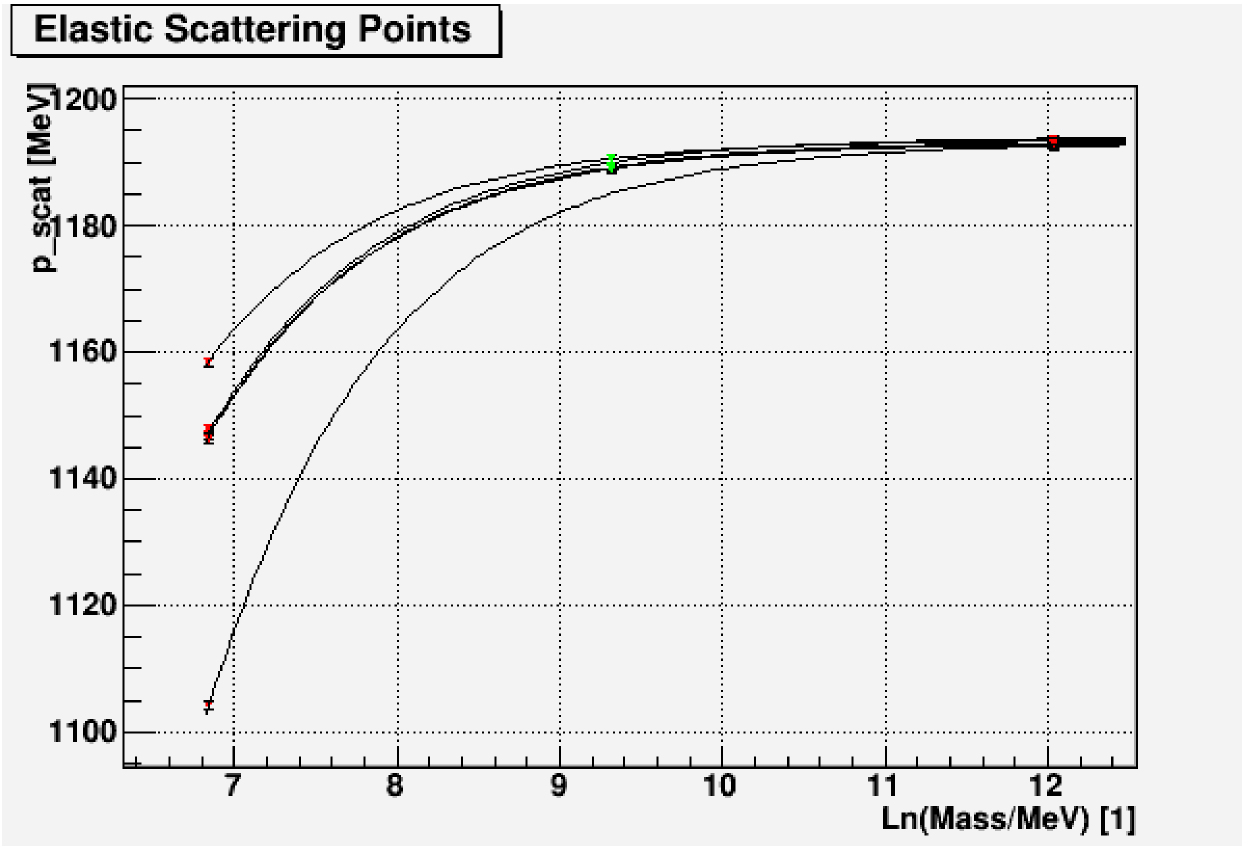
Motivation

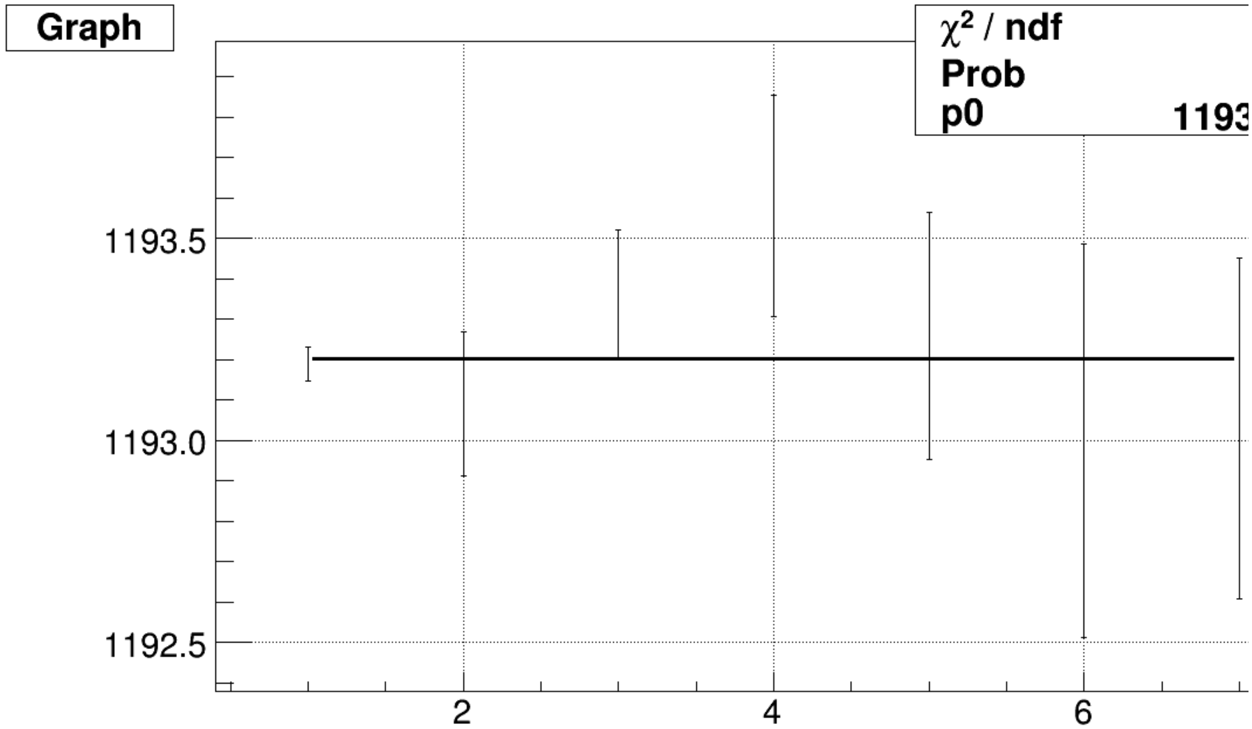
In[75]:=
```
Import["/home/miham/JLab/e04007/ElasticFitAll.png", ImageSize → {600, 400}]
```

Out[75]=



In[77]:= `Import["/home/miham/JLab/e04007/final_fit.png", ImageSize → {700, 400}]`

Out[77]=

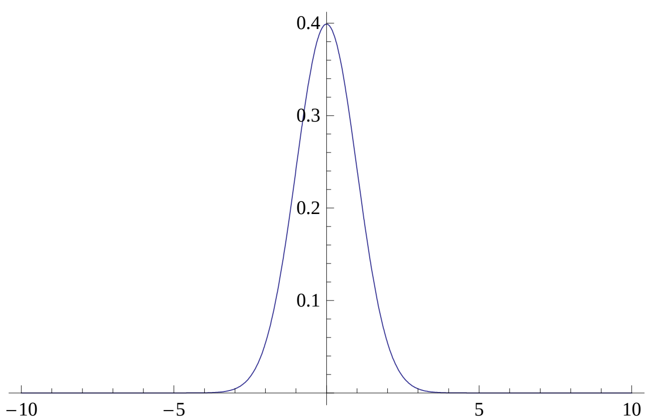


Convolution of two Gaussian Functions

In[14]:= `Gauss[x_, b_, σ_] :=` $\frac{1}{\sqrt{2\pi}\,\sigma} E^{\left(\frac{-(x-b)^2}{2\sigma^2}\right)}$

