSignal(OriginalSS):
  OriginalSS := SaveSearchSignal():
  loop:= true:
  if targetR > sourceMax then
     recCodes:= [12, 11, 7, 3, 4, 8]:
     n:= Sol2Code():
     recCodes:= CycleUntil(recCodes, n);
  elif targetR < sourceMax then
     recCodes:= [12, 11, 5, 1, 2, 6]:
     n:= Sol2Code():
     recCodes:= CycleUntil(recCodes, n);
  else
    printf("Tagent is Source ring should not happen\n"):
    printf("Exiting SolveHard() after %g\n", time()-St);
    return []:
  end if:
  BestKnownSolution := none:
  do
    n:= recCodes[1]:
    if DebugFlowControlOn then
      printf("Try code %d\n", n);
    end if:
    Code2Sol(n);
    sol := SolveOne():
    DoneCode():
    if nops(sol) > 0 then
      if ValidateSolution() then
        AddPlotRay():
        if DebugTimeOn then
          printf("Exiting SolveHard() after %g\n", time()-St);
        end if:
        return sol:
      end if:
```

```
    else
                if DebugTimeOn then
                   printf("Exiting SolveHard() after %g\n", time()
-St);
                end if:
                return []:
    end if:
    recCodes := [op(2..nops(recCodes), recCodes)]:
  until nops(recCodes) = 0:

  if DebugFlowControlOn then
    printf("Warning: Going to unreccommended types\n");
  end if:
  for n from 1 to 12 do
    if TypeToDo[n] then
      printf("Try unreccommended code %d\n", n);
      Code2Sol(n);
      sol := SolveOne():
      DoneCode():
      if ValidateSolution() then
        AddPlotRay():
        if DebugTimeOn then
          printf("Exiting SolveHard() after %g\n", time()-St);
        end if:
        return sol:
      else
                if DebugTimeOn then
                   printf("Exiting SolveHard() after %g\n", time()
-St);
                end if:
                return []:
      end if:
    end if:
  end do:
  if DebugOn then
    printf("Could not find a validate solution\n");
  end if:
  AcceptBestSolution():

  if SearchSignal = none then
    printf("No solution\n");
    sol:=[]:
  else   #  [sourceSat, subs(ris, r), subs(ris, rm),
sourceBranch, rayTypev];
    if Get(RayType) = Infalling then
      sol:=[Get(SourceSat), Get(Risr), Get(RisK2), Get
(SourceBranch), Get(RayType)]:
    else
      sol:=[Get(SourceSat), Get(Risr), Get(Risrm), Get
(SourceBranch), Get(RayType)]:
    end if:
  end if:
  if DebugTimeOn then
    printf("Exiting SolveHard() after %g\n", time()-St);
  end if:
  #  Possiamo ricostruire la BestSolution  if any se no
  sol:
```

```
  end:
```

```
> Real
```

Real **(6.1)**

```
> # perché i - su clock e branch?

  DrawGuess := proc(Through, branchv, clockv, rmv)
   global Plots:
   local eq:

   # deal with infalling as well

    ScatteringBranchThrough(Through[1], Through[2], Through[3],
  branchv, clockv ):
    subs(rh=rhv, %):
    subs(rm=rmv, %):
    subs(r=Through[1], %):
    printf("P=%a= %a\n", %, [Through[1]*cos(Through[2]), Through[1]
  *sin(Through[2]), Through[3]]);
    ScatteringBranchThrough(Through[1], Through[2], Through[3],
  branchv, clockv):
    subs(rh=rhv, %):
    subs(rm=rmv, %):
    [op(%), r=rmv..Through[1]]:
    spacecurve(%, axes=box, color = orange, linestyle = solid,
  thickness = 3):
    Plots := [op(Plots), %];

    ScatteringOtherBranchThrough(Through[1], Through[2], Through
  [3], branchv, clockv):
    subs(rh=rhv, %):
    subs(rm=rmv, %):
    [op(%), r=rmv..3*10^7]:
    spacecurve(%, axes=box, color = orange, linestyle = solid,
  thickness = 3):
    Plots := [op(Plots), %];
  end:
```

```
> AcceptSolution:= proc()
    local sourceSat, gen, k;
    global NextSignalAvailable, ListPlots, Plots, OtherPlots,
  ListP, ListHints, ListTau:

    sourceSat := SearchSignal[SourceSat];
    gen := SearchSignal[Gen];

    ListPlots[gen+1] := [op(ListPlots[gen+1]), op(Plots)]:
    Plots:= []:
    OtherPlots:= []:

    k := 3*NextSignalAvailable[sourceSat+1] +sourceSat + 1:
    ListHints[k]:= CreateHint():
    ListP[k]:= Get(RisP):
    ListTau[k]:= tauSat(sourceSat, Get(Risr), Get(RisBranch));
    NextSignalAvailable[sourceSat+1]:= NextSignalAvailable
```

```
  [sourceSat+1] +1;
    k:
  end:

> SetBasePoint:= proc(Sat, rv, branchv)
   local n:
   global Plots, OtherPlots, NextSignalAvailable, ListHints, ListP,
  ListTau, ListPlots, MaxSignals, MaxGenerations:

   if ProducePlots then
    Plots:=[]:
    OtherPlots:=[]:

   end if:
   NextSignalAvailable:=CreateList(3, 0);
   #ListHints:=CreateList(MaxSignals, none);
   ListP:=CreateList(MaxSignals, none);
   ListTau:=CreateList(MaxSignals, none);
   ListPlots:= CreateList(MaxGenerations, []):

   n := 3*NextSignalAvailable[Sat+1]:
   [r, teSat(Sat, r, branchv), tSat(Sat, r, branchv)]:
   subs(r=rv, %):
   evalf(%):
   ListP[n+1]:= %:
   tauSat(Sat, rv, branchv):
   evalf(%):
   ListTau[n+1]:= %:
   NextSignalAvailable[Sat+1]:= NextSignalAvailable[Sat+1]+1:
   if ProducePlots then
     AddPlotPointPolar(ListP[n+1], Sat, 0):  # AddPlotPointPolar:=
  proc(P, PSat, G)
     ListPlots[1] := Plots:
     Plots:=[]:
     OtherPlots:=[]:
   end if:
   n+1:
  end:

> Cascade:= proc(Sat, rv, branchv)
   local Gen, NewSignals, OldSignals, n, St, event:
   global ListHints, ListTau, ListP, MaxGenerations, ErrorFlag:

   St:= time():

   ErrorFlag := false;

   Gen := 0:
   NewSignals:= []:

   n:= SetBasePoint(Sat, rv, branchv):
   NewSignals:= [op(NewSignals), n]:

   do
     Gen := Gen+1:
     #printf("Start Generation %d\n", Gen):
```

```
    OldSignals := NewSignals:
    NewSignals:= []:

    for n in OldSignals do
      event:= NextEvent(n mod 3):
      SolveHard((n-1) mod 3, ListP[n], n mod 3, Gen, ListHints
[event]):
      if nops(%) > 0 then
       AcceptSolution():
       NewSignals:= [op(NewSignals), %]:

       if DebugOn then
         printf("\nTau %a\n\n", ListTau);
       end if:
      else
       printf("\nNo solution found\n\n");
       #if DebugTimeOn then
         printf("Cascade time %g\n", time()-St);
       #end if:
       return "NotFound";
      end if:

      event:= NextEvent((n+1) mod 3):
      SolveHard((n-1) mod 3, ListP[n], (n+1) mod 3, Gen, ListHints
[event]):
      if nops(%) > 0 then
       AcceptSolution():
       NewSignals:= [op(NewSignals), %]:

       if DebugOn then
         printf("\nTau %a\n\n", ListTau);
       end if:
      else
       printf("\nNo solution found\n\n");
       #if DebugTimeOn then
         printf("Cascade time %g\n", time()-St);
       #end if:
       return "NotFound";
      end if:
    end do:
   until Gen = MaxGenerations-1:
   #if DebugTimeOn then
     printf("Cascade time %g\n", time()-St);
   #end if:
   ListTau:
  end:
```

```
> #f:= proc(r)
  # cos(r);
  #end:
  #
  #plot(f(r), r=-10..10);
```

Ok, now we can go for the signals

```
> Plots:=[];
  OtherPlots:=[];
```

$$Plots := [\,]$$

$$OtherPlots := [\,] \tag{69}$$

```
> ListHints:=CreateList(MaxSignals, none);
  ListTau:=CreateList(31, 0);
  ListP:=CreateList(31, []);
```

$$ListHints := [none, none]$$

$$ListTau := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$ListP := [[\,], [\,]] \tag{70}$$

```
> BaseSat:= BaseSatOriginal;
  BaseR:= BaseROriginal;
  BaseBranch:= BaseBranchOriginal;
  MaxT:= 1/2;
```

$$BaseSat := 0$$

$$BaseR := 23889000$$

$$BaseBranch := 0$$

$$MaxT := \frac{1}{2} \tag{71}$$

```
> Cascade(BaseSat, BaseR, BaseBranch);

1 --> 0  target = [23889000.,
2.63255643460110699996795137350840014798369334124130390697258788045752
40036514658,
10510.45814019397412810699008640310343641626781186985297188547521331685
4268817156]
one interval r =
24850248.88301387793152980010731189259063775326624740582311934804445258
2659327516 ..
62125696192963148851975981344143373766904097099512684578887050146463665
94952221/25000000000000000000000000000000000000000000000000000000000000
0000000000000000
Time Approximations 0.072.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.533028) |   P  <--- S
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.03965e+07
k=5.43204e+14   scos=2.39474e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
```

```
24850278.477185259540790392537657349506761638839805073831554820058585466379808884, rm=
20396532.000224296639883813760276210489500276684826063879778948414248171569162281}, {r =
24850248.883013877931529800107311892590637753266247405823119348044452582659327516 ..
24850278.477185259540790392537657349506761638839805073831554820058585466379808884, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 23889000.}, avoid={});
Accepted {r=2.48503e+07, rm=2.03964e+07} with Delta=1.0e-71
Equations at solution: [-.16e-70, -.10e-70, .17e-74]
Solution in 1.133s

Time Plot 2.861 s.
Exiting SolveHard() after 6.883
r=2.48503e+07 in
[24850248.883013877931529800107311892590637753266247405823119348044452582659327516 ..
62125696192963148851975981344143373766904097099512684578887050146463665949522 21/25000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03964e+07) in
[298960418182500/22468879468420441 .. 23889000.]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none]

2 --> 0  target = [23889000.,
2.63255643460110699996795137350840014798369334124130390697258788045752400365146 58,
10510.458140193974128106990086403103436416267811869852971885475213316854268817156]
one interval r =
25795878.077598356622876839415216948133846234600100346975637254314929166021300019 ..
25795907507808464278385632905911146002077381970654650449671463787784438810032137/1000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.053.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.63282) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.28604e+07
k=4.31733e+14   scos=4.39724e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
```

```
25795907.507808464278385632905911146002077381970654650449671463787784
4438810032137, rm=
22860405.032127814773769893952491881153412646677278957019340325727661
064403120086}, {r =
25795878.077598356622876839415216948133846234600100346975637254314921
9166021300019 ..
25795907.507808464278385632905911146002077381970654650449671463787784
4438810032137, rm =
.133055330419428732850022379412935116857570484811026812518551495926248
4481163562730e-1 .. 23889000.}, avoid={});
Accepted {r=2.57959e+07, rm=2.28604e+07} with Delta=1.6e-71
Equations at solution: [-.16e-70, -.16e-70, .82e-74]
Solution in 0.757s

Time Plot 2.645 s.
Exiting SolveHard() after 5.531
r=2.57959e+07 in
[25795878.077598356622876839415216948133846234600100346975637254314929
29166021300019 ..
257959075078084642783856329059111460020773819706546504496714637877844
38810032137/100000000000000000000000000000000000000000000000000000000
000000000000000]
Scattering ray (rm=2.28604e+07) in
[298960418182500/22468879468420441 .. 23889000.]: target and source
on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none]

2 --> 1  target =
[24850273.219190804417272987728022459024343182060859251424473865940104981285822392,
1.476968812052973768854931710912166405070997186502133826726857758214736395668662 4,
10510.369303882849439649629860128958025144381629311043902724336372114864851076446]
one interval r =
25795872.848587050647418641713444891292226023926734551769554753173974927791658948 ..
257959022789138379548668069776355182162284856805732296126355731653092186423744 67/100000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.049.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.760172) |   P  <--- S
```

```
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.47435e+07
k=-2.37383e+14   scos=5.95462e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
25795902.278913837954866806977635518216228485680573229612635573165309218642374467, rm=
24743527.370892339694285694358075293956897645351909113274151202622822031823343152}, {r =
25795872.8485870506474186417134448912922260239267345517695547531739749277916589484 ..
25795902.278913837954866806977635518216228485680573229612635573165309218642374467, rm =
.13305533041942873285002237941293511685757048481102681251855149592624481163562730e-1 ..
24850273.2191908044172729877280224590243431820608592514244738659401049812858223921}, avoid={});
Accepted {r=2.57959e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.12371e-70, 0., .43e-74]
Solution in 0.751s

Time Plot 1.743 s.
Exiting SolveHard() after 4.441
r=2.57959e+07 in
[25795872.8485870506474186417134448912922260239267345517695547531739749277916589484 ..
25795902278913837954866806977635518216228485680573229612635573165309218642374467/100000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 ..
24850273.2191908044172729877280224590243431820608592514244738659401049812858223921]: target and source on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.4581367893620784201476418232459678890859571140640573686618602853980606788777,
10510.3693005868253261847550114570798419055347488169747097599503420529230106916233,
10510.3951407618026543118853677681756205010999248765625670819771437489244884027144, none, none,
10510.3372998384311948022480110656631887015100574095580656851195830261222493841344, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none,
none, none, none, none, none]

0 --> 1  target =
[24850273.2191908044172729877280224590243431820608592514244738659401049812858223921,
1.47696881205297376885493171091216640507099718650213382672685775821473639566866246,
10510.3693038828494396496298601289580251443816293110439027243363721148648510764461]
one interval r =
23888964.972988903871843358170878332157794462984509713929509312179433
```

```
8177296054672 ..
23888994715693741781812262211108019986286975328804224521399716790800
04606401871/10000000000000000000000000000000000000000000000000000000
0000000000000000
Time Approximations 0.073.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.466972) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.03965e+07
k=-5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888994.715693741781812262211108019986286975328804224521399716790800
0046064018710, rm=
20396533.250243558247425959596656325963673488085149128323795254610298
138523330687}, {r =
23888964.972988903871843358170878332157794462984509713929509312179438
177296054672 ..
23888994.715693741781812262211108019986286975328804224521399716790800
0046064018710, rm =
.133055330419428732850022379412935116857570484811026812518551495926244
81163562730e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.03966e+07} with Delta=8e-72
Equations at solution: [.14e-70, .8e-71, -.49e-74]
Solution in 1.139s

