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

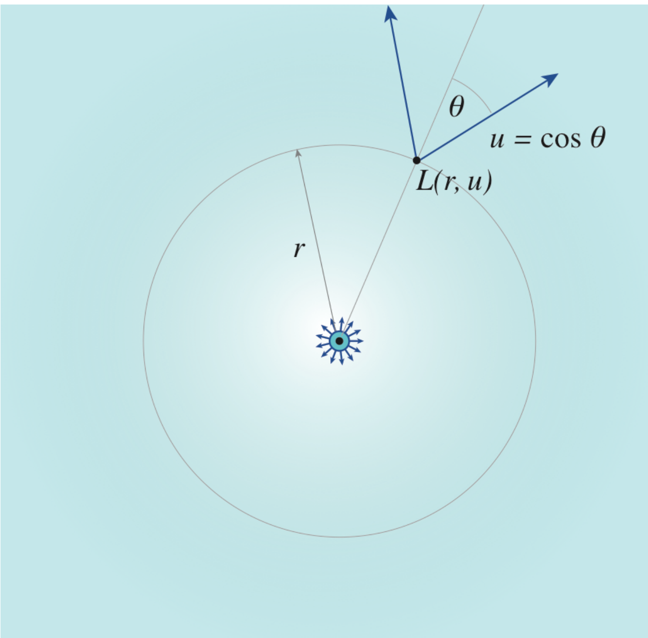
www.eugenedeon.com

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[1]:= SetDirectory[Import["~/.hitchhikerpath"]]