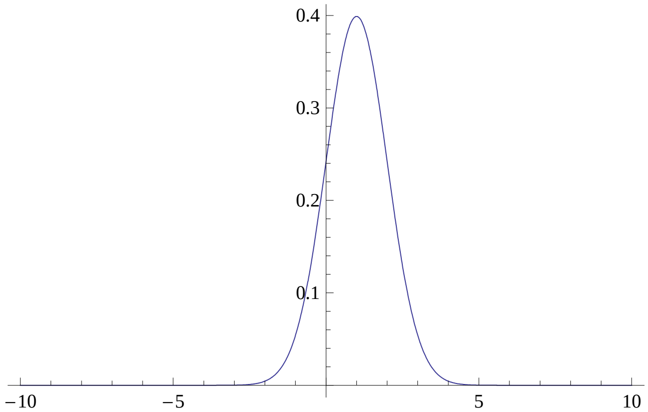
In[15]:= `Plot[Gauss[x, 0, 1], {x, -10, 10}, PlotRange → All]`

Out[15]=



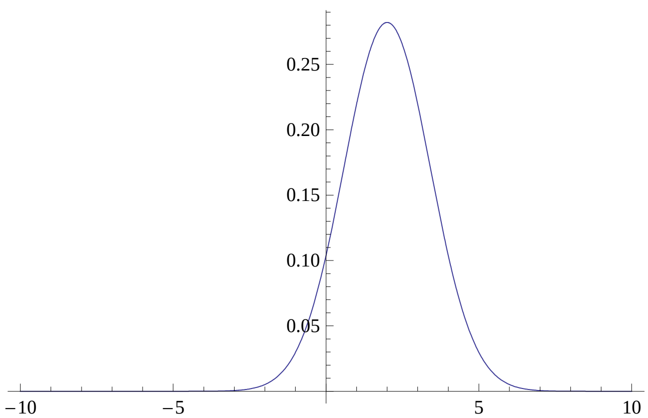
```
In[16]:= N[Integrate[Gauss[x, 0, 1], {x, -100, 100}]]

Out[16]= 1.

In[17]:= convol[y_] := N[Integrate[Gauss[x, 0, 1] DiracDelta[y - x - 1], {x, -1000, 1000}]]

In[18]:= convol[1]

Out[18]= 0.398942

In[19]:= Plot[convol[y], {y, -10, 10}, PlotRange → All]
```

Out[19]=




```
In[20]:= convol2[y_] := N[Integrate[Gauss[x, 0, 1] Gauss[y - x, 2, 1], {x, -1000, 1000}]]

In[21]:= convol2[1]

Out[21]= 0.219696

In[22]:= Plot[convol2[y], {y, -10, 10}, PlotRange → All]
```

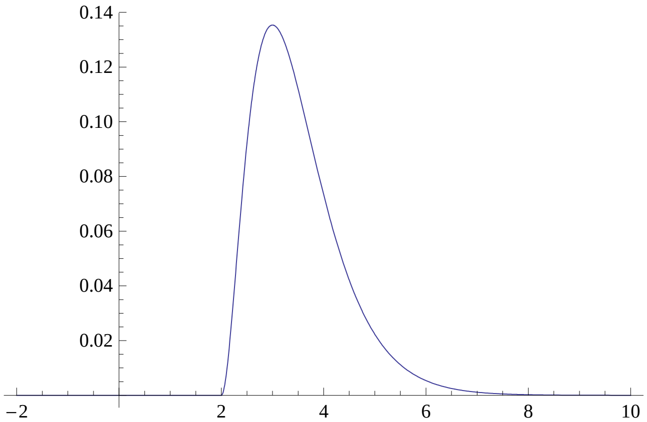
Out[22]=



Convolution with Gauss

$$\texttt{FunWithTail}[x\_, b\_, \sigma\_] := E^{\frac{-(x-b)}{\sigma}} (x-b)^2 \frac{1}{1 + E^{\frac{-(x-b)}{0.001}}}$$

```
Plot[FunWithTail[x, 2, 0.5, 0], {x, -2, 10}, PlotRange → All]
```




```
FindMaximum[FunWithTail[x, 2, 0.5, 0], {x, 4}]
```

```
{0.135335, {x → 3.}}
```

```
NIntegrate[FunWithTail[x, 2, 0.5, 0] * x, {x, 0, 1000}, MinRecursion → 7] /
 NIntegrate[FunWithTail[x, 2, 0.5, 0], {x, 0, 1000}, MinRecursion → 7]
```

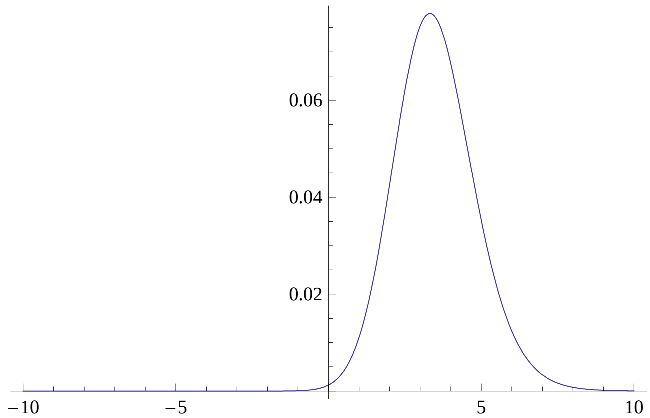
```
3.5
```

```
convol3[y_] :=
 NIntegrate[Gauss[x, 0, 1] FunWithTail[y - x, 2, 0.5], {x, -100, 100}, MinRecursion → 7]
```

```
convol3[1]
```

```
0.0111072
```

```
Plot[convol3[y], {y, -10, 10}]
```




```
Last[SortBy[Table[{x, convol3[x]}, {x, 0, 10, 0.1}], Last]]
```

{3.3, 0.0779028}

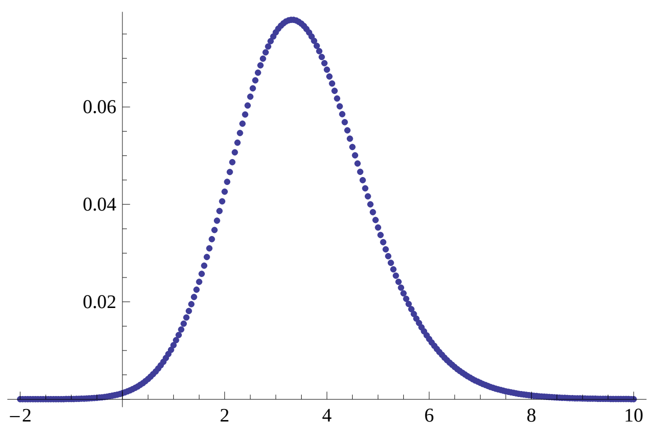
Next Step - width dependence

```
convol4[y_, sigma_] := NIntegrate[Gauss[x, 0, sigma] FunWithTail[y - x, 2, 0.5],
  {x, -100, 100}, MinRecursion → 10, MaxRecursion → 20]
```

```
convol4[1, 0.1]
```

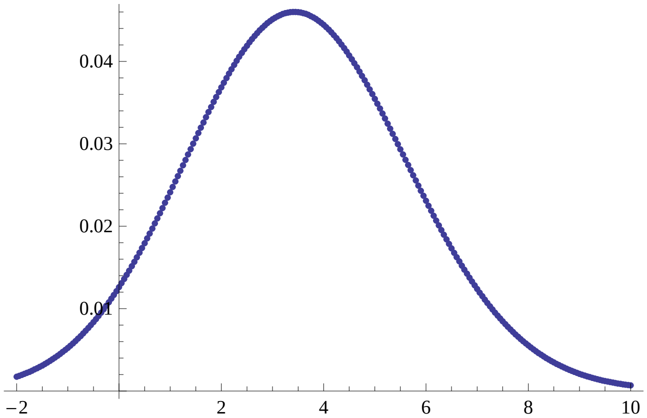
1.37299×10^{-27}

```
values = Table[{x, convol4[x, 1]}, {x, -2.0, 10, 0.05}];
kat1 = ListPlot[values]
res1 = {Last[SortBy[values, Last]], 1}
```