Time Plot 3.046 s.
Exiting SolveHard() after 7.255
r=2.38890e+07 in
[23888964.972988903871843358170878332157794462984509713929509312179438
177296054672 ..
23888994715693741781812262211108019986286975328804224521399716790800
04606401871/10000000000000000000000000000000000000000000000000000000
0000000000000000]
Scattering ray (rm=2.03966e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862, none,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none,
none, none, none, none, none]

0 --> 2  target =
```

```
[25795903.799855115613167816421136950934913413279478744256818225561433990471282725,
1.856318223578679476954424130626621939871238748002263945811895082007122497556430
4,
10510.395143880065381003717929828926679920743292184252332093840475489693255428460]
one interval r =
23888966.510128753957811446979262486544023219692132003449131349488829434580905420 ..
23888996252755494636886759336945942538679604882373757474837317619667074366746491/1000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.094.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.36718) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.28604e+07   k=-4.31732e+14   scos=4.39724e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=23888996.252755494636886759336945942538679604882373757474837317619667074366746490, rm=22860406.587218420274840081630072849595003929123588140063376215202036680336637728}, {r =
23888966.510128753957811446979262486544023219692132003449131349488829434580905420 ..
23888996.252755494636886759336945942538679604882373757474837317619667074366746490, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627
30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.28605e+07} with Delta=2.0e-71
Equations at solution: [-.21e-70, -.20e-70, .57e-74]
Solution in 1.102s

Time Plot 2.416 s.
Exiting SolveHard() after 6.722
r=2.38890e+07 in
[23888966.510128753957811446979262486544023219692132003449131349488829434580905420 ..
23888996252755494636886759336945942538679604882373757474837317619667074366746491/1000000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.28605e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
```

```
10510.280466243503212699779442463518560403462952440579496003243406499338828043862, none,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none,
none, none, none, none]

1 --> 2  target =
[25795903.799855115613167816421136950934913413279478744256818225561433990471282725,
1.856318223578679476954424130626621939871238748002263945811895082007122497556430 4,
10510.395143880065381003717929828926679920743292184252332093840475489693255428460]
one interval r =
24850245.154289192585435018030677800645302009432857707504831647690330169250113834 ..
155314217178736352341082027687693133046642453429078591715155473875612262765177 9/625000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.062.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.239828) |   S  ---> P
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.47435e+07
k=2.37383e+14   scos=5.95462e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850274.748597816374573124430030901287462792548652574674424875820097962042428464, rm=
24743529.265720902408294571636111826367144442111052367213996415244067209730795537}, {r =
24850245.154289192585435018030677800645302009432857707504831647690330169250113834 ..
24850274.748597816374573124430030901287462792548652574674424875820097962042428464, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 25089000.}, avoid={});
Accepted {r=2.48503e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.27218e-70, 0., .50e-74]
Solution in 0.907s

Time Plot 1.661 s.
Exiting SolveHard() after 5.163
r=2.48503e+07 in
[24850245.154289192585435018030677800645302009432857707504831647690330169250113834 ..
155314217178736352341082027687693133046642453429078591715155473875612262765177 9/625000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 .. 25089000]: target and source
on the different branches.
Counterclockwise ray.
```

```
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405, none, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none,
none, none, none, none]

0 --> 2  target =
[25795900.395338079209419782810620237858015406692056489784607156615134366355600634,
1.856309428678043266814392150358620934816230054831584091377257835325811423045924 3,
10510.337302956679025004480167145618529294238990535370050445779121258334094178406]
one interval r =
23888963.069349248283354790317066732648095593294059036755587842761540841819471964 ..
2388899281215080305602286687243794493588194127896937677204498454038744330667171 9/1000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.077.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.36718) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.28604e+07
k=-4.31731e+14   scos=4.39724e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888992.812150803056022866872437944935881941278969376772044984540387443306671719, rm=
22860408.015023051069920923651012588348636526050794870022750504890630133589665063}, {r =
23888963.069349248283354790317066732648095593294059036755587842761540841819471964 ..
23888992.812150803056022866872437944935881941278969376772044984540387443306671719, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.28605e+07} with Delta=1.3e-71
Equations at solution: [.13e-70, .13e-70, -.70e-74]
Solution in 1.066s
```

```
Time Plot 2.525 s.
Exiting SolveHard() after 6.63
r=2.38890e+07 in
[23888963.069349248283354790317066732648095593294059036755587842761540841819471964 ..
238889928121508030560228668724379449358819412789693767720449845403874433066717 19/1000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.28605e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405, none, none,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903, none, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none,
none]

1 --> 2  target =
[25795900.395338079209419782810620237858015406692056489784607156615134366355600634,
1.856309428678043266814392150358620934816230054831584091377257835325811423045924 3,
10510.337302956679025004480167145618529294238990535370050445779121258334094178406]
one interval r =
24850241.730694626315262371655060156169158798188774563495097726107755820642979712 ..
248502713251292582485547002045078784313460030683463195197610428496250632809343 67/1000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.079.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.239829) |   S  ---> P
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.47435e+07
k=2.37384e+14   scos=5.95462e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
```

```
24850271.325129258248554700204507878431346003068346319519761042849625063280934367, rm=
24743525.024282879137257063253111191767886662705398667216257139049831713593136770}, {r =
24850241.730694626315262371655060156169158798188774563495097726107755820642979712 ..
24850271.325129258248554700204507878431346003068346319519761042849625063280934367, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 25089000.}, avoid={});
Accepted {r=2.48503e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.9897e-71, 0., .27e-74]
Solution in 1.166s

Time Plot 1.939 s.
Exiting SolveHard() after 6.003
r=2.48503e+07 in
[24850241.730694626315262371655060156169158798188774563495097726107755820642979712 ..
248502713251292582485547002045078784313460030683463195197610428496250632809343 67/100000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 .. 25089000]: target and source
on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735, none,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none,
none]

1 --> 0  target =
[23888989.431463286441758075013476009976495241419465530684772791703951580355188968,
2.632526185857223384039587002846869614086544102903100412172420916615882605874558 9,
10510.280469648065922750468374268484541854503062186366215334085180 78
```

```
1924247656144]
one interval r =
24850238.366726412891618455244035235576145536206538384771803753358216869220719093 ..
24850267961284855396516969692859822637115303766407898898534192078666490105341993/1000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.077.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.533028) |   P  <--- S
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.03965e+07
k=5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850267.961284855396516969692859822637115303766407898898534192078666490105341993, rm=
20396534.500196784303134935983362736171100538588964824987498636157387081264240983}, {r =
24850238.366726412891618455244035235576145536206538384771803753358216869220719093 ..
24850267.961284855396516969692859822637115303766407898898534192078666490105341993, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888989.431463286441758075013476009976495241419465530684772791703951580355188968}, avoid={});
Accepted {r=2.48503e+07, rm=2.03964e+07} with Delta=0
Equations at solution: [0., 0., .5e-75]
Solution in 0.969s

Time Plot 3.35 s.
Exiting SolveHard() after 7.377
r=2.48503e+07 in
[24850238.366726412891618455244035235576145536206538384771803753358216869220719093 ..
24850267961284855396516969692859822637115303766407898898534192078666490105341993/1000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03964e+07) in
[298960418182500/22468879468420441 ..
23888989.431463286441758075013476009976495241419465530684772791703951580355188968]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
```

```
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735, none,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none]

2 --> 0  target =
[23888989.431463286441758075013476009976495241419465530684772791703951580355188968,
2.632526185857223384039587002846869614086544102903100412172420916615882605874558 9,
10510.280469648065922750468374268484541854503062186366215334085180781924247656144]
one interval r =
25795867.619677233672718710490902589798807363007222318233332199455807649897518121 ..
257958970501206925825518533128317262409832018271535440678386866826020698007264 1/100000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.059.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.632821) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.28604e+07
k=4.31731e+14   scos=4.39725e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
25795897.050120692582551853312831726240983201827153544067838686682602069800726410, rm=
22860409.417933255509477580798085182428197302383990341775256420160110691348257476}, {r =
25795867.619677233672718710490902589798807363007222318233332199455807649897518121 ..
25795897.050120692582551853312831726240983201827153544067838686682602069800726410, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888989.431463286441758075013476009976495241419465530684772791703951580355188968}, avoid={});
Accepted {r=2.57959e+07, rm=2.28604e+07} with Delta=8e-72
Equations at solution: [.9e-71, .8e-71, -.29e-74]
Solution in 0.793s

Time Plot 2.716 s.
Exiting SolveHard() after 5.62
r=2.57959e+07 in
[25795867.619677233672718710490902589798807363007222318233332199455807649897518121 ..
```

```
257958970501206925825518533128317262409832018271535440678386866826020
6980072641/100000000000000000000000000000000000000000000000000000000000
0000000000000000]
Scattering ray (rm=2.28604e+07) in
[298960418182500/22468879468420441 ..
23888989.431463286441758075013476009976495241419465530684772791703951
580355188968]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none, none, none, none]

1 --> 0  target =
[23888992.505584726779415621997642067162438691016418128023667007019209856215676407,
2.632534984402235609731825352106670503718625783279793205133964743247527260430002 9,
10510.332149209078941724870613491427128929695525415338296800581465548607539118490]
one interval r =
24850241.425644461959298570828556342292613209685637581381038212700629741459234113 ..
49700542040180642720577743446912703281708583682652212123506112401422597792809 59/200000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.072.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.533028) |   P  <--- S
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.03965e+07
k=5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Counterclockwise
```

```
(Scattering) fsolve(eqs, {r=
24850271.020090321360288871723456351640854291841326106061753056200711298896404795, rm=
20396533.773037528014744673599524082948940669962029842113285550096987594209939137}, {r =
24850241.425644461959298570828556342292613209685637581381038212700629741459234113 ..
24850271.020090321360288871723456351640854291841326106061753056200711298896404795, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888992.505584726779415621997642067162438691016418128023667007019209856215676407}, avoid={});
Accepted {r=2.48503e+07, rm=2.03964e+07} with Delta=7e-72
Equations at solution: [.13e-70, .7e-71, -.41e-74]
Solution in 0.917s

Time Plot 3.248 s.
Exiting SolveHard() after 7.209
r=2.48503e+07 in
[24850241.425644461959298570828556342292613209685637581381038212700629741459234113 ..
49700542040180642720577743446912703281708583682652212123506112401422597792809 59/20000000000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03964e+07) in
[298960418182500/22468879468420441 ..
23888992.505584726779415621997642067162438691016418128023667007019209856215676407]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442, none, none,
10510.243309711722465167299850323791263507661727545284505306255219533
```

```
1652691067106, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none]

2 --> 0  target =
[23888992.505584726779415621997642067162438691016418128023667007019209856215676407,
2.6325349844022356097318253521066705037186257832797932051339647432475272604300029,
10510.332149209078941724870613491427128929695525415338296800581465548607539118490]
one interval r =
25795870.661612458008816048100464941705626671634262727124894536526558603278339321 ..
161224375574925272123721696299781502078912601681832941032050262950267200269456 3/625000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.056.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.63282) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.28604e+07
k=4.31731e+14   scos=4.39724e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
25795900.091988043539795471407965040332626016269093270565128042072042752043113008, rm=
22860408.142242371050994137542975770012421012763231553910510838440430342393447635}, {r =
25795870.661612458008816048100464941705626671634262727124894536526558603278339321 ..
25795900.091988043539795471407965040332626016269093270565128042072042752043113008, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888992.505584726779415621997642067162438691016418128023667007019209856215676407}, avoid={});
Accepted {r=2.57959e+07, rm=2.28604e+07} with Delta=1.8e-71
Equations at solution: [.18e-70, .18e-70, -.73e-74]
Solution in 0.847s

Time Plot 2.63 s.
Exiting SolveHard() after 5.478
r=2.57959e+07 in
[25795870.661612458008816048100464941705626671634262727124894536526558603278339321 ..
161224375574925272123721696299781502078912601681832941032050262950267200269456 3/625000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.28604e+07) in
[298960418182500/22468879468420441 ..
23888992.505584726779415621997642067162438691016418128023667007019209856215676407]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.
```

```
Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835, none,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none, none]

2 --> 1  target =
[24850272.854491891325264026585501935466574204438032729469169003972314267398432904,
1.476967822333829106402975511591159392522331299468469928463572634962460439425873 0,
10510.363142143218307851278306862080235851820069751649557554713206271304166408797]
one interval r =
25795872.485898932723465493525440119575329907586933130194946112744727124141284993 ..
25795901916233812803910537040455846516523994834305270243351951359980582132880041/1000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.239.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.760172) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.47435e+07
k=-2.37383e+14   scos=5.95462e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
25795901.916233812803910537040455846516523994834305270243351951359980582132880041, rm=
24743526.919055902243862721119925703593228369636971277286759273823809371840576077}, {r =
25795872.485898932723465493525440119575329907586933130194946112744727124141284993 ..
```

```
25795901.916233812803910537040455846516523994834305270243351951359980
0582132880041, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
24850272.854491891325264026585501935466574204438032729469169003972314267398432904}, avoid={});
Accepted {r=2.57959e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [.2475e-71, 0., -.24e-74]
Solution in 0.795s

Time Plot 1.903 s.
Exiting SolveHard() after 4.854
r=2.57959e+07 in
[25795872.485898932723465493525440119575329907586933130194946112744727124141284993 ..
257959019162338128039105370404558465165239948343052702433519513599800582132880041/100000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 ..
24850272.854491891325264026585501935466574204438032729469169003972314267398432904]: target and source on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835, none,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749, none, none, none, none, none, none, none, none, none,
none, none, none, none, none, none, none]
```

```
0 --> 1  target =
[24850272.854491891325264026585501935466574204438032729469169003972314267398432904,
1.476967822333829106402975511591159392522331299468469928463572634962460439425873
0,
10510.363142143218307851278306862080235851820069751649557554713206271304166408797]
one interval r =
23888964.606445871675751680198871149124303974927015118671565079874289842994852086 ..
11944497174584666184046584051055488189609798195610202172983302324857869137922981/5000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.084.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.466972) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.03965e+07
k=-5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888994.349169332368093168102110976379219596391220404345966604649715738275845962, rm=
20396533.336944275755383184011587013111974834842880250885353043006018862321447582}, {r =
23888964.606445871675751680198871149124303974927015118671565079874289842994852086 ..
23888994.349169332368093168102110976379219596391220404345966604649715738275845962, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627
30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.03966e+07} with Delta=3e-72
Equations at solution: [.5e-71, .3e-71, -.6e-75]
Solution in 1.244s

Time Plot 3.154 s.
Exiting SolveHard() after 7.503
r=2.38890e+07 in
[23888964.606445871675751680198871149124303974927015118671565079874289842994852086 ..
11944497174584666184046584051055488189609798195610202172983302324857869137922981/5000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03966e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143741
```

