Infinite Flatland medium, Isotropic Point Source, Isotropic Scattering

Exponential Random Flight

This is code to accompany the book:

A Hitchhiker's Guide to Multiple Scattering

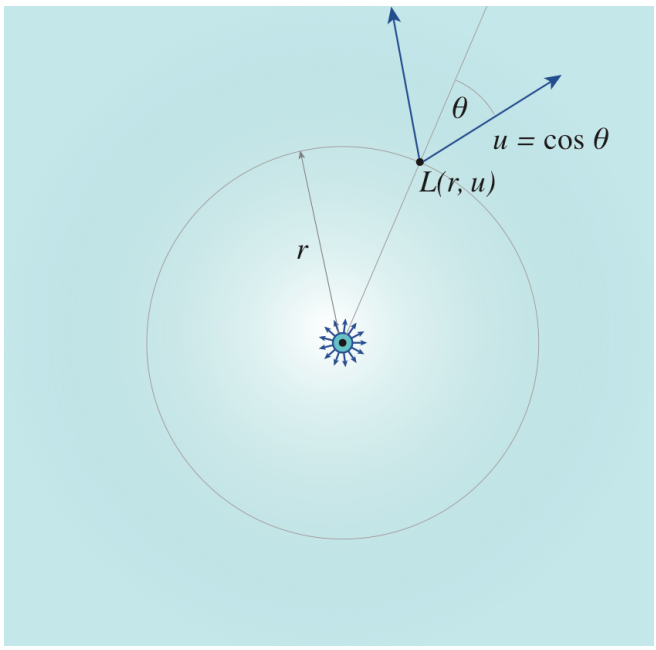
www.eugenedeon.com

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[1541]:=
```
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

Analytic solutions

Fluence: exact solution (1)

In[1542]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{exact1}[r\_, \Sigma t\_, \alpha\_] := \frac{\text{Exp}[-\Sigma t\, r]}{2\,\text{Pi}\, r} + \frac{\Sigma t}{2\,\text{Pi}}\,\text{NIntegrate}\left[\frac{\alpha\, z\, \text{BesselJ}[0, r\, z\, \Sigma t]}{1 + z^2 - \alpha\sqrt{1+z^2}}, \{z, 0, \text{Infinity}\}, \text{Method} \to \text{"LevinRule"}\right]$$

Fluence: exact solution (2)

In[1543]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{exact2}[r\_, \Sigma t\_, \alpha\_] := \Sigma t\,\alpha\,\frac{\text{BesselK}\left[0, r\,\Sigma t\sqrt{1-\alpha^2}\right]}{\text{Pi}} + \frac{\Sigma t}{2\,\text{Pi}}\,\text{NIntegrate}\left[\frac{z\,\text{BesselJ}[0, \Sigma t\, r\, z]}{\sqrt{1+z^2}+\alpha}, \{z, 0, \text{Infinity}\}, \text{Method} \to \text{"LevinRule"}\right]$$

Fluence: exact solution (3)

In[1544]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{exact3a}[r\_, \Sigma t\_, \alpha\_] := \text{NIntegrate}\left[\frac{\Sigma t}{2\,\text{Pi}}\,\frac{k\,\text{BesselJ}[0, k\,\Sigma t\, r]}{\sqrt{k^2+1}-\alpha}, \{k, 0, \text{Infinity}\}, \text{Method} \to \text{"LevinRule"}\right]$$

In[1545]:=
$$\text{besselk}[n\_, x\_] := \sqrt{\frac{2}{\text{Pi}\, x}}\,\text{BesselK}[n+1/2, x];$$

$$\text{infflatlandisopointisoscatter`}\phi\text{exact3b}[r\_, \Sigma t\_, \alpha\_, M\_] := \frac{\text{Exp}[-\Sigma t\, r]}{2\,\text{Pi}\, r} + \frac{\alpha\,\Sigma t}{2\,\text{Pi}}\,\text{BesselK}\left[0, \Sigma t\sqrt{1-\alpha^2}\, r\right] + \frac{\Sigma t}{2\,\text{Pi}}\,\text{Sum}\left[\frac{\alpha^{2n}\, n!}{(2n)!}\,(2\,\Sigma t\, r)^n\,\text{besselk}[n-1, \Sigma t\, r], \{n, 1, M\}\right]$$

Fluence: exact solution (4)

In[1547]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{exact4}[r\_, \Sigma t\_, \alpha\_, M\_] := \frac{\text{Exp}[-\Sigma t\, r]}{2\,\text{Pi}\, r} + \frac{\alpha\,\Sigma t}{2\,\text{Pi}}\,\text{BesselK}\left[0, \Sigma t\sqrt{1-\alpha^2}\, r\right] + \frac{\Sigma t}{2\,\text{Pi}}\,\text{Sum}\left[\frac{\alpha^{2n}\, n!}{(2n)!}\,(2\,\Sigma t\, r)^n\sqrt{\frac{2}{\text{Pi}\,\Sigma t\, r}}\,\text{BesselK}\left[n-1/2, \Sigma t\, r\right], \{n, 1, M\}\right]$$

