Infinite 3D medium, Isotropic Point Source, Isotropic Scattering

Exponential Random Flight

This is code to accompany the book:

A Hitchhiker's Guide to Multiple Scattering

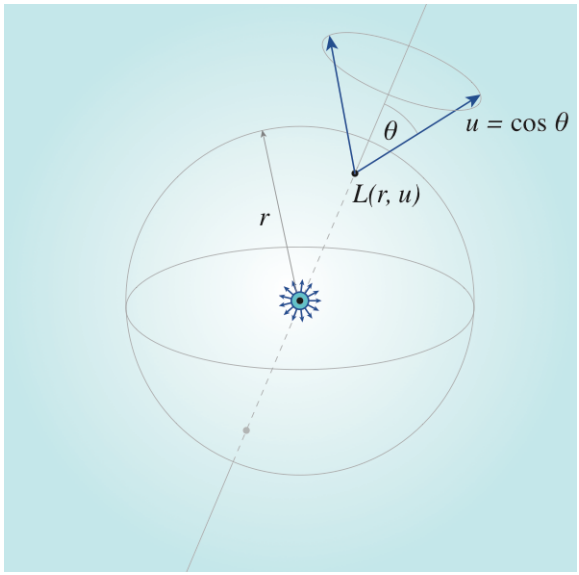
www.eugenedeon.com

Path Setup

Put a file at ~/.hitchhikerpath with the path to your hitchhiker repo so that these worksheets can find the MC data from the C++ simulations for verification

In[424]:=
```
SetDirectory[Import["~/.hitchhikerpath"]]
```

Notation



α - single-scattering albedo
Σt - extinction coefficient
r - radial position coordinate in medium (distance from point source at origin)
u = cos θ - direction cosine

Util

In[425]:=
```
SurfaceArea[d_, r_] := d Pi^(d/2) / Gamma[d/2 + 1] r^(d-1)
```

Diffusion modes

In[426]:=
```
diffusionMode[v_, d_, r_] := (2 π)^(-d/2) r^(1-d/2) v^(-1-d/2) BesselK[1/2 (-2 + d), r/v]
```

Analytic solutions

Caseology quantities

In[427]:=
```
CaseN0[c_, v0_] := 1/2 c v0^3 (c/(v0^2 - 1) - 1/v0^2)
```

In[428]:=
```
Casev0[c_?NumericQ] := FindRoot[c v ArcTanh[1/v] - 1 == 0,
    {v, 1.00000000001, 10^10}, Method → "Brent"][[1]][[2]]
```

In[429]:=
```
CaseN[c_, v_] := v (Caseλ[v, c]^2 + (π c v/2)^2)
```

In[430]:=
```
Caseλ[v_, c_] := 1 - c v ArcTanh[v]
```

Fluence: exact solution (1)

[Bothe 1942]

In[431]:=
```
inf3Disopointisoscatter`ϕexact1a[r_, Σt_, α_] :=
 1/(2 Pi^2 r) NIntegrate[z ArcTan[z / Σt] / (z - α Σt ArcTan[z / Σt]) Sin[r z],
  {z, 0, Infinity}, Method → "ExtrapolatingOscillatory"]
```

[Case et al. 1953]

In[432]:=
```
inf3Disopointisoscatter`ϕexact1b[r_, Σt_, α_] := Exp[-Σt r]/(4 Pi r^2) + α Σt/(2 Pi^2 r)
  NIntegrate[ArcTan[z]^2 / (z - α ArcTan[z]) Sin[r Σt z], {z, 0, Infinity}, Method → "LevinRule"]
```

Rigorous diffusion approximation

In[433]:=
```
inf3Disopointisoscatter`ϕrigourousDiffusion[r_, Σt_, α_] :=
 Σt/(4 Pi r) E^(-r Σt/#)/(# CaseN0[α, #]) &[Casev0[α]]
```

Fluence: exact solution (2)

[Davison 1947]

In[434]:=
```
inf3Disopointisoscatter`ϕexact2a[r_, Σt_, α_] :=
 inf3Disopointisoscatter`ϕrigourousDiffusion[r, Σt, α] +
  Σt/(4 Pi r) NIntegrate[e^(-Σt r y)/((α^2 π^2)/(4 y^2) + (1 - α/(2 y) Log[(y+1)/(y-1)])^2), {y, 1, Infinity}]
```

[Case and Zwiefel 1967]

In[435]:=
```
inf3Disopointisoscatter`ϕexact2b[r_, Σt_, α_] :=
 inf3Disopointisoscatter`ϕrigourousDiffusion[r, Σt, α] +
  Σt/(4 Pi r) NIntegrate[e^(-Σt r/v)/(v CaseN[α, v]), {v, 0, 1}]
```

n-th scattered fluence

In[436]:=
```
inf3Disopointisoscatter`ϕexact1[r_, Σt_, α_, n_] :=
 (α Σt)^n/(2 π^2 r) NIntegrate[ArcTan[z/Σt]^(1+n) Sin[r z]/z^n,
   {z, 0, Infinity}, Method → "ExtrapolatingOscillatory"]
```

In[437]:=
```
inf3Disopointisoscatter`ϕexact2[r_, Σt_, α_, n_] := α^n Σt/(2^(n+3) Pi^2 I r) Chop[NIntegrate[
  Exp[-r z Σt]/z^n ((Log[(z + 1)/(z - 1)] + I Pi)^(n+1) - (Log[(z + 1)/(z - 1)] - I Pi)^(n+1)), {z, 1, Infinity}]]
```

In[757]:=
```
inf3Disopointisoscatter`ϕGaussian[r_, Σt_, α_, n_] := 3 Sqrt[3] e^(-3 r^2 Σt^2/(4 (1+n))) α^n Σt^2/(8 Sqrt[(1 + n)^3] π^(3/2))
```