```
8924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749, none, none, none, none, none, none, none, none, none, none, none, none, none, none, none]

1 --> 0  target =
[23888989.064963294858729111970894888683501076586660467924668123380769484276372562,
2.6325251368883642273252814761467181826551481302273948129005259281256560590901287,
10510.274308374551431192984552171396681118779413552775477074496989412433845386004]
one interval r =
24850238.002039281126732109493901653848541850359591807518365233250253603831943387 ..
24850267596611145734046853279389904345249135980652569697075446747081695896372831/1000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.067.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.533028) |   P  <--- S
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.03965e+07
k=5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850267.596611145734046853279389904345249135980652569697075446747081695896372831, rm=
20396534.586888389570725256722427441142953873091305023146316035982318801630568349}, {r =
24850238.002039281126732109493901653848541850359591807518365233250253603831943387 ..
24850267.596611145734046853279389904345249135980652569697075446747081695896372831, rm =
```

```
.133055330419428732850022379412935116857570484811026812518551495926244811635627
30e-1 ..
23888989.064963294858729111970894888683501076586660467924668123380769484276372562
}, avoid={});
Accepted {r=2.48503e+07, rm=2.03964e+07} with Delta=6e-72
Equations at solution: [.11e-70, .6e-71, -.27e-74]
Solution in 1.083s

Time Plot 2.817 s.
Exiting SolveHard() after 6.408
r=2.48503e+07 in
[24850238.002039281126732109493901653848541850359591807518365233250253603831943387 ..
24850267596611145734046853279389904345249135980652569697075446747081695896372831/10000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03964e+07) in
[298960418182500/22468879468420441 ..
23888989.064963294858729111970894888683501076586660467924668123380769484276372562]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749, none,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685, none, none, none, none, none, none, none, none, none,
```

```
none, none, none, none, none]

2 --> 0  target =
[23888989.064963294858729111970894888683501076586660467924668123380769484276372562,
2.6325251368883642273252814761467181826551481302273948129005259281256560590901287,
10510.27430837455143119298455217139668111877941355277547707449698941243384538600 4]
one interval r =
25795867.257015114488019800954920625938902146240359676380044884700324644157043768 ..
1289794834373333259475618738609751260273902553871682778955482797870611426037372 7/500000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.052.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.632821) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.28604e+07
k=4.31731e+14   scos=4.39725e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
25795896.687466665189512374772195025205478051077433655579109655957412228520747454, rm=
22860409.570020845748004123797040669679807224912764040725723905304881959194544410}, {r =
25795867.257015114488019800954920625938902146240359676380044884700324644157043768 ..
25795896.687466665189512374772195025205478051077433655579109655957412228520747454, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888989.064963294858729111970894888683501076586660467924668123380769484276372562}, avoid={});
Accepted {r=2.57959e+07, rm=2.28604e+07} with Delta=1.7e-71
Equations at solution: [.19e-70, .17e-70, -.72e-74]
Solution in 0.721s

Time Plot 2.495 s.
Exiting SolveHard() after 5.058
r=2.57959e+07 in
[25795867.257015114488019800954920625938902146240359676380044884700324644157043768 ..
1289794834373333259475618738609751260273902553871682778955482797870611426037372 7/500000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.28604e+07) in
[298960418182500/22468879468420441 ..
23888989.064963294858729111970894888683501076586660467924668123380769484276372562]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
```

```
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749, none,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757, none, none, none, none, none, none, none, none, none, none, none, none]

2 --> 1  target =
[24850269.431013098612417336168793346654307842174812940040551374094504585980289428,
1.476958531724740943417498052240302069247650787458343239692673024680932620293967 3,
10510.305301183186374863252537160653653381983377356798419220604855056204888460917]
one interval r =
25795869.081298910026823540035715138185796189340313182434712759365873935876391196 ..
257958985117097566516608560416665916451802474047930207447412314661476915964542 39/1000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.054.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.760171) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.47435e+07
k=-2.37384e+14   scos=5.95462e+14
branch  outgoing at target, Clockwise
```

```
(Scattering) fsolve(eqs, {r=
25795898.511709756651660856041666591645180247404793020744741231466147
691596454239, rm=
24743522.677606446489867529440340394446619808455392375338537274783339
106372964790}, {r =
25795869.081298910026823540035715138185796189340313182434712759365873
935876391196 ..
25795898.511709756651660856041666591645180247404793020744741231466147
691596454239, rm =
.1330553304194287328500223794129351168575704848110268125185514959262
4481163562730e-1 ..
24850269.431013098612417336168793346654307842174812940040551374094504
585980289428}, avoid={});
Accepted {r=2.57959e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [.7423e-71, 0., .5e-75]
Solution in 0.786s

Time Plot 1.901 s.
Exiting SolveHard() after 4.761
r=2.57959e+07 in
[25795869.081298910026823540035715138185796189340313182434712759365873935876391196 ..
2579589851170975665166085604166659164518024740479302074474123146614
7691596454239/100000000000000000000000000000000000000000000000000000
00000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 ..
24850269.431013098612417336168793346654307842174812940040551374094504
585980289428]: target and source on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.26914997049867919619036826661055919547009552987210304269829965
```

```
2044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749, none,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757, none, none,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798, none, none, none, none, none, none, none, none, none,
none]

0 --> 1  target =
[24850269.43101309861241733616879334665430784217481294004055137409450458598028942 8,
1.476958531724740943417498052240302069247650787458343239692673024680932620293966 7,
10510.305301183186374863252537160653653381983377356798419220604855056204888460917]
one interval r =
23888961.165652998179489481194491782105062003321286697286084665705570691486326417 ..
11944495454275635754124958411961317098889495458536863070114786926401271062317473/5000000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.078.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.466972) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.03965e+07
k=-5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888990.908551271508249916823922634197778990917073726140229573852802542124634946, rm=
20396534.150805266948090020909158878305333639588229203068569422968135140289229571}, {r =
23888961.165652998179489481194491782105062003321286697286084665705570691486326417 ..
23888990.908551271508249916823922634197778990917073726140229573852802542124634946, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.03966e+07} with Delta=7e-72
Equations at solution: [-.10e-70, -.7e-71, .49e-74]
Solution in 1.205s

Time Plot 3.158 s.
Exiting SolveHard() after 7.931
r=2.38890e+07 in
[23888961.165652998179489481194491782105062003321286697286084665705570691486326417 ..
11944495454275635754124958411961317098889495458536863070114786926401
```

```
271062317473/50000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03966e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757, none, none,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798, none, none, none, none, none, none, none, none, none,
none]

2 --> 1  target =
[24850262.703230786449258755050024335940913287243759119970981549364199930550085514,
1.476940274031262782949409178233029376485245729461138398991518625295424402817292 7,
10510.191633491721325673156556652831945706825452642623241218944399341310368909164]
```

```
one interval r =
25795862.390633584467481284895779334802780621939676265097596423154255048060158792 ..
257958918211937121808284515123359957494555594688981578629426374007613884556285 73/1000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.05.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.76017) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.47435e+07
k=-2.37384e+14   scos=5.95461e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
25795891.821193712180828451512335995749455559468898157862942637400761388455628573, rm=
24743514.342363886011954994984202256676323439280320727527834124401366925435053647}, {r =
25795862.390633584467481284895779334802780621939676265097596423154255048060158792 ..
25795891.821193712180828451512335995749455559468898157862942637400761388455628573, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
24850262.703230786449258755050024335940913287243759119970981549364199930550085514}, avoid={});
Accepted {r=2.57959e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [.7423e-71, 0., -.1e-75]
Solution in 0.763s

Time Plot 1.997 s.
Exiting SolveHard() after 4.782
r=2.57959e+07 in
[25795862.390633584467481284895779334802780621939676265097596423154255048060158792 ..
257958918211937121808284515123359957494555594688981578629426374007613884556285 73/1000000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 ..
24850262.703230786449258755050024335940913287243759119970981549364199930550085514]: target and source on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158 11
```

```
1596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757, none, none,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798, none, none,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547, none, none, none, none, none, none, none]

0 --> 1  target =
[24850262.703230786449258755050024335940913287243759119970981549364199930550085514,
1.476940274031262782949409178233029376485245729461138398991518625295424402817292 7,
10510.191633491721325673156556652831945706825452642623241218944399341310368909164]
one interval r =
23888954.403829020662026386116701986641127945548736260331018062087938502606724616 ..
2388898414707082340708204737703207331095929556966177320501372311983093326946487 9/100000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.086.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.466972) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.03965e+07
k=-5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888984.147070823407082047377032073310959295569661773205013723119830933269464879, rm=
20396535.750140228934100301447943259655850204427400787942160872385 97
```

```
6758911440626}, {r =
23888954.403829020662026386116701986641127945548736260331018062087938502606724616 ..
23888984.147070823407082047377032073310959295569661773205013723119830933269464879, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.03966e+07} with Delta=1.0e-71
Equations at solution: [-.16e-70, -.10e-70, .28e-74]
Solution in 1.23s

Time Plot 3.041 s.
Exiting SolveHard() after 7.801
r=2.38890e+07 in
[23888954.403829020662026386116701986641127945548736260331018062087938502606724616 ..
2388898414707082340708204737703207331095929556966177320501372311983093326946 4879/10000000000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03966e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
```

```
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436, none,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798, none, none,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547, none, none, none, none, none, none, none]

0 --> 2  target =
[25795893.342154006242210885900569181077627754118842710104481931271820385360589939,
1.856291208228086723992130094043171908944807830067155600328242596439183280433 1478,
10510.217473607110852613231325453082004905889586796056437087741272906547086753046]
one interval r =
23888955.941003649134871758122059421399551704531581655570823985848059787852705279 ..
2388898568416735818079539166696781319751177432922720838297671089569011829254440 9/100000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.08.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.367179) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.28604e+07
k=-4.31730e+14   scos=4.39725e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888985.684167358180795391666967813197511774329227208382976710895690118292544409, rm=
22860410.972943515066489777210197383491016863160385005794687375358135971302157351}, {r =
23888955.941003649134871758122059421399551704531581655570823985848059787852705279 ..
23888985.684167358180795391666967813197511774329227208382976710895690118292544409, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.28605e+07} with Delta=1.5e-71
Equations at solution: [-.14e-70, -.15e-70, .79e-74]
Solution in 1.226s

Time Plot 2.288 s.
Exiting SolveHard() after 7.111
r=2.38890e+07 in
[23888955.941003649134871758122059421399551704531581655570823985848059787852705279 ..
2388898568416735818079539166696781319751177432922720838297671089569011829254440 9/100000000000000000000000000000000000000000000000000000000000000000000000000]
```

```
Scattering ray (rm=2.28605e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436, none,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709, none,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547, none, none, none, none, none, none, none]

1 --> 2  target =
[25795893.342154006242210885900569181077627754118842710104481931271820388360589939,
1.856291208228086723992130094043171908944807830067155600328242596439
```

```
1832804331478,
10510.217473607110852613231325453082004905889586796056437087741272906547086753046]
one interval r =
24850234.637969121731419204322081176607078050469696513503633393318519988523908869 ..
124251321163323986959085230147675698841575450898894929985033956207717787800187 21/500000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.066.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.23983) |   S  ---> P
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.47435e+07
k=2.37384e+14   scos=5.95462e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850264.232664797391817046029535139768315090179778985997006791241543557560037442, rm=
24743516.237222804649593611553930024285496555453184450956357912323535322652025971}, {r =
24850234.637969121731419204322081176607078050469696513503633393318519988523908869 ..
24850264.232664797391817046029535139768315090179778985997006791241543557560037442, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 25089000.}, avoid={});
Accepted {r=2.48503e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.22269e-70, 0., .67e-74]
Solution in 1.085s

Time Plot 1.92 s.
Exiting SolveHard() after 5.857
r=2.48503e+07 in
[24850234.637969121731419204322081176607078050469696513503633393318519988523908869 ..
124251321163323986959085230147675698841575450898894929985033956207717787800187 21/500000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 .. 25089000]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158 11
```

1596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709, none,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547, none, none, none, none, none, none, none]

2 --> 1 target =
[24850265.762053591998707367035934971525267294336362937733362229292081894905182929,
1.476948574973150678424900072270774720401672905597257080356978927366521980043214 4,
10510.243313007712922544404098851923307936883306160852773790306572890601929368569]
one interval r =
25795865.432578216165701796735284371596600400450195193087124021579142756508338589 ..
128979474315352368278583529240254764647377398552095942874204063491493884678 3927/500000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.055.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.76017) | P <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.47435e+07

```
k=-2.37384e+14   scos=5.95462e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
25795894.863070473655716705848050952929475479710419188574840812698298776935167854, rm=
24743518.132026153427657935751052138671826169065954230638667004848149034151671196}, {r =
25795865.432578216165701796735284371596600400450195193087124021579142756508338589 ..
25795894.863070473655716705848050952929475479710419188574840812698298776935167854, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
24850265.762053591998707367035934971525267294336362937733362229292081894905182929}, avoid={});
Accepted {r=2.57959e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.22270e-70, 0., .80e-74]
Solution in 0.837s

Time Plot 2.034 s.
Exiting SolveHard() after 4.751
r=2.57959e+07 in
[25795865.432578216165701796735284371596600400450195193087124021579142756508338589 ..
12897947431535236827858352924025476464737739855209594287420406349149388467583927/500000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 ..
24850265.762053591998707367035934971525267294336362937733362229292081894905182929]: target and source on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417681
```

```
9768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709, none,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547, none, none,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389, none, none, none, none]

0 --> 1  target =
[24850265.762053591998707367035934971525267294336362937733362229292081894905182929,
1.476948574973150678424900072270774720401672905597257080356978927366521980043214 4,
10510.243313007712922544404098851923307936883306160852773790306572890601929368569]
one interval r =
23888957.478131720926353533119832999260972433311915678924734093971795828031175567 ..
59722468053043344669467034199130496688892252778420947669395135769241943322482 07/25000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.079.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.466972) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.03965e+07
k=-5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888987.221217337867786813679652198675556901111368379067758054307696777328992828, rm=
20396535.023003025298438936691475010503319787036530189283759969972195880941806625}, {r =
23888957.478131720926353533119832999260972433311915678924734093971795828031175567 ..
23888987.221217337867786813679652198675556901111368379067758054307696
```

```
6777328992828, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627
30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.03966e+07} with Delta=9e-72
Equations at solution: [.16e-70, .9e-71, -.52e-74]
Solution in 1.168s

Time Plot 3.185 s.
Exiting SolveHard() after 7.892
r=2.38890e+07 in
[23888957.47813172092635353311983299926097243331191567892473409397179582803117556
7 ..
59722468053043344669467034199130496688892252778420947669395135769241943322482070
/25000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03966e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
```