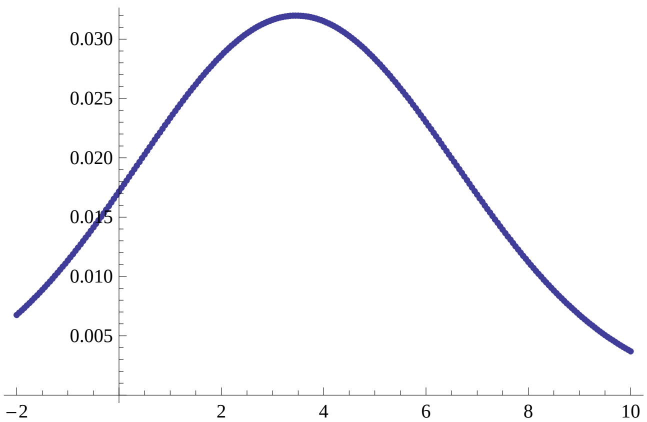
{{3.3, 0.0779028}, 1}

```
values = Table[{x, convol4[x, 2]}, {x, -2.0, 10, 0.05}];
kat2 = ListPlot[values]
res2 = {Last[SortBy[values, Last]], 2}
```



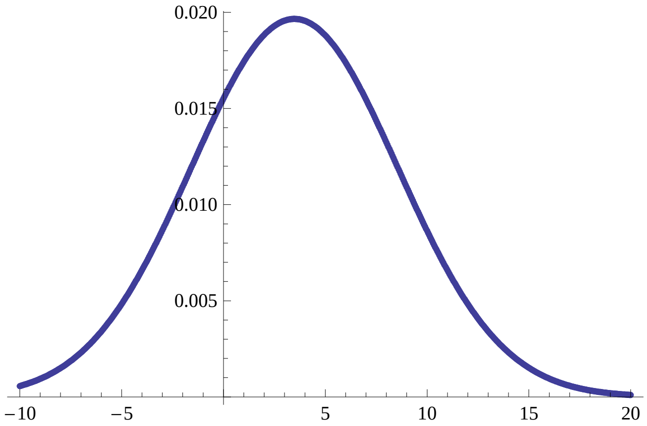
{{3.45, 0.0459985}, 2}

```
values = Table[{x, convol4[x, 3]}, {x, -2.0, 10, 0.05}];
kat3 = ListPlot[values]
res3 = {Last[SortBy[values, Last]], 3}
```



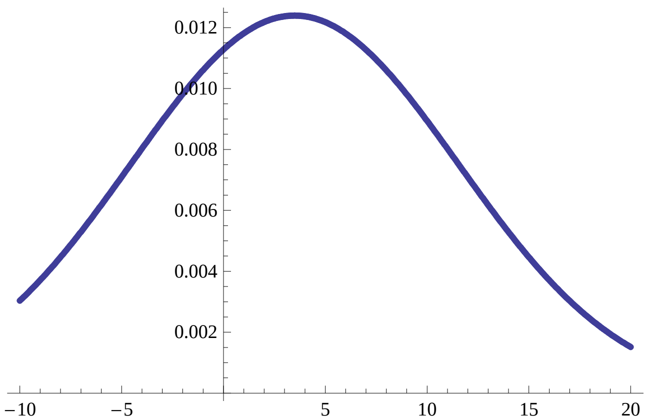
{{3.45, 0.0319834}, 3}

```
values = Table[{x, convol4[x, 5]}, {x, -10.0, 20, 0.05}];
kat4 = ListPlot[values]
res4 = {Last[SortBy[values, Last]], 5}
```



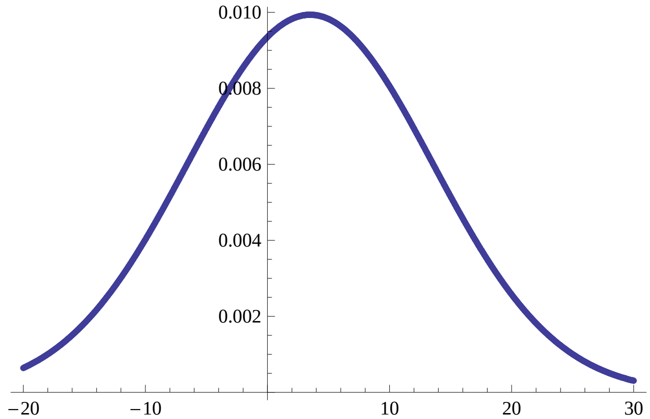

```
{{3.5, 0.0196584}, 5}
```

```
values = Table[{x, convol4[x, 8]}, {x, -10.0, 20, 0.05}];
kat5 = ListPlot[values]
res5 = {Last[SortBy[values, Last]], 8}
```



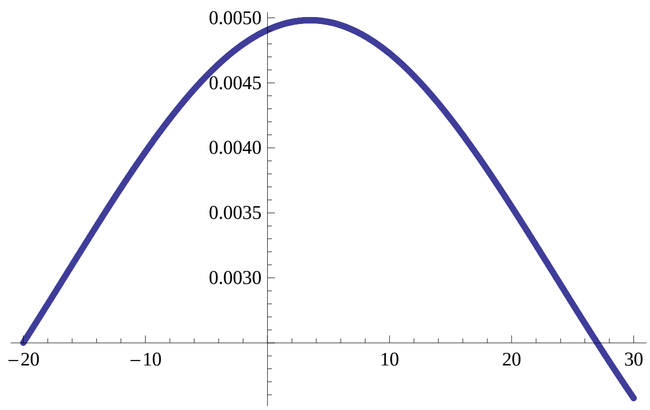

```
{{3.5, 0.0123949}, 8}
```

```
values = Table[{x, convol4[x, 10]}, {x, -20.0, 30, 0.05}];
kat6 = ListPlot[values]
res6 = {Last[SortBy[values, Last]], 10}
```



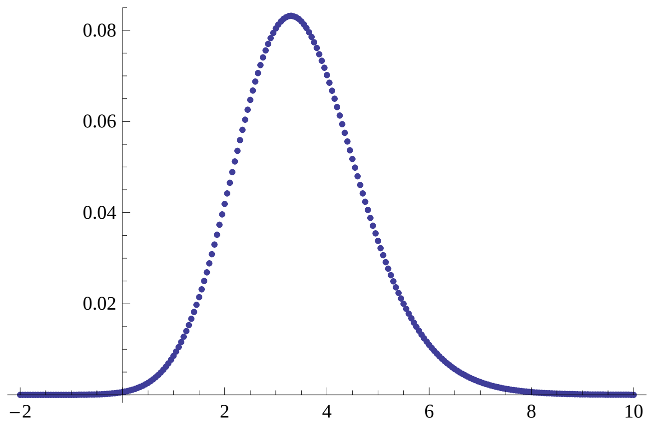
{{3.5, 0.0099365}, 10}

```
values = Table[{x, convol4[x, 20]}, {x, -20.0, 30, 0.05}];
kat7 = ListPlot[values]
res7 = {Last[SortBy[values, Last]], 20}
```



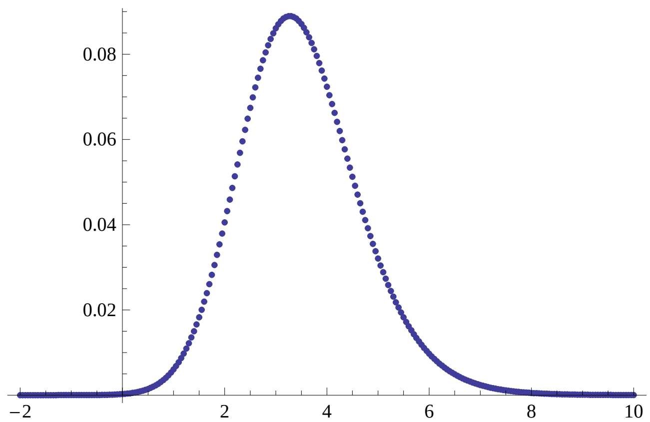
{{3.5, 0.00498211}, 20}

```
values = Table[{x, convol4[x, 0.9]}, {x, -2.0, 10, 0.05}];
kat8 = ListPlot[values]
res8 = {Last[SortBy[values, Last]], 0.9}
```



