```
> #Vec:=n->Vector(n,datatype=float,i->i);
> restart;
> dLum:=(Xo,Yo,Xpo,Ypo,Zo,z) ->
  exp(-(Zo-2*z)*((1/2)*Zo-z)/Sigz^2-(Xpo*z+Xo)*((1/2)*Xpo*z+(1/2)*Xo)/(Sigx^2+Sigxp^2*z^2)-(Yo+Ypo*z)*((
  1/2)*Ypo*z+(1/2)*Yo)/(Sigy^2+Sigyp^2*z^2))/sqrt((Sigx^2+Sigxp^2*z^2)*(Sigy^2+Sigyp^2*z^2)*Sigz^2);
```

$$\mathrm{dLum} := (\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}, z) \to \frac{\mathbf{e}^{\left(-\frac{(\mathrm{Zo}-2z)(1/2\,\mathrm{Zo}-z)}{\mathrm{Sigz}^2}-\frac{(\mathrm{Xpo}\,z+\mathrm{Xo})(1/2\,\mathrm{Xpo}\,z+1/2\,\mathrm{Xo})}{\mathrm{Sigx}^2+\mathrm{Sigxp}^2 z^2}-\frac{(\mathrm{Yo}+\mathrm{Ypo}\,z)(1/2\,\mathrm{Ypo}\,z+1/2\,\mathrm{Yo})}{\mathrm{Sigy}^2+\mathrm{Sigyp}^2 z^2}\right)}}{\sqrt{(\mathrm{Sigx}^2+\mathrm{Sigxp}^2 z^2)(\mathrm{Sigy}^2+\mathrm{Sigyp}^2 z^2)\,\mathrm{Sigz}^2}}$$

```
> #dLum(Xo,Yo,Xpo,Ypo,Zo,z);
  exp(
  -1/2*(Zo-2*z)*(Zo-2*z)/Sigz^2
  -1/2*(Xpo*z+Xo)*(Xpo*z+Xo)/(Sigx^2+Sigxp^2*z^2)
  -1/2*(Yo+Ypo*z)*(Ypo*z+Yo)/(Sigy^2+Sigyp^2*z^2)
  ) /
  sqrt((Sigx^2+Sigxp^2*z^2)*(Sigy^2+Sigyp^2*z^2)*Sigz^2):
  dL:= unapply(%, Xo,Yo,Xpo,Ypo,Zo,z);
```

$$\mathrm{dL} := (\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}, z) \to \frac{\mathbf{e}^{\left(-1/2\frac{(\mathrm{Zo}-2z)^2}{\mathrm{Sigz}^2}-1/2\frac{(\mathrm{Xpo}\,z+\mathrm{Xo})^2}{\mathrm{Sigx}^2+\mathrm{Sigxp}^2 z^2}-1/2\frac{(\mathrm{Yo}+\mathrm{Ypo}\,z)^2}{\mathrm{Sigy}^2+\mathrm{Sigyp}^2 z^2}\right)}}{\sqrt{(\mathrm{Sigx}^2+\mathrm{Sigxp}^2 z^2)(\mathrm{Sigy}^2+\mathrm{Sigyp}^2 z^2)\,\mathrm{Sigz}^2}}$$

```
> 'dLum'(Xo,Yo,Xpo,Ypo,Zo,z) = 'dL'(Xo,Yo,Xpo,Ypo,Zo,z); is(%);
```

$$\mathrm{dLum}(\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}, z) = \mathrm{dL}(\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}, z)$$

true

```
> ln(1/sqrt(t)):
  a+%:
  exp(%):
  %=simplify(%);
```

$$\mathbf{e}^{(a-1/2\ln(t))} = \frac{\mathbf{e}^a}{\sqrt{t}}$$

```
> logIntegrand:=
  '-1/2*(Zo-2*z)*(Zo-2*z)/Sigz^2
  -1/2*(Xpo*z+Xo)*(Xpo*z+Xo)/(Sigx^2+Sigxp^2*z^2)
  -1/2*(Yo+Ypo*z)*(Ypo*z+Yo)/(Sigy^2+Sigyp^2*z^2)' -
  '1/2*ln((Sigx^2+Sigxp^2*z^2)*(Sigy^2+Sigyp^2*z^2)*Sigz^2)';
```

$$\mathrm{logIntegrand} := -\frac{(\mathrm{Zo}-2z)^2}{2\,\mathrm{Sigz}^2}-\frac{(\mathrm{Xpo}\,z+\mathrm{Xo})^2}{2\,(z^2\,\mathrm{Sigxp}^2+\mathrm{Sigx}^2)}-\frac{(\mathrm{Ypo}\,z+\mathrm{Yo})^2}{2\,(z^2\,\mathrm{Sigyp}^2+\mathrm{Sigy}^2)}-\frac{1}{2}\ln((z^2\,\mathrm{Sigxp}^2+\mathrm{Sigx}^2)(z^2\,\mathrm{Sigyp}^2+\mathrm{Sigy}^2)\,\mathrm{Sigz}^2)$$

```
> 0='dL(Xo,Yo,Xpo,Ypo,Zo,z)' - 'exp(logIntegrand)'; expand(%): simplify(%): is(%);
```