```
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709, none,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366, none,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389, none, none, none, none]

0 --> 2  target =
[25795896.384025236462043554521143962625150202072068766655166471847306690017008087,
1.856299066265858109439961038732420613266225861568166780791938030365261704001064 3,
10510.269153088728957915240932041622063069281303487600060285492992216852289282423]
one interval r =
23888959.015296233423216277888246720032122360754969751045479295014311273994923743 ..
2388898875830375563821150496748780957320524162054735095091019320813584320256812 3/10000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.078.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.367179) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.28604e+07
k=-4.31731e+14   scos=4.39725e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888988.758303755638211504967487809573205241620547350950910193208135843202568123, rm=
22860409.697276000699351501209198687350367441866768155148821960670974082102362602}, {r =
23888959.015296233423216277888246720032122360754969751045479295014311273994923743 ..
23888988.758303755638211504967487809573205241620547350950910193208135843202568123, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.28605e+07} with Delta=1.8e-71
Equations at solution: [.18e-70, .18e-70, -.60e-74]
Solution in 1.128s

Time Plot 2.528 s.
Exiting SolveHard() after 7.349
r=2.38890e+07 in
[23888959.015296233423216277888246720032122360754969751045479295014 3
```

```
11273994923743 ..
23888988758303755638211504967487809573205241620547350950910193208135
843202568123/10000000000000000000000000000000000000000000000000000
000000000000000000]
Scattering ray (rm=2.28605e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.4581367893620784201476418232459678890859571140640573686618602
85398060678877,
10510.36930058682532618475501145707984190553474881697470975995034205
2923010691623,
10510.39514076180265431188536776817562050109992487656256708197714374
8924488402714,
10510.28046624350321269977944246351856040346295244057949600324340649
9338828043862,
10510.36313884719584037870310490843560318767696161571405820201315811
1596828075483,
10510.33729983843119480224801106566318870151005740955806568511958302
6122249384134,
10510.33214580450188010420862572621511168710980484526671327340295923
4689817843405,
10510.30529788717935851166146004846522188372805981346527537300081528
0930259875735,
10510.21747048889388355633134516173064799183671307195327565813975362
3450875733963,
10510.27430496999043214669943582338314255426296477168685756090451827
3368393987903,
10510.19163019574467350866023355115823461466098797782490184630241768
9768542682442,
10510.26914997049867919619036826661055919547009552987210304269829965
2044386240835,
10510.27430450924162391988961859334736837101260699050289046168193064
6919262954694,
10510.24330971172246516729985032379126350766172754528450530625521953
1652691067106,
10510.33113809489794039050928383230867120785727712157262355037843838
4989475607749,
10510.21646359961425315844565703156723283980093091699075595631003331
4286488234396,
10510.18546892759931749020200814673581567476188091970763763373534800
8969812620685,
10510.21130922484649961353409212373837542059929602552401035520345281
5403406093757,
10510.10279600720328866736745742636090098020355987728916087661346065
5051217727436,
10510.18546846172437709400611835600963192644801395545421255225863427
2975381010926,
10510.27329709823633588538635678513651685712798485711533910949055353
7012903531798,
10510.15447580454509166514822644094075185823429261899046103679028046
4792167775709, none,
10510.15962933478753614217473212128017617751575969879025193566508699
```

```
5871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366, none,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389,
10510.206155206755466562722350838002121935268626958867837068310101841985610568670, none, none, none]

1 --> 2  target =
[25795896.384025236462043554521143962625150202072068766655166471847306690017008087,
1.856299066265858109439961038732420613266225861568166780791938030365261704001064 3,
10510.269153088728957915240932041622063069281303487600060285492992216852289282423]
one interval r =
24850237.696896654593059252555086865019732229044228170665031526424055888933003744 ..
49700534582959499549005698617187774188521989613757258297029658626553065205789 1/200000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.067.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.239829) |   S  ---> P
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.47435e+07
k=2.37384e+14   scos=5.95462e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850267.291479749774502849308593887094260994806878629148514829313276532602894550, rm=
24743520.026876242452666784142764225520055873424484577134732050586471517126695137}, {r =
24850237.696896654593059252555086865019732229044228170665031526424055888933003744 ..
24850267.291479749774502849308593887094260994806878629148514829313276532602894550, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 25089000.}, avoid={});
Accepted {r=2.48503e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.29692e-70, 0., .62e-74]
Solution in 0.851s

Time Plot 1.962 s.
Exiting SolveHard() after 5.758
r=2.48503e+07 in
[24850237.696896654593059252555086865019732229044228170665031526424055888933003744 ..
49700534582959499549005698617187774188521989613757258297029658626553065205789 1/200000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 .. 25089000]: target and source
on the different branches.
Counterclockwise ray.
Ray outgoing at target.
```

Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709,
10510.237147976070612910869561445654930884879154264342111346367660632691133192332,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366, none,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389,
10510.206155206755466562722350838002121935268626958867837068310101841985610568670, none, none, none]

```
0 --> 2  target =
[25795900.032657306271041714959905408399445918763440377304111345037372077510764244,
1.856308491764749423379286180252072848057657146150544732184897334739662938261488 2,
10510.331141213144183682435342856894055031568320924221851460967162386148861618445]
one interval r =
23888962.702804792062994360815060768354472027818557223137547823708596279399186144 ..
119444962228124847314038193891870467045178077743702913452162558195520783228415 23/500000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.084.

Try code 11
I search for an scattering ray on opposite branches with 0<sv<1
(0.36718) |   S  ---> P
rGuessMin=2.38890e+07   rGuessMax=2.38890e+07   rmGuess=2.28604e+07
k=-4.31731e+14   scos=4.39724e+14
branch  outgoing at target, Clockwise
(Scattering) fsolve(eqs, {r=
23888992.445624969462807638778374093409035615548740582690432511639104156645683046, rm=
22860408.167124546519654373550892665144929879560962333397609532498345093587873799}, {r =
23888962.702804792062994360815060768354472027818557223137547823708596279399186144 ..
23888992.445624969462807638778374093409035615548740582690432511639104156645683046, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 24089000.}, avoid={});
Accepted {r=2.38890e+07, rm=2.28605e+07} with Delta=4e-72
Equations at solution: [-.5e-71, -.4e-71, .26e-74]
Solution in 1.176s

Time Plot 2.488 s.
Exiting SolveHard() after 7.213
r=2.38890e+07 in
[23888962.702804792062994360815060768354472027818557223137547823708596279399186144 ..
119444962228124847314038193891870467045178077743702913452162558195520783228415 23/500000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.28605e+07) in
[298960418182500/22468879468420441 .. 24089000]: target and source
on the different branches.
Clockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143744
```

8924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709,
10510.237147976070612910869561445654930884879154264342111346367660632691133192332,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366, none,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389,
10510.206155206755466562722350838002121935268626958867837068310101841985610568670, none, none,
10510.268143235923361803876779297199167447349721035242514349969741594979357082062]

1 --> 2 target =
[25795900.032657306271041714959905408399445918763440377304111345037372077510764244,
1.856308491764749423379286180252072848057657146150544732184897334739662938261488 2,

```
10510.331141213144183682435342856894055031568320924221851460967162386148861618445]
one interval r =
24850241.365981199580869141298968300537653745066457496440293413829932366235069164 ..
124251354802146274521242827242256868568370337431091887999763986788040516604187 27/500000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.064.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.239829) |   S  ---> P
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.47435e+07
k=2.37384e+14   scos=5.95462e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850270.960429254904248565448451373713674067486218377599952797357608103320837454, rm=
24743524.57244522382720072140810521049044724167672709029761354687726465048772178 3}, {r =
24850241.365981199580869141298968300537653745066457496440293413829932366235069164 ..
24850270.960429254904248565448451373713674067486218377599952797357608103320837454, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 .. 25089000.}, avoid={});
Accepted {r=2.48503e+07, rm=2.47435e+07} with Delta=0
Equations at solution: [-.9897e-71, 0., .35e-74]
Solution in 0.839s

Time Plot 1.92 s.
Exiting SolveHard() after 5.646
r=2.48503e+07 in
[24850241.365981199580869141298968300537653745066457496440293413829932366235069164 ..
124251354802146274521242827242256868568370337431091887999763986788040516604187 27/500000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.47435e+07) in
[298960418182500/22468879468420441 .. 25089000]: target and source
on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
```

10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709,
10510.237147976070612910869561445654930884879154264342111346367660632691133192332,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366,
10510.299136139742341903099012793255153018895791128393265891321555549917550385094,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389,
10510.206155206755466562722350838002121935268626958867837068310101841985610568670, none, none,
10510.268143235923361803876779297199167447349721035242514349969741594979357082062]

1 --> 0 target =
[23888989.064935887458805150578028018979347329468236391407713929141531200106547427,
2.632525136809920840883118370193127663554505490826022800883496847229889557285011 0,
10510.274307913802622838223387483739826214224754916607510828122809773709735235713]
one interval r =

```
24850238.002012009297443899059149613849245370866516344156368923684875924661464387 ..
155314172478649218178009426833046224700447973065961349281936632226944970534298 1/625000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.074.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.533028) |   P  <--- S
rGuessMin=2.48502e+07   rGuessMax=2.48503e+07   rmGuess=2.03965e+07
k=5.43203e+14   scos=2.39475e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
24850267.596583874908481508293287395952071675690553815885109861156311195285487696, rm=
20396534.586894872485811451524925439493705899049058916528119027149202747567960502}, {r =
24850238.002012009297443899059149613849245370866516344156368923684875924661464387 ..
24850267.596583874908481508293287395952071675690553815885109861156311195285487696, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888989.064935887458805150578028018979347329468236391407713929141531200106547427}, avoid={});
Accepted {r=2.48503e+07, rm=2.03964e+07} with Delta=5e-72
Equations at solution: [-.9e-71, -.5e-71, .14e-74]
Solution in 1.01s

Time Plot 3.201 s.
Exiting SolveHard() after 7.003
r=2.48503e+07 in
[24850238.002012009297443899059149613849245370866516344156368923684875924661464387 ..
155314172478649218178009426833046224700447973065961349281936632226944970534298 1/625000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.03964e+07) in
[298960418182500/22468879468420441 ..
23888989.064935887458805150578028018979347329468236391407713929141531200106547427]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
```

10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709,
10510.237147976070612910869561445654930884879154264342111346367660632691133192332,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366,
10510.299136139742341903099012793255153018895791128393265891321555549917550385094,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389,
10510.206155206755466562722350838002121935268626958867837068310101841985610568670,
10510.185468466850910647076271951105011015295307456972624297787224601501502034101731 none,
10510.268143235923361803876779297199167447349721035242514349969741594979357082062]

2 --> 0 target =
[23888989.064935887458805150578028018979347329468236391407713929141531200106547427,
2.632525136809920840883118370193127663554505490826022800883496847229889557285011 0,
10510.274307913802622838223387483739826214224754916607510828122809777

```
3709735235713]
one interval r =
25795867.256987994094512439931682951906269072176398293285115926555982641207403225 ..
257958966874395454011197610631778148728669542597513977097617402764637048685795247/100000000000000000000000000000000000000000000000000000000000000000000000000000
Time Approximations 0.052.

Try code 12
I search for an scattering ray on opposite branches with 0<sv<1
(0.632821) |   P  <--- S
rGuessMin=2.57959e+07   rGuessMax=2.57959e+07   rmGuess=2.28604e+07
k=4.31731e+14   scos=4.39725e+14
branch  outgoing at target, Counterclockwise
(Scattering) fsolve(eqs, {r=
25795896.687439545401119761063177814872866954259751397709617402764637048685795247, rm=
22860409.570032219065271388629815366251454950533331304606733892783149264792137530}, {r =
25795867.256987994094512439931682951906269072176398293285115926555982641207403225 ..
25795896.687439545401119761063177814872866954259751397709617402764637048685795247, rm =
.133055330419428732850022379412935116857570484811026812518551495926244811635627 30e-1 ..
23888989.064935887458805150578028018979347329468236391407713929141531200106547427}, avoid={});
Accepted {r=2.57959e+07, rm=2.28604e+07} with Delta=5e-72
Equations at solution: [.5e-71, .5e-71, -.20e-74]
Solution in 0.738s

Time Plot 2.586 s.
Exiting SolveHard() after 5.221
r=2.57959e+07 in
[25795867.256987994094512439931682951906269072176398293285115926555982641207403225 ..
257958966874395454011197610631778148728669542597513977097617402764637048685795247/100000000000000000000000000000000000000000000000000000000000000000000000000000]
Scattering ray (rm=2.28604e+07) in
[298960418182500/22468879468420441 ..
23888989.064935887458805150578028018979347329468236391407713929141531200106547427]: target and source on the different branches.
Counterclockwise ray.
Ray outgoing at target.
Solve Side.