Moments

In[483]:=
```
inf3Disopointisoscatter`ϕm[c_, Σt_, m_?IntegerQ, n_] :=
 Limit[Simplify[(-1)^(m/2) (2 Gamma[(3+m)/2]/Gamma[(1+m)/2]) D[(c Σt ArcTan[z/Σt]/z)^(1+n)/(c Σt), {z, m}]], z → 0]
```

In[484]:=
```
TableForm[Table[inf3Disopointisoscatter`ϕm[α, Σt, m, n], {m, 0, 6, 2}]]
```

Out[484]//TableForm=

$$\frac{\alpha^n}{\Sigma t}$$
$$\frac{2\,(1+n)\,\alpha^n}{\Sigma t^3}$$
$$\frac{4\,(1+n)\,(18+5n)\,\alpha^n}{3\,\Sigma t^5}$$
$$\frac{8\,(1+n)\,(810+343n+35n^2)\,\alpha^n}{9\,\Sigma t^7}$$

In[485]:=
```
inf3Disopointisoscatter`ϕm[c_, Σt_, m_?IntegerQ] :=
 Limit[Simplify[(-1)^(m/2) (2 Gamma[(3+m)/2]/Gamma[(1+m)/2]) D[ArcTan[z/Σt]/(z - c Σt ArcTan[z/Σt]), {z, m}]], z → 0]
```

In[486]:=
```
TableForm[Table[inf3Disopointisoscatter`ϕm[α, Σt, m], {m, 0, 6, 2}]]
```

Out[486]//TableForm=

$$\frac{1}{\Sigma t-\alpha\,\Sigma t}$$
$$\frac{2}{(-1+\alpha)^2\,\Sigma t^3}$$
$$\frac{8\,(-9+4\,\alpha)}{3\,(-1+\alpha)^3\,\Sigma t^5}$$
$$\frac{16\,(135-144\,\alpha+44\,\alpha^2)}{3\,(-1+\alpha)^4\,\Sigma t^7}$$

Classical diffusion approximation

In[442]:=
```
inf3Disopointisoscatter`ϕDiffusion[r_, Σt_, α_] :=
 1/(Σt (1 - α)) diffusionMode[1/(Sqrt[3 (1 - α)] Σt), 3, r]
```

In[443]:=
```
FullSimplify[inf3Disopointisoscatter`ϕDiffusion[r, Σt, α],
 Assumptions → α > 0 && α < 1 && Σt > 0]
```

Out[443]=

$$\frac{3\,e^{-r\sqrt{3-3\,\alpha}\;\Sigma t}\,\Sigma t}{4\,\pi\,r}$$

Grosjean-style diffusion approximation

In[444]:=
```
inf3Disopointisoscatter`ϕGrosjean[r_, Σt_, α_] :=
 Exp[-r Σt]/(4 Pi r^2) + α/(Σt (1 - α)) diffusionMode[Sqrt[2 - α]/(Sqrt[3 (1 - α)] Σt), 3, r]
```

In[445]:=
```
FullSimplify[inf3Disopointisoscatter`ϕGrosjean[r, Σt, α],
 Assumptions → α > 0 && α < 1 && Σt > 0]
```

Out[445]=

$$\frac{e^{-r\,\Sigma t}-\frac{3\,e^{-r\sqrt{3+\frac{3}{-2+\alpha}}\;\Sigma t}\,r\,\alpha\,\Sigma t}{-2+\alpha}}{4\,\pi\,r^2}$$

Angular ϕ Integral

Note: this form leaves out the singular term $\frac{e^{-r\Sigma_t}}{4\pi r^2}\delta(u-1)$, because it doesn't plot:

In[598]:=
```
inf3Disopointisoscatter`Lintegral[r_, u_, Σt_, α_, ϕ_] :=
 α Σt/(4 Pi) NIntegrate[ϕ[Sqrt[r^2 + t^2 - 2 r t u], Σt, α] Exp[-Σt t], {t, 0, Infinity}]
```

Angular Classical diffusion approximation

In[588]:=
```
inf3Disopointisoscatter`Ldiffusion[r_, u_, Σt_, α_] :=
 1/(4 Pi) inf3Disopointisoscatter`ϕDiffusion[r, Σt, α] +
  1/(4 Pi) u 3 e^(-r Sqrt[3 - 3 α] Σt) (1 + r Sqrt[3 - 3 α] Σt)/(4 π r^2)
```

load MC data

In[447]:=
```
inf3Disopointisoscatter`ppoints[xs_, dr_, maxx_, Σt_] :=
 Table[{dr (i) - 0.5 dr, xs[[i]] / Σt}, {i, 1, Length[xs]}][[1 ;; -2]]
```

In[609]:=
```
inf3Disopointisoscatter`ppointsu[xs_, du_, Σt_] :=
 Table[{-1.0 + du (i) - 0.5 du, xs[[i]] / (2 Σt)}, {i, 1, Length[xs]}][[1 ;; -1]]
```

In[449]:=
```
inf3Disopointisoscatter`fs =
  FileNames["code/3D_medium/infinite3Dmedium/Isotropicpointsource/data/
     inf3D_isotropicpoint_isotropicscatter*"];
```

In[450]:=
```
inf3Disopointisoscatter`index[x_] := Module[{data, α, Σt},
   data = Import[x, "Table"];
   Σt = data[[1, 13]];
   α = data[[2, 3]];
   {α, Σt, data}];
inf3Disopointisoscatter`simulations =
  inf3Disopointisoscatter`index /@ inf3Disopointisoscatter`fs;
inf3Disopointisoscatter`alphas =
 Union[#[[1]] & /@ inf3Disopointisoscatter`simulations]
```

Out[452]= {0.01, 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999}

In[453]:=
```
inf3Disopointisoscatter`muts =
 Union[#[[2]] & /@ inf3Disopointisoscatter`simulations]
```