Out[1]= /Users/eug/Documents/research/hitchhikersscatter
```

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

Fluence: exact solution (2)

In[3]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{exact2}[r\_, \Sigma t\_, \alpha\_] := \Sigma t\,\alpha\,\frac{\text{BesselK}\left[0, r\,\Sigma t\sqrt{1-\alpha^2}\right]}{\text{Pi}} + \frac{\Sigma t}{2\,\text{Pi}}\,\text{NIntegrate}\left[\frac{z\,\text{BesselJ}[0, \Sigma t\, r\, z]}{\sqrt{1+z^2}+\alpha}, \{z, 0, \text{Infinity}\}, \text{Method} \to \text{"LevinRule"}\right]$$

Fluence: exact solution (3)

In[4]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{exact3a}[r\_, \Sigma t\_, \alpha\_] := \text{NIntegrate}\left[\frac{\Sigma t}{2\,\text{Pi}}\,\frac{k\,\text{BesselJ}[0, k\,\Sigma t\, r]}{\sqrt{k^2+1}-\alpha}, \{k, 0, \text{Infinity}\}, \text{Method} \to \text{"LevinRule"}\right]$$

In[5]:=
$$\text{besselk}[n\_, x\_] := \sqrt{\frac{2}{\text{Pi}\, x}}\,\text{BesselK}[n + 1/2, x];$$
$$\text{infflatlandisopointisoscatter`}\phi\text{exact3b}[r\_, \Sigma t\_, \alpha\_, M\_] := \frac{\text{Exp}[-\Sigma t\, r]}{2\,\text{Pi}\, r} + \frac{\alpha\,\Sigma t}{2\,\text{Pi}}\,\text{BesselK}\left[0, \Sigma t\sqrt{1-\alpha^2}\; r\right] + \frac{\Sigma t}{2\,\text{Pi}}\,\text{Sum}\left[\frac{\alpha^{2n}\, n!}{(2n)!}\,(2\,\Sigma t\, r)^n\,\text{besselk}[n-1, \Sigma t\, r], \{n, 1, M\}\right]$$

Classical diffusion approximation

In[7]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{Diffusion}[r\_, \Sigma t\_, \alpha\_] := \frac{\Sigma t\,\text{BesselK}\left[0, \sqrt{2-2\alpha}\; r\,\Sigma t\right]}{\pi}$$

Rigorous diffusion approximation

In[8]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{RigorousDiffusion}[r\_, \Sigma t\_, \alpha\_] := \frac{\Sigma t\,\alpha\,\text{BesselK}\left[0, \sqrt{1-\alpha^2}\; r\,\Sigma t\right]}{\pi}$$

Grosjean-style diffusion approximation

In[9]:=
$$\text{infflatlandisopointisoscatter`}\phi\text{Grosjean}[r\_, \Sigma t\_, \alpha\_] := \frac{\text{Exp}[-r\,\Sigma t]}{2\,\text{Pi}\, r} + \frac{\alpha\,\Sigma t}{(2-\alpha)\,\text{Pi}}\,\text{BesselK}\left[0, r\,\Sigma t\left(\sqrt{2}\,\frac{\sqrt{1-\alpha}}{\sqrt{2-\alpha}}\right)\right]$$

n-th scattered fluence

In[10]:= infflatlandisopointisoscatter`ϕ[r_, Σt_, α_, n_] :=

$$\frac{2^{\frac{1}{2}(-1-n)}\,\alpha^n\, r^{\frac{1}{2}(-1+n)}\,\Sigma t^{\frac{1+n}{2}}\,\mathrm{BesselK}\left[\frac{1}{2}(-1+n),\, r\,\Sigma t\right]}{\pi\,\mathrm{Gamma}\left[\frac{1+n}{2}\right]}$$

Moments

In[11]:= infflatlandisopointisoscatter`ϕm[Σt_, α_, m_] := $\left(1-\alpha^2\right)^{-1-\frac{m}{2}}\Sigma t^{-1-m}$

$$\left(2^m\,\alpha\,\mathrm{Gamma}\left[1+\frac{m}{2}\right]^2+\mathrm{Gamma}[1+m]\,\mathrm{Hypergeometric2F1}\left[-\frac{1}{2},\,-\frac{m}{2},\,\frac{1}{2},\,\alpha^2\right]\right)$$

In[12]:= infflatlandisopointisoscatter`ϕm[Σt_, α_, n_, m_] :=

$$\frac{2^m\,\alpha^n\,\Sigma t^{-1-m}\,\mathrm{Gamma}\left[1+\frac{m}{2}\right]\,\mathrm{Gamma}\left[\frac{1}{2}(1+m+n)\right]}{\mathrm{Gamma}\left[\frac{1+n}{2}\right]}$$

Angular phi integral

In[140]:= infflatlandisopointisoscatter`Lintegral[r_, u_, Σt_, α_, ϕ_] :=

$$\frac{\alpha\,\Sigma t}{2\,\mathrm{Pi}}\,\mathrm{NIntegrate}\left[\phi\left[\sqrt{r^2+t^2-2\,r\,t\,u},\,\Sigma t,\,\alpha\right]\mathrm{Exp}[-\Sigma t\, t],\,\{t,\,0,\,\mathrm{Infinity}\}\right]$$

Angular Classical diffusion approximation

In[192]:= infflatlandisopointisoscatter`Ldiffusion[r_, u_, Σt_, α_] := $\frac{1}{2\,\pi^2}$

$$\Sigma t\left(\mathrm{BesselK}\left[0,\, r\sqrt{2-2\,\alpha}\;\Sigma t\right]+u\sqrt{2-2\,\alpha}\;\mathrm{BesselK}\left[1,\, r\sqrt{2-2\,\alpha}\;\Sigma t\right]\right)$$

load MC data

In[49]:= infflatlandisopointisoscatter`ppoints[xs_, dr_, maxx_, Σt_] :=
Table[{dr (i) - 0.5 dr, xs[[i]] / Σt}, {i, 1, Length[xs]}][[1 ;; -2]]

In[110]:= infflatlandisopointisoscatter`ppointsu[xs_, du_, Σt_] :=

$$\mathrm{Table}\left[\left\{-1.0+du\,(i)-0.5\,du,\,\sqrt{1-\left(-1.0+du\,(i)-0.5\,du\right)^2}\;xs[[i]]\,/\,(2\,\Sigma t)\right\},\,\{i,\,1,\,\mathrm{Length}[xs]\}\right][[1\;;;\,-1]]$$

In[51]:= infflatlandisopointisoscatter`fs =
FileNames["code/flatland/infiniteFlatland/Isotropicpointsource/data/
infflatland_isotropicpoint_isotropicscatter*"];

```
In[52]:= 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[54]= {0.01, 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999}

In[55]:= infflatlandisopointisoscatter`muts =
          Union[#[[2]] & /@ infflatlandisopointisoscatter`simulations]