Tau
[10510.458136789362078420147641823245967889085957114064057368661860285398060678877,
10510.369300586825326184755011457079841905534748816974709759950342052923010691623,
10510.395140761802654311885367768175620501099924876562567081977143748924488402714,
10510.280466243503212699779442463518560403462952440579496003243406499338828043862,
```

```
10510.363138847195840378703104908435603187676961615714058202013158111596828075483,
10510.337299838431194802248011065663188701510057409558065685119583026122249384134,
10510.332145804501880104208625726215111687109804845266713273402959234689817843405,
10510.305297887179358511661460048465221883728059813465275373000815280930259875735,
10510.217470488893883556331345161730647991836713071953275658139753623450875733963,
10510.274304969990432146699435823383142554262964771686857560904518273368393987903,
10510.191630195744673508660233551158234614660987977824901846302417689768542682442,
10510.269149970498679196190368266610559195470095529872103042698299652044386240835,
10510.274304509241623919889618593347368371012606990502890461681930646919262954694,
10510.243309711722465167299850323791263507661727545284505306255219531652691067106,
10510.331138094897940390509283832308671207857277121572623550378438384989475607749,
10510.216463599614253158445657031567232839800930916990755956310033314286488234396,
10510.185468927599317490202008146735815674761880919707637633735348008969812620685,
10510.211309224846499613534092123738375420599296025524010355203452815403406093757,
10510.102796007203288667367457426360900980203559877289160876613460655051217727436,
10510.185468461724377094006118356009631926448013955454212552258634272975381010926,
10510.273297098236335885386356785136516857127984857115339109490553537012903531798,
10510.154475804545091665148226440940751858234292618990461036790280464792167775709,
10510.237147976070612910869561445654930884879154264342111346367660632691133192332,
10510.159629334787536142174732121280176177515759698790251935665086995871307653547,
10510.154475478159472862128332712674359405553020449648954059824628127722428715366,
10510.299136139742341903099012793255153018895791128393265891321555549917550385094,
10510.211308883506859153864938589789302105312912485695546327089817730357756822389,
10510.206155206755466562722350838002121935268626958867837068310101841985610568670,
10510.185468466850910647076271951105011015295307456972624297787224600150203410173,
10510.211308764098399222591598939240094426256382811202692218579994334398538168300,
10510.268143235923361803876779297199167447349721035242514349969741594979357082062]

Cascade time 187.985
```

[**(72)**

10510.458136789362078420147641823245967889085957114064057368661860285398060678877,

10510.369300586825326184755011457079841905534748816974709759950342052923010691623,

10510.395140761802654311885367768175620501099924876562567081977143748924488402714,

10510.280466243503212699779442463518560403462952440579496003243406499338828043862,

10510.363138847195840378703104908435603187676961615714058202013158111596828075483,

10510.337299838431194802248011065663188701510057409558065685119583026122249384134,

10510.332145804501880104208625726215111687109804845266713273402959234689817843405,

10510.305297887179358511661460048465221883728059813465275373000815280930259875735,

10510.217470488893883556331345161730647991836713071953275658139753623450875733963,

10510.274304969990432146699435823383142554262964771686857560904518273368393987903,

10510.191630195744673508660233551158234614660987977824901846302417689768542682442,

10510.269149970498679196190368266610559195470095529872103042698299652044386240835,

10510.274304509241623919889618593347368371012606990502890461681930646919262954694,

10510.243309711722465167299850323791263507661727545284505306255219531652691067106,

10510.331138094897940390509283832308671207857277121572623550378438384989475607749,

10510.216463599614253158445657031567232839800930916990755956310033314286488234396,

10510.185468927599317490202008146735815674761880919707637633735348008969812620685,

10510.211309224846499613534092123738375420599296025524010355203452815403406093757,

10510.102796007203288667367457426360900980203559877289160876613460655051217710\

```
    27436,
    10510.18546846172437709400611835600963192644801395545421255225863427297538\
    10926,
    10510.27329709823633588538635678513651685712798485711533910949055353701290\
    31798,
    10510.15447580454509166514822644094075185823429261899046103679028046479216\
    75709,
    10510.23714797607061291086956144565493088487915426434211134636766063269113\
    92332,
    10510.15962933478753614217473212128017617751575969879025193566508699587130\
    53547,
    10510.15447547815947286212833271267435940555302044964895405982462812772242\
    15366,
    10510.29913613974234190309901279325515301889579112839326589132155554991755\
    85094,
    10510.21130888350685915386493858978930210531291248569554632708981773035775\
    22389,
    10510.20615520675546656272235083800212193526862695886783706831010184198561\
    68670,
    10510.18546846685091064707627195110501101529530745697262429778722460015020\
    10173,
    10510.21130876409839922259159893924009442625638281120269221857999433439853\
    68300,
    10510.26814323592336180387677929719916744734972103524251434996974159497935\
    82062 ]
```

```
> Hints40:= ListHints:
  Tau40:=ListTau:
  P40:=ListP:
```

```
> PurifyTau := proc(Tau)
    local T, n, Tr:
    T:= Tau[1..3]:
    Tr:=CreateList(31, 0):

    for n from 0 to nops(Tau)-1 do
      Tr[n+1] := -Tau[n+1] +T[1+ (n mod 3)]:
```

```
    end do:
    Tr[4..MaxSignals];
  end:
```

```
> FirstSignals:=PurifyTau(Tau40);
```

FirstSignals := **(73)**

```
[
0.17767054585886572036819935972740748562300467348456136541845378605923263501\
5,
0.00616173962948580605190654864423871785778720126065155793718394132618261614\
0,
0.05784092337145950963735670251243179958986746700450139685756072280223901858\
0,
0.12599098486019831593901609703085620197615226879734409525890105070824283547\
2,
0.06400269964596767309355140861462002180668900350943438694952677199275081588\
8,
0.17767027290877075555402260644497250926321180460929142383739012547361266875\
1,
0.18383181937164627344820599986282533482299234237719980775734201202966669097\
4,
0.17767039108065267609477790592160729087376083914980791364792436315446800918\
1,
0.12599079130397511569499950156506130562982934669046403927884409688010216187\
9,
0.18383228012045450025802322989859951807335012356116690697992963847879772418\
3,
0.12599087510286101745516113328857839787302127169020445369512252127031962451\
7,
0.06400266690471392137608393586694929324264775498994353159870536393501279496\
5,
0.24167318974782526170198479167873504928502619707330141235182697111157244448\
1,
0.18383165922600869455300331034402623077286789726707212621499404395319807093\
8,
0.18383153695615469835127564443724508050062885103855672677369093352108230895\
7,
0.35534078215878975278018439688506690888239723677489649204839963034684295144\
1,
```

```
0.18383212510094909074889310107020997908673486152049720769170777994762968069\
7,
0.12184366356631842649901098303910364397194001944722797248659021191158487091\
6,
0.30366098481698675499941538230521603085166449507359633187157982060589290316\
8,
0.13215261075471327388545001142491102065559455263259841358268142023187749929\
1,
0.23551142701511816971063564689544432358416517777231514631205675305318074916\
7,
0.30366131120260555801930911057160848353293666441510330883723215767563196351\
1,
0.07016444708298428165599866382468888663895768858144386862878650300546030652\
9,
0.18383187829579515802042917838631839578701239086702075488732601856673158032\
5,
0.25198158260661185742529098524384595381733015519622030035175844341245011020\
7,
0.18383211997441553767873950597483089023944136000208546216311745277280728145\
0,
0.18383199770425508929376882893552607484354206535987486339714941452595023441\
4,
0.18999355343871661627086252604680044173623607882154301869211869041870359681\
5 ]
```

```
> ObsSignals:= PurifyTau(Tau40);
```

ObsSignals := **(74)**

```
[
0.17767054585886572036819935972740748562300467348456136541845378605923263501\
5,
0.00616173962948580605190654864423871785778720126065155793718394132618261614\
0,
0.05784092337145950963735670251243179958986746700450139685756072280223901858\
0,
0.12599098486019831593901609703085620197615226879734409525890105070824283547\
2,
0.06400269964596767309355140861462002180668900350943438694952677199275081588\
8,
0.17767027290877075555402260644497250926321180460929142383739012547361266875\
```

1,
0.18383181937164627344820599986282533482299234237719980775734201202966669097\
4,
0.17767039108065267609477790592160729087376083914980791364792436315446800918\
1,
0.12599079130397511569499950156506130562982934669046403927884409688010216187\
9,
0.18383228012045450025802322989859951807335012356116690697992963847879772418\
3,
0.12599087510286101745516113328857839787302127169020445369512252127031962451\
7,
0.06400266690471392137608393586694929324264775498994353159870536393501279496\
5,
0.24167318974782526170198479167873504928502619707330141235182697111157244448\
1,
0.18383165922600869455300331034402623077286789726707212621499404395319807093\
8,
0.18383153695615469835127564443724508050062885103855672677369093352108230895\
7,
0.35534078215878975278018439688506690888239723677489649204839963034684295144\
1,
0.18383212510094909074889310107020997908673486152049720769170777994762968069\
7,
0.12184366356631842649901098303910364397194001944722797248659021191158487091\
6,
0.30366098481698675499941538230521603085166449507359633187157982060589290316\
8,
0.13215261075471327388545001142491102065559455263259841358268142023187749929\
1,
0.23551142701511816971063564689544432358416517777231514631205675305318074916\
7,
0.30366131120260555801930911057160848353293666441510330883723215767563196351\
1,
0.07016444708298428165599866382468888663895768858144386862878650300546030652\
9,
0.18383187829579515802042917838631839578701239086702075488732601856673158032\
5,
0.25198158260661185742529098524384595381733015519622030035175844341245011020\

```
    7,
    0.183832119974415537678739505974830890239441360002085462163117452772807281 45\
    0,
    0.183831997704255089293768828935526074843542065359874863397149414525950234 41\
    4,
    0.189993553438716616270862526046800441736236078821543018692118690418703596 81\
    5 ]
```

```
> te0in, t0in, tau0in, rm0v, rp0v := op(Sat0):


  te0in := te0in -TeOffset;
  t0in := t0in -TOffset;
  tau0in := tau0in -TauOffset;

  ComputeSat0():
```

$$te0in := -2.10895765900280812689084414296181633395083177467878627014343044840622126927043 10$$

$$t0in := -12010.45814019397412810699008640310343641626781186985297188547521331685426881 7156$$

$$tau0in := -12010.45813678936207842014764182324596788908595711406405736866186028539806067 8877 \quad (75)$$

```
> Sat0Norm:=[te0in, t0in, tau0in, rm0v, rp0v];
```

$$Sat0Norm := [-2.10895765900280812689084414296181633395083177467878627014343044840622126927043 10, -12010.45814019397412810699008640310343641626781186985297188547521331685426881 7156, -12010.45813678936207842014764182324596788908595711406405736866186028539806067 8877, 23289000, 24089000] \quad (76)$$

```
> rSat(BaseSat, BaseR, BaseBranch);
  teSat(BaseSat, BaseR, BaseBranch);
  tSat(BaseSat, BaseR, BaseBranch);
```

$$23889000$$

$$0.$$

$$-9.24 \times 10^{-76} \quad (77)$$

```
> te1in, t1in, tau1in, rm1v, rp1v := op(Sat1):


  te1in := te1in -TeOffset;
  t1in := t1in -TOffset;
  tau1in := tau1in -TauOffset;

  ComputeSat1():
```

$$te1in := -3.156155210199405873045058604054983962016554907803821543801745312508826738\backslash 0325006$$

$$t1in := -12110.458140193974128106990086403103436416267811869852971885475213316854268817156$$

$$tau1in := -12110.458136789362078420147641823245967889085957114064057368661860285398060678877 \quad (78)$$

```
> Sat1Norm:=[te1in, t1in, tau1in, rm1v, rp1v];
```

$$Sat1Norm := [-3.156155210199405873045058604054983962016554907803821543801745312508826738\backslash 0325006, -12110.458140193974128106990086403103436416267811869852971885475213316854268817156, -12110.458136789362078420147641823245967889085957114064057368661860285398060678877, 24289000, 25089000] \quad (79)$$

```
> te2in, t2in, tau2in, rm2v, rp2v := op(Sat2):


  te2in := te2in -TeOffset;
  t2in := t2in -TOffset;
  tau2in := tau2in -TauOffset;

  ComputeSat2():
```

$$te2in := -2.632556434601106999967951373508400147983693341241303906972587880457524003\backslash 6514658$$

$$t2in := -11910.458140193974128106990086403103436416267811869852971885475213316854268817156$$

$$tau2in := \quad (80)$$

```
    −11910.458136789362078420147641823245967889085957114064057368661860285398\
    0678877
> Sat2Norm:=[te2in, t2in, tau2in, rm2v, rp2v];
```

$$Sat2Norm := \left[-2.6325564346011069999679513735084001479836933412413039069725878804575240036514658,\ -11910.458140193974128106990086403103436416267811869852971885475213316854268817156,\ -11910.458136789362078420147641823245967889085957114064057368661860285398\ 0678877,\ 25289000,\ 26089000 \right] \tag{81}$$

```
> ProducePlots:= true;
  DebugOn := false;
  DebugTimeOn := true;
  DebugFlowControlOn := false;
  DebugSolutionsOn := false;
```

$$ProducePlots := true$$
$$DebugOn := false$$
$$DebugTimeOn := true$$
$$DebugFlowControlOn := false$$
$$DebugSolutionsOn := false \tag{82}$$

```
> #ListTau:=CreateList(31, 0);
  #ListP:=CreateList(31, []);
  ##ListHints:=CreateList(31, none);

  #Cascade(BaseSat, BaseR, BaseBranch);
```

```
> ProducePlots:= false;
```

$$ProducePlots := false \tag{83}$$

```
> #[xS(BaseSat, BaseBranch),yS(BaseSat, BaseBranch), tS(BaseSat,
  BaseBranch) ]:
  #subs(r=BaseR, %):
  #evalf(%):
  #P0:=%;
```

```
> Rumor:= proc(x, si, n:= 1)
    if type(x, list) then
      return map(Rumor, x, si, n):
    else
      x+Sample(Normal(0,si), n)[1]:
      Q(%):
    end if:
  end:

> FitFunction:= proc(s1, s2)
    local n, k, P:
    P:= 0.0:
    n := nops(s1):
    printf("counts: %d, %d\n ", n, nops(s2));
    for k from 1 to n do
      P:= P + (s1[n]-s2[n])^2:
    end do:
    P:
  end:

> Shake:=proc(si:= 10^(-10))
   global te0in, t0in, tau0in, rm0v, rp0v:
   global te1in, t1in, tau1in, rm1v, rp1v:
   global te2in, t2in, tau2in, rm2v, rp2v:
   global H, BaseR, MaxT, ListTau, ListP, BaseSat, BaseBranch,
  ObsSignals, ListHints, HintsOriginal:
   local  NewSignals:

   #ListHints:= HintsOriginal:
   ListHints:=CreateList(31, none):
   ListTau:=CreateList(31, 0):
   ListP:=CreateList(31, []):
   #
   Rumor(BaseROriginal, si):
   BaseR:=%:

   Rumor(Sat0, si):
   te0in, t0in, tau0in, rm0v, rp0v := op(%):
   te0in := te0in -TeOffset;
   t0in := t0in -TOffset;
   tau0in := tau0in -TauOffset;
   ComputeSat0():

   Rumor(Sat1, si):
   te1in, t1in, tau1in, rm1v, rp1v := op(%):
   te1in := te1in -TeOffset;
   t1in := t1in -TOffset;
   tau1in := tau1in -TauOffset;
   ComputeSat1():

   Rumor(Sat2, si):
   te2in, t2in, tau2in, rm2v, rp2v := op(%):
   te2in := te2in -TeOffset;
   t2in := t2in -TOffset;
   tau2in := tau2in -TauOffset;
```

```
ComputeSat2():

MaxT:=1:
Cascade(BaseSat, BaseR, BaseBranch):
NewSignals:= PurifyTau(ListTau):
FitFunction(NewSignals, ObsSignals):
end:
```

```
> ObsSignals;
```

[0.17767054585886572036819935972740748562300467348456136541845378605923263501 5, **(84)**

0.00616173962948580605190654864423871785778720126065155793718394132618261614\

0,

0.05784092337145950963735670251243179958986746700450139685756072280223901858\

0,

0.12599098486019831593901609703085620197615226879734409525890105070824283547\

2,

0.06400269964596767309355140861462002180668900350943438694952677199275081588\

8,

0.17767027290877075555402260644497250926321180460929142383739012547361266875\

1,

0.18383181937164627344820599986282533482299234237719980775734201202966669097\

4,

0.17767039108065267609477790592160729087376083914980791364792436315446800918\

1,

0.12599079130397511569499950156506130562982934669046403927884409688010216187\

9,

0.18383228012045450025802322989859951807335012356116690697992963847879772418\

3,

0.12599087510286101745516113328857839787302127169020445369512252127031962451\

7,

0.06400266690471392137608393586694929324264775498994353159870536393501279496\

5,

0.24167318974782526170198479167873504928502619707330141235182697111157244448\

1,

0.18383165922600869455300331034402623077286789726707212621499404395319807093\

8,

0.18383153695615469835127564443724508050062885103855672677369093352108230895\

7,

0.35534078215878975278018439688506690888239723677489649204839963034684295144\