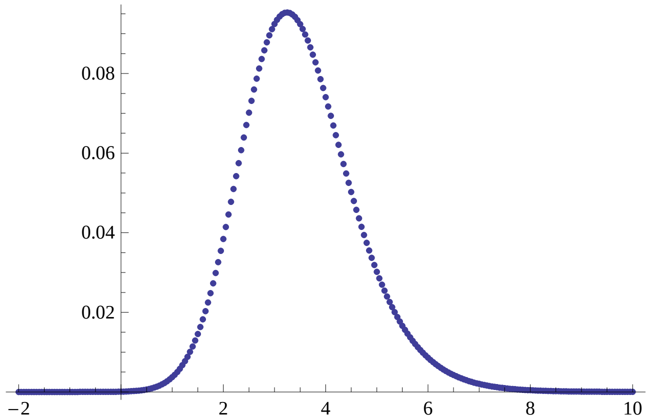
{{3.3, 0.0831682}, 0.9}

```
values = Table[{x, convol4[x, 0.8]}, {x, -2.0, 10, 0.05}];
kat9 = ListPlot[values]
res9 = {Last[SortBy[values, Last]], 0.8}
```



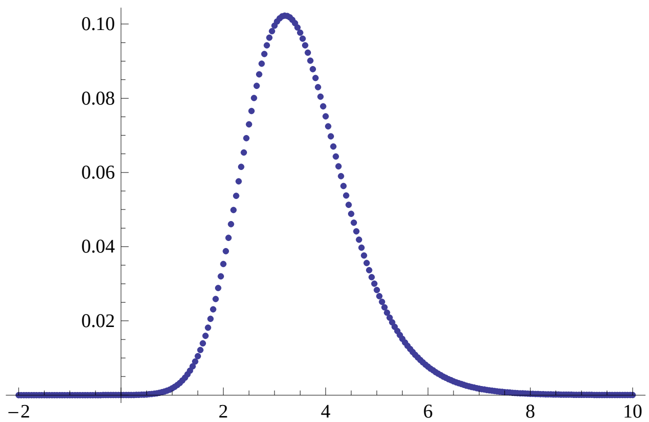
{{3.25, 0.0889517}, 0.8}

```
values = Table[{x, convol4[x, 0.7]}, {x, -2.0, 10, 0.05}];
kat10 = ListPlot[values]
res10 = {Last[SortBy[values, Last]], 0.7}
```



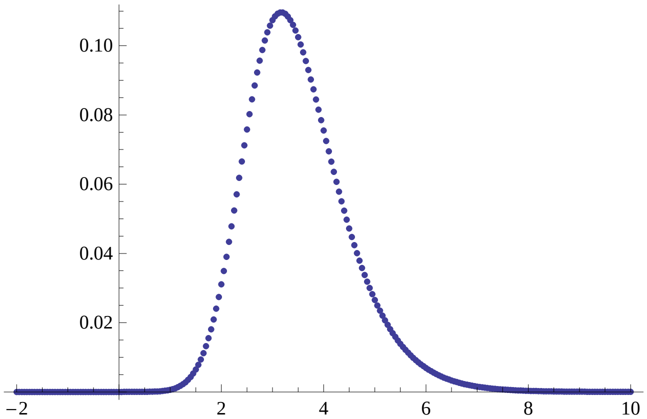
{{3.25, 0.09535}, 0.7}

```
values = Table[{x, convol4[x, 0.6]}, {x, -2.0, 10, 0.05}];
kat11 = ListPlot[values]
res11 = {Last[SortBy[values, Last]], 0.6}
```



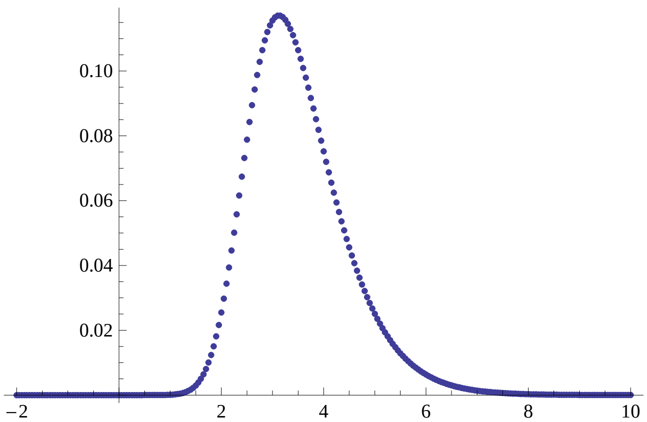
{{3.2, 0.102273}, 0.6}

```
values = Table[{x, convol4[x, 0.5]}, {x, -2.0, 10, 0.05}];
kat12 = ListPlot[values]
res12 = {Last[SortBy[values, Last]], 0.5}
```



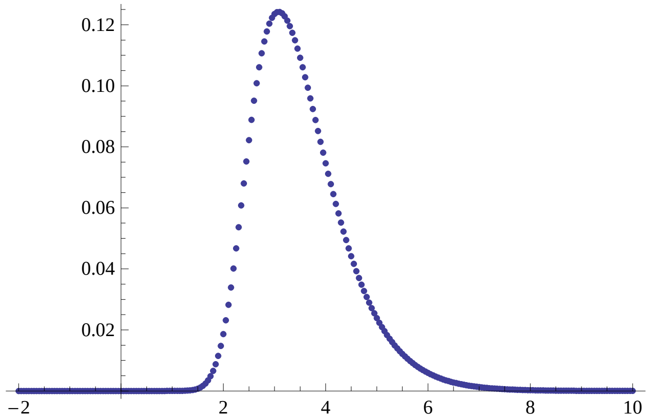

```
{{3.15, 0.109609}, 0.5}
```

```
values = Table[{x, convol4[x, 0.4]}, {x, -2.0, 10, 0.05}];
kat13 = ListPlot[values]
res13 = {Last[SortBy[values, Last]], 0.4}
```



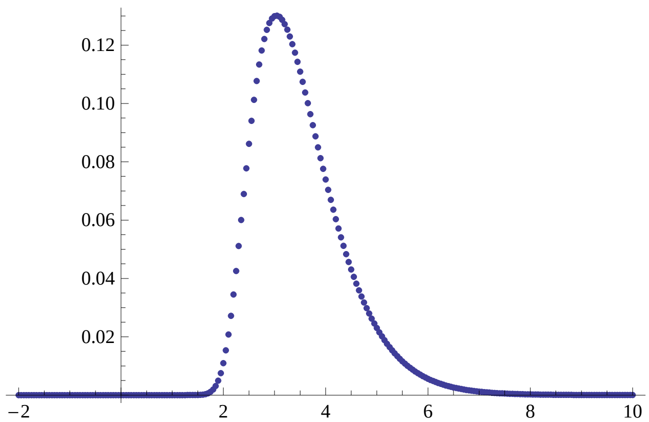

```
{{3.15, 0.117105}, 0.4}
```

```
values = Table[{x, convol4[x, 0.3]}, {x, -2.0, 10, 0.05}];
kat14 = ListPlot[values]
res14 = {Last[SortBy[values, Last]], 0.3}
```



{{3.1, 0.124217}, 0.3}

```
values = Table[{x, convol4[x, 0.2]}, {x, -2.0, 10, 0.05}];
kat15 = ListPlot[values]
res15 = {Last[SortBy[values, Last]], 0.2}
```



{{3.05, 0.130122}, 0.2}

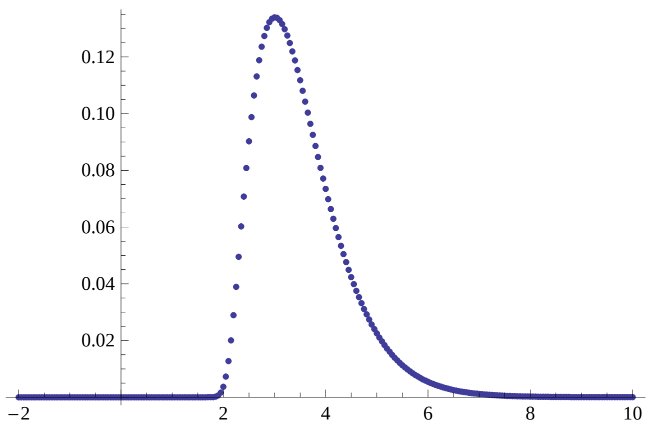
```
values = Table[{x, convol4[x, 0.1]}, {x, -2.0, 10, 0.05}];
kat16 = ListPlot[values]
res16 = {Last[SortBy[values, Last]], 0.1}
```

