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

© 2015 Eugene d'Eon

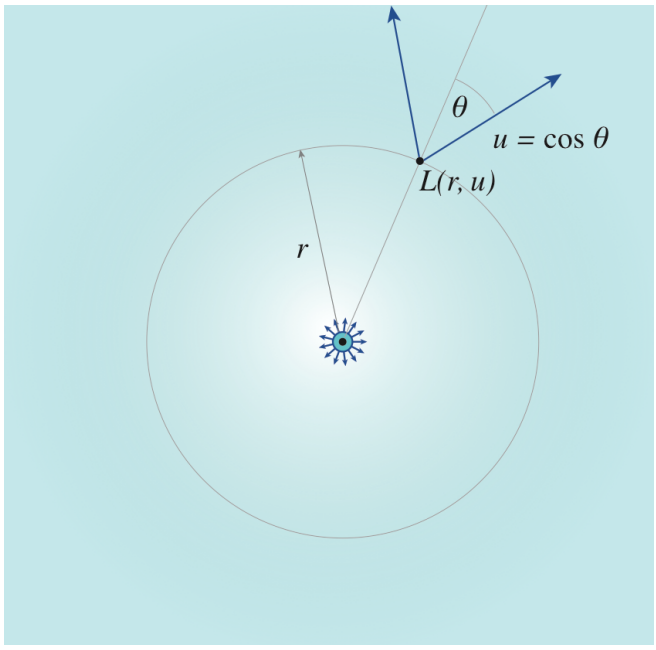
www.eugenedeon.com

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[281]:=
```
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[282]:=
```
infflatlandisopointisoscatter`ϕexact1[r_, Σt_, α_] := Exp[-Σt r]/(2 Pi r) +
  Σt/(2 Pi) NIntegrate[α z BesselJ[0, r z Σt]/(1 + z^2 - α Sqrt[1 + z^2]), {z, 0, Infinity}, Method → "LevinRule"]
```

Fluence: exact solution (2)

In[283]:=
```
infflatlandisopointisoscatter`ϕexact2[r_, Σt_, α_] :=
  Σt α BesselK[0, r Σt Sqrt[1 - α^2]]/Pi +
  Σt/(2 Pi) NIntegrate[z BesselJ[0, Σt r z]/(Sqrt[1 + z^2] + α), {z, 0, Infinity}, Method → "LevinRule"]
```

Fluence: exact solution (3)

In[299]:=
```
infflatlandisopointisoscatter`ϕexact3a[r_, Σt_, α_] :=
  NIntegrate[Σt/(2 Pi) k BesselJ[0, k Σt r]/(Sqrt[k^2 + 1] - α), {k, 0, Infinity}, Method → "LevinRule"]
```

In[312]:=
```
besselk[n_, x_] := Sqrt[2/(Pi x)] BesselK[n + 1/2, x];
infflatlandisopointisoscatter`ϕexact3b[r_, Σt_, α_, M_] :=
  Exp[-Σt r]/(2 Pi r) + α Σt/(2 Pi) BesselK[0, Σt Sqrt[1 - α^2] r] +
  Σt/(2 Pi) Sum[α^(2 n) n!/(2 n)! (2 Σt r)^n besselk[n - 1, Σt r], {n, 1, M}]
```

Grosjean-style diffusion approximation

In[284]:=
```
infflatlandisopointisoscatter`ϕGrosjean[r_, Σt_, α_] :=
  Exp[-r Σt]/(2 Pi r) + α Σt/((2 - α) Pi) BesselK[0, r Σt (Sqrt[2] Sqrt[1 - α]/Sqrt[2 - α])]
```

n-th scattered fluence

In[285]:=
```
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])
```

load MC data

In[286]:=
```
infflatlandisopointisoscatter`ppoints[xs_, dr_, maxx_, Σt_] :=
  Table[{dr (i) - 0.5 dr, xs[[i]]/Σt}, {i, 1, Length[xs]}][[1 ;; -2]]
```

In[287]:=
```
infflatlandisopointisoscatter`fs =
  FileNames["code/flatland/infiniteFlatland/Isotropicpointsource/data/
     infflatland_isotropicpoint_isotropicscatter*"];
```

In[288]:=
```
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[290]=
```
{0.01, 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999}
```