```
    1,
    0.183832125100949090748893101070209979086734861520497207691707779947629680691\
    7,
    0.121843663566318426499010983039103643971940019447227972486590211911584870916\
    6,
    0.303660984816986754999415382305216030851664495073596331871579820605892903161\
    8,
    0.132152610754713273885450011424911020655594552632598413582681420231877499291\
    1,
    0.235511427015118169710635646895444323584165177772315146312056753053180749161\
    7,
    0.303661311202605558019309110571608483532936664415103308837232157675631963511\
    1,
    0.070164447082984281655998663824688886638957688581443868628786503005460306521\
    9,
    0.183831878295795158020429178386318395787012390867020754887326018566731580321\
    5,
    0.251981582606611857425290985243845953817330155196220300351758443412450110201\
    7,
    0.183832119974415537678739505974830890239441360002085462163117452772807281451\
    0,
    0.183831997704255089293768828935526074843542065359874863397149414525950234411\
    4,
    0.189993553438716616270862526046800441736236078821543018692118690418703596811\
    5]
```

```
> # I shake the parameters by 10^(-10), the Fit function should
  respond by 10^(-20)


  Shake();

Exiting SolveHard() after 3.729
Exiting SolveHard() after 2.594
Exiting SolveHard() after 2.886
Exiting SolveHard() after 4.609
Exiting SolveHard() after 4.634
Exiting SolveHard() after 3.83
Exiting SolveHard() after 4.785
Exiting SolveHard() after 3.438
Exiting SolveHard() after 3.686
Exiting SolveHard() after 2.815
Exiting SolveHard() after 3.735
Exiting SolveHard() after 2.517
```

```
Exiting SolveHard() after 2.88
Exiting SolveHard() after 4.837
Exiting SolveHard() after 3.805
Exiting SolveHard() after 2.815
Exiting SolveHard() after 2.612
Exiting SolveHard() after 4.547
Exiting SolveHard() after 2.9
Exiting SolveHard() after 4.676
Exiting SolveHard() after 4.456
Exiting SolveHard() after 3.726
Exiting SolveHard() after 2.895
Exiting SolveHard() after 4.658
Exiting SolveHard() after 4.458
Exiting SolveHard() after 3.778
Exiting SolveHard() after 4.777
Exiting SolveHard() after 3.838
Exiting SolveHard() after 3.517
Exiting SolveHard() after 2.858
Cascade time 111.374
counts: 28, 28
```

$$6.990564831488567040405360521905944403552249880337277433557073252296819667856937 \times 10^{-21} \quad \textbf{(85)}$$

```
> Signals:=PurifyTau(ListTau):
>
  SMax:=0:
  for n from 1 to nops(ObsSignals) do
    if SMax < abs(ObsSignals[n]-Signals[n]) then
     SMax:= abs(ObsSignals[n]-Signals[n]):
    end if:
  end do:
  SMax;
```

$$3.237673422683712716061390478613121143338260748250780242693265792405 7 \times 10^{-11} \quad \textbf{(86)}$$

```
> Num:=0:

> myF := proc(r0, rm0, rp0, te1, t1, rm1, rp1, te2, t2, rm2, rp2)
    global te0in, t0in, tau0in, rm0v, rp0v:
    global te1in, t1in, tau1in, rm1v, rp1v:
    global te2in, t2in, tau2in, rm2v, rp2v:
    global ObsSignals, ListP, ListTau, ListHints, Num:
    local NewSignals, teOff, tOff, res:

  if not type(evalf(r0), float) then
     return evaln(myF(r0, rm0, rp0, te1, t1, rm1, rp1, te2, t2, rm2,
  rp2)):
  end if:
  Num:= Num+1:
  printf("Evaluate %d:\n", Num);
  printf(" r0=%a rm0=%a rp0=%a \n", r0, rm0, rp0):
  printf(" te1=%a t1=%a rm1=%a rp1=%a \n", te1, t1, rm1, rp1):
  printf(" te2=%a t2=%a rm2=%a rp2=%a \n", te2, t2, rm2, rp2):
    #ListHints:= HintsOriginal:
    ListHints:=CreateList(31, none):
```

```
  ListTau:=CreateList(31, 0):
  ListP:=CreateList(31, []):
#printf("Step0\n");

  te0in, t0in, rm0v, rp0v := 0, 0, rm0, rp0:
  tau0in := 0:
  ComputeSat0():
#printf("Sat 0\n");

  tSat(BaseSatOriginal, r0, BaseBranchOriginal):
  simplify(%):
  tOff:=%:
  teSat(BaseSatOriginal, r0, BaseBranchOriginal):
  simplify(%):
  teOff:=%:

  te0in:= -teOff:
  t0in:= -tOff:
#printf("Offsets\n");

  te1in, t1in, rm1v, rp1v := te1, t1, rm1, rp1:
  tau1in := 0:
  ComputeSat1():
#printf("Sat 1\n");

  te2in, t2in, rm2v, rp2v := te2, t2, rm2, rp2:
  tau2in := 0:
  ComputeSat2():
#printf("Sat 2\n");

#printf("Cascade\n"):
  Cascade(BaseSatOriginal, r0, BaseBranchOriginal):      #
BaseBranch :=0
  if % = "NotFound" then
    return 1:
  else
    NewSignals:= PurifyTau(ListTau):
    res := FitFunction(NewSignals, ObsSignals):
    printf("chi^2 = %a\n", res);
    return res:
  end if:
#printf("End Cascade\n"):
end:
```

```
> Sat0Norm;
```

$$\left[-2.108957659002808126890844142961816333950831774678786270143430448406221269\backslash 2704310,\ -12010.458140193974128106990086403103436416267811869852971885475213316854268817156,\ -12010.458136789362078420147641823245967889085957114064057368661860285398060678877, 23289000, 24089000\right] \quad (87)$$

```
> ste0in, st0in, stau0in, srm0v, srp0v := op(Sat0Norm);
  ste1in, st1in, stau1in, srm1v, srp1v := op(Sat1Norm);
  ste2in, st2in, stau2in, srm2v, srp2v := op(Sat2Norm);
```

$$ste0in, st0in, stau0in, srm0v, srp0v :=$$
$$-2.108957659002808126890844142961816333950831774678786270143430448406221269\backslash 2704310,$$
$$-12010.458140193974128106990086403103436416267811869852971885475213316854268\backslash 8817156,$$
$$-12010.458136789362078420147641823245967889085957114064057368661860285398060\backslash 678877, 23289000, 24089000$$
$$ste1in, st1in, stau1in, srm1v, srp1v :=$$
$$-3.156155210199405873045058604054983962016554907803821543801745312508826738\backslash 0325006,$$
$$-12110.458140193974128106990086403103436416267811869852971885475213316854268\backslash 8817156,$$
$$-12110.458136789362078420147641823245967889085957114064057368661860285398060\backslash 678877, 24289000, 25089000$$
$$ste2in, st2in, stau2in, srm2v, srp2v := \quad \textbf{(88)}$$
$$-2.632556434601106999967951373508400147983693341241303906972587880457524003\backslash 6514658,$$
$$-11910.458140193974128106990086403103436416267811869852971885475213316854268\backslash 8817156,$$
$$-11910.458136789362078420147641823245967889085957114064057368661860285398060\backslash 678877, 25289000, 26089000$$

```
> #printlevel:=4;
> # This should be closed to a solution

  myF(BaseROriginal, srm0v, srp0v, ste1in, st1in, srm1v, srp1v,
  ste2in, st2in, srm2v, srp2v);
Evaluate 1:
 r0=23889000 rm0=23289000 rp0=24089000
 te1=
-3.156155210199405873045058604054983962016554907803821543801745312508826738
0325006 t1=
-12110.458140193974128106990086403103436416267811869852971885475213316854268
8817156 rm1=24289000 rp1=25089000
 te2=
-2.632556434601106999967951373508400147983693341241303906972587880457524003
6514658 t2=
-11910.458140193974128106990086403103436416267811869852971885475213316854268
8817156 rm2=25289000 rp2=26089000
```

```
Exiting SolveHard() after 4.014
Exiting SolveHard() after 2.947
Exiting SolveHard() after 2.555
Exiting SolveHard() after 4.678
Exiting SolveHard() after 4.72
Exiting SolveHard() after 3.778
Exiting SolveHard() after 4.768
Exiting SolveHard() after 3.494
Exiting SolveHard() after 3.807
Exiting SolveHard() after 2.823
Exiting SolveHard() after 4.023
Exiting SolveHard() after 3.084
Exiting SolveHard() after 3.07
Exiting SolveHard() after 5
Exiting SolveHard() after 3.671
Exiting SolveHard() after 2.977
Exiting SolveHard() after 2.984
Exiting SolveHard() after 4.85
Exiting SolveHard() after 3.07
Exiting SolveHard() after 4.566
Exiting SolveHard() after 4.789
Exiting SolveHard() after 3.951
Exiting SolveHard() after 2.998
Exiting SolveHard() after 4.544
Exiting SolveHard() after 4.729
Exiting SolveHard() after 4.022
Exiting SolveHard() after 5.027
Exiting SolveHard() after 3.997
Exiting SolveHard() after 4.036
Exiting SolveHard() after 2.637
Cascade time 115.642
counts: 28, 28
 chi^2 = 0.
```

$$0. \tag{89}$$

```
> tSat(BaseSatOriginal, BaseROriginal, BaseBranchOriginal):
    simplify(%):
  %;
```

$$-2.24 \times 10^{-76} \tag{90}$$

```
> teSat(BaseSatOriginal, BaseROriginal, BaseBranchOriginal):
    simplify(%):
  %;
```

$$0. \tag{91}$$

```
> ListTau;
```

$$[\tag{92}$$

12010.458136789362078420147641823245967889085957114064057368661860285398060678877,

12110.369300586825326184755011457079841905534748816974709759950342052923010691623,

11910.395140761802654311885367768175620501099924876562567081977143748924488

02714,
12010.280466243503212699779442463518560403462952440579496003243406499338828 0\
43862,
12110.363138847195840378703104908435603187676961615714058202013158111596828 0\
75482,
11910.337299838431194802248011065663188701510057409558065685119583026122249 3\
84134,
12010.332145804501880104208625726215111687109804845266713273402959234689817 8\
43405,
12110.305297887179358511661460048465221883728059813465275373000815280930259 8\
75734,
11910.217470488893883556331345161730647991836713071953275658139753623450875 7\
33963,
12010.274304969990432146699435823383142554262964771686857560904518273368393 9\
87887,
12110.191630195744673508660233551158234614660987977824901846302417689768542 6\
82442,
11910.269149970498679196190368266610559195470095529872103042698299652044386 2\
40835,
12010.274304509241623919889618593347368371012606990502890461681930646919262 9\
54694,
12110.243309711722465167299850323791263507661727545284505306255219531652691 0\
67106,
11910.331138094897940390509283832308671207857277121572623550378438384989475 6\
07749,
12010.216463599614253158445657031567232839800930916990755956310033314286488 2\
34396,
12110.185468927599317490202008146735815674761880919707637633735348008969812 6\
20671,
11910.211309224846499613534092123738375420599296025524010355203452815403406 0\
93738,
12010.102796007203288667367457426360900980203559877289160876613460655051217 7\
27436,
12110.185468461724377094006118356009631926448013955454212552258634272975381 0\
10926,
11910.273297098236335885386356785136516857127984857115339109490553537012903 5\
31798,
12010.154475804545091665148226440940751858234292618990461036790280464792167 7\

75709,
12110.237147976070612910869561445654930884879154264342111346367660632691133192332,
11910.159629334787536142174732121280176177515759698790251935665086995871307653547,
12010.154475478159472862128332712674359405553020449648954059824628127722428715366,
12110.299136139742341903099012793255153018895791128393265891321555549917550385094,
11910.211308883506859153864938589789302105312912485695546327089817730357756822389,
12010.206155206755466562722350838002121935268626958867837068310101841985610568670,
12110.185468466850910647076271951105011015295307456972624297787224600150203410174,
11910.211308764098399222591598939240094426256382811202692218579994334398538168300,
12010.268143235923361803876779297199167447349721035242514349969741594979357082062]

```
> Signals:=PurifyTau(ListTau);
  ObsSignals;
```

Signals :=

[
0.17767054585886572036819935972740748562300467348456136541845378605923263501\
5,
0.00616173962948580605190654864423871785778720126065155793718394132618261614\
1,
0.05784092337145950963735670251243179958986746700450139685756072280223901858\
0,
0.12599098486019831593901609703085620197615226879734409525890105070824283547\
2,
0.06400269964596767309355140861462002180668900350943438694952677199275081588\
9,
0.17767027290877075555402260644497250926321180460929142383739012547361266875\
1,
0.18383181937164627344820599986282533482299234237719980775734201202966669099\
0,
0.17767039108065267609477790592160729087376083914980791364792436315446800918\

1,
0.125990791303975115694999501565061305629829346690464039278844096880102161879\
9,
0.183832280120454500258023229898599518073350123561166906979929638478797724183\
3,
0.125990875102861017455161133288578397873021271690204453695122521270319624517\
7,
0.064002666904713921376083935866949293242647754989943531598705363935012794965\
5,
0.241673189747825261701984791678735049285026197073301412351826971111572444481\
1,
0.183831659226008694553003310344026230772867897267072126214994043953198070952\
2,
0.183831536956154698351275644437245080500628851038556726773690933521082308976\
6,
0.355340782158789752780184396885066908882397236774896492048399630346842951441\
1,
0.183832125100949090748893101070209979086734861520497207691707779947629680697\
7,
0.121843663566318426499010983039103643971940019447227972486590211911584870916\
6,
0.303660984816986754999415382305216030851664495073596331871579820605892903168\
8,
0.132152610754713273885450011424911020655594552632598413582681420231877499291\
1,
0.235511427015118169710635646895444323584165177772315146312056753053180749167\
7,
0.303661311202605558019309110571608483532936664415103308837232157675631963511\
1,
0.070164447082984281655998663824688886638957688581443868628786503005460306529\
9,
0.183831878295795158020429178386318395787012390867020754887326018566731580325\
5,
0.251981582606611857425290985243845953817330155196220300351758443412450110207\
7,
0.183832119974415537678739505974830890239441360002085462163117452772807281449\
9,
0.183831997704255089293768828935526074843542065359874863397149414525950234417\

4,
0.189993553438716616270862526046800441736236078821543018692118690418703596815]

[0.17767054585886572036819935972740748562300467348456136541845378605923263501 5, **(93)**
0.006161739629485806051906548644238717857787201260651557937183941326182616140, 0.057840923371459509637356702512431799589867467004501396857560722802239018580, 0.125990984860198315939016097030856201976152268797344095258901050708242835472, 0.064002699645967673093551408614620021806689003509434386949526771992750815888, 0.177670272908770755554022606444972509263211804609291423837390125473612668751, 0.183831819371646273448205999862825334822992342377199807757342012029666690974, 0.177670391080652676094777905921607290873760839149807913647924363154468009181, 0.125990791303975115694999501565061305629829346690464039278844096880102161879, 0.183832280120454500258023229898599518073350123561166906979929638478797724183, 0.125990875102861017455161133288578397873021271690204453695122521270319624517, 0.064002666904713921376083935866949293242647754989943531598705363935012794965, 0.241673189747825261701984791678735049285026197073301412351826971111572444481, 0.183831659226008694553003310344026230772867897267072126214994043953198070938, 0.183831536956154698351275644437245080500628851038556726773690933521082308957, 0.355340782158789752780184396885066908882397236774896492048399630346842951441, 0.183832125100949090748893101070209979086734861520497207691707779947629680697, 0.121843663566318426499010983039103643971940019447227972486590211911584870916,