Out[453]= {1, 3}

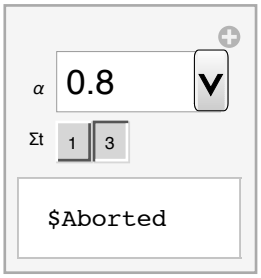
In[454]:=
```
inf3Disopointisoscatter`numcollorders =
  inf3Disopointisoscatter`simulations[[1]][[3]][[2, 13]];
inf3Disopointisoscatter`maxr =
 inf3Disopointisoscatter`simulations[[1]][[3]][[2, 5]];
inf3Disopointisoscatter`dr =
 inf3Disopointisoscatter`simulations[[1]][[3]][[2, 7]];
inf3Disopointisoscatter`numr =
  Floor[inf3Disopointisoscatter`maxr / inf3Disopointisoscatter`dr];
```

Compare Deterministic and MC

Fluence - Exact solution (1a) comparison to MC

In[541]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points = Quiet[{#[[1]], 4 Pi #[[1]]^2
         inf3Disopointisoscatter`ϕexact1a[#[[1]], Σt, α]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (1a)\nInfinite 3D, isotropic point
       source, isotropic scattering, fluence ϕ[r], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.8}, inf3Disopointisoscatter`alphas},
{{Σt, 3}, inf3Disopointisoscatter`muts}]
```

Out[542]=



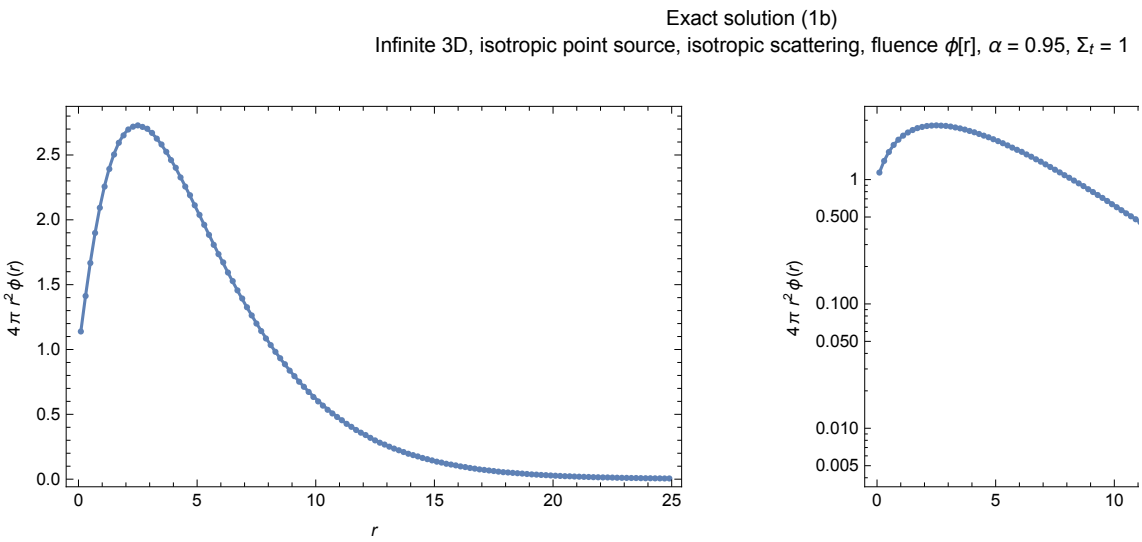
Fluence - Exact solution (1b) comparison to MC

In[534]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points = Quiet[{#[[1]], 4 Pi #[[1]]^2
         inf3Disopointisoscatter`ϕexact1b[#[[1]], Σt, α]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (1b)\nInfinite 3D, isotropic point
       source, isotropic scattering, fluence ϕ[r], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.95}, inf3Disopointisoscatter`alphas},
{Σt, inf3Disopointisoscatter`muts}]
```