In[291]:=
```
infflatlandisopointisoscatter`muts =
 Union[#[[2]] & /@ infflatlandisopointisoscatter`simulations]
```

Out[291]=
```
{1, 3}
```

In[292]:=
```
infflatlandisopointisoscatter`numcollorders =
  infflatlandisopointisoscatter`simulations[[1]][[3]][[2, 13]];
```

Compare Deterministic and MC

Fluence - Exact solution (1) comparison to MC

In[293]:=
```
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}]
```

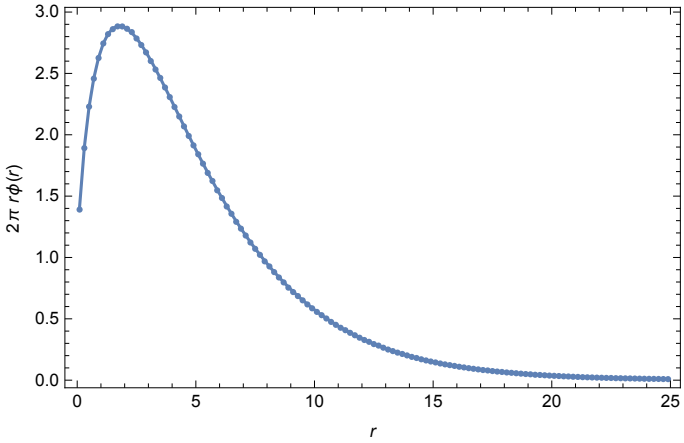
Out[294]=

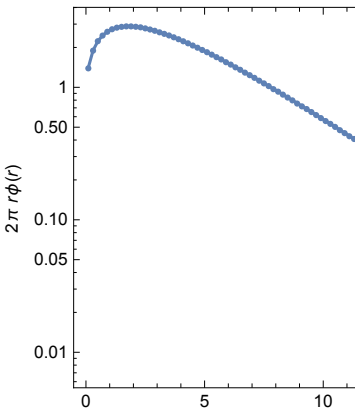
α 0.95

Σt 1 3

Exact solution (1)

Infinite Flatland, isotropic point source, isotropic scattering, fluence $\phi[r]$, $\alpha = 0.95$, $\Sigma_t =$