Classical diffusion approximation

In[1548]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{Diffusion}[r\_, \Sigma t\_, \alpha\_] := \frac{\Sigma t\,\text{BesselK}\left[0, \sqrt{2-2\,\alpha}\, r\,\Sigma t\right]}{\pi}$$

Rigorous diffusion approximation

In[1549]:=
```
infflatlandisopointisoscatter`ϕRigorousDiffusion[r_, Σt_, α_] :=
 Σt α BesselK[0, √(1 - α^2) r Σt] / π
```

Grosjean-style diffusion approximation

In[1550]:=
```
infflatlandisopointisoscatter`ϕGrosjean[r_, Σt_, α_] :=
 Exp[-r Σt] / (2 Pi r) + α Σt / ((2 - α) Pi) BesselK[0, r Σt (√2 √(1 - α) / √(2 - α))]
```

Approximate series expansion for low α and low r:

In[1551]:=
```
infflatlandisopointisoscatter`ϕapprox1[r_, c_] :=
 Exp[-r] / (2 Pi r) + c BesselK[0, r √(1 - c^2)] / (2 Pi) - 1/(8 π)
  c E^-r (c r (4 + 3 r) + (-4 + r (-4 + (-3 + c^2) r)) ArcTanh[c])
```

n-th scattered fluence

In[1552]:=
```
infflatlandisopointisoscatter`ϕ[r_, Σt_, α_, n_] :=
 2^(1/2 (-1-n)) α^n r^(1/2 (-1+n)) Σt^((1+n)/2) BesselK[1/2 (-1 + n), r Σt] / (π Gamma[(1+n)/2])
```

Alternate form suitable for deriving series expansions - valid for n > 0 and n even

In[1553]:=
```
infflatlandisopointisoscatter`ϕalternate[r_, α_, n_?EvenQ] := Chop[
 1/(32 Gamma[(1+n)/2]) (i α)^n (-√π (4 HypergeometricPFQRegularized[{1/2}, {3/2, 3/2 - n/2}, r^2/4] +
    r^2 HypergeometricPFQRegularized[{3/2}, {5/2, 5/2 - n/2}, r^2/4]) +
  2 r (2 HypergeometricPFQRegularized[{1, 1}, {1/2, 2, 2 - n/2}, r^2/4] +
    HypergeometricPFQRegularized[{1, 1}, {3/2, 2, 2 - n/2}, r^2/4]))]
```

In[1554]:=
```
infflatlandisopointisoscatter`ϕalternate[r_, Σt_, α_, n_?EvenQ] :=
 Σt infflatlandisopointisoscatter`ϕalternate[Σt r, α, n]
```

Approximate Gaussian preserving 0th and 2nd radial moments:

In[1555]:=
```
infflatlandisopointisoscatter`ϕGaussianApprox[r_, Σt_, α_, n_] := Σt α^n E^-((r Σt)^2/(2+2 n)) / (2 π + 2 n π)
```

Moments

In[1556]:=
```
infflatlandisopointisoscatter`ϕm[Σt_, α_, m_] := (1 - α^2)^(-1-m/2) Σt^(-1-m)
  (2^m α Gamma[1 + m/2]^2 + Gamma[1 + m] Hypergeometric2F1[-1/2, -m/2, 1/2, α^2])
```

In[1557]:=
```
infflatlandisopointisoscatter`ϕm[Σt_, α_, n_, m_] :=
 2^m α^n Σt^(-1-m) Gamma[1 + m/2] Gamma[1/2 (1 + m + n)] / Gamma[(1+n)/2]
```

Angular phi integral

In[1558]:=
```
infflatlandisopointisoscatter`Lintegral[r_, u_, Σt_, α_, ϕ_] :=
 α Σt / (2 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[1559]:=
```
infflatlandisopointisoscatter`Ldiffusion[r_, u_, Σt_, α_] :=
 1/(2 π^2) Σt (BesselK[0, r Sqrt[2 - 2 α] Σt] + u Sqrt[2 - 2 α] BesselK[1, r Sqrt[2 - 2 α] Σt])
```