Out[535]=

α 0.95

Σt 1 3



Fluence - Exact solution (2a) comparison to MC

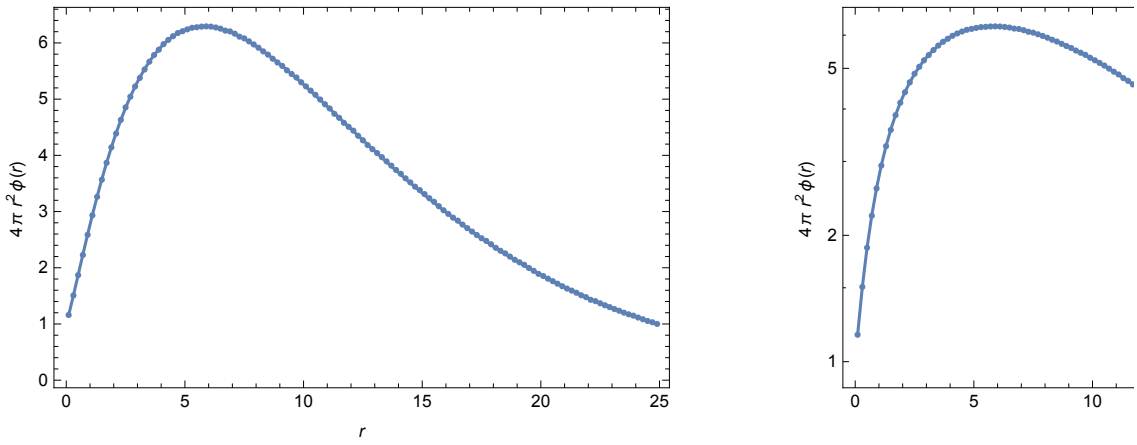
In[546]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points = Quiet[{#[[1]], 4 Pi #[[1]]^2
         inf3Disopointisoscatter`ϕexact2a[#[[1]], Σt, α]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (2a)\nInfinite 3D, isotropic point
       source, isotropic scattering, fluence ϕ[r], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.99}, inf3Disopointisoscatter`alphas},
{Σt, inf3Disopointisoscatter`muts}]
```

Out[547]=

α 0.99

Σt 1 3

Exact solution (2a)
Infinite 3D, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.99, Σ_t = 1



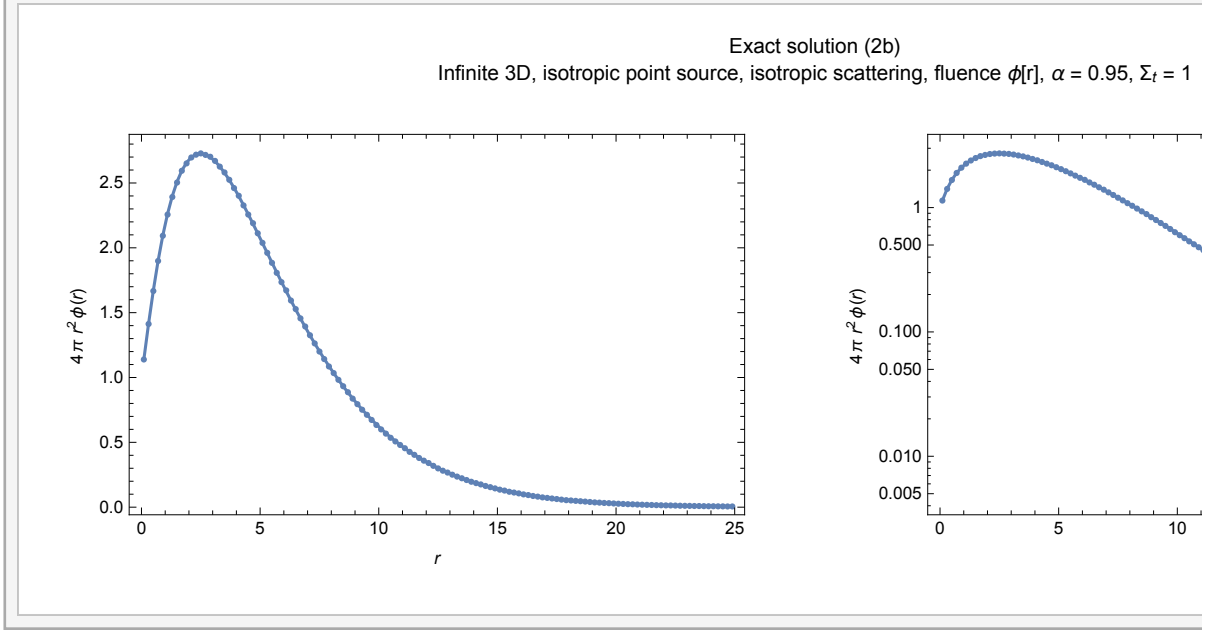
Fluence - Exact solution (2b) comparison to MC

In[699]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points = Quiet[{#[[1]], 4 Pi #[[1]]^2
         inf3Disopointisoscatter`ϕexact2b[#[[1]], Σt, α]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (2b)\nInfinite 3D, isotropic point
       source, isotropic scattering, fluence ϕ[r], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.95}, inf3Disopointisoscatter`alphas},
{Σt, inf3Disopointisoscatter`muts}]
```

Out[700]=

α 0.95

Σt 1 3



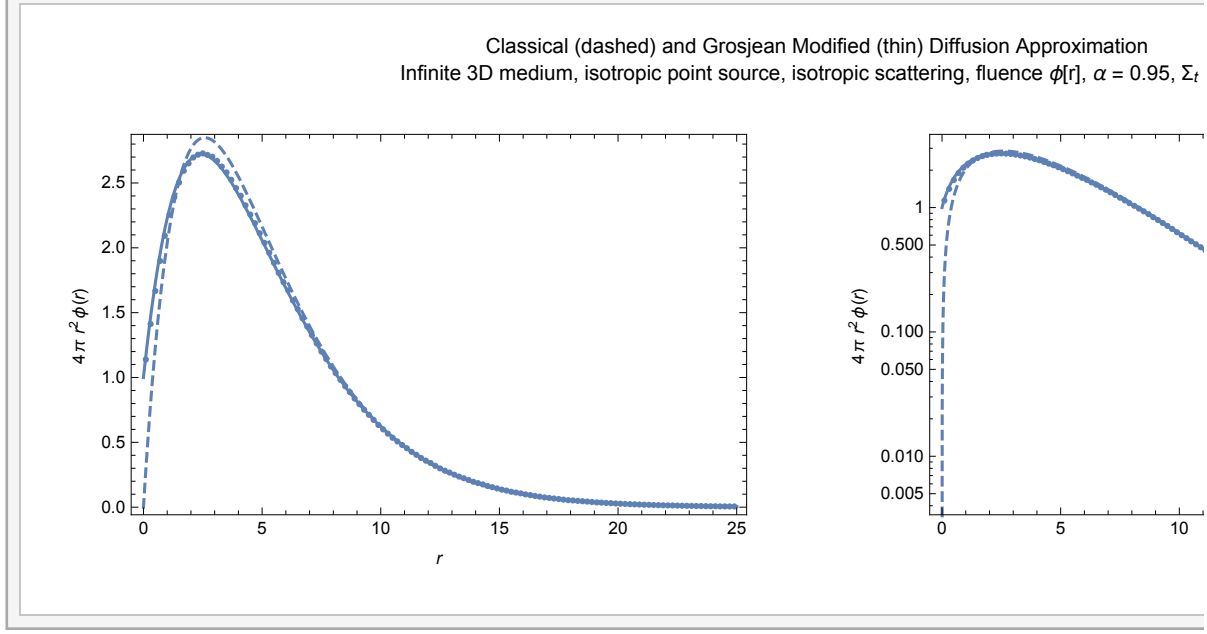
Fluence - Diffusion approximations (Classical and Grosjean) comparison to MC