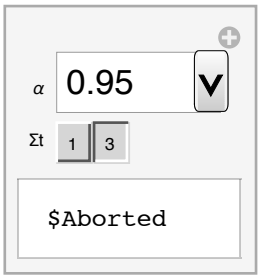




Fluence - Exact solution (2) comparison to MC

In[295]:=
```
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[296]=



Fluence - Exact solution (3a) comparison to MC

In[302]:=
```
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}]
```

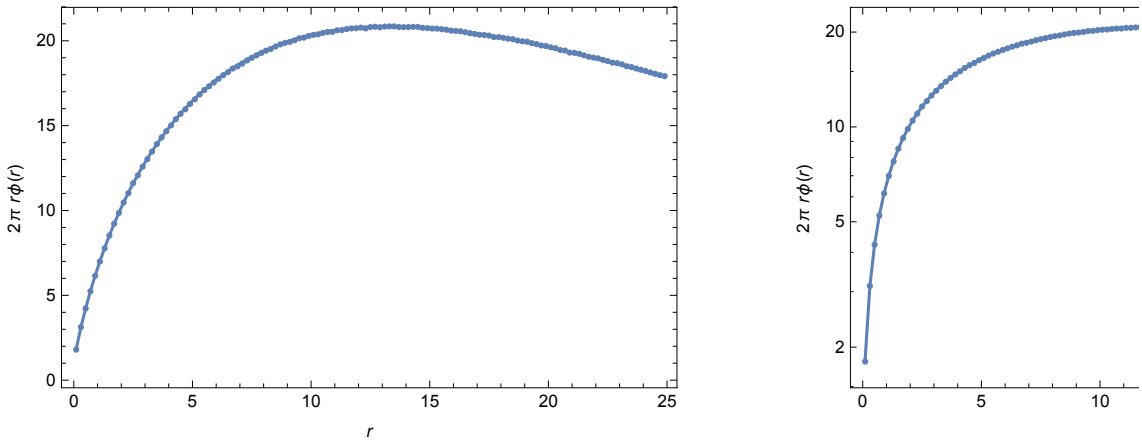
Out[303]=

α 0.999

Σt 1 3

Exact solution (3a)

Infinite Flatland, isotropic point source, isotropic scattering, fluence ϕ[r], α = 0.999, Σ_t =



Fluence - Exact solution (3b) comparison to MC

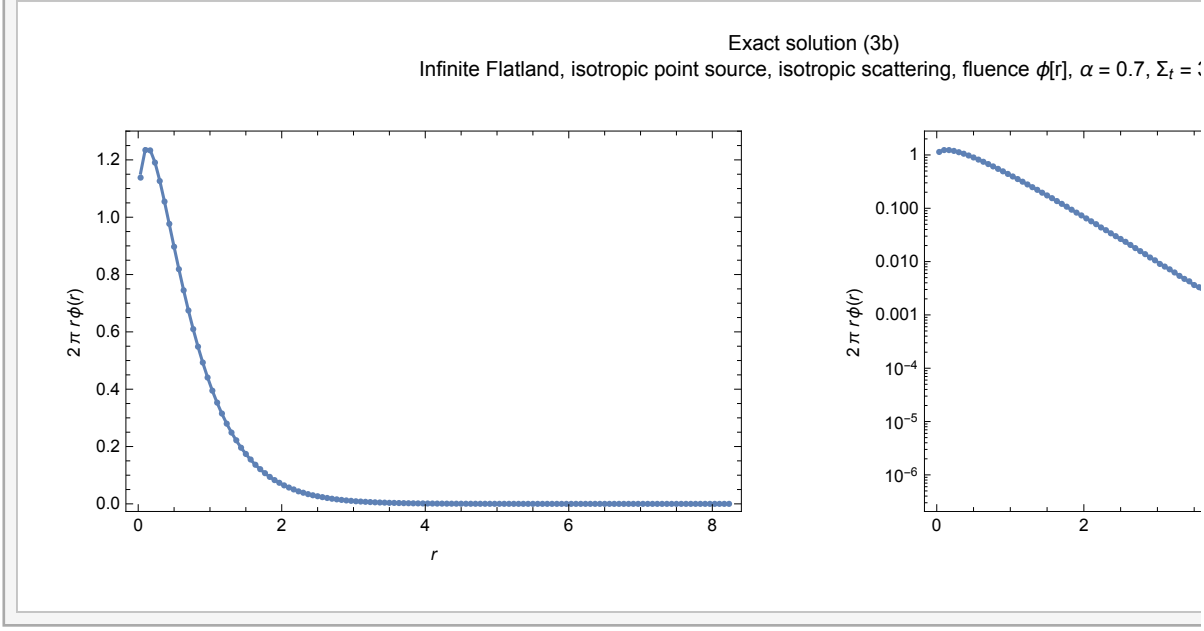
In[314]:=
```
Clear[alpha, Σt];
Manipulate[
 If[Length[infflatlandisopointisoscatter`simulations] > 0,
  Module[{data, maxr, dr, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[infflatlandisopointisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxr = 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[315]=

α 0.7

Σt 1 3

M 10



Fluence - Grosjean Modified Diffusion comparison to MC

In[297]:=
```
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],
      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],
      Frame → True,
      FrameLabel -> {{2 Pi r ϕ[r],}, {r,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800], PlotLabel ->
     "Grosjean Modified 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, 0.95}, infflatlandisopointisoscatter`alphas},
{Σt, infflatlandisopointisoscatter`muts}]
```

Out[298]=

α 0.95

Σt 1 3