$$0 = \mathrm{dL}(\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}, z) - \mathbf{e}^{\mathrm{logIntegrand}}$$

true

```
>
>
> Lum:=(Xo,Yo,Xpo,Ypo,Zo) -> Int(dLum(Xo,Yo,Xpo,Ypo,Zo,z),z=-infinity..infinity);
```

$$\mathrm{Lum} := (\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}) \to \int_{-\infty}^{\infty} \mathrm{dLum}(\mathrm{Xo}, \mathrm{Yo}, \mathrm{Xpo}, \mathrm{Ypo}, \mathrm{Zo}, z)\,dz$$

```
>
> pre:=[epsyH=rH*epsxH,
  epsyL=rL*epsxL,
  sigxL=sqrt(epsxL*betxL),
  sigxpL=sqrt(epsxL/betxL),
  sigyL=sqrt(epsyL*betyL),
  sigypL=sqrt(epsyL/betyL),
  sigxH=sqrt(epsxH*betxH),
  sigxpH=sqrt(epsxH/betxH),
  sigyH=sqrt(epsyH*betyH),
  sigypH=sqrt(epsyH/betyH),
  Sigx=sqrt(sigxH^2+sigxL^2),
  Sigy=sqrt(sigyH^2+sigyL^2),
  Sigxp=sqrt(sigxpH^2+sigxpL^2),
  Sigyp=sqrt(sigypH^2+sigypL^2),
  Sigz=sqrt(szH^2+szL^2)];
```

$$\mathrm{pre} := \left[\mathrm{epsyH} = \mathrm{rH}\,\mathrm{epsxH},\ \mathrm{epsyL} = \mathrm{rL}\,\mathrm{epsxL},\ \mathrm{sigxL} = \sqrt{\mathrm{epsxL}\,\mathrm{betxL}},\ \mathrm{sigxpL} = \sqrt{\frac{\mathrm{epsxL}}{\mathrm{betxL}}},\ \mathrm{sigyL} = \sqrt{\mathrm{epsyL}\,\mathrm{betyL}},\ \mathrm{sigypL} = \sqrt{\frac{\mathrm{epsyL}}{\mathrm{betyL}}},\right.$$
$$\left.\mathrm{sigxH} = \sqrt{\mathrm{epsxH}\,\mathrm{betxH}},\ \mathrm{sigxpH} = \sqrt{\frac{\mathrm{epsxH}}{\mathrm{betxH}}},\ \mathrm{sigyH} = \sqrt{\mathrm{epsyH}\,\mathrm{betyH}},\ \mathrm{sigypH} = \sqrt{\frac{\mathrm{epsyH}}{\mathrm{betyH}}},\ \mathrm{Sigx} = \sqrt{\mathrm{sigxH}^2+\mathrm{sigxL}^2},\ \mathrm{Sigy} = \sqrt{\mathrm{sigyH}^2+\mathrm{sigyL}^2},\right.$$

$$\text{Sigxp} = \sqrt{\text{sigxpH}^2 + \text{sigxpL}^2},\ \text{Sigyp} = \sqrt{\text{sigypH}^2 + \text{sigypL}^2},\ \text{Sigz} = \sqrt{\text{szH}^2 + \text{szL}^2}\ \Big]$$

```
> params:=
[betxL=25E-3,betxH=32E-3,betyL=0.27E-3,betyH=0.3E-3,epsxL=3.2E-9,epsxH=4.6E-9,szL=6E-3,szH=5E-3,rL=0.2
7E-2,rH=0.28E-2];
#Params:=map(rhs, params);
```

$$\text{params} := [\text{betxL} = 0.025,\ \text{betxH} = 0.032,\ \text{betyL} = 0.00027,\ \text{betyH} = 0.0003,\ \text{epsxL} = 0.32\ 10^{-8},\ \text{epsxH} = 0.46\ 10^{-8},\ \text{szL} = 0.006,\ \text{szH} = 0.005,\ \text{rL} = 0.0027,\ \text{rH} = 0.0028]$$

```
> 
> # double precision
exp(z)=1e-308; map(ln, %): z= fsolve(%); #map(exp, %);
```

$$e^z = 0.1\ 10^{-307}$$

$$z = -709.196208642166$$

```
> # single precision
exp(z)=1e-38; map(ln, %): z= fsolve(%); #map(exp, %);
```