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. >>

NIntegrate::ncvb : NIntegrate failed to converge to prescribed accuracy after 20 recursive bisections in x near {x} = {-3.29022}. NIntegrate obtained 5.859793269572613‘15.954589770191005*^-546 and 6.409691485196267‘15.954589770191005*^-551 for the integral and error estimates. >>

NIntegrate::slwcon : Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. >>

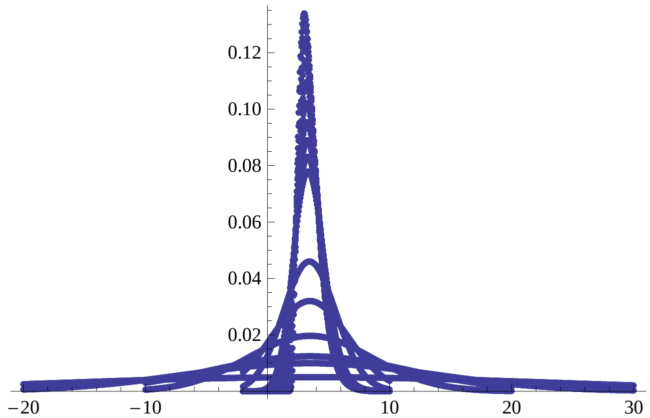
```
NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 20 recursive
   bisections in x near {x} = {-3.24022}. NIntegrate
   obtained 7.16083417832366‘15.954589770191005*^-539 and
   9.04066308804714‘15.954589770191005*^-544 for the integral and error estimates. >>

NIntegrate::slwcon :
 Numerical integration converging too slowly; suspect one of the following:
   singularity, value of the integration is 0, highly
   oscillatory integrand, or WorkingPrecision too small. >>

General::stop :
 Further output of NIntegrate::slwcon will be suppressed during this calculation. >>

NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 20 recursive
   bisections in x near {x} = {-3.19022}. NIntegrate
   obtained 6.815565565130211‘15.954589770191005*^-532 and
   1.168830685960116‘15.954589770191005*^-536 for the integral and error estimates. >>

General::stop :
 Further output of NIntegrate::ncvb will be suppressed during this calculation. >>

NIntegrate::eincr :
 The global error of the strategy GlobalAdaptive has increased more than 400
   times. The global error is expected to decrease monotonically after a
   number of integrand evaluations. Suspect one of the following: the working
   precision is insufficient for the specified precision goal; the integrand
   is highly oscillatory or it is not a (piecewise) smooth function; or the
   true value of the integral is 0. Increasing the value of the GlobalAdaptive
   option MaxErrorIncreases might lead to a convergent numerical integration.
   NIntegrate obtained 4.735299338899453‘15.954589770191005*^-321 and
   8.883685526692691‘15.954589770191005*^-327 for the integral and error estimates. >>
```




```
{{3., 0.133982}, 0.1}
```

Results

```
Show[kat1, kat2, kat3, kat4, kat5, kat6, kat7, kat8, kat9,
 kat10, kat11, kat12, kat13, kat14, kat15, kat16, PlotRange → All]
```



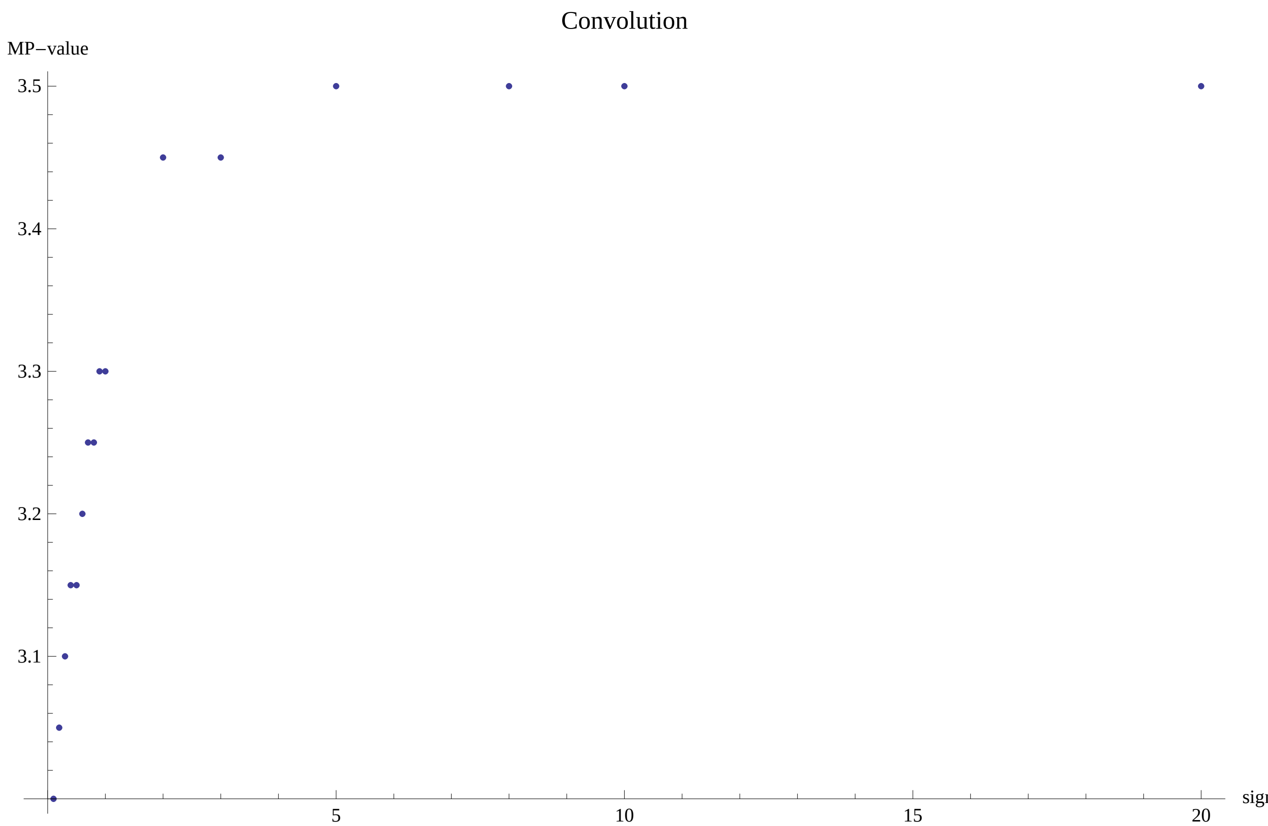

```
tocke = {res1, res2, res3, res4, res5, res6,
  res7, res8, res9, res10, res11, res12, res13, res14, res15, res16}
```

```
{{{3.3, 0.0779028}, 1}, {{3.45, 0.0459985}, 2}, {{3.45, 0.0319834}, 3}, {{3.5, 0.0196584}, 5},
 {{3.5, 0.0123949}, 8}, {{3.5, 0.0099365}, 10}, {{3.5, 0.00498211}, 20},
 {{3.3, 0.0831682}, 0.9}, {{3.25, 0.0889517}, 0.8}, {{3.25, 0.09535}, 0.7},
 {{3.2, 0.102273}, 0.6}, {{3.15, 0.109609}, 0.5}, {{3.15, 0.117105}, 0.4},
 {{3.1, 0.124217}, 0.3}, {{3.05, 0.130122}, 0.2}, {{3., 0.133982}, 0.1}}
```

```
coordinates = Map[{#[[2]], #[[1]][[1]]} &, tocke]
```

```
{{1, 3.3}, {2, 3.45}, {3, 3.45}, {5, 3.5}, {8, 3.5}, {10, 3.5}, {20, 3.5}, {0.9, 3.3}, {0.8, 3.25},
 {0.7, 3.25}, {0.6, 3.2}, {0.5, 3.15}, {0.4, 3.15}, {0.3, 3.1}, {0.2, 3.05}, {0.1, 3.}}
```

```
ListPlot[coordinates, ImageSize → {600, 500},
 AxesLabel → {"sigma", "MP-value"}, PlotLabel → "Convolution"]
```



How this affects real measurements & Energy Losses :

The approximate width of the energy loss distribution for hydrogen is approximatey 0.3 MeV. The width of the momentum distribution is approximately 4.0 MeV. The ratio in widths is approximately 10. That means that we are in the regime, where Most probable value for the Energy Loses equals Mean Energy losses.