In[561]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      Plot[4 Pi r^2 inf3Disopointisoscatter`ϕGrosjean[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      Plot[4 Pi r^2 inf3Disopointisoscatter`ϕDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All, PlotStyle → Dashed],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      LogPlot[4 Pi r^2 inf3Disopointisoscatter`ϕGrosjean[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      LogPlot[4 Pi r^2 inf3Disopointisoscatter`ϕDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All, PlotStyle → Dashed],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   pp = Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
     PlotLabel -> "Classical (dashed) and Grosjean Modified (thin) Diffusion
         Approximation\nInfinite 3D medium, isotropic point
         source, isotropic scattering, fluence ϕ[r], α = " <>
       ToString[α] <> ", Σt = " <> ToString[Σt]]
  ]
  ,
  Text[
   "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
     ensure the data path is setup correctly."]
 ]
 , {{α, 0.95}, inf3Disopointisoscatter`alphas},
 {Σt, inf3Disopointisoscatter`muts}]
```

Out[562]=

α 0.95

Σt 1 3



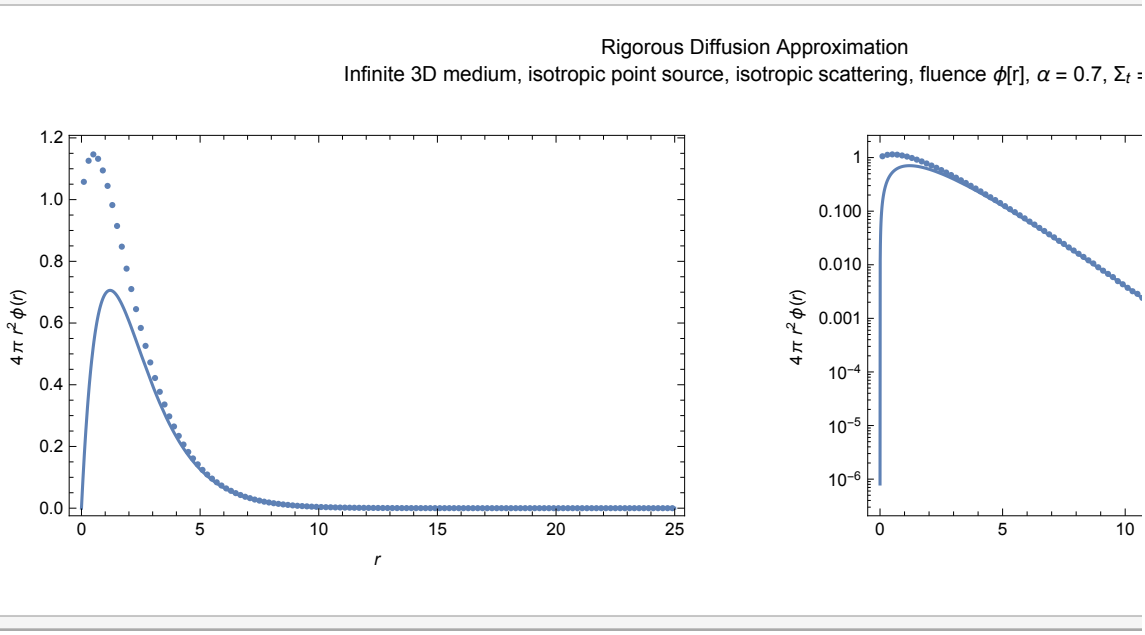
Fluence - Diffusion approximation (Rigorous) comparison to MC

In[553]:=
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      Plot[4 Pi r^2 inf3Disopointisoscatter`ϕrigourousDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      LogPlot[4 Pi r^2 inf3Disopointisoscatter`ϕrigourousDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800], PlotLabel ->
     "Rigorous Diffusion Approximation\nInfinite 3D medium, isotropic
        point source, isotropic scattering, fluence ϕ[r], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.7}, inf3Disopointisoscatter`alphas},
{Σt, inf3Disopointisoscatter`muts}]
```

Out[553]=

α 0.7

Σt 1 3



N-th order fluence / scalar flux

N-th collided Fluence - Exact solution (1) comparison to MC

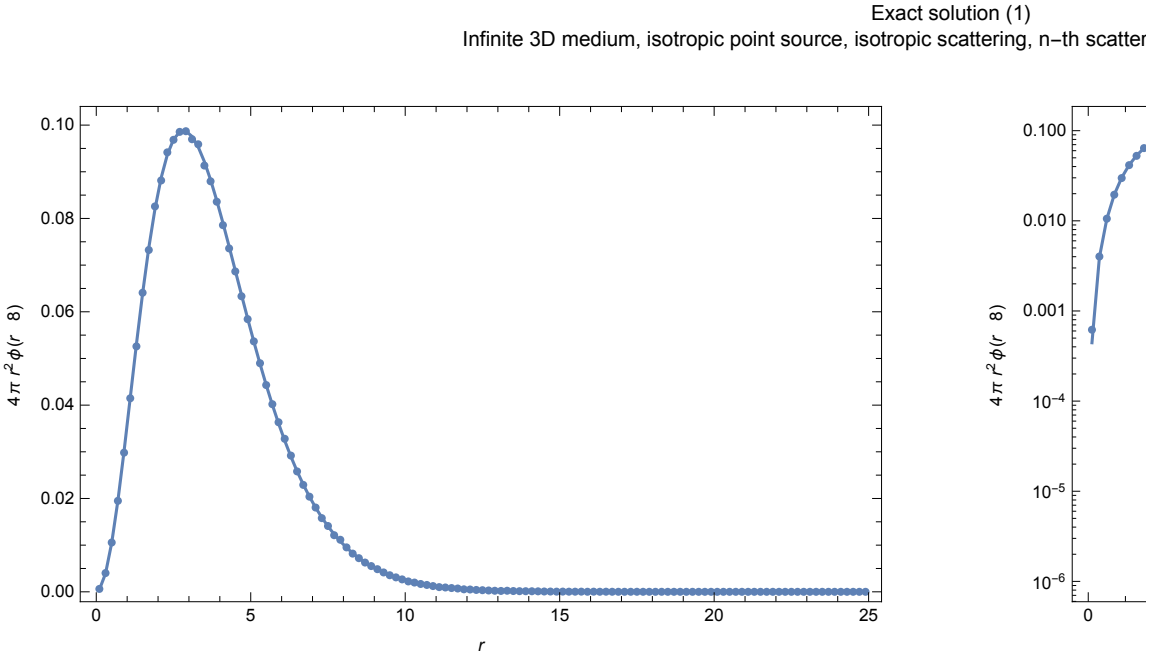
In[557]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[inf3Disopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];
   numorders = data[[2, 13]];