$$e^z = 0.1\ 10^{-37}$$

$$z = -87.4982335337737$$

```
> 
> P:=proc(Xo,Yo,Xpo,Ypo,Zo)
option hfloat;
local z, zmax, zlower, zupper, theIntegrand, logIntegrand,
cutoff, eps, f,
epsyH, epsyL, sigxL, sigxpL, sigyL, sigypL, sigxH, sigxpH, sigyH, sigypH, Sigx, Sigy, Sigxp, Sigyp,
Sigz;
global params, Params;

cutoff:= - 709.0; # double precision
#cutoff:= -  87.0; # single precision
eps:= 6; # for integration

epsyH:=rH*epsxH;
epsyL:=rL*epsxL;
sigxL:=sqrt(epsxL*betxL);
sigxpL:=sqrt(epsxL/betxL);
sigyL:=sqrt(epsyL*betyL);
sigypL:=sqrt(epsyL/betyL);
sigxH:=sqrt(epsxH*betxH);
sigxpH:=sqrt(epsxH/betxH);
sigyH:=sqrt(epsyH*betyH);
sigypH:=sqrt(epsyH/betyH);
Sigx:=sqrt(sigxH^2+sigxL^2);
Sigy:=sqrt(sigyH^2+sigyL^2);
Sigxp:=sqrt(sigxpH^2+sigxpL^2);
Sigyp:=sqrt(sigypH^2+sigypL^2);
Sigz:=sqrt(szH^2+szL^2);

logIntegrand:=
  -1/2*(Zo-2*z)*(Zo-2*z)/Sigz^2
  -1/2*(Xpo*z+Xo)*(Xpo*z+Xo)/(Sigx^2+Sigxp^2*z^2)
  -1/2*(Yo+Ypo*z)*(Ypo*z+Yo)/(Sigy^2+Sigyp^2*z^2)
  - 1/2*ln((Sigx^2+Sigxp^2*z^2)*(Sigy^2+Sigyp^2*z^2)*Sigz^2);
logIntegrand:=eval(logIntegrand, params);

zlower:=fsolve(cutoff=logIntegrand, z=-1,  z=-infinity .. 0);

if type(zlower, numeric) then NULL else zlower:=-1.0 end if;
zupper:=fsolve(cutoff=logIntegrand, z=-1 .. infinity, avoid={z=zlower});

if type(zupper, numeric) then NULL else zupper:=+1.0 end if;

if 5 < nargs then
  print('zlower' = zlower);
  print('zupper' = zupper);
  plot([logIntegrand],z=-1e-2 .. 1e-2, title="log(Integrand)", color=[red,blue]); print(%);
end if;



theIntegrand:= exp(
  -1/2*(Zo-2*z)*(Zo-2*z)/Sigz^2
  -1/2*(Xpo*z+Xo)*(Xpo*z+Xo)/(Sigx^2+Sigxp^2*z^2)
  -1/2*(Yo+Ypo*z)*(Ypo*z+Yo)/(Sigy^2+Sigyp^2*z^2)
  ) /
  sqrt((Sigx^2+Sigxp^2*z^2)*(Sigy^2+Sigyp^2*z^2)*Sigz^2);
theIntegrand:=eval(theIntegrand, params);
f:=unapply(%, z);
```