---

load MC data

In[1560]:=
```
infflatlandisopointisoscatter`ppoints[xs_, dr_, maxx_, Σt_] :=
 Table[{dr (i) - 0.5 dr, xs[[i]] / Σt}, {i, 1, Length[xs]}][[1 ;; -2]]
```

In[1561]:=
```
infflatlandisopointisoscatter`ppointsu[xs_, du_, Σt_] :=
 Table[{-1.0 + du (i) - 0.5 du, Sqrt[1 - (-1.0 + du (i) - 0.5 du)^2] xs[[i]] / (2 Σt)},
   {i, 1, Length[xs]}][[1 ;; -1]]
```

In[1562]:=
```
infflatlandisopointisoscatter`fs =
  FileNames["code/flatland/infiniteFlatland/Isotropicpointsource/data/
     infflatland_isotropicpoint_isotropicscatter*"];
```

In[1563]:=
```
infflatlandisopointisoscatter`index[x_] := Module[{data, α, Σt},
   data = Import[x, "Table"];
   Σt = data[[1, 13]];
   α = data[[2, 3]];
   {α, Σt, data}];
infflatlandisopointisoscatter`simulations =
  infflatlandisopointisoscatter`index /@ infflatlandisopointisoscatter`fs;
infflatlandisopointisoscatter`alphas =
 Union[#[[1]] & /@ infflatlandisopointisoscatter`simulations]
```

Out[1565]=
```
{0.01, 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999}
```

In[1566]:=
```
infflatlandisopointisoscatter`muts =
 Union[#[[2]] & /@ infflatlandisopointisoscatter`simulations]
```

Out[1566]=
```
{1, 3}
```

In[1567]:=
```
infflatlandisopointisoscatter`numcollorders =
  infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 13]];
infflatlandisopointisoscatter`maxr =
 infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 5]];
infflatlandisopointisoscatter`dr =
 infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 7]];
infflatlandisopointisoscatter`numr =
  Floor[infflatlandisopointisoscatter`maxr / infflatlandisopointisoscatter`dr];
```