   pointsϕ = data[[9 + numorders + n + 1]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points = Quiet[{#[[1]], 4 Pi #[[1]]^2
         inf3Disopointisoscatter`ϕexact1[#[[1]], Σt, α, n]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r | n],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r | n],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000], PlotLabel ->
     "Exact solution (1)\nInfinite 3D medium, isotropic point source,
        isotropic scattering, n-th scattered fluence ϕ[r|n], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.9}, inf3Disopointisoscatter`alphas},
{Σt, inf3Disopointisoscatter`muts},
{{n, 8}, Range[If[NumberQ[inf3Disopointisoscatter`numcollorders],
   inf3Disopointisoscatter`numcollorders, 1]]}]
```

α 0.9

Σt 1 3

n 8

Out[558]=



N-th collided Fluence - Exact solution (2) comparison to MC

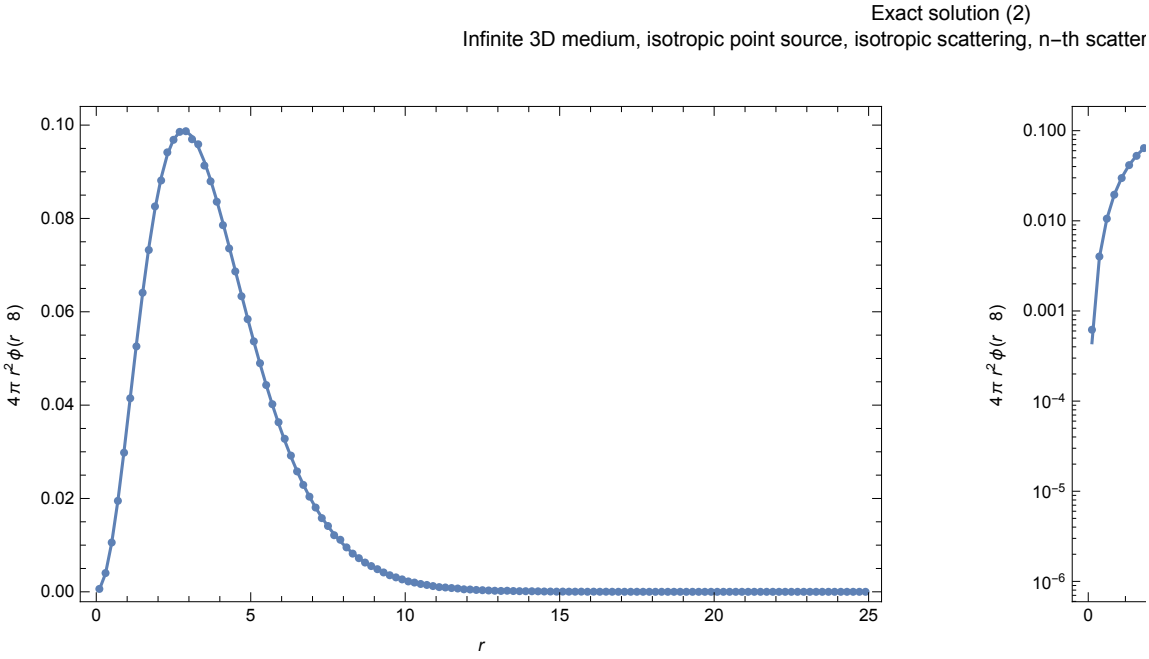
In[701]:=
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[inf3Disopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];
   numorders = data[[2, 13]];

   pointsϕ = data[[9 + numorders + n + 1]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points = Quiet[{#[[1]], 4 Pi #[[1]]^2
         inf3Disopointisoscatter`ϕexact2[#[[1]], Σt, α, n]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r | n],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r | n],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000], PlotLabel ->
     "Exact solution (2)\nInfinite 3D medium, isotropic point source,
        isotropic scattering, n-th scattered fluence ϕ[r|n], α = " <>
      ToString[α] <> ", Σt = " <> ToString[Σt]]
 ]
 ,
 Text[
  "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
    ensure the data path is setup correctly."]
]
, {{α, 0.9}, inf3Disopointisoscatter`alphas},
{Σt, inf3Disopointisoscatter`muts},
{{n, 8}, Range[If[NumberQ[inf3Disopointisoscatter`numcollorders],
   inf3Disopointisoscatter`numcollorders, 1]]}]
```