Out[55]= {1, 3}

In[56]:= infflatlandisopointisoscatter`numcollorders =
           infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 15]];
         infflatlandisopointisoscatter`maxr =
          infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 7]];
         infflatlandisopointisoscatter`dr =
          infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 9]];
         infflatlandisopointisoscatter`numr =
           Floor[infflatlandisopointisoscatter`maxr / infflatlandisopointisoscatter`dr];
```

Compare Deterministic and MC

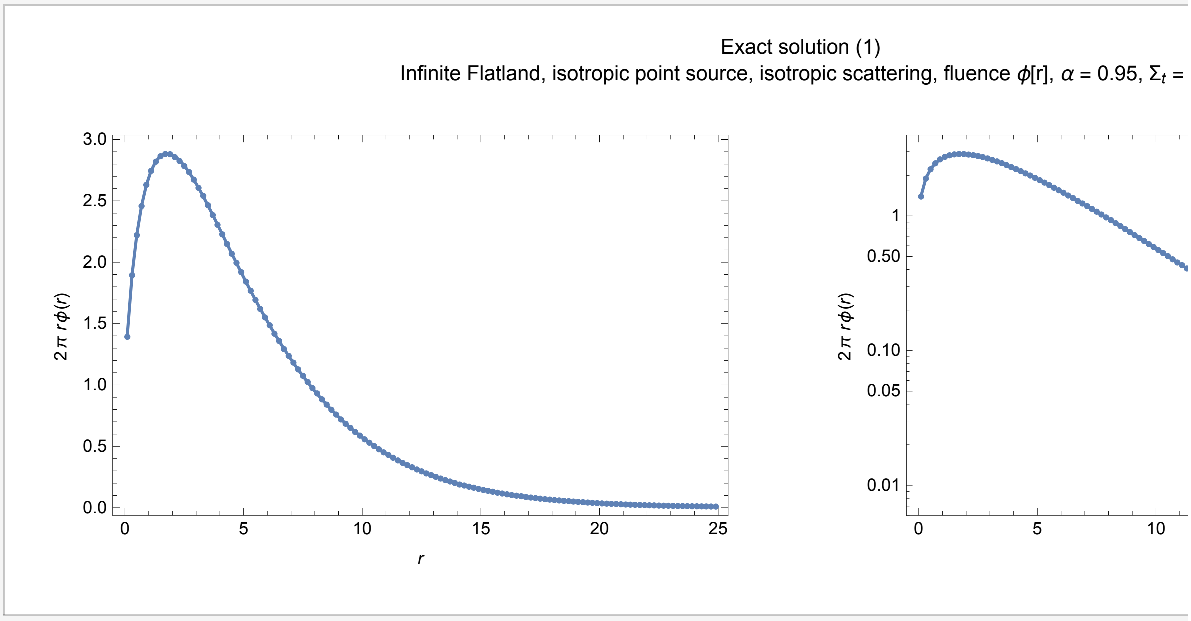
Fluence - Exact solution (1) comparison to MC

In[21]:=
```
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, 7]];
   dr = data[[2, 9]];

   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[22]=

α 0.95

Σt 1 3



Fluence - Exact solution (2) comparison to MC

In[23]:=

```
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, 7]];
   dr = data[[2, 9]];

   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}]
```