Compare Deterministic and MC

Fluence - Exact solution (1) comparison to MC

In[1570]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

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

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (1)\nInfinite Flatland, 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}, infflatlandisopointisoscatter`alphas},
{Σt, infflatlandisopointisoscatter`muts}]
```

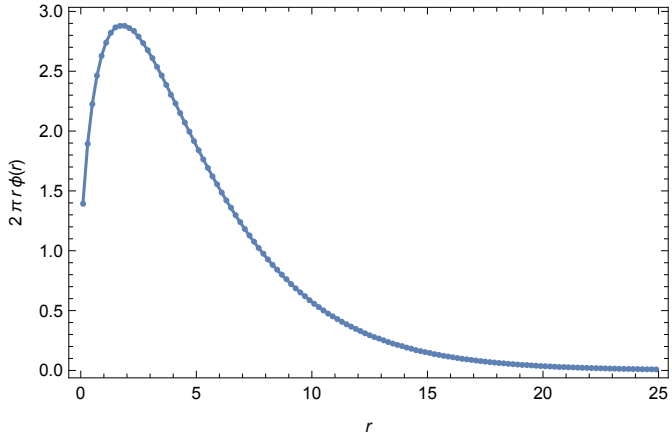
Out[1571]=

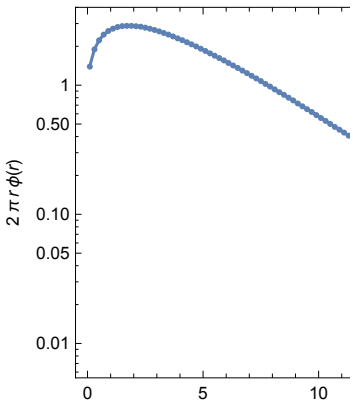
α 0.95

Σt 1 3

Exact solution (1)

Infinite Flatland, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.95, Σ_t =

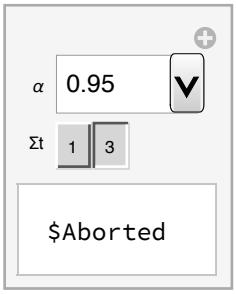




Fluence - Exact solution (2) comparison to MC

In[1572]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

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

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (2)\nInfinite Flatland, 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}, infflatlandisopointisoscatter`alphas},
{{Σt, 3}, infflatlandisopointisoscatter`muts}]
```

Out[1573]=



Fluence - Exact solution (3a) comparison to MC

In[1574]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points =
    Quiet[{#[[1]], 2 Pi #[[1]] infflatlandisopointisoscatter`ϕexact3a[
          #[[1]], Σt, α]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (3a)\nInfinite Flatland, 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}, infflatlandisopointisoscatter`alphas},
{{Σt, 3}, infflatlandisopointisoscatter`muts}]
```

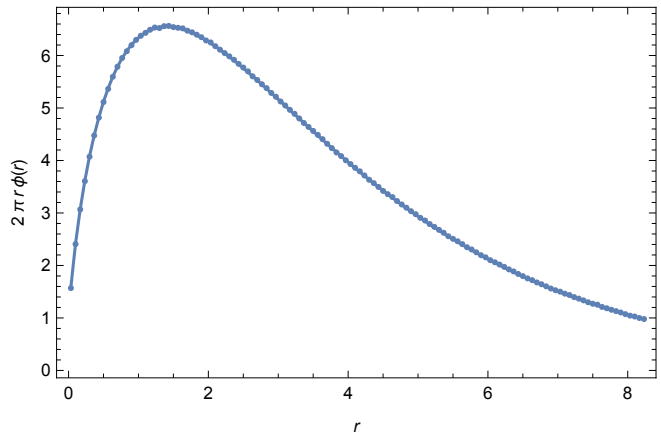
Out[1575]=

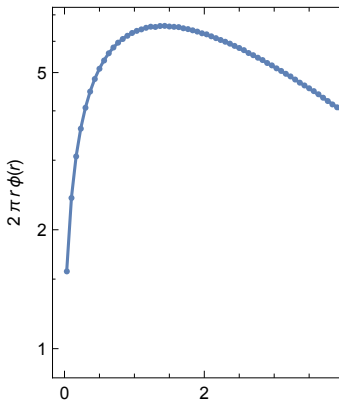
α 0.99

Σt 1 3

Exact solution (3a)

Infinite Flatland, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.99, Σ_t =





Fluence - Exact solution (3b) comparison to MC

In[1576]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   exact1points =
    Quiet[{#[[1]], 2 Pi #[[1]] infflatlandisopointisoscatter`ϕexact3b[
          #[[1]], Σt, α, M]}] & /@ plotpointsϕ;

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (3b)\nInfinite Flatland, 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}, infflatlandisopointisoscatter`alphas},
{{Σt, 3}, infflatlandisopointisoscatter`muts}, {{M, 10}, Range[20]}]
```