Energy Losses

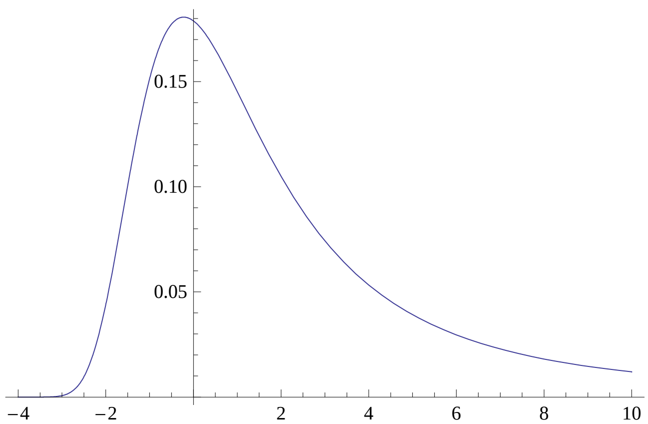
In[23]:=
```
Phi[λ_] := N[1/π
    NIntegrate[E^(-u Log[u] - u λ) Sin[π u], {u, 0, 100},
     MaxRecursion → 6, AccuracyGoal → 4]]
   UnitStep[λ + 4] UnitStep[30 - λ]
```

In[24]:= `Phi[0]`

Out[24]= 0.178854

In[25]:= `Plot[Phi[x], {x, -4, 10}]`

Out[25]=



In[26]:= `momentum = 1193.5`

Out[26]= 1193.5

In[27]:=
$$beta[p\_] := \frac{p}{\sqrt{p^2 + 0.511^2}}$$

$$gamma[p\_] := \frac{1}{\sqrt{1 - beta[p]^2}}$$

In[29]:=
$$Xi[x\_, \rho\_, \beta\_, A\_, Z\_] := 0.1535 * \rho \frac{Z}{A} \left(\frac{1}{\beta}\right)^2 x$$

In[30]:=
```
Epsilon[β_, I_] := Exp[β^2 + Log[((1.0 - β^2) * I^2)/(2 * 0.511 * β^2)]]
```

In[31]:=
```
Lambda[Δ_, x_, ρ_, β_, A_, Z_, I_] := 1/Xi[x, ρ, β, A, Z]
  (Δ - Xi[x, ρ, β, A, Z] (Log[Xi[x, ρ, β, A, Z]] - Log[Epsilon[β, I]] + 1 - 0.577))
```

In[32]:=
```
f[Δ_, x_, ρ_, β_, A_, Z_, I_] := Phi[Lambda[Δ, x, ρ, β, A, Z, I]]/Xi[x, ρ, β, A, Z]
```

In[33]:=
```
beta[momentum]
```

Out[33]= 1.

In[34]:=
```
Epsilon[beta[momentum], 21.8 * 10^-6]
```

Out[34]= 2.31715×10^{-16}

In[35]:=
```
Lambda[2.0, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8 * 10^-6]
```

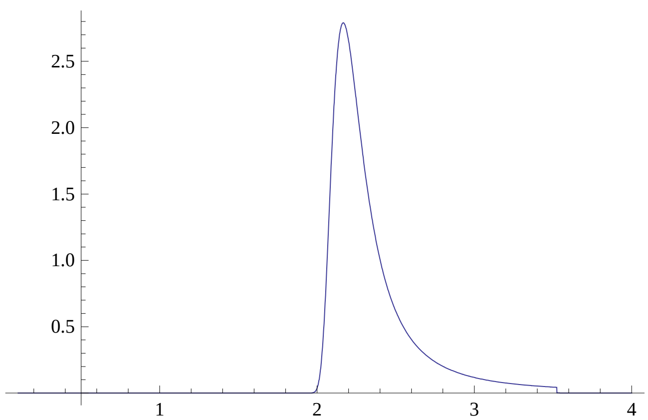
Out[35]= -2.79561

In[36]:=
```
f[2.0, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8 * 10^-6]
```

Out[36]= 0.0368642

In[37]:=
```
Plot[f[x, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8 * 10^-6], {x, 0.1, 4}, PlotRange → All]
```

Out[37]=



```
FindMaximum[f[x, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8*10^-6], {x, 2.1}]
```

NIntegrate::inumr :
The integrand $e^{-15.4455\,u\,(-2.181+x)-u\,\mathrm{Log}[u]}$ Sin[π u] has evaluated to non-numerical values for all sampling points in the region with boundaries {{0, 100}}. >>

NIntegrate::inumr :
The integrand $2.71828^{-15.4455\,u\,(-2.181+x)-1.\,u\,\mathrm{Log}[u]}$ Sin[3.14159 u] has evaluated to non-numerical values for all sampling points in the region with boundaries {{0., 100.}}. >>

NIntegrate::inumr :
The integrand $2.71828^{-15.4455\,u\,(-2.181+x)-1.\,u\,\mathrm{Log}[u]}$ Sin[3.14159 u] has evaluated to non-numerical values for all sampling points in the region with boundaries {{0., 100.}}. >>

General::stop :
Further output of NIntegrate::inumr will be suppressed during this calculation. >>

{2.79033, {x → 2.16657}}

```
f[2.2, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8*10^-6]
```

2.65763

```
Last[SortBy[Table[
   {x, f[x, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8*10^-6]}, {x, 2.0, 2.2, 0.001}], Last]]
```

{2.167, 2.7903}

In[38]:=
```
DeltaMP [x_, ρ_, β_, A_, Z_, I_] := Xi[x, ρ, β, A, Z] * (Log[Xi[x, ρ, β, A, Z] / Epsilon[β, I]] + 0.198 - 0.0)
```

In[39]:=
```
ELOSSH2 = DeltaMP[ 6.0, 0.071, beta[momentum], 1.01, 1, 21.8*10^-6]
```

Out[39]= 2.16643

```
NIntegrate[f[dE, 6.0, 0.071, beta[momentum], 1.01, 1, 21.8*10^-6]*dE, {dE, 0.01, 4}]
```

```
NIntegrate::inumr :
 The integrand ⅇ^(-15.4455 (-2.181+dE) u-u Log[u]) Sin[π u] has evaluated to non-numerical
   values for all sampling points in the region with boundaries {{0, 100}}. >>

NIntegrate::inumr :
 The integrand 2.71828^(-15.4455 (-2.181+dE) u-1. u Log[u]) Sin[3.14159 u] has evaluated
   to non-numerical values for all sampling points
   in the region with boundaries {{0., 100.}}. >>

NIntegrate::inumr :
 The integrand 2.71828^(-15.4455 (-2.181+dE) u-1. u Log[u]) Sin[3.14159 u] has evaluated
   to non-numerical values for all sampling points
   in the region with boundaries {{0., 100.}}. >>

General::stop :
 Further output of NIntegrate::inumr will be suppressed during this calculation. >>

NIntegrate::slwcon :
 Numerical integration converging too slowly; suspect one of the following:
   singularity, value of the integration is 0, highly
   oscillatory integrand, or WorkingPrecision too small. >>

NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 9 recursive bisections
   in dE near {dE} = {3.52457}. NIntegrate obtained 2.2120599511267165`
   and 0.000031572671841471137` for the integral and error estimates. >>

2.21206
```

In[41]:=
```
Ffac[τ_, β_] := 1 - β^2 + (τ^2/8 - (2 τ + 1) Log[2])/(τ + 1)^2
```

In[42]:=
```
Cfac[i_, η_] := (0.422377 η^-2 + 0.0304043 η^-4 - 0.00038106 η^-6)*10^6*i^2 +
  (3.850190 η^-2 - 0.1667989 η^-4 + 0.00157955 η^-6)*10^9 i^3
```

In[43]:=
```
deltafac[X_, C0_, a_, m_, X1_, X0_] :=
 If[X ≤ X0, 0, If[X ≤ X1, 4.6052 X + C0 + a (X1 - X)^m, 4.6052 X + C0]]
```