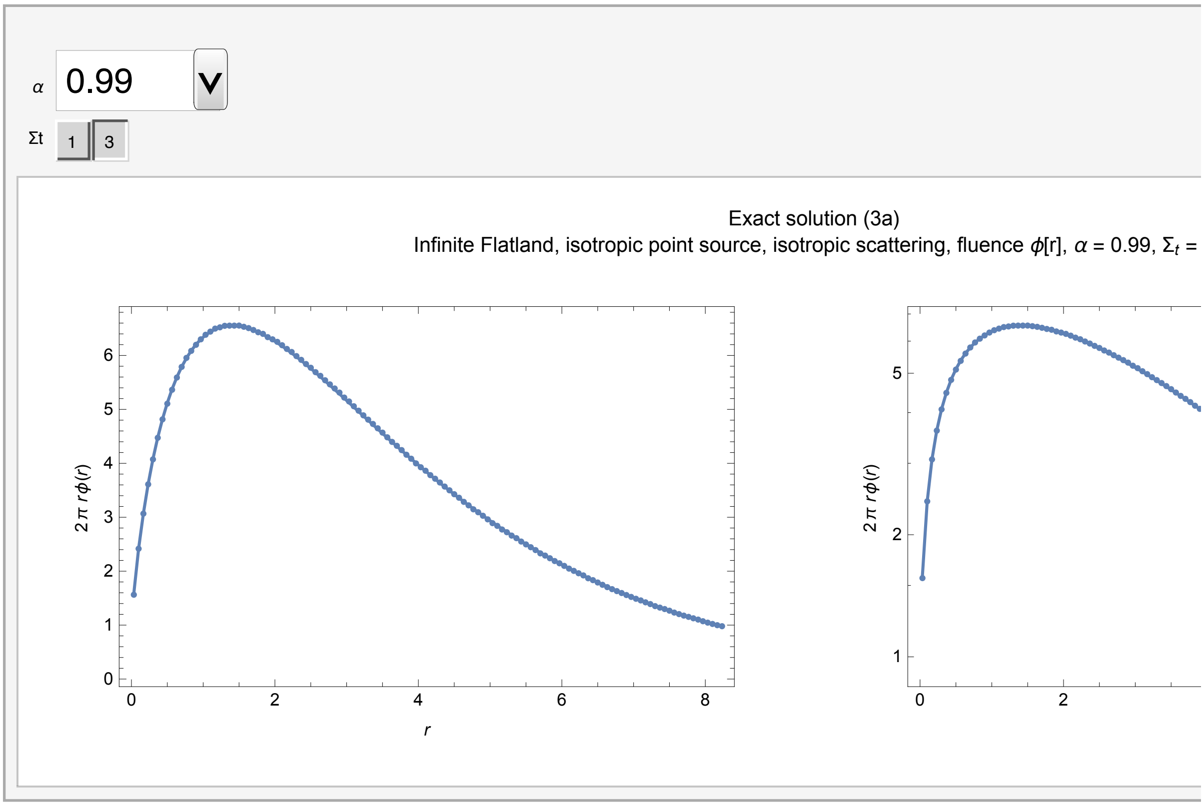
Out[24]=

```
α  0.99
Σt  1  3

$Aborted
```

Fluence - Exact solution (3a) comparison to MC

In[25]:=
```
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, 7]];
   dr = data[[2, 9]];

   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[26]=

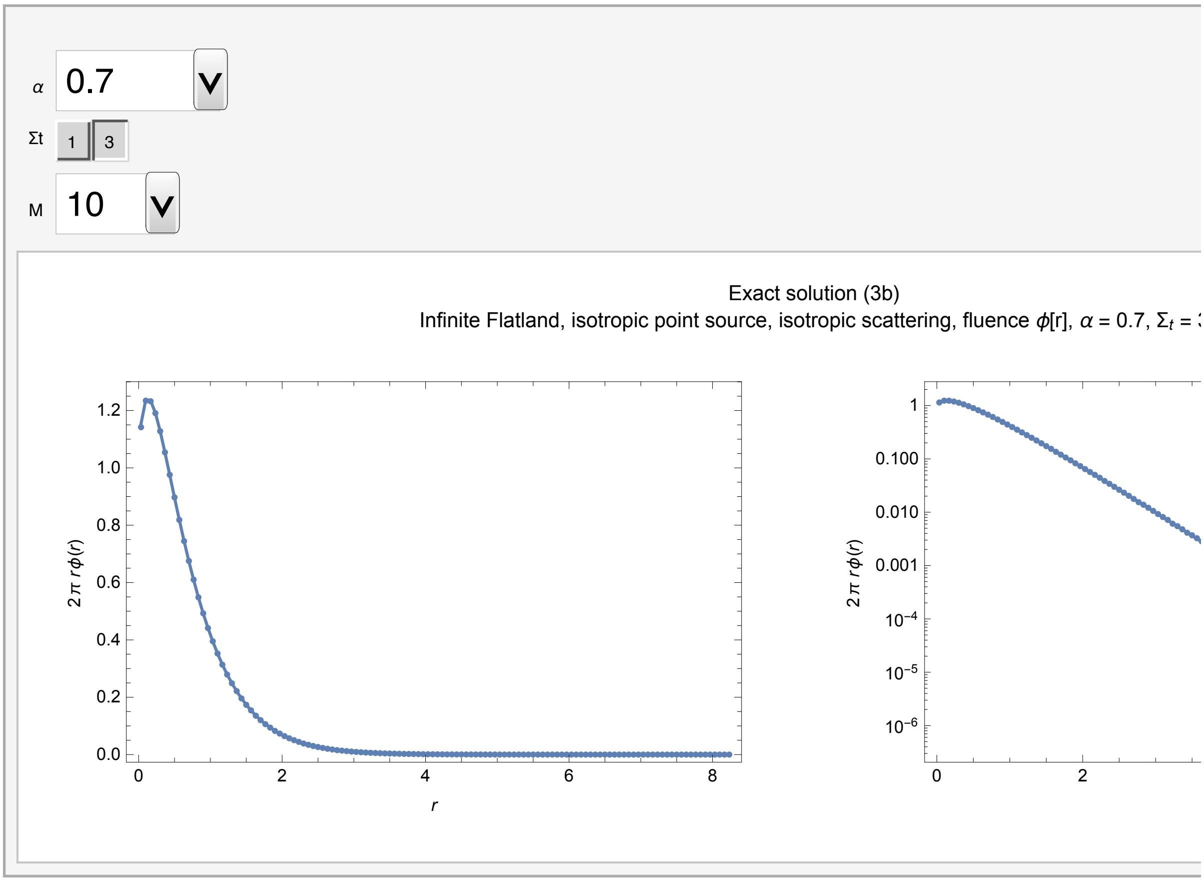


α 0.99
Σt 1 3
Exact solution (3a)
Infinite Flatland, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.99, Σt =
2π rϕ(r)
r

Fluence - Exact solution (3b) comparison to MC

In[27]:=
```
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, 7]];
   dr = data[[2, 9]];