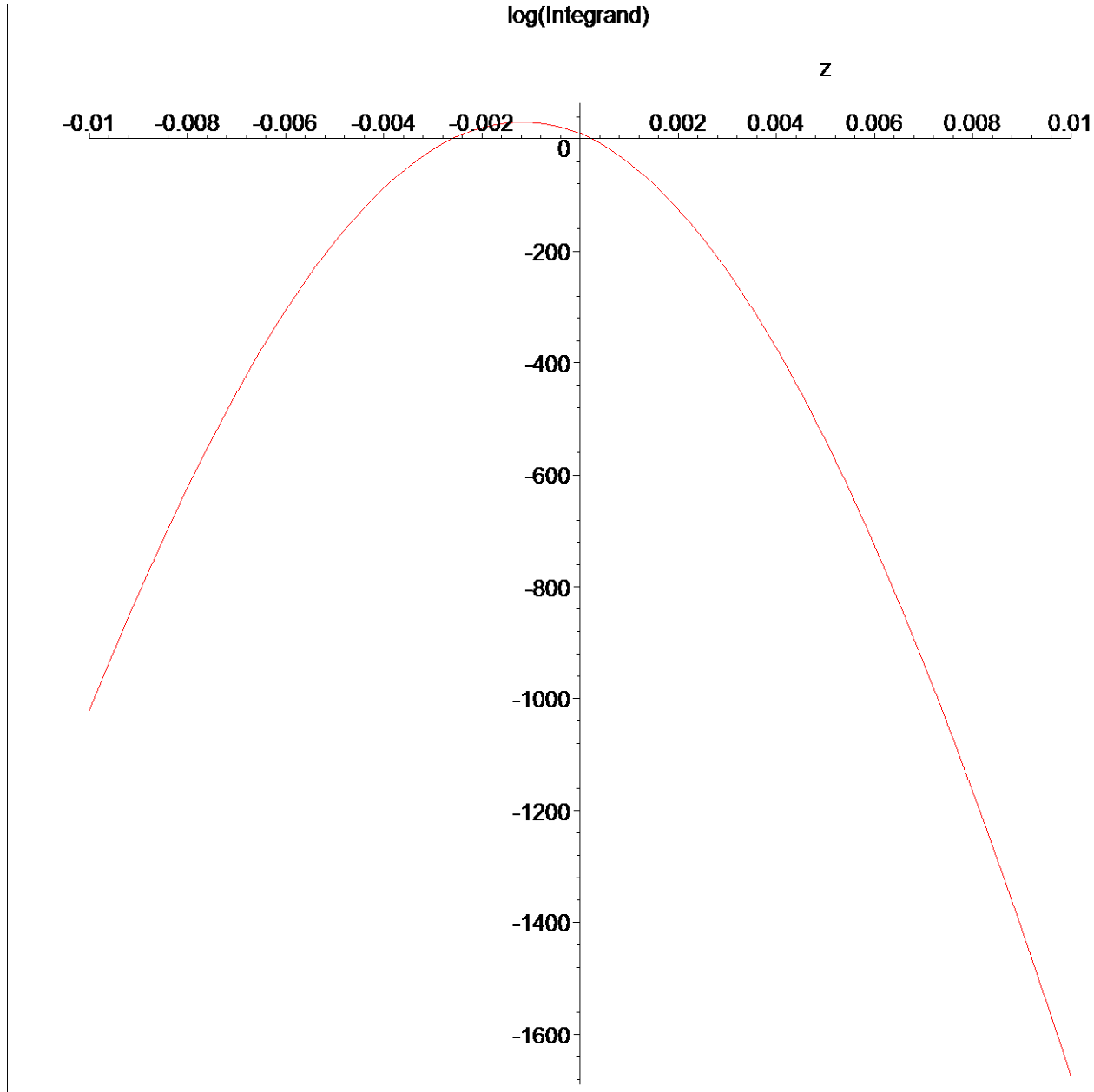
```
  Int(f,  min(zlower, zupper) .. max(zlower, zupper), method = _d01ajc);
  #Int(f,  min(zlower, zupper) .. max(zlower, zupper), method = _d01ajc, epsilon = eps);
  evalf(%);

  end proc;# maplemint(P);
P := proc(Xo, Yo, Xpo, Ypo, Zo)
local z, zmax, zlower, zupper, theIntegrand, logIntegrand, cutoff, eps, f, epsyH, epsyL, sigxL, sigxpL, sigyL, sigypL, sigxH, sigxpH, sigyH, sigypH,
Sigx, Sigy, Sigxp, Sigyp, Sigz;
global params, Params;
option hfloat;
    cutoff := –709.;
    eps := 6;
    epsyH := rH*epsxH;
    epsyL := rL*epsxL;
    sigxL := sqrt( epsxL*betxL );
    sigxpL := sqrt( epsxL / betxL );
    sigyL := sqrt( epsyL*betyL );
    sigypL := sqrt( epsyL / betyL );
    sigxH := sqrt( epsxH*betxH );
    sigxpH := sqrt( epsxH / betxH );
    sigyH := sqrt( epsyH*betyH );
    sigypH := sqrt( epsyH / betyH );
    Sigx := sqrt( sigxH^2 + sigxL^2 );
    Sigy := sqrt( sigyH^2 + sigyL^2 );
    Sigxp := sqrt( sigxpH^2 + sigxpL^2 );
    Sigyp := sqrt( sigypH^2 + sigypL^2 );
    Sigz := sqrt( szH^2 + szL^2 );
    logIntegrand := –1 / 2*( Zo – 2*z )*( Zo – 2*z ) / Sigz^2 – 1 / 2*( Xpo*z + Xo )*( Xpo*z + Xo ) / ( Sigx^2 + Sigxp^2*z^2 )
        – 1 / 2*( Yo + Ypo*z )*( Yo + Ypo*z ) / ( Sigy^2 + Sigyp^2*z^2 ) – 1 / 2*ln( ( Sigx^2 + Sigxp^2*z^2 )*( Sigy^2 + Sigyp^2*z^2 )*Sigz^2 );
    logIntegrand := eval( logIntegrand, params );
    zlower := fsolve( cutoff = logIntegrand, z = -1, z = –∞ .. 0 );
    if type( zlower, numeric ) then NULL else zlower := –1. end if;
    zupper := fsolve( cutoff = logIntegrand, z = -1 .. ∞, avoid = { z = zlower } );
    if type( zupper, numeric ) then NULL else zupper := 1. end if;
    if 5 < nargs then
        print( 'zlower' = zlower );
        print( 'zupper' = zupper );
        plot( [ logIntegrand ], z = –0.01 .. 0.01, title = "log(Integrand)", color = [ red, blue ] );
        print( % )
    end if;
    theIntegrand := exp( –1 / 2*( Zo – 2*z )*( Zo – 2*z ) / Sigz^2 – 1 / 2*( Xpo*z + Xo )*( Xpo*z + Xo ) / ( Sigx^2 + Sigxp^2*z^2 )
        – 1 / 2*( Yo + Ypo*z )*( Yo + Ypo*z ) / ( Sigy^2 + Sigyp^2*z^2 ) ) / sqrt( ( Sigx^2 + Sigxp^2*z^2 )*( Sigy^2 + Sigyp^2*z^2 )*Sigz^2 );
    theIntegrand := eval( theIntegrand, params );
    f := unapply( %, z );
    Int( f, min( zlower, zupper ) .. max( zlower, zupper ), method = _d01ajc );
    evalf( % )
end proc
> data:= 1e-4,0,0.083,0,0;
  P(data, info);
```