α 0.9

Σt 1 3

n 8

Out[701]=



N-th collided Fluence - Approximations

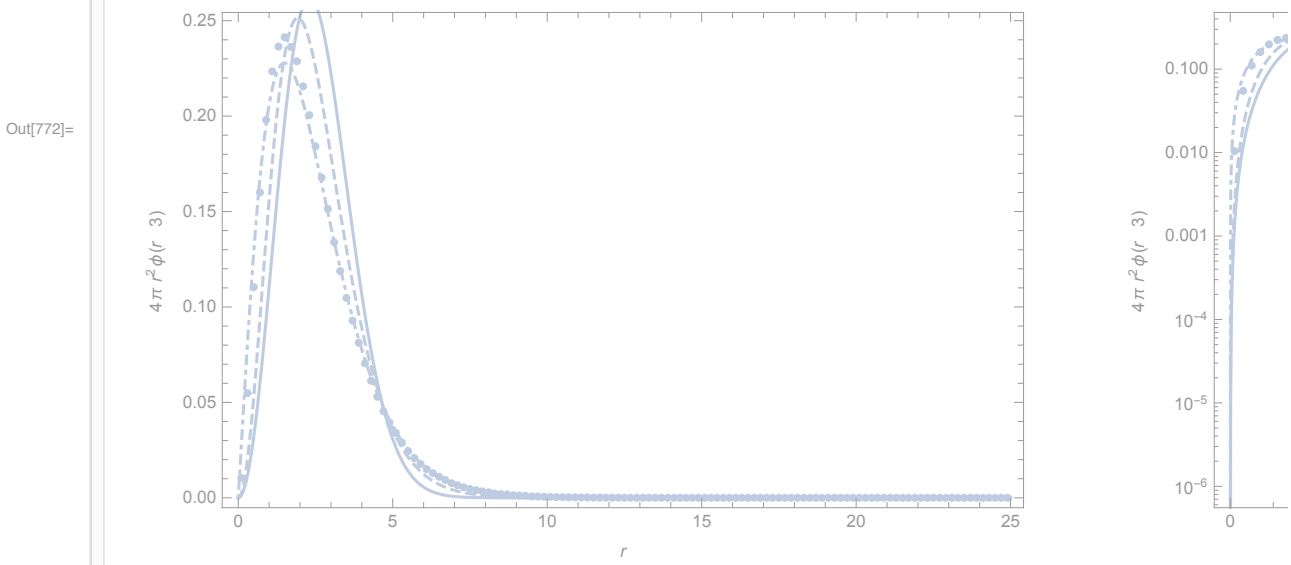
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[inf3Disopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];
   numorders = data[[2, 13]];

   pointsϕ = data[[9 + numorders + n + 1]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   seriesclassical = α^n SeriesCoefficient[
      inf3Disopointisoscatter`ϕDiffusion[r, Σt, C], {C, 0, n}];
   seriesG = α^n SeriesCoefficient[inf3Disopointisoscatter`ϕGrosjean[r, Σt, C],
      {C, 0, n}];

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      Plot[4 Pi r^2 inf3Disopointisoscatter`ϕGaussian[r, Σt, α, n],
       {r, 0, maxr}, PlotRange → All],
      Plot[4 Pi r^2 seriesclassical, {r, 0, maxr},
       PlotRange → All, PlotStyle → Dashed],
      Plot[4 Pi r^2 seriesG, {r, 0, maxr}, PlotRange → All, PlotStyle → DotDashed],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r | n],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      LogPlot[
       4 Pi r^2 inf3Disopointisoscatter`ϕGaussian[r, Σt, α, n], {r, 0, maxr}],
      LogPlot[4 Pi r^2 seriesclassical, {r, 0, maxr},
       PlotRange → All, PlotStyle → Dashed],
      LogPlot[4 Pi r^2 seriesG, {r, 0, maxr}, PlotRange → All,
       PlotStyle → DotDashed],
      Frame → True,
      FrameLabel -> {{4 Pi r^2 ϕ[r | n],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000],
    PlotLabel -> "Approximate solutions, Gaussian (thin) Classical Diffusion
         (Dashed) Grosjean (Dot-Dashed)\nInfinite 3D medium, isotropic
        point source, isotropic scattering, n-th scattered fluence
        ϕ[r|n], α = " <> ToString[α] <> ", Σt = " <> ToString[Σt]]
  ]
  ,
  Text[
   "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
     ensure the data path is setup correctly."]
 ]
 , {{α, 0.9}, inf3Disopointisoscatter`alphas},
 {Σt, inf3Disopointisoscatter`muts},
 {{n, 3}, Range[If[NumberQ[inf3Disopointisoscatter`numcollorders],
    inf3Disopointisoscatter`numcollorders, 1]]}]
```

α 0.9

Σt 1 3

n 3

Out[772]=



Compare moments of ϕ

In[524]:=
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, nummoments, ϕmoments, ks, analytic, j},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   nummoments = data[[2, 15]];
   ϕmoments = N[{data[[6]]/Σt}];
   ks = Table[k, {k, 0, nummoments - 1}];
   analytic = Table[inf3Disopointisoscatter`ϕm[α, Σt, k], {k, ks}];
   j = Join[{ks}, {analytic}, ϕmoments];

   TableForm[
    Join[{{"k", "analytic", "MC"}}, Transpose[j]]
   ]
  ],
  Text["Uh oh!  Couldn't find MC data.
     Try to evaluate this entire notebook and ensure the data path is setup
      correctly."]
 ]
 , {{α, 0.95}, inf3Disopointisoscatter`alphas},
 {{Σt, 3}, inf3Disopointisoscatter`muts}]
```

Out[524]=

α 0.7

Σt 1 3

k	analytic	MC
0	3.33333	3.33223
1	0. + 0. i	6.59823
2	22.2222	22.2373
3	(- i) ∞	103.736
4	612.346	614.906

n-th collided moments of ϕ

In[526]:=
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, nummoments, ϕmoments, ks, analytic, j},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   nummoments = data[[2, 15]];
   ϕmoments = N[{data[[9 + n]]}/Σt];
   ks = Table[k, {k, 0, nummoments - 1}];
   analytic = Table[inf3Disopointisoscatter`ϕm[α, Σt, k, n], {k, ks}];
   j = Join[{ks}, {analytic}, ϕmoments];