   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]}]
```

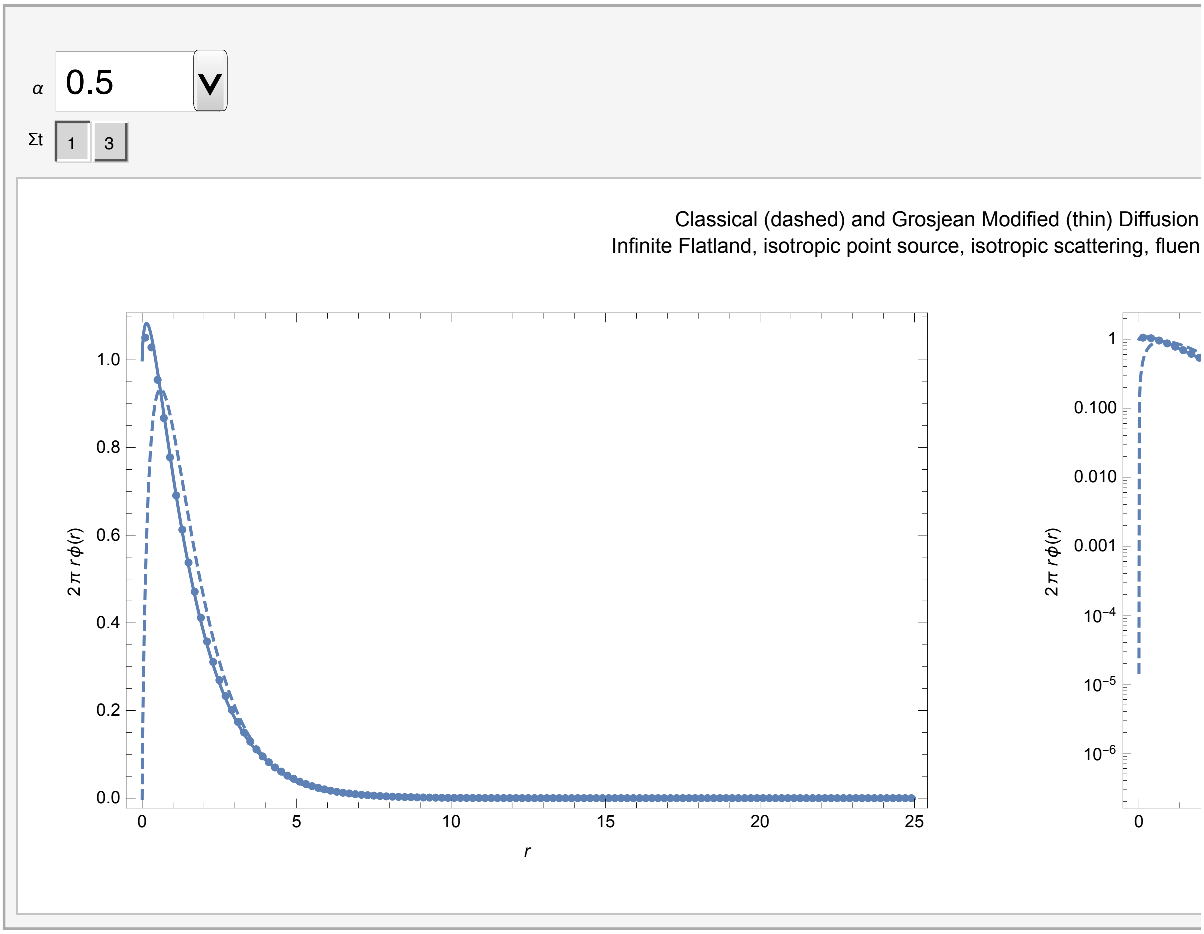
Out[28]=


α 0.7
Σt 1 3
M 10
Exact solution (3b)
Infinite Flatland, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.7, Σt = 3
2π rϕ(r)
r
0 2 4 6 8
0.0 0.2 0.4 0.6 0.8 1.0 1.2
1 0.100 0.010 0.001 10^-4 10^-5 10^-6
0 2

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

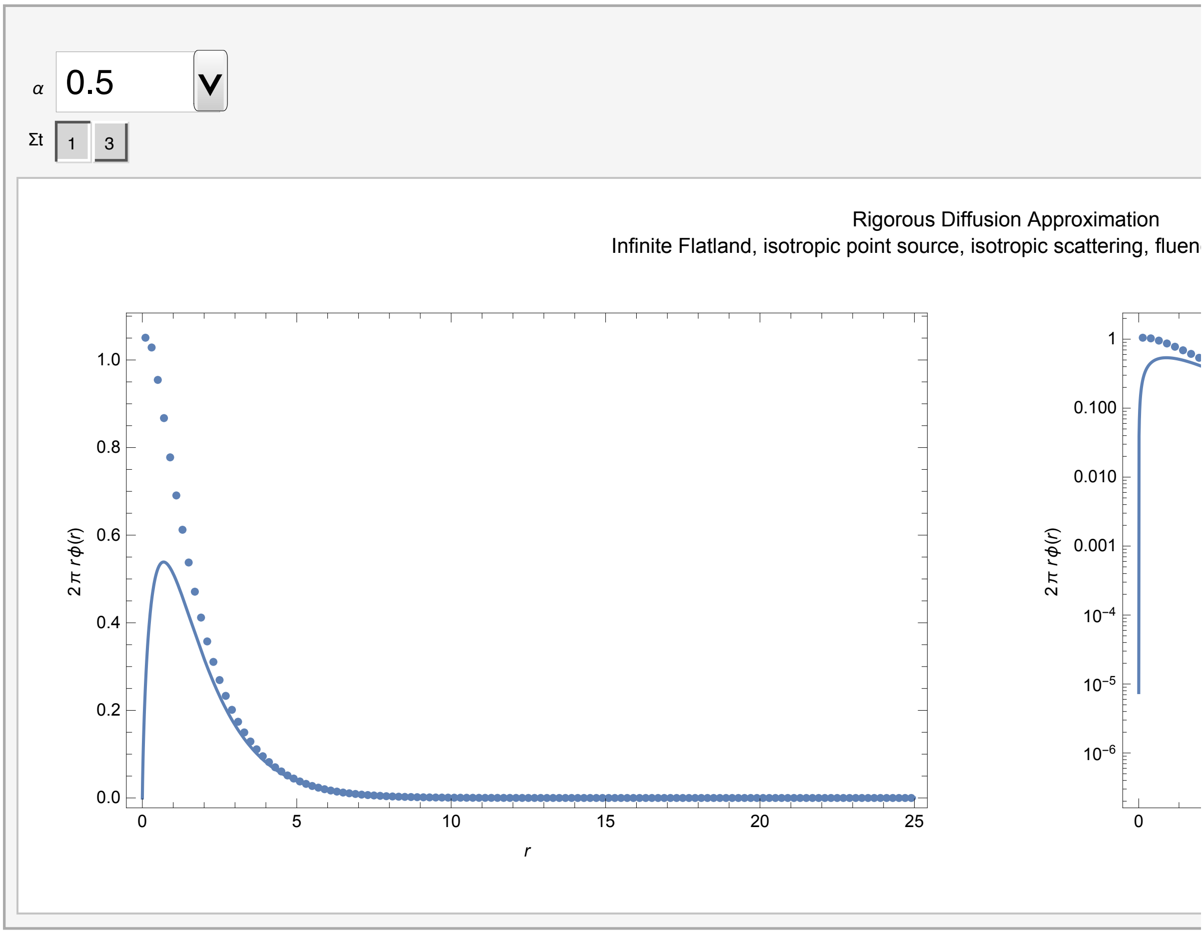
In[29]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = data[[2, 7]];
   dr = data[[2, 9]];

   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[30]=


α 0.5
Σt 1 3
Classical (dashed) and Grosjean Modified (thin) Diffusion
Infinite Flatland, isotropic point source, isotropic scattering, fluen
2 π r ϕ(r)
r

Fluence - Diffusion approximation (Rigorous) comparison to MC

In[31]:=
```
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, 7]];
   dr = data[[2, 9]];