In[44]:=
```
BetheBlochElectron[p_, x_, A_, Z_, i_, ρ_, C0_, X0_, X1_, a_, m_] :=
 0.1535*ρ*Z/A 1/β^2*x (Log[(τ^2 (τ + 2))/(2 (i/0.511)^2)] + Ffac[τ, β] -
      deltafac[X, C0, a, m, X1, X0] - 2*Cfac[i, η]/Z) /. {X → Log[10, η]} /. {η → β*γ} /.
   {γ → 1/Sqrt[1 - β^2]} /. {τ → (Sqrt[p^2 + 0.511^2] - 0.511)/0.511} /. {β → p/Sqrt[p^2 + 0.511^2]}
```

In[45]:=
```
BetheBlochElectron[1193.0, 6.0, 1.01, 1.0,
 21.8*10^-6, 0.071, -3.2632, 0.4759, 1.9215, 0.13483, 5.624]
```

Out[45]= 1.97946

MP Energy Losses with density corrections

In[46]:=
```
LambdaDen[Δ_, p_, x_, A_, Z_, i_, ρ_, C0_, X0_, X1_, a_, m_] :=
```
$$\frac{1}{\text{Xi}[x,\rho,\beta,A,Z]}(\Delta - \text{Xi}[x,\rho,\beta,A,Z](\text{Log}[\text{Xi}[x,\rho,\beta,A,Z]] - \text{Log}[\text{Epsilon}[\beta,i]] + 1 - 0.577 - \text{deltafac}[X, C0, a, m, X1, X0])) /. \{X \to \text{Log}[10,\eta]\} /. \{\eta \to \beta * \gamma\} /. \left\{\gamma \to \frac{1}{\sqrt{1-\beta^2}}\right\} /. \left\{\tau \to \frac{\sqrt{p^2+0.511^2} - 0.511}{0.511}\right\} /. \left\{\beta \to \frac{p}{\sqrt{p^2+0.511^2}}\right\}$$

In[47]:=
```
fDen[Δ_, p_, x_, A_, Z_, i_, ρ_, C0_, X0_, X1_, a_, m_] :=
```
$$\frac{\text{Phi}[\text{LambdaDen}[\Delta, p, x, A, Z, i, \rho, C0, X0, X1, a, m]]}{\text{Xi}[x,\rho,\beta,A,Z]} /. \left\{\beta \to \frac{p}{\sqrt{p^2+0.511^2}}\right\}$$

In[48]:=
```
LambdaDen[1.4, 1193.0, 6.0, 1.01, 1.0,
 21.8*10^-6, 0.071, -3.2632, 0.4759, 1.9215, 0.13483, 5.624]
```

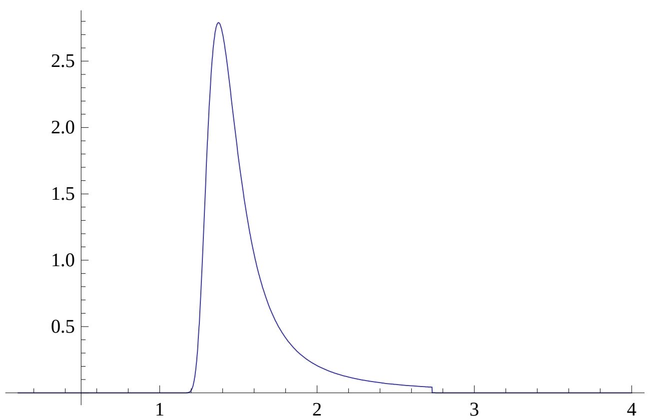
Out[48]= 0.186025

In[49]:=
```
fDen[1.4, 1193.0, 6.0, 1.01, 1.0, 21.8*10^-6, 0.071, -3.2632, 0.4759, 1.9215, 0.13483, 5.624]
```

Out[49]= 2.70335

In[50]:=
```
Plot[fDen[x, 1193.0, 6.0, 1.01, 1.0, 21.8*10^-6, 0.071, -3.2632, 0.4759, 1.9215, 0.13483, 5.624],
 {x, 0.1, 4}, PlotRange → All]
```

Out[50]=



```
FindMaximum[fDen[x, 1193.0, 6.0, 1.01, 1.0,
  21.8*10^-6, 0.071, -3.2632, 0.4759, 1.9215, 0.13483, 5.624], {x, 1.2}]
```

```
NIntegrate::inumr :
 The integrand e^(-15.4455 u (-1.38796+x) <<1>> u <<1>> <<1>>) Sin[π u] has evaluated to non-numerical
   values for all sampling points in the region with boundaries {{0, 100}}. >>

NIntegrate::inumr :
 The integrand 2.71828^(-15.4455 u (-1.38796+x) - <<1>>) Sin[3.14159 u] has evaluated to non-numerical
   values for all sampling points in the region with boundaries {{0., 100.}}. >>

NIntegrate::inumr :
 The integrand 2.71828^(-15.4455 u (-1.38796+x) - <<1>>) Sin[3.14159 u] has evaluated to non-numerical
   values for all sampling points in the region with boundaries {{0., 100.}}. >>

General::stop :
 Further output of NIntegrate::inumr will be suppressed during this calculation. >>

{2.79033, {x → 1.37353}}
```

In[51]:=
```
DeltaMPWithDensity[p_, x_, A_, Z_, i_, ρ_, C0_, X0_, X1_, a_, m_] :=
 Xi[x, ρ, β, A, Z] * (Log[Xi[x, ρ, β, A, Z] / Epsilon[β, i]] + 0.198 - deltafac[X, C0, a, m, X1, X0]) /.
     { X → Log[10, η]} /. {η → β*γ} /. {γ → 1/Sqrt[1 - β^2]} /.
   {τ → (Sqrt[p^2 + 0.511^2] - 0.511)/0.511} /. {β → p/Sqrt[p^2 + 0.511^2]}
```

In[52]:=
```
DeltaMPWithDensity[1193.0, 6.0, 1.01, 1.0,
 21.8*10^-6, 0.071, -3.2632, 0.4759, 1.9215, 0.13483, 5.624 ]
```

Out[52]= 1.37339

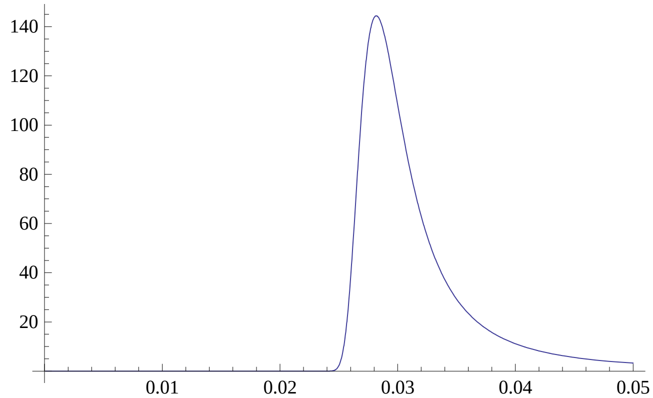
Tantalum

In[53]:=
```
ELOSSTa = DeltaMP[ 0.001213, 16.654, beta[momentum], 180.948, 73.0, 718*10^-6]
```

Out[53]= 0.0281801

In[56]:=
```
Plot[f[x, 0.001213, 16.654, beta[momentum], 180.948, 73.0, 718 * 10^-6],
  {x, 0.0001, 0.05}, PlotRange → All]
```

Out[56]=



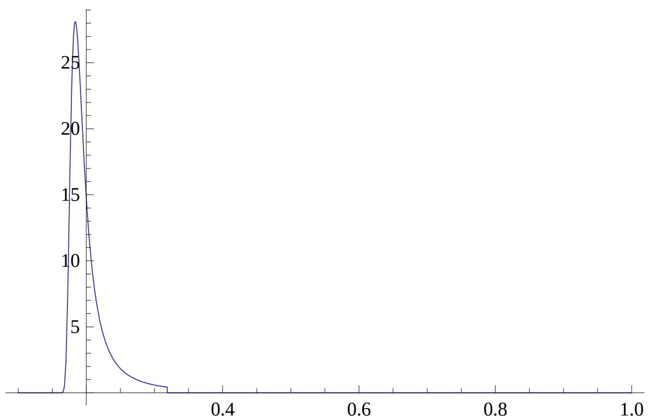
Carbon

In[57]:=
```
ELOSSC12 = DeltaMP[ 0.0419, 2.00, beta[momentum], 12.01, 6.0, 78.0 * 10^-6]
```