```
0.30366098481698675499941538230521603085166449507359633187157982060589290316\
8,
0.13215261075471327388545001142491102065559455263259841358268142023187749929\
1,
0.23551142701511816971063564689544432358416517777231514631205675305318074916\
7,
0.30366131120260555801930911057160848353293666441510330883723215767563196351\
1,
0.07016444708298428165599866382468888663895768858144386862878650300546030652\
9,
0.18383187829579515802042917838631839578701239086702075488732601856673158032\
5,
0.25198158260661185742529098524384595381733015519622030035175844341245011020\
7,
0.18383211997441553767873950597483089023944136000208546216311745277280728145\
0,
0.18383199770425508929376882893552607484354206535987486339714941452595023441\
4,
0.18999355343871661627086252604680044173623607882154301869211869041870359681\
5 ]
```

```
> SMax:=0:
  for n from 1 to nops(Signals) do
    if SMax < abs(ObsSignals[n]-Signals[n]) then
     SMax:= abs(ObsSignals[n]-Signals[n]):
    end if:
  end do:
  SMax;
```

$$1.9 \times 10^{-74} \quad \textbf{(94)}$$

```
> # Ok we have the function myF ,
  # the 11 unknowns
  r0, te0, t0, rm0, rp0;
  te1, t1, rm1, rp1;
  te2, t2, rm2, rp2;
  # and an approximate solution (BaseROriginal, srm0v, srp0v, ste1in,
  st1in, srm1v, srp1v, ste2in, st2in, srm2v, srp2v)
```

$$r0, te0, t0, rm0, rp0$$
$$te1, t1, rm1, rp1$$
$$te2, t2, rm2, rp2 \quad \textbf{(95)}$$

```
> DebugTimeOn := false;
```

$$DebugTimeOn := false \quad \textbf{(96)}$$

```
> Digits:=25;
  MaxT:=10;
```

$$Digits := 25$$

$$MaxT := 10 \tag{97}$$

```
> delta:=10^(-10);
```

$$\delta := \frac{1}{10000000000} \tag{98}$$

```
> Rumor([BaseROriginal, srm0v, srp0v, ste1in, st1in, srm1v, srp1v,
  ste2in, st2in, srm2v, srp2v],10^(-5)):
  gr, grm0v, grp0v, gte1in, gt1in, grm1v, grp1v, gte2in, gt2in,
  grm2v, grp2v:= op(%):

  BaseROriginal,evalf(gr);
  evalf(grm0v);
  evalf(grp0v);
  evalf(gte1in);
  evalf(gt1in);
  evalf(grm1v);
  evalf(grp1v);
  evalf(gte2in);
  evalf(gt2in);
  evalf(grm2v);
  evalf(grp2v);
```

$$23889000,\ 2.3889000000010884146006 38 \times 10^{7}$$

$$2.3288999999989580939229 99 \times 10^{7}$$

$$2.4089000000007536423637 20 \times 10^{7}$$

$$-3.1561613672099713960200 00$$

$$-12110.458149204337997930 21$$

$$2.4289000000000212034005 24 \times 10^{7}$$

$$2.5089000000003140553029 81 \times 10^{7}$$

$$-2.6325649424929075231000 00$$

$$-11910.458124875869893552 98$$

$$2.5288999999999711249281 01 \times 10^{7}$$

$$2.6089000000011612975283 28 \times 10^{7} \tag{99}$$

```
> #printlevel:=6;
  infolevel[Optimization]:= 4:

> Num:=0;
  St:= time():
  Sol:=Minimize(
     myF(r0, rm0, rp0, te1, t1, rm1, rp1, te2, t2, rm2, rp2),
     initialpoint = {r0=gr, rm0=grm0v, rp0=grp0v, te1=gte1in, t1=
  gt1in, rm1=grm1v, rp1=grp1v, te2=gte2in, t2=gt2in, rm2=grm2v, rp2=
  grp2v},
```

```
    r0=BaseROriginal-delta..BaseROriginal+delta,
    rm0= srm0v-delta..srm0v+delta,
    rp0= srp0v-delta..srp0v+delta,
    te1= ste1in-delta..ste1in+delta,
    t1 = st1in-delta..st1in+delta,
    rm1= srm1v-delta..srm1v+delta,
    rp1= srp1v-delta..srp1v+delta,
    te2= ste2in-delta..ste2in+delta,
    t2 = st2in-delta..st2in+delta,
    rm2= srm2v-delta..srm2v+delta,
    rp2= srp2v-delta..srp2v+delta,
    assume = nonnegative,
    iterationlimit = 300
  );

  TotalTime:= time()-St;
```

$$Num := 0$$

```
NLPSolve: calling NLP solver
NLPSolve: using method=sqp
NLPSolve: number of problem variables 11
NLPSolve: number of nonlinear inequality constraints 0
NLPSolve: number of nonlinear equality constraints 0
NLPSolve: number of general linear constraints 0
NLPSolve: feasibility tolerance set to 0.3162277660e-12
NLPSolve: optimality tolerance set to 0.1000000000e-17
NLPSolve: iteration limit set to 300
NLPSolve: infinite bound set to 0.10e21
NLPSolve: trying evalf mode
Evaluate 1:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 24.514
counts: 28, 28
 chi^2 = .651036523650401226402383 2e-20
Evaluate 2:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 43.811
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854 96e-20
Evaluate 3:
 r0=23889000.000000000100050000000000000000000 rm0=
23289000.00000000010000000 rp0=24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
```

```
Cascade time 45.012
counts: 28, 28
 chi^2 = .651036524402728129805971670868026127587 54e-20
Evaluate 4:
 r0=23889000.000000000099950000000000000000000 rm0=
23289000.00000000010000000 rp0=24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 44.663
counts: 28, 28
 chi^2 = .651036524419266083120164608858821630951 39e-20
Evaluate 5:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 44.599
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854 96e-20
Evaluate 6:
 r0=23889000.00000000010000000 rm0=
23289000.000000000100050000000000000000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 45.584
counts: 28, 28
 chi^2 = .651036524410997181990236721708616223295 47e-20
Evaluate 7:
 r0=23889000.00000000010000000 rm0=
23289000.000000000099950000000000000000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 46.349
counts: 28, 28
 chi^2 = .651036524410997030935842519135979553449 37e-20
Evaluate 8:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 45.638
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854 96e-20
Evaluate 9:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
```

```
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000100050000000000000000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 43.439
counts: 28, 28
 chi^2 = .651036524342350781275825213856591764786664e-20
Evaluate 10:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000099950000000000000000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 42.916
counts: 28, 28
 chi^2 = .651036524479643431653874396631959877803184e-20
Evaluate 11:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 42.521
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854960e-20
Evaluate 12:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100050000000000000000000 rp2=
26089000.00000000010000000
Cascade time 44.871
counts: 28, 28
 chi^2 = .651036524408668988561690362813176385459760e-20
Evaluate 13:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000099950000000000000000000 rp2=
26089000.00000000010000000
Cascade time 43.922
counts: 28, 28
 chi^2 = .651036524413325224364394321590196610351640e-20
Evaluate 14:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
```

```
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100000000 rp2=26089000.000000000100000000
Cascade time 46.669
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854960e-20
Evaluate 15:
 r0=23889000.000000000100000000 rm0=23289000.000000000100000000 rp0=
24089000.000000000100050000000000000000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000100000000 rp1=25089000.000000000100000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100000000 rp2=26089000.000000000100000000
Cascade time 48.486
counts: 28, 28
 chi^2 = .651036524410997330226949107669931957884100e-20
Evaluate 16:
 r0=23889000.000000000100000000 rm0=23289000.000000000100000000 rp0=
24089000.000000000099950000000000000000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000100000000 rp1=25089000.000000000100000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100000000 rp2=26089000.000000000100000000
Cascade time 46.726
counts: 28, 28
 chi^2 = .651036524410996882699130218565617042874130e-20
Evaluate 17:
 r0=23889000.000000000100000000 rm0=23289000.000000000100000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000100000000 rp1=25089000.000000000100000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100000000 rp2=26089000.000000000100000000
Cascade time 44.421
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854960e-20
Evaluate 18:
 r0=23889000.000000000100000000 rm0=23289000.000000000100000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000100000000 rp1=
25089000.000000000100050000000000000000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100000000 rp2=26089000.000000000100000000
Cascade time 48.813
counts: 28, 28
 chi^2 = .651036524392990897074210434818708428851240e-20
Evaluate 19:
 r0=23889000.000000000100000000 rm0=23289000.000000000100000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000100000000 rp1=
25089000.000000000099950000000000000000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000100000000 rp2=26089000.000000000100000000
Cascade time 49.245
counts: 28, 28
 chi^2 = .651036524429003315852120373427027200933820e-20
```

```
Evaluate 20:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 50.026
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854960e-20
Evaluate 21:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000010000000 rp2=
26089000.000000000100050000000000000000000
Cascade time 50.108
counts: 28, 28
 chi^2 = .651036524398685117800157416009494015438000e-20
Evaluate 22:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000010000000 rp2=
26089000.000000000099950000000000000000000
Cascade time 49.954
counts: 28, 28
 chi^2 = .651036524423309095126039438557289926686530e-20
Evaluate 23:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106699
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 49.627
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854960e-20
Evaluate 24:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210099405873045059 t1=
-12110.458140193874078106990000000000000000000 rm1=
24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106699
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 50.069
counts: 28, 28
 chi^2 = .651036411363339091977316485422459289780260e-20
Evaluate 25:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210099405873045059 t1=
```

```
-12110.458140193874178106990000000000000000 rm1=
24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 48.765
counts: 28, 28
 chi^2 = .651036637458664935892791529676190071153 00e-20
Evaluate 26:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 49.249
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854 96e-20
Evaluate 27:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=
-11910.458140193874078106990000000000000000 rm2=
25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 50.026
counts: 28, 28
 chi^2 = .651036533817464226307443921215034064544 96e-20
Evaluate 28:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=
-11910.458140193874178106990000000000000000 rm2=
25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 50.651
counts: 28, 28
 chi^2 = .651036515004530054573061283272979298751 39e-20
Evaluate 29:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 53.257
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854 96e-20
Evaluate 30:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099355873045059000000000000000000 t1=
-12110.45814019387412810699 rm1=24289000.00000000010000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412810699
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
```

```
Cascade time 54.147
counts: 28, 28
 chi^2 = .651741492919573374643020314311132467968 63e-20
Evaluate 31:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099455873045059000000000000000 t1=
-12110.458140193874128106 99 rm1=24289000.00000000010000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106 99
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 54.259
counts: 28, 28
 chi^2 = .650331937380023206113268416875759824446 89e-20
Evaluate 32:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874128106 99
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 50.869
counts: 28, 28
 chi^2 = .651036524410997106463040218158729752854 96e-20
Evaluate 33:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501056999967951000000000000000000 t2=
-11910.458140193874128106 99 rm2=25289000.00000000010000000 rp2=
26089000.00000000010000000
Cascade time 53.722
counts: 28, 28
 chi^2 = .650982888021165792565330051775372992480 85e-20
Evaluate 34:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210099405873045059 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501156999967951000000000000000000 t2=
-11910.458140193874128106 99 rm2=25289000.00000000010000000 rp2=
26089000.00000000010000000
Cascade time 53.04
counts: 28, 28
 chi^2 = .651090163010367156318605165650177829324 85e-20
Evaluate 35:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988 28
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 31.159
counts: 28, 28
 chi^2 = .180064831477076621058883 0e-20
Evaluate 36:
```

```
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988280
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 54.888
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 37:
 r0=23889000.000000000100050000000000000000000 rm0=
23289000.00000000010000000 rp0=24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988280
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.843
counts: 28, 28
 chi^2 = .18006482713676724670009064153532708097064e-20
Evaluate 38:
 r0=23889000.000000000099950000000000000000000 rm0=
23289000.00000000010000000 rp0=24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988280
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.894
counts: 28, 28
 chi^2 = .18006482712806976489660088604087949663059e-20
Evaluate 39:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988280
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 54.617
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 40:
 r0=23889000.00000000010000000 rm0=
23289000.000000000100050000000000000000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988280
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 53.559
counts: 28, 28
 chi^2 = .18006482713241846607778069350845612202910e-20
Evaluate 41:
 r0=23889000.00000000010000000 rm0=
23289000.000000000099950000000000000000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988280
```

```
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.446
counts: 28, 28
 chi^2 = .18006482713241854551885845562813329198517e-20
Evaluate 42:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 54.688
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 43:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010005000000000000000000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.212
counts: 28, 28
 chi^2 = .18006482716852032280651956107035344962562e-20
Evaluate 44:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000009995000000000000000000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.763
counts: 28, 28
 chi^2 = .18006482709631668879373876666236625967473e-20
Evaluate 45:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 56.251
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 46:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010005000000000000000000000 rp2=
26089000.00000000010000000
Cascade time 55.268
counts: 28, 28
 chi^2 = .18006482713364288723821018093608405145306e-20
```