   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}]
```

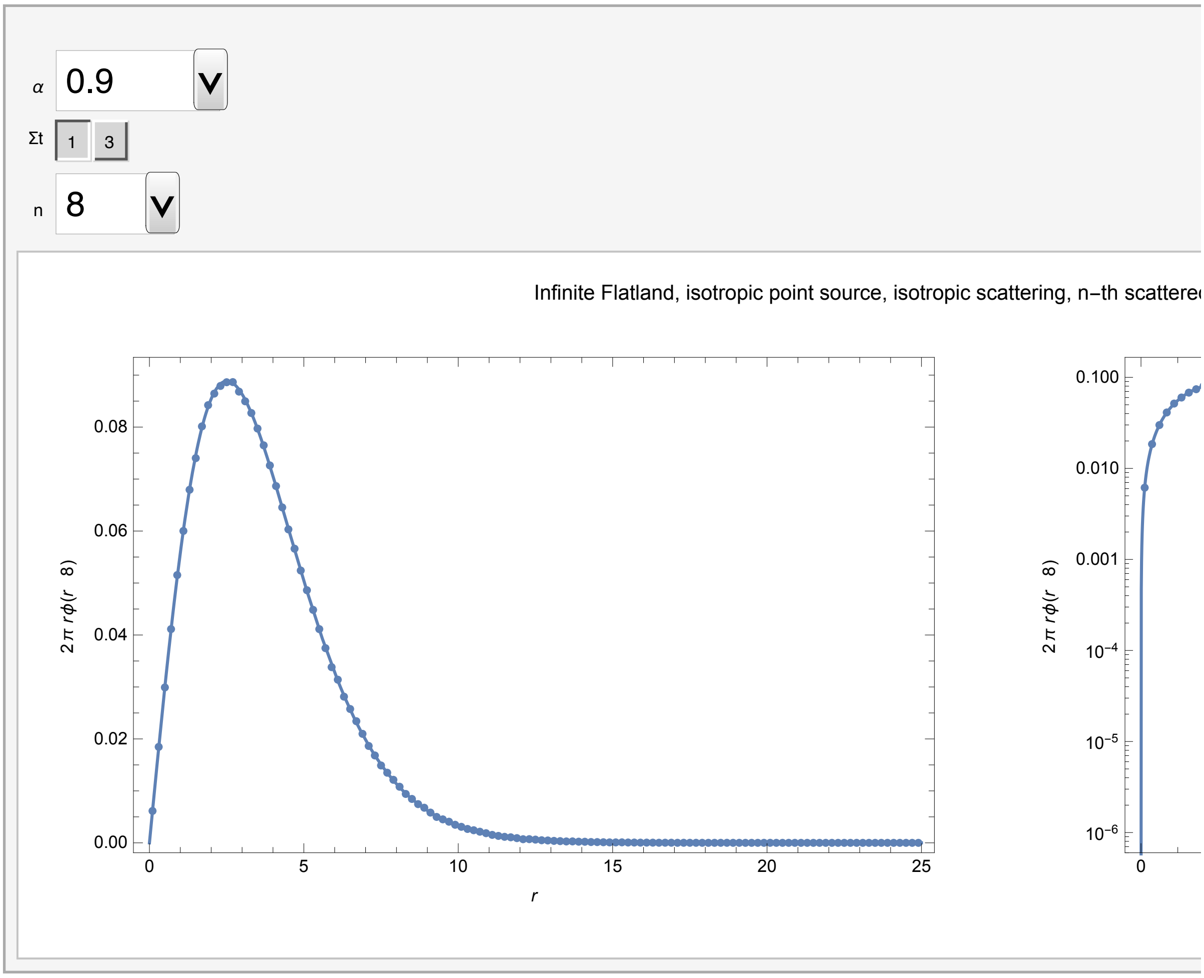


α 0.5
Σt 1 3
Rigorous Diffusion Approximation
Infinite Flatland, isotropic point source, isotropic scattering, fluen
2π r ϕ(r)
r
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
1
0.100
0.010
0.001
10^-4
10^-5
10^-6

N-th order fluence / scalar flux

In[33]:=
```
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, 7]];
   dr = data[[2, 9]];
   numorders = data[[2, 15]];