data := 0.0001, 0, 0.083, 0, 0

zlower = -0.00846236983607901

zupper = 0.00591230729754396



$$0.111129754183574\ 10^{11}$$

```
> 'Lum'(data);
  Int( op(1, %), op(2, %), method=_Sinc):
  eval(%, pre): eval(%, pre): eval(%, pre): eval(%, params);
  evalf(%);
```

$$\mathrm{Lum}(0.0001, 0, 0.083, 0, 0)$$

$$\mathrm{Int}\left(\frac{128.036879932896\ \mathbf{e}^{\left(-32786.8852459016\ z^2 - \frac{(0.083\ z + 0.0001)\,(0.0415000000000000\ z + 0.0000500000000000000)}{0.271750000000000\ 10^{-6}\ z^2 + 0.2272\ 10^{-9}}\right)}}{\sqrt{(0.271750000000000\ 10^{-6}\ z^2 + 0.2272\ 10^{-9})\,(0.749333333333333\ 10^{-7}\ z^2 + 0.61968\ 10^{-14})}},\ z = -\infty\ ..\ \infty,\ \mathrm{method} = \_\mathrm{Sinc}\right)$$

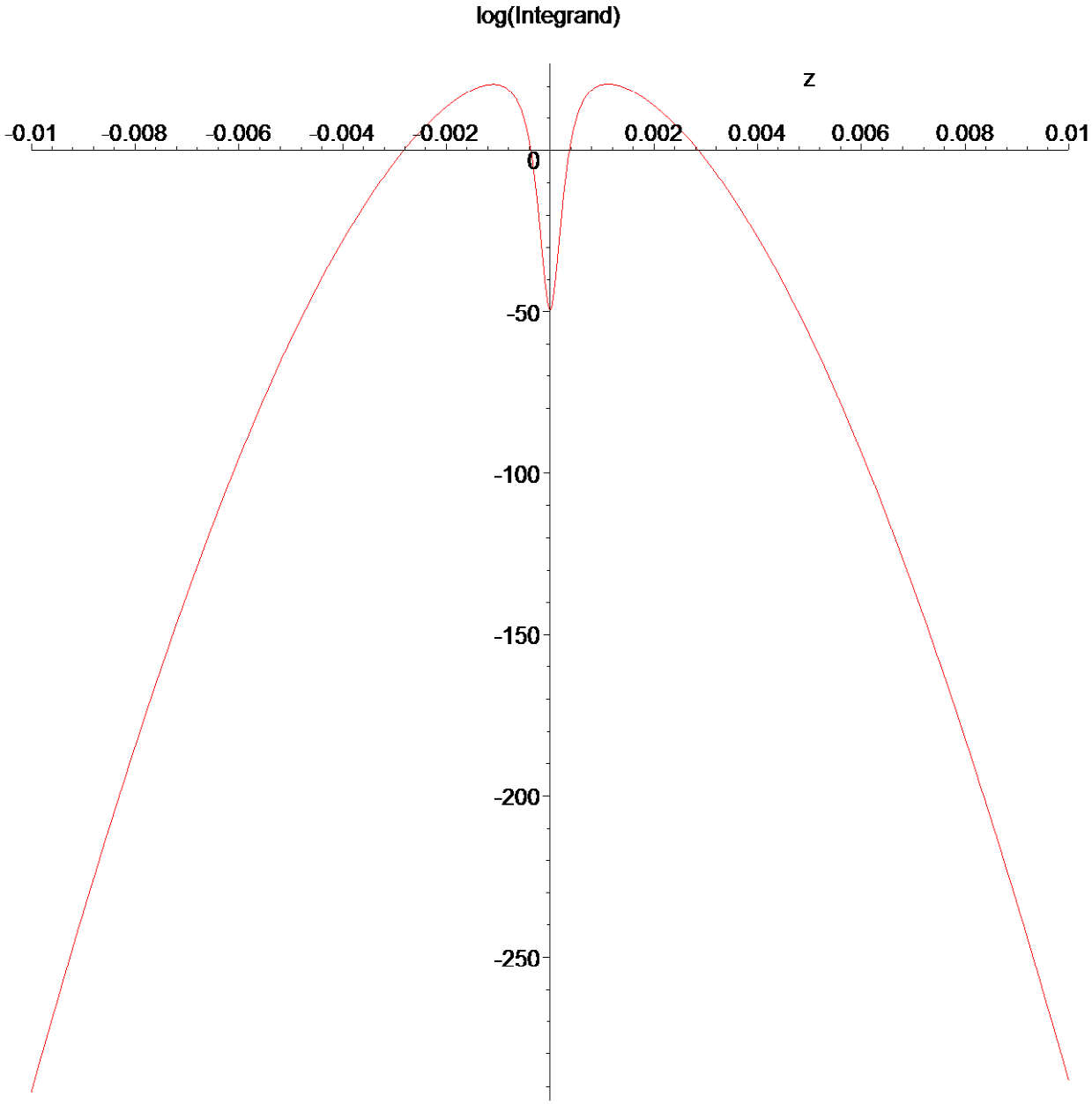
$$0.111129754183571\ 10^{11}$$

```
> data:=1e-6,1e-6,0.04,1e-5,1e-2;
  P(data, info);
```