   TableForm[
    Join[{{"k", "analytic", "MC"}}, Transpose[j]]
   ]
  ],
  Text["Uh oh!  Couldn't find MC data.
     Try to evaluate this entire notebook and ensure the data path is setup
      correctly."]
 ]
 , {{α, 0.95}, inf3Disopointisoscatter`alphas},
 {{Σt, 3}, inf3Disopointisoscatter`muts},
 {{n, 11}, Range[If[NumberQ[inf3Disopointisoscatter`numcollorders],
    inf3Disopointisoscatter`numcollorders, 1]]}
]
```

Out[526]=

α 0.95

Σt 1 3

n 6

k	analytic	MC
0	0.245031	0.244959
1	0. + 0. i	0.268885
2	0.381159	0.380743
3	0.	0.66009
4	1.35523	1.35476

Angular Distributions

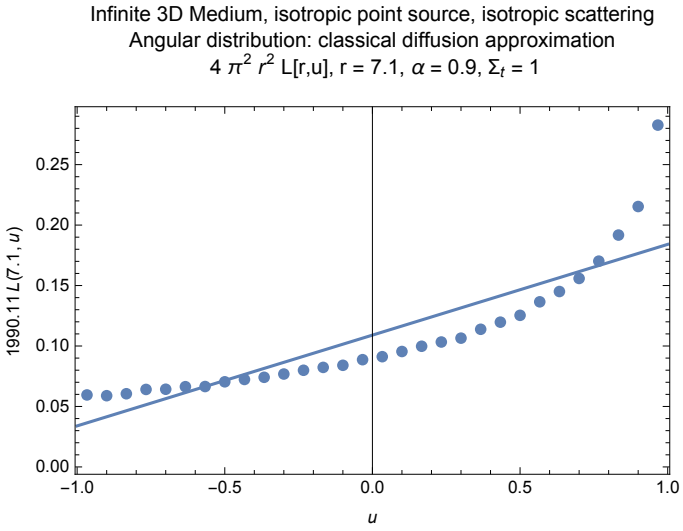
In[704]:=
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, numorders, pointsu, plotpointsu, du, r, dr},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   numorders = data[[2, 13]];
   du = data[[2, 9]];
   dr = data[[2, 7]];

   pointsu = data[[9 + 2 numorders + n]];

   r = dr * n - 0.5 dr;

   (* divide by Σt to convert collision density into fluence *)
   plotpointsu = inf3Disopointisoscatter`ppointsu[pointsu, du, Σt];
   Show[
    ListPlot[plotpointsu, PlotRange → All,
     Frame → True,
     FrameLabel -> {{4 Pi^2 r^2 L[r, u],}, {u,}}],
    Plot[4 Pi r^2 Pi inf3Disopointisoscatter`Ldiffusion[r, u, Σt, α],
     {u, -1, 1}, PlotRange → All
    ],
    PlotLabel -> "Infinite 3D Medium, isotropic point source,
        isotropic scattering\nAngular distribution: classical
        diffusion approximation\n 4 π^2 r^2 L[r,u], r = " <>
      ToString[r] <> ", α = " <> ToString[α] <> ", Σ_t = " <> ToString[Σt]
   ]

  ],
  Text[
   "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
     ensure the data path is setup correctly."]
 ]
 , {{α, 0.9}, inf3Disopointisoscatter`alphas},
 {{Σt, 1}, inf3Disopointisoscatter`muts}, {{n, 36}, Range[If[
    NumberQ[inf3Disopointisoscatter`numr], inf3Disopointisoscatter`numr, 1]]}]
```

Out[704]=

α 0.9

Σt 1 3

n 36



Angular Distribution: Integral of Grosjean's Diffusion Approximation

In[705]:=
```
Manipulate[
 If[Length[inf3Disopointisoscatter`simulations] > 0,
  Module[{data, numorders, pointsu, plotpointsu, du, r, dr},
   data = SelectFirst[
      inf3Disopointisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   numorders = data[[2, 13]];
   du = data[[2, 9]];
   dr = data[[2, 7]];

   pointsu = data[[9 + 2 numorders + n]];

   r = dr * n - 0.5 dr;

   (* divide by Σt to convert collision density into fluence *)
   plotpointsu = inf3Disopointisoscatter`ppointsu[pointsu, du, Σt];
   pp = Show[
     ListPlot[plotpointsu, PlotRange → All,
      Frame → True,
      FrameLabel -> {{4 Pi^2 r^2 L[r, u],}, {u,}}],
     Plot[4 Pi r^2 Pi inf3Disopointisoscatter`Lintegral[r, u, Σt, α,
        inf3Disopointisoscatter`ϕGrosjean], {u, -1, 1}, PlotRange → All
     ],
     PlotLabel -> "Infinite 3D Medium, isotropic point source, isotropic
         scattering\nAngular distribution: Integral of Grosjean
         diffusion approximation\n 4 π^2 r^2 L[r,u], r = " <>
       ToString[r] <> ", α = " <> ToString[α] <> ", Σ_t = " <> ToString[Σt]
    ]

  ],
  Text[
   "Uh oh!  Couldn't find MC data.  Try to evaluate this entire notebook and
     ensure the data path is setup correctly."]
 ]
 , {{α, 0.9}, inf3Disopointisoscatter`alphas},
 {{Σt, 1}, inf3Disopointisoscatter`muts}, {{n, 36}, Range[If[
    NumberQ[inf3Disopointisoscatter`numr], inf3Disopointisoscatter`numr, 1]]}]
```

Out[705]=

α 0.9

Σt 1 3

n 36