   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[34]=


α 0.9
Σt 1 3
n 8
Infinite Flatland, isotropic point source, isotropic scattering, n–th scattere
2π rϕ(r 8)
r
0.00
0.02
0.04
0.06
0.08
0
5
10
15
20
25
0.100
0.010
0.001
10⁻⁴
10⁻⁵
10⁻⁶

Compare moments of ϕ

In[35]:=
```
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, 17]];
   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[35]=

α 0.95

Σt 1 3

k	analytic	MC
0	6.66667	6.66647
1	10.931	10.9305
2	29.6296	29.6046
3	111.733	111.425
4	539.918	536.417

n-th collided moments of ϕ

In[36]:=
```
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, 17]];
   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[36]=

α 0.8

Σt 1 3

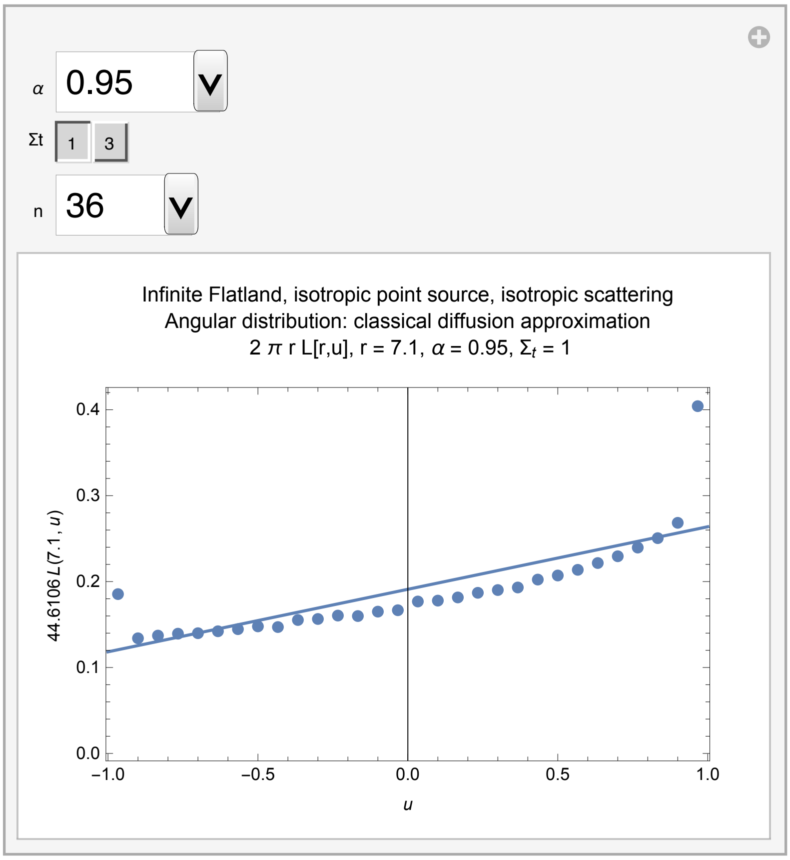
n 11

k	analytic	MC
0	0.0286331	0.0286669
1	0.0405845	0.0406517
2	0.076355	0.0765137
3	0.175866	0.176313
4	0.475098	0.476537

Angular distributions

In[186]:=
```
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, 15]];
   du = data[[2, 11]];
   dr = data[[2, 9]];

   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[186]=


α 0.95
Σt 1 3
n 36
Infinite Flatland, isotropic point source, isotropic scattering
Angular distribution: classical diffusion approximation
2 π r L[r,u], r = 7.1, α = 0.95, Σt = 1
44.6106 L(7.1, u)
u

In[152]:=
```
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, 15]];
   du = data[[2, 11]];
   dr = data[[2, 9]];

   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`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[152]=

α 0.9

Σt 1 3

n 44