$$\mathrm{data} := 0.1\ 10^{-5}, 0.1\ 10^{-5}, 0.04, 0.00001, 0.01$$

$$\mathrm{zlower} = -0.0165146920609301$$

$$\mathrm{zupper} = 0.0166106063302982$$



893240.517411864

```
> 'Lum'(data);
  #Int( op(1, %), op(2, %), method=_Sinc, epsilon = 10):
  Int( op(1, %), op(2, %), method=_Sinc):
  eval(%, pre): eval(%, pre): eval(%, pre): eval(%, params):
  evalf(%);
```

$$Lum(0.1\ 10^{-5}, 0.1\ 10^{-5}, 0.04, 0.00001, 0.01)$$

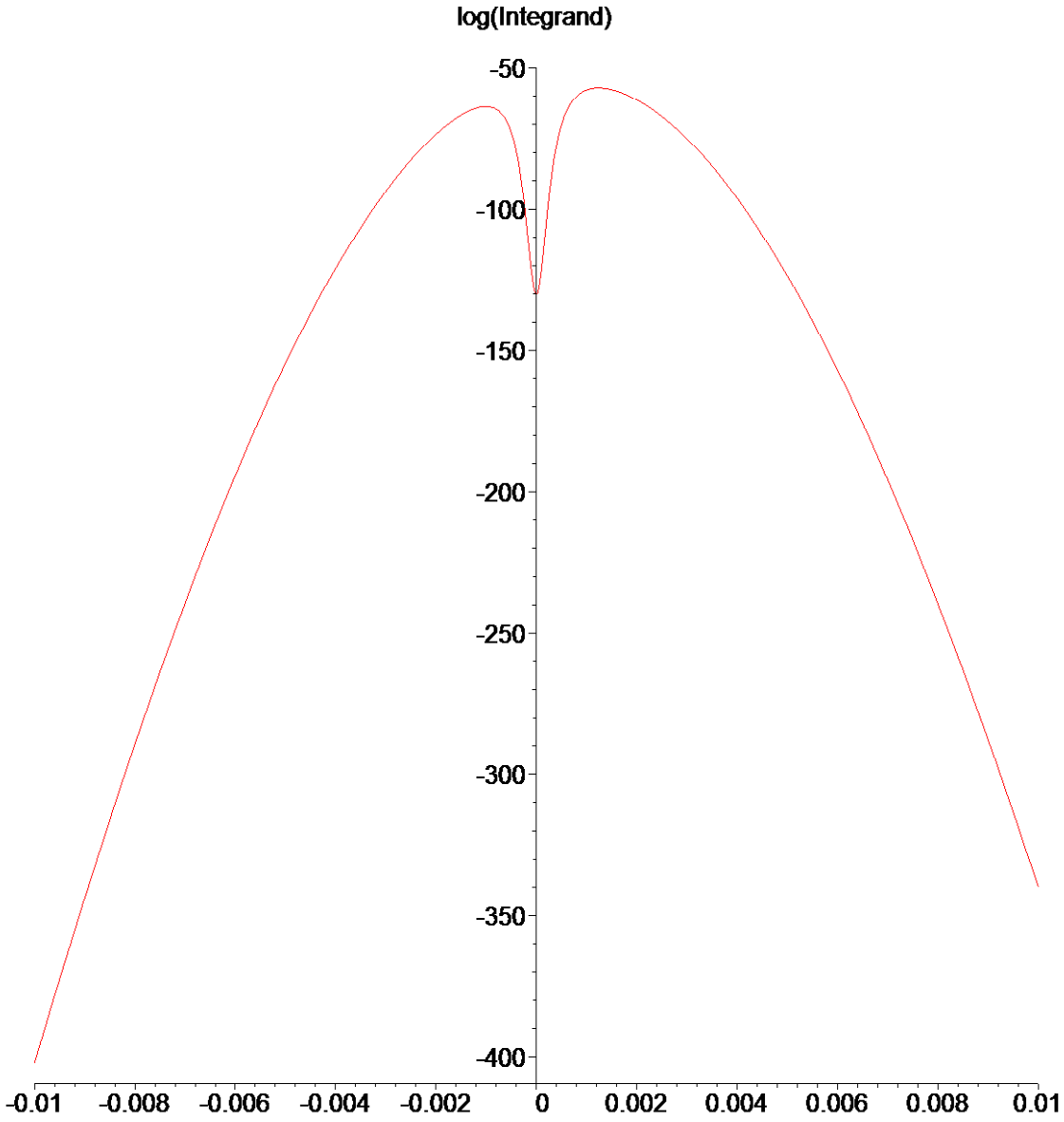
893240.517411864

```
> 
> data:=1e-6,1e-6,0.04,1e-5,1e-1;
  P(data, info);
```