Out[1577]=

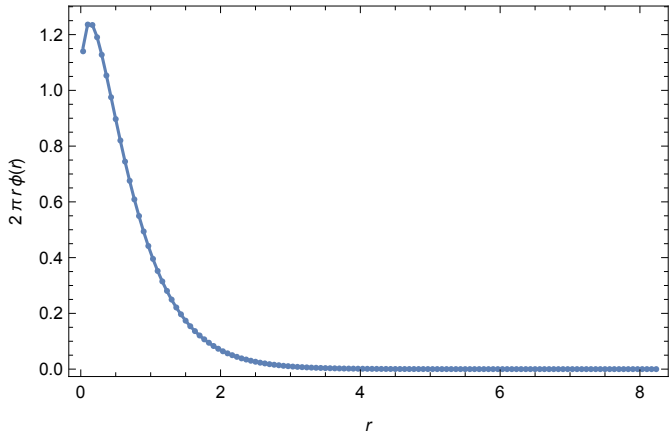
α 0.7

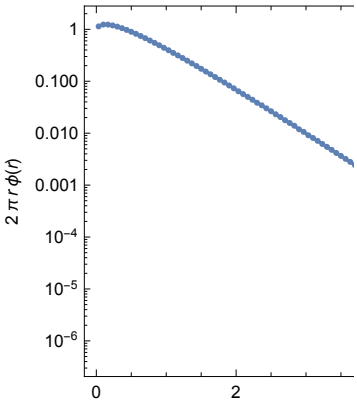
Σt 1 3

M 10

Exact solution (3b)

Infinite Flatland, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.7, Σ_t = 3





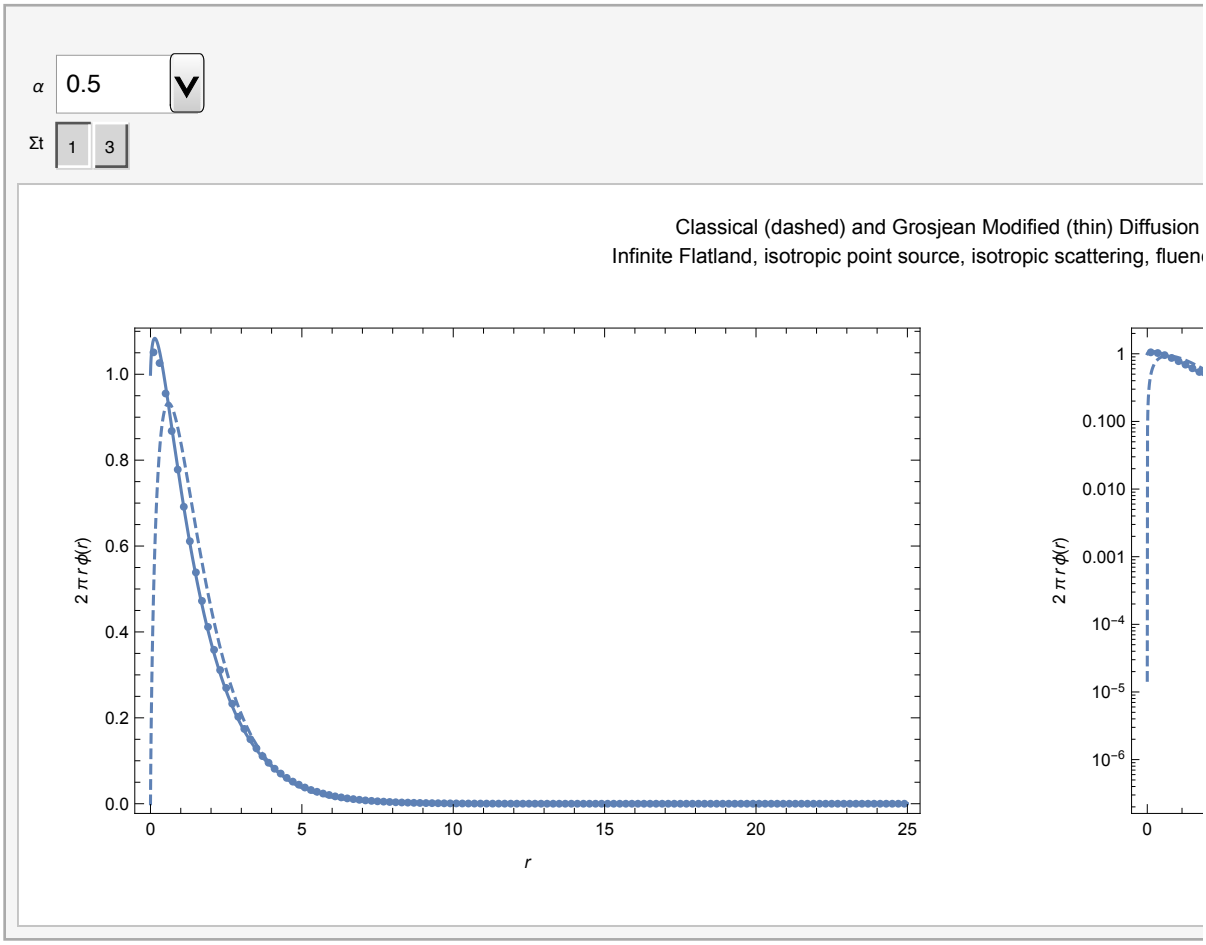
Fluence - Diffusion approximations (Classical and Grosjean) comparison to MC