Out[57]= 0.183805

In[58]:=
```
Plot[f[x, 0.0419, 2.00, beta[momentum], 12.01, 6.0, 78.0 * 10^-6], {x, 0.1, 1}, PlotRange → All]
```

Out[58]=



In[59]:=
```
Last[SortBy[Table[{x, f[x, 0.0419, 2.00, beta[momentum], 12.01, 6.0, 78.0 * 10^-6]},
   {x, 0.1, 0.4, 0.001}], Last]]
```

```
NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 6 recursive
   bisections in u near {u} = {53.125}. NIntegrate obtained -4.76879 × 10^10
   and 132443.90271459252‘ for the integral and error estimates. >>

NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 6 recursive
   bisections in u near {u} = {42.1875}. NIntegrate obtained 544.05859375‘
   and 109.9419385633012‘ for the integral and error estimates. >>

NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 6 recursive
   bisections in u near {u} = {35.9375}. NIntegrate obtained 5.875‘
   and 0.2917840310626295‘ for the integral and error estimates. >>

General::stop :
 Further output of NIntegrate::ncvb will be suppressed during this calculation. >>
```

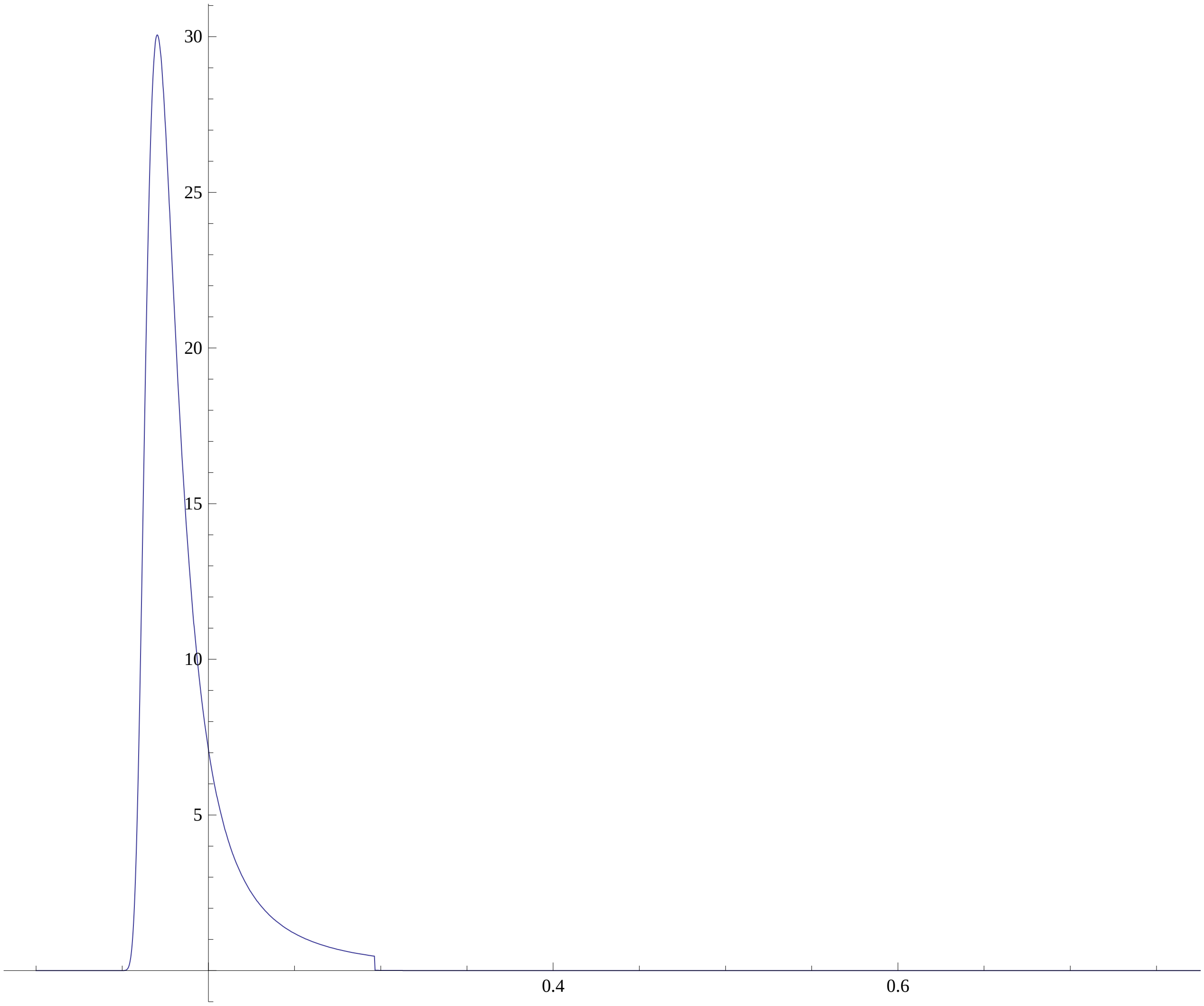
Out[59]= {0.184, 28.1072}

Air

In[60]:=
```
ELOSSAir = DeltaMP[ 65.1, 1.2048 * 10^-3, beta[momentum], 14 289.15, 7133.0, 85.7 * 10^-6]
```

Out[60]= 0.170363

```
Plot[f[x, 65.1, 1.2048 * 10^-3, beta[momentum], 14289.15, 7133.0, 85.7 * 10^-6],
 {x, 0.1, 1}, PlotRange → All]
```



```
Last[SortBy[Table[{x, f[x, 65.1, 1.2048*10^-3, beta[momentum], 14289.15, 7133.0, 85.7*10^-6]},
   {x, 0.1, 0.4, 0.001}], Last]]
```

```
NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 6 recursive
   bisections in u near {u} = {40.625}. NIntegrate obtained -9.34397×10^6
   and 2350.5110036470214‘ for the integral and error estimates. >>

NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 6 recursive
   bisections in u near {u} = {35.9375}. NIntegrate obtained 73.75‘
   and 2.9808854884653035‘ for the integral and error estimates. >>

NIntegrate::ncvb :
 NIntegrate failed to converge to prescribed accuracy after 6 recursive
   bisections in u near {u} = {32.8125}. NIntegrate obtained 0.341796875‘
   and 0.007731248021171422‘ for the integral and error estimates. >>

General::stop :
 Further output of NIntegrate::ncvb will be suppressed during this calculation. >>

{0.17, 30.0332}
```

Aluminium Window

```
In[61]:= ELOSSAl1 = DeltaMP[ 0.0127, 2.6989, beta[momentum], 26.982, 13.0, 166.0*10^-6]

Out[61]= 0.0663174

In[62]:= ELOSSAl2 = DeltaMP[ 0.0113, 2.6989, beta[momentum], 26.982, 13.0, 166.0*10^-6]

Out[62]= 0.0587433

In[63]:= ELOSSAl3 = DeltaMP[ 0.04064, 2.6989, beta[momentum], 26.982, 13.0, 166.0*10^-6]

Out[63]= 0.221651
```

Kapton Window

```
In[64]:= ELOSSKapton = DeltaMP[ 0.03556, 1.42, beta[momentum], 382.33, 196.0, 79.6*10^-6]

Out[64]= 0.11158
```

All Together

```
In[65]:= TantalumLosses = ELOSSTa + ELOSSAl3 + ELOSSAir + ELOSSKapton

Out[65]= 0.531773
```

```
In[66]:= CarbonLosses = ELOSSC12 + ELOSSAl3 + ELOSSAir + ELOSSKapton
Out[66]= 0.687398
In[67]:= HydrogenLosses = ELOSSAl1 + ELOSSH2 + ELOSSAl2 + ELOSSAl3 + ELOSSAir + ELOSSKapton
Out[67]= 2.79508
```