$$data := 0.1\ 10^{-5}, 0.1\ 10^{-5}, 0.04, 0.00001, 0.1$$

zlower = -0.0146631399593735

zupper = 0.0161088492826565



$$0.999403491234243\ 10^{-28}$$

```
> 'Lum'(data);
  #Int( op(1, %), op(2, %), method=_Sinc, epsilon=6):
  Int( op(1, %), op(2, %), method=_Sinc):
  eval(%, pre): eval(%, pre): eval(%, pre): eval(%, params):
  evalf(%);
```

$$\mathrm{Lum}(0.1\ 10^{-5}, 0.1\ 10^{-5}, 0.04, 0.00001, 0.1)$$

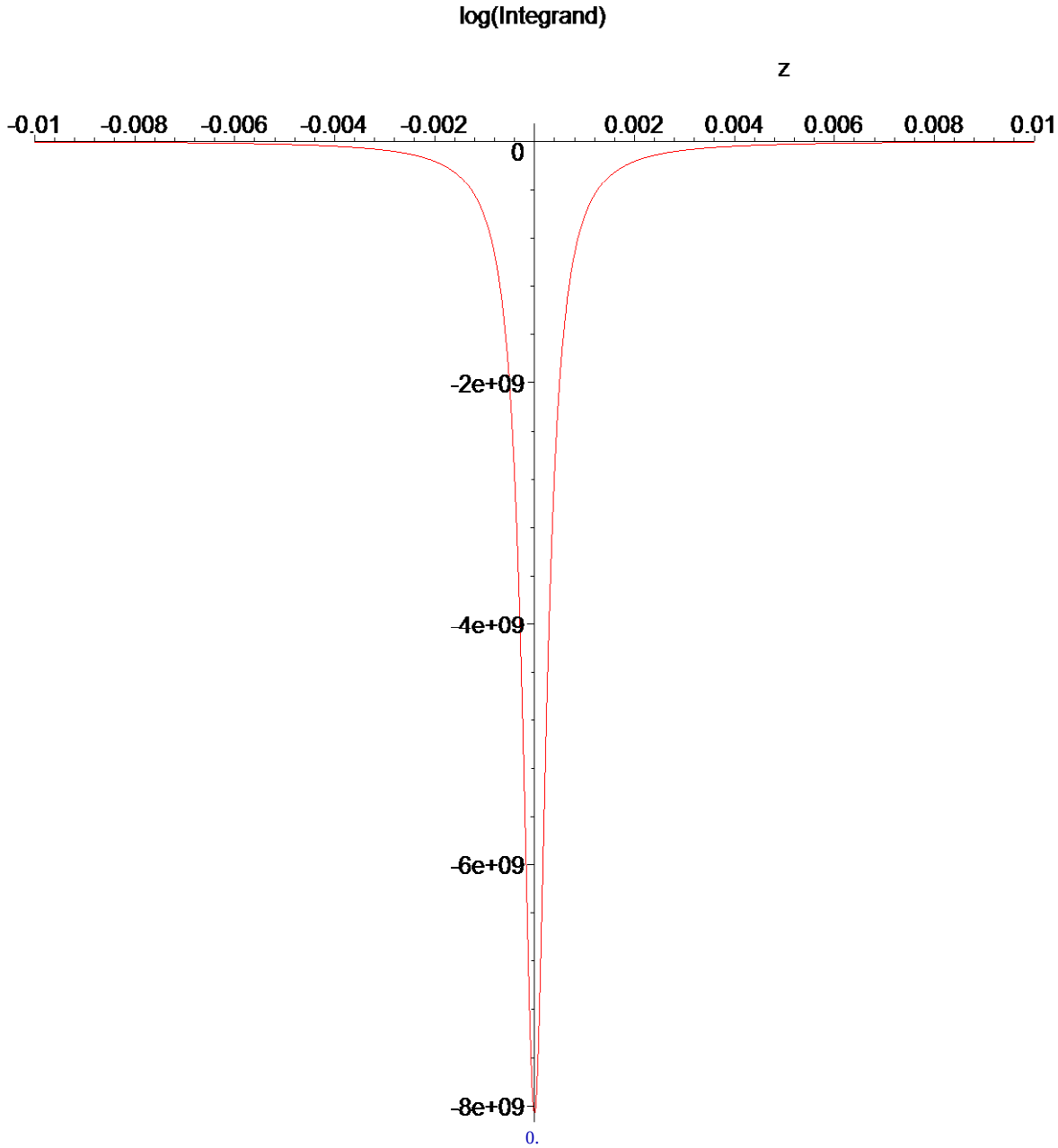
$$0.224617773610585\ 10^{-31}$$

```
> data:=1e-6,1e-2,0.04,1e-5,1e-1;
  P(data, info);
```

$$\mathrm{data} := 0.1\ 10^{-5}, 0.01, 0.04, 0.00001, 0.1$$

zlower = -1.

zupper = 1.