In[1578]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      Plot[2 Pi r infflatlandisopointisoscatter`ϕGrosjean[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      Plot[2 Pi r infflatlandisopointisoscatter`ϕDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All, PlotStyle → Dashed],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      LogPlot[2 Pi r infflatlandisopointisoscatter`ϕGrosjean[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      LogPlot[2 Pi r infflatlandisopointisoscatter`ϕDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All, PlotStyle → Dashed],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000],
    PlotLabel -> "Classical (dashed) and Grosjean Modified (thin)
        Diffusion Approximation\nInfinite Flatland, 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, 0.5}, infflatlandisopointisoscatter`alphas},
 {Σt, infflatlandisopointisoscatter`muts}]
```

Out[1579]=

α 0.5

Σt 1 3



Fluence - Diffusion approximation (Rigorous) comparison to MC

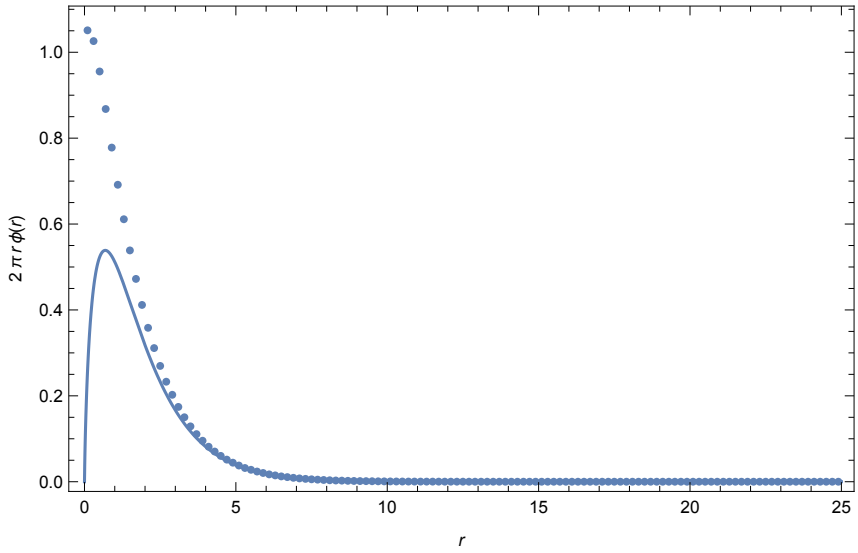
In[1580]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 5]];
   dr = data[[2, 7]];

   pointsϕ = data[[4]];

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

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      Plot[2 Pi r infflatlandisopointisoscatter`ϕRigorousDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      LogPlot[2 Pi r infflatlandisopointisoscatter`ϕRigorousDiffusion[r, Σt, α],
       {r, 0, maxr}, PlotRange → All],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000], PlotLabel ->
     "Rigorous Diffusion Approximation\nInfinite Flatland, 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, 0.5}, infflatlandisopointisoscatter`alphas},
{Σt, infflatlandisopointisoscatter`muts}]
```

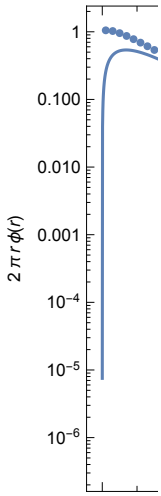
Out[1581]=

α 0.5

Σt 1 3

Rigorous Diffusion Approximation
Infinite Flatland, isotropic point source, isotropic scattering, fluen