```
Evaluate 47:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988 28
rm2=25289000.000000000099950000000000000000000 rp2=
26089000.000000000010000000
Cascade time 57.218
counts: 28, 28
 chi^2 = .180064827131194124358433220711445268911 79e-20
Evaluate 48:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988 28
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 56.075
counts: 28, 28
 chi^2 = .180064827132418505798319597022279155213 33e-20
Evaluate 49:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000100050000000000000000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988 28
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 56.704
counts: 28, 28
 chi^2 = .180064827132418388118550366603618398158 43e-20
Evaluate 50:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000099950000000000000000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988 28
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 57.15
counts: 28, 28
 chi^2 = .180064827132418623478088692717058534371 90e-20
Evaluate 51:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.000000000010000000 rp1=25089000.000000000010000000
 te2=-2.632556434501106999967951 t2=-11910.458140193874129988 28
rm2=25289000.000000000010000000 rp2=26089000.000000000010000000
Cascade time 59.747
counts: 28, 28
 chi^2 = .180064827132418505798319597022279155213 33e-20
Evaluate 52:
 r0=23889000.000000000010000000 rm0=23289000.000000000010000000 rp0=
24089000.000000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.000000000010000000 rp1=
25089000.000000000100050000000000000000000
```

```
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 58.192
counts: 28, 28
 chi^2 = .18006482714188815875680592946413019902411e-20
Evaluate 53:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=
25089000.000000000099950000000000000000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 57.131
counts: 28, 28
 chi^2 = .18006482712294885284008236006999760028518e-20
Evaluate 54:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.477
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 55:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=
26089000.000000000100050000000000000000000
Cascade time 56.887
counts: 28, 28
 chi^2 = .18006482713889350880100247988135674590595e-20
Evaluate 56:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=
26089000.000000000099950000000000000000000
Cascade time 55.996
counts: 28, 28
 chi^2 = .18006482712594350279575313241253897309554e-20
Evaluate 57:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106 99
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.398
counts: 28, 28
```

```
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 58:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=
-12110.458140193874078106990000000000000000 rm1=
24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 56.167
counts: 28, 28
 chi^2 = .18006488658536613838791006599936434256236e-20
Evaluate 59:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=
-12110.458140193874178106990000000000000000 rm1=
24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 57.301
counts: 28, 28
 chi^2 = .18006476767948068815275435447180390439442e-20
Evaluate 60:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 55.376
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 61:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=
-11910.458140193874079988280000000000000000 rm2=
25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 57.564
counts: 28, 28
 chi^2 = .18006482218545958110994415950077915516368e-20
Evaluate 62:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.458140193874128106699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=
-11910.458140193874179988280000000000000000 rm2=
25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 56.37
counts: 28, 28
 chi^2 = .18006483207937749844111762232423332018135e-20
Evaluate 63:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
```

```
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 56.444
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 64:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.15615521024031142700007600000000000000000 t1=
-12110.45814019387412810699 rm1=24289000.00000000010000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 56.945
counts: 28, 28
 chi^2 = .17969436790555548633459060796159701515487e-20
Evaluate 65:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.15615521024041142700007600000000000000000 t1=
-12110.45814019387412810699 rm1=24289000.00000000010000000 rp1=
25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 56.947
counts: 28, 28
 chi^2 = .18043566783688388168936523587680034258063e-20
Evaluate 66:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.632556434501106999967951 t2=-11910.45814019387412998828
rm2=25289000.00000000010000000 rp2=26089000.00000000010000000
Cascade time 58.525
counts: 28, 28
 chi^2 = .18006482713241850579831959702227915521333e-20
Evaluate 67:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.63255643450105699999679510000000000000000000 t2=
-11910.45814019387412998828 rm2=25289000.00000000010000000 rp2=
26089000.00000000010000000
Cascade time 58.458
counts: 28, 28
 chi^2 = .18009303675394627161123071367880764252513e-20
Evaluate 68:
 r0=23889000.00000000010000000 rm0=23289000.00000000010000000 rp0=
24089000.00000000010000000
 te1=-3.156155210240361427000076 t1=-12110.45814019387412810699
rm1=24289000.00000000010000000 rp1=25089000.00000000010000000
 te2=-2.63255643450115699999679510000000000000000000 t2=
```

```
-11910.458140193874129988 28 rm2=25289000.000000000100000000 rp2=
26089000.000000000100000000
Cascade time 61.445
counts: 28, 28
 chi^2 = .180036619720429462027504987689414484765 40e-20
attemptsolution: number of major iterations taken 1
```

$$Sol := [1.8006483147707662105888 30 \times 10^{-21}, [r0 = 2.3889000000000000100000 00 \times 10^{7},$$
$$rm0 = 2.3289000000000000100000 00 \times 10^{7}, rm1 = 2.4289000000000000100000 00 \times 10^{7},$$
$$rm2 = 2.5289000000000000100000 00 \times 10^{7}, rp0 = 2.4089000000000000100000 00 \times 10^{7},$$
$$rp1 = 2.5089000000000000100000 00 \times 10^{7}, rp2 = 2.6089000000000000100000 00 \times 10^{7}, t1$$
$$= -12110.458140193874128106 99, t2 = -11910.458140193874129988 28, te1$$
$$= -3.1561552102403614270000 76, te2 = -2.6325564345011069999679 51]]$$

$$TotalTime := 3585.830 \quad \textbf{(100)}$$

```
> Sol;
```

$$[1.8006483147707662105888 30 \times 10^{-21}, [r0 = 2.3889000000000000100000 00 \times 10^{7}, rm0 \quad \textbf{(101)}$$
$$= 2.3289000000000000100000 00 \times 10^{7}, rm1 = 2.4289000000000000100000 00 \times 10^{7}, rm2$$
$$= 2.5289000000000000100000 00 \times 10^{7}, rp0 = 2.4089000000000000100000 00 \times 10^{7}, rp1$$
$$= 2.5089000000000000100000 00 \times 10^{7}, rp2 = 2.6089000000000000100000 00 \times 10^{7}, t1$$
$$= -12110.458140193874128106 99, t2 = -11910.458140193874129988 28, te1$$
$$= -3.1561552102403614270000 76, te2 = -2.6325564345011069999679 51]]$$

```
>
  subs(Sol[2], r0):
  fr0:=%;
  subs(Sol[2], rm0):
  frm0:=%;
  subs(Sol[2], rp0):
  frp0:=%;
  subs(Sol[2], te1):
  fte1:=%;
  subs(Sol[2], t1):
  ft1:=%;
  subs(Sol[2], rm1):
  frm1:=%;
  subs(Sol[2], rp1):
  frp1:=%;

  subs(Sol[2], te2):
  fte2:=%;
  subs(Sol[2], t2):
  ft2:=%;
  subs(Sol[2], rm2):
  frm2:=%;
  subs(Sol[2], rp2):
  frp2:=%;
```

$$fr0 := 2.3889000000000000010000000 \times 10^{7}$$

$$frm0 := 2.3289000000000000010000000 \times 10^{7}$$

$$frp0 := 2.4089000000000000010000000 \times 10^{7}$$

$$fte1 := -3.1561552102403614270000 76$$

$$ft1 := -12110.458140193874128106 99$$

$$frm1 := 2.4289000000000000010000000 \times 10^{7}$$

$$frp1 := 2.5089000000000000010000000 \times 10^{7}$$

$$fte2 := -2.6325564345011069999679$$

$$ft2 := -11910.458140193874129988 28$$

$$frm2 := 2.5289000000000000010000000 \times 10^{7}$$

$$frp2 := 2.6089000000000000010000000 \times 10^{7} \quad \textbf{(102)}$$

```
> te0in, t0in, tau0in, rm0v, rp0v := op(Sat0Norm);
  te1in, t1in, tau1in, rm1v, rp1v := op(Sat1Norm);
  te2in, t2in, tau2in, rm2v, rp2v := op(Sat2Norm);
```

$$te0in, t0in, tau0in, rm0v, rp0v :=$$
$$-2.10895765900280812689084414296181633395083177467878627014343044840622126\backslash$$
$$92704310,$$
$$-12010.458140193974128106990086403103436416267811869852971885475213316854 2\backslash$$
$$68817156,$$
$$-12010.458136789362078420147641823245967889085957114064057368661860285398 0\backslash$$
$$60678877, 23289000, 24089000$$
$$te1in, t1in, tau1in, rm1v, rp1v :=$$
$$-3.156155210199405873045058604054983962016554907803821543801745312508826 73\backslash$$
$$80325006,$$
$$-12110.458140193974128106990086403103436416267811869852971885475213316854 2\backslash$$
$$68817156,$$
$$-12110.458136789362078420147641823245967889085957114064057368661860285398 0\backslash$$
$$60678877, 24289000, 25089000$$
$$te2in, t2in, tau2in, rm2v, rp2v := \quad \textbf{(103)}$$
$$-2.6325564346011069999679513735084001479836933412413039069725878804575240 0\backslash$$
$$36514658,$$
$$-11910.458140193974128106990086403103436416267811869852971885475213316854 2\backslash$$
$$68817156,$$
$$-11910.458136789362078420147641823245967889085957114064057368661860285398 0\backslash$$
$$60678877, 25289000, 26089000$$

```
> printf("r0: %a -> %a\n",evalf(gr-BaseROriginal), fr0-BaseROriginal)
```

```
  ;
  printf("rm0: %a -> %a\n",evalf(grm0v-rm0v), frm0-rm0v);
  printf("rp0: %a -> %a\n",evalf(grp0v-rp0v), frp0-rp0v);


  printf("te1: %a -> %a\n",evalf(gte1in-te1in), fte1-te1in);
  printf("t1: %a -> %a\n",evalf(gt1in-t1in), ft1-t1in);
  printf("rm1: %a -> %a\n",evalf(grm1v-rm1v), frm1-rm1v);
  printf("rp1: %a -> %a\n",evalf(grp1v-rp1v), frp1-rp1v);

  printf("te2: %a -> %a\n",evalf(gte2in-te2in), fte2-te2in);
  printf("t2: %a -> %a\n", evalf(gt2in-t2in), ft2-t2in);
  printf("rm2: %a -> %a\n",evalf(grm2v-rm2v), frm2-rm2v);
  printf("rp2: %a -> %a\n",evalf(grp2v-rp2v), frp2-rp2v);


r0: .10884146006380000000000000e-4 -> .10000000e-9
rm0: -.10419060770010000000000000e-4 -> .10000000e-9
rp0: .75364236372000000000000000e-5 -> .10000000e-9
te1: -.615701056552297494 1e-5 -> -.40955553955017e-10
t1: -.901036386982322e-5 -> .10000000000e-9
rm1: .21203400524000000000000000e-6 -> .10000000e-9
rp1: .31405530298100000000000000e-5 -> .10000000e-9
te2: -.8507891800523132049e-5 -> .100000000000000e-9
t2: .1531810423455401e-4 -> .9999811871e-10
rm2: -.28875071899000000000000000e-6 -> .10000000e-9
rp2: .11612975283280000000000000e-4 -> .10000000e-9
```

```
evaluations  68
 chi^2 = .17978285244188114755053870798059038240e-20

Digits          23
delta           10^(-10)
rumor           10^(-5)



1.80065536083987407070 34*10^(-21)
                r0 = 2.38890000000000000100000*10^7
                rm0 = 2.32890000000000000100000*10^7
                rm1 = 2.42890000000000000100000*10^7
                rm2 = 2.52890000000000000100000*10^7
                rp0 = 2.40890000000000000100000*10^7
                rp1 = 2.50890000000000000100000*10^7
                rp2 = 2.60890000000000000100000*10^7
                t1 = -12110.458140193874128107
                t2 = -11910.458140193874129988
                te1 = -3.1561552102403614270001
```

te2 = -2.6325564345011069999680

r0: -.35579746540000000000000e-5 -> .100000e-9
rm0: -.58827024770000000000000e-5 -> .100000e-9
rp0: .20474739692000000000000e-4 -> .100000e-9
te1: -.5863735374749549e-6 -> -.409555539550e-10
t1: -.1206749369002e-5 -> .100000000e-9
rm1: .34596449190000000000000e-5 -> .100000e-9
rp1: .72683169520000000000000e-5 -> .100000e-9
te2: .17963968468497968e-4 -> .1000000000000e-9
t2: -.3050710869727e-5 -> .99998119e-10
rm2: -.56256175870000000000000e-5 -> .100000e-9
rp2: -.14869288650000000000000e-5 -> .100000e-9

evaluations 68
chi^2 = .87750803796860969019599041021954840e-20

Digits 22
delta 10^(-10)
rumor 10^(-5)

6.5106025059056210372 35*10^(-21),
r0 = 2.3889000000000000010000*10^7
rm0 = 2.3289000000000000010000*10^7
rp0 = 2.4089000000000000010000*10^7

te1 = -3.156155210099405873045
t1 = -12110.45814019387412811
rm1 = 2.4289000000000000010000*10^7
rp1 = 2.5089000000000000010000*10^7

te2 = -2.632556434501106999968
t2 = -11910.45814019387412811
rm2 = 2.5289000000000000010000*10^7
rp2 = 2.6089000000000000010000*10^7

r0: -.40871496100000000000000e-5 -> .10000e-9
rm0: .20193581800000000000000e-4 -> .10000e-9
rp0: -.24958576300000000000000e-5 -> .10000e-9
te1: .27626323822163045e-4 -> .100000000000e-9
t1: .978578606084e-5 -> .10000000e-9
rm1: .17976227600000000000000e-5 -> .10000e-9

rp1: .3529761040000000000000e-5 -> .10000e-9
te2: -.6224840301700032e-5 -> .100000000000e-9
t2: -.29236662196e-6 -> .10000000e-9
rm2: .17277048340000000000e-4 -> .10000e-9
rp2: .72537693000000000000e-5 -> .10000e-9

delta = 10^(-8)
Rumor 10^(-5)
Digits 20

Evaluation 102
chi^2 = .17277256090550040959165806299905e-22

$$\left[7.133659021639612 \times 10^{-25},\ \left[\begin{array}{l} r0 = 2.3889000000000010000 \times 10^{7},\\ \\ rm0 = 2.3289000000000010000 \times 10^{7},\\ rp0 = 2.4089000000000010000 \times 10^{7},\\ \\ te1 = -3.1561552094359249329,\\ t1 = -12110.458140183974128,\\ rm1 = 2.4289000000000010000 \times 10^{7},\\ rp1 = 2.5089000000000010000 \times 10^{7},\\ \\ te2 = -2.6325564246011070000,\\ t2 = -11910.458140183974251,\\ rm2 = 2.5289000000000010000 \times 10^{7},\\ rp2 = 2.6089000000000010000 \times 10^{7} \end{array}\right]\right]$$

r0: .11778242000000000000e-4 -> .10000e-7
rm0: .22612880000000000000e-5 -> .10000e-7
rp0: -.12673570000000000000e-5 -> .10000e-7
te1: -.92561408661270e-5 -> .7634809401e-9
t1: -.4315953028e-5 -> .10000000e-7
rm1: -.89925510000000000000e-5 -> .10000e-7
rp1: -.13540443000000000000e-4 -> .10000e-7
te2: -.59528164510000e-5 -> .100000000000e-7

t2: .11118658037e-4 -> .9999877e-8
rm2: .741868700000000000e-5 -> .10000e-7
rp2: -.793983100000000000e-5 -> .10000e-7