0.

```
> 'Lum'(data);
  Int( op(1, %), op(2, %), method=_Sinc, digits=6):
  eval(%, pre): eval(%, pre): eval(%, pre): eval(%, params):
  evalf(%);
```

$$Lum(0.1\ 10^{-5}, 0.01, 0.04, 0.00001, 0.1)$$

0.

```
> 
> cycles:=40:periods:=50;
  L:=[0$cycles];
  times:=[0.0$periods];
```

$$periods := 50$$

$$L := [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$times := [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]$$

```
> rand()/999999999999*1e-4;
```

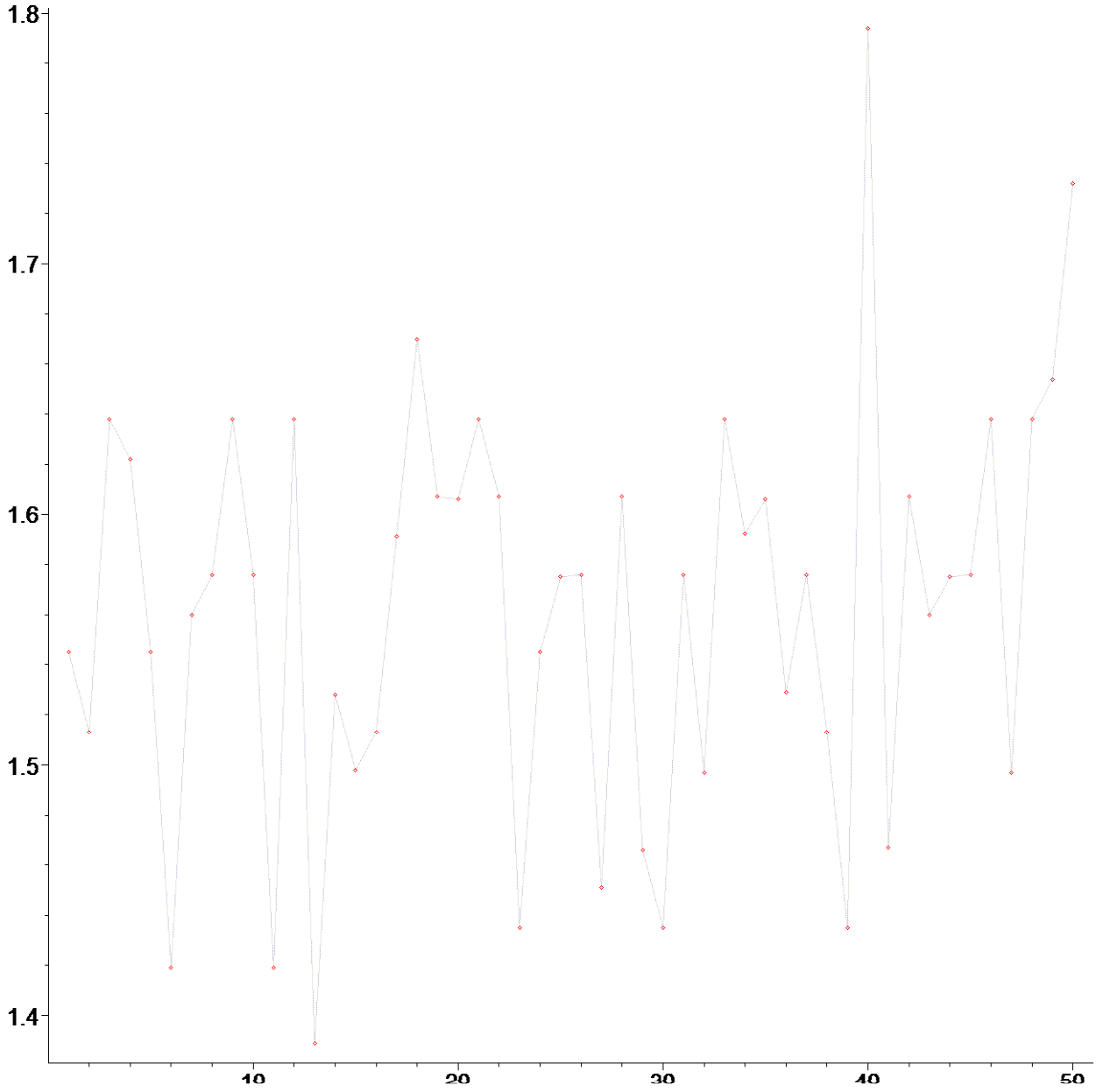
$$0.0000895038201152895$$

```
> with(RandomTools):
> 
> for j from 1 to periods do
  st:=time():
    for i from 1 to cycles do
      t:=rand()/999999999999*1e-4;
  #t:=Generate(float(range=0.0 .. 1e-4));
```

```
      L[i]:=P(t,0,0.083,0,0);
    end do:
  times[j]:=time()-st;
  end do:
```

```
> #L;
> times:
  Statistics:-PointPlot(times, color="LightGrey", style=line):
  Statistics:-PointPlot(times, color=red):
  plots:-display(%,%%);

  `average time per period` = convert(times, `+`)/periods;
  `average time per integral` = rhs(%)/cycles;
```



average time per period = 1.56252000000000

average time per integral = 0.0390630000000000

```
>
```