N-th order fluence / scalar flux

In[1582]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppoints[pointsϕ, dr, maxr, Σt];

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      Plot[2 Pi r infflatlandisopointisoscatter`ϕ[r, Σt, α, n],
       {r, 0, maxr}, PlotRange → All],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r | n],}, {r,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      LogPlot[2 Pi r infflatlandisopointisoscatter`ϕ[r, Σt, α, n],
       {r, 0, maxr}, PlotRange → All],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r | n],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000],
    PlotLabel -> "Infinite Flatland, 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}, infflatlandisopointisoscatter`alphas},
 {Σt, infflatlandisopointisoscatter`muts},
 {{n, 8}, Range[If[NumberQ[infflatlandisopointisoscatter`numcollorders],
    infflatlandisopointisoscatter`numcollorders, 1]]}]
```

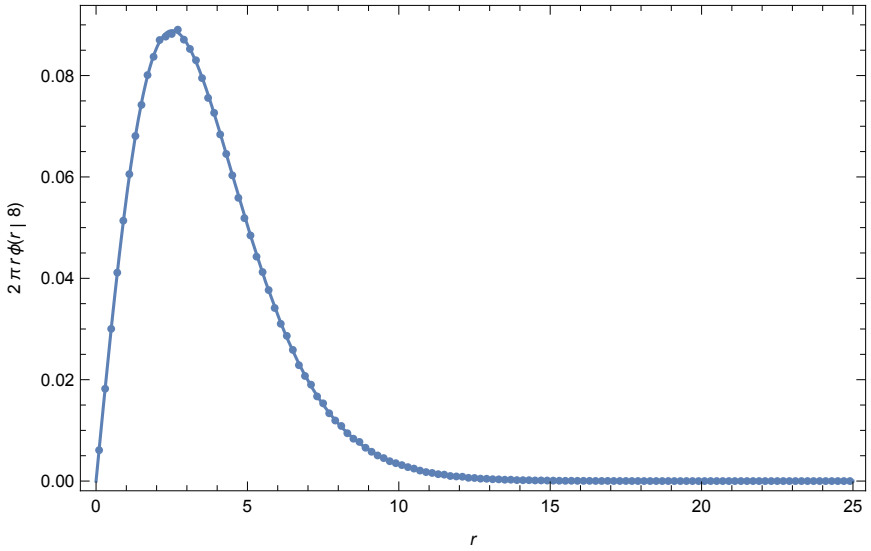
Out[1583]=

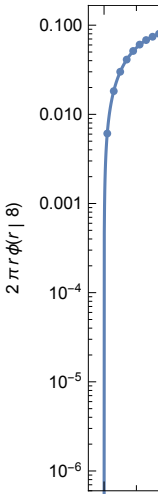
α 0.9

Σt 1 3

n 8

Infinite Flatland, isotropic point source, isotropic scattering, n−th scattered





Compare moments of ϕ

In[1584]:=
```
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, nummoments, ϕmoments, ks, analytic, j},
   data = SelectFirst[infflatlandisopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   nummoments = data[[2, 15]];
   ϕmoments = N[{data[[6]]/Σt}];
   ks = {Table[k, {k, 0, nummoments - 1}]};
   analytic = Table[infflatlandisopointisoscatter`ϕm[Σt, α, 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}, infflatlandisopointisoscatter`alphas},
 {{Σt, 3}, infflatlandisopointisoscatter`muts}]
```

Out[1584]=

α 0.95

Σt 1 3

k	analytic	MC
0	6.66667	6.65803
1	10.931	10.9132
2	29.6296	29.5695
3	111.733	111.445
4	539.918	538.247

n-th collided moments of ϕ

In[1585]:=
```
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, ϕmoments, ks, analytic, j, nummoments},
   data = SelectFirst[infflatlandisopointisoscatter`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[Quiet[N[infflatlandisopointisoscatter`ϕm[Σt, α, n, 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, 0.8}, infflatlandisopointisoscatter`alphas},
 {{Σt, 3}, infflatlandisopointisoscatter`muts},
 {{n, 11}, Range[If[NumberQ[infflatlandisopointisoscatter`numcollorders],
    infflatlandisopointisoscatter`numcollorders, 1]]}]
```

Out[1585]=

α 0.8

Σt 1 3

n 11

k	analytic	MC
0	0.0286331	0.0286687
1	0.0405845	0.040645
2	0.076355	0.0764887
3	0.175866	0.176249
4	0.475098	0.476663

Angular distributions

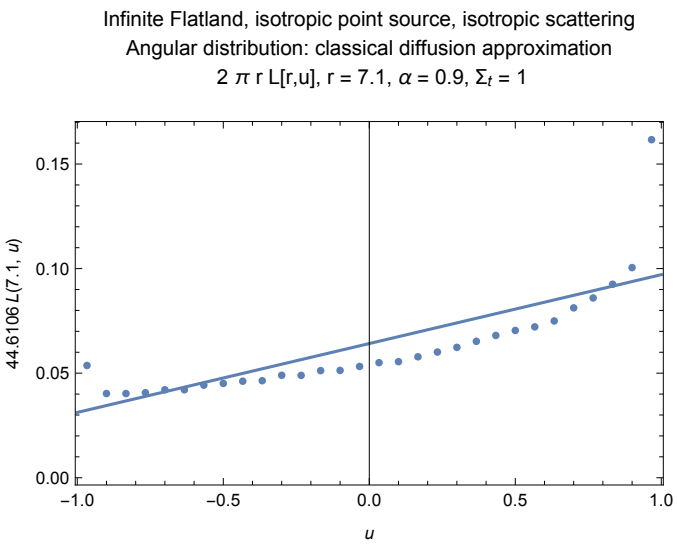
In[1586]:=
```
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, numorders, pointsu, plotpointsu, du, r, dr},
   data = SelectFirst[infflatlandisopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   numorders = data[[2, 13]];
   du = data[[2, 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 = infflatlandisopointisoscatter`ppointsu[pointsu, du, Σt];
   Show[
    ListPlot[plotpointsu, PlotRange → All,
     Frame → True,
     FrameLabel -> {{2 Pi r L[r, u],}, {u,}}],
    Plot[2 Pi r infflatlandisopointisoscatter`Ldiffusion[r, u, Σt, α],
     {u, -1, 1}, PlotRange → All
    ],
    PlotLabel -> "Infinite Flatland, isotropic point source,
        isotropic scattering\nAngular distribution: classical
        diffusion approximation\n 2 π r L[r,u], r = " <>
      ToString[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, 0.9}, infflatlandisopointisoscatter`alphas},
 {{Σt, 1}, infflatlandisopointisoscatter`muts},
 {{n, 36}, Range[If[NumberQ[infflatlandisopointisoscatter`numr],
    infflatlandisopointisoscatter`numr, 1]]}]
```

Out[1586]=

α 0.9

Σt 1 3

n 36



In[1590]:=
```
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, numorders, pointsu, plotpointsu, du, r, dr},
   data = SelectFirst[infflatlandisopointisoscatter`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 = infflatlandisopointisoscatter`ppointsu[pointsu, du, Σt];
   Quiet[Show[
     ListPlot[plotpointsu, PlotRange → All,
      Frame → True,
      FrameLabel -> {{2 Pi r L[r, u],}, {u,}}],
     Plot[2 Pi r infflatlandisopointisoscatter`Lintegral[r, u, Σt, α,
        infflatlandisopointisoscatter`ϕGrosjean], {u, -1, 1}, PlotRange → All],
     PlotLabel -> "Infinite Flatland, isotropic point source,
          isotropic scattering\nAngular distribution:
          Fluence integral Grosjean\n 2 π r L[r,u], r = " <>
       ToString[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, 0.9}, infflatlandisopointisoscatter`alphas},
 {{Σt, 1}, infflatlandisopointisoscatter`muts},
 {{n, 53}, Range[If[NumberQ[infflatlandisopointisoscatter`numr],
    infflatlandisopointisoscatter`numr, 1]]}]
```

Out[1590]=

α 0.9

Σt 1 3

n 53