Infinite 3D medium, Isotropic Plane Source, Isotropic Scattering

Exponential Random Flight

This is code to accompany the book:

A Hitchhiker's Guide to Multiple Scattering

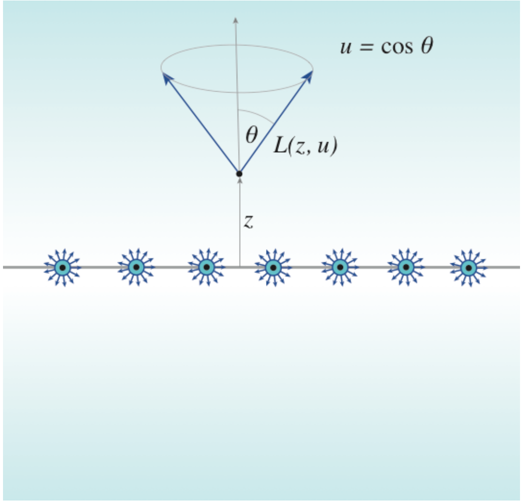
www.eugenedeon.com

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

Notation



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

Analytic solutions

Caseology quantities

In[5]:=
$$\mathtt{CaseN0[c\_,\ v0\_]} := \frac{1}{2}\, c\, v0^{3} \left(\frac{c}{v0^{2}-1} - \frac{1}{v0^{2}}\right)$$

In[6]:=
$$\mathtt{Casev0[c\_?NumericQ]} := \mathtt{FindRoot}\left[c\, v\, \mathtt{ArcTanh}\left[\frac{1}{v}\right] - 1 == 0,\ \left\{v,\ 1.00000000001,\ 10^{10}\right\},\ \mathtt{Method} \to \text{"Brent"}\right][[1]][[2]]$$

In[7]:=
$$\mathtt{CaseN[c\_,\ v\_]} := v \left(\mathtt{Case}\lambda[v,\ c]^{2} + \left(\frac{\pi\, c\, v}{2}\right)^{2}\right)$$

In[8]:=
$$\mathtt{Case}\lambda[v\_,\ c\_] := 1 - c\, v\, \mathtt{ArcTanh}[v]$$

In[9]:=
$$\mathtt{Case}\psi\mathtt{0}[u\_,\ v0\_,\ c\_,\ z\_] := \frac{c}{2}\, \frac{v0}{v0 - \mathtt{Sign}[z]\, u}$$

Rigorous diffusion approximation

In[10]:=
$$\mathtt{inf3Disoplaneisoscatter`}\phi\mathtt{rigourousDiffusion}[z\_,\ \Sigma t\_,\ \alpha\_] := \frac{1}{2}\, \frac{E^{-\mathtt{Abs}[z]\, \Sigma t/\#}}{\mathtt{CaseN0}[\alpha,\ \#]}\ \&[\mathtt{Casev0}[\alpha]]$$

Fluence: exact solution

[Case and Zwiefel 1967]

In[11]:=
$$\mathtt{inf3Disoplaneisoscatter`}\phi\mathtt{exact}[z\_,\ \Sigma t\_,\ \alpha\_] := \mathtt{inf3Disoplaneisoscatter`}\phi\mathtt{rigourousDiffusion}[z,\ \Sigma t,\ \alpha] + \frac{1}{2}\, \mathtt{NIntegrate}\left[\frac{e^{-\Sigma t\, \mathtt{Abs}[z]\, /v}}{\mathtt{CaseN}[\alpha,\ v]},\ \{v,\ 0,\ 1\}\right]$$

Nth-scattered fluence

In[12]:=
$$\mathtt{inf3Disoplaneisoscatter`}\phi\mathtt{exact1}[x\_,\ \Sigma t\_,\ \alpha\_,\ n\_] := \frac{(\alpha\, \Sigma t)^{n}}{\pi}\, \mathtt{NIntegrate}\left[\frac{\mathtt{ArcTan}\left[\frac{z}{\Sigma t}\right]^{n+1} \mathtt{Cos}[x\, z]}{z^{n+1}},\ \{z,\ 0,\ \mathtt{Infinity}\},\ \mathtt{Method} \to \text{"ExtrapolatingOscillatory"}\right]$$

In[13]:=
$$\mathtt{inf3Disoplaneisoscatter`}\phi\mathtt{exact2}[x\_,\ \Sigma t\_,\ \alpha\_,\ n\_] := \frac{\alpha^{n}}{\pi}\, \mathtt{Chop}\left[\mathtt{NIntegrate}\left[i\, 2^{-2-n}\, e^{-\mathtt{Abs}[x]\, z\, \Sigma t} \left(\left(\frac{-i\, \pi + \mathtt{Log}\left[\frac{z+1}{z-1}\right]}{z}\right)^{1+n} - \left(\frac{i\, \pi + \mathtt{Log}\left[\frac{z+1}{z-1}\right]}{z}\right)^{1+n}\right),\ \{z,\ 1,\ \mathtt{Infinity}\}\right]\right]$$

In[14]:=
$$\mathtt{inf3Disoplaneisoscatter`}\phi\mathtt{Gaussian}[x\_,\ \Sigma t\_,\ \alpha\_,\ n\_] := \frac{e^{-\frac{3\, x^{2}\, \Sigma t^{2}}{4\, (1+n)}} \sqrt{\frac{3}{\pi}}\, \alpha^{n}}{2\, \sqrt{1+n}}$$

Classical Diffusion Approximation

In[15]:=
$$\texttt{inf3Disoplaneisoscatter`}\phi\texttt{Diffusion[x\_, }\Sigma\texttt{t\_, }\alpha\texttt{\_] :=}\ \frac{3\,e^{-\texttt{Abs[x]}\sqrt{3-3\,\alpha}\ \Sigma\texttt{t}}}{2\,\sqrt{3-3\,\alpha}}$$

Grosjean Modified Diffusion Approximation

In[16]:=
$$\texttt{inf3Disoplaneisoscatter`}\phi\texttt{Grosjean[x\_, }\Sigma\texttt{t\_, }\alpha\texttt{\_] :=}$$
$$-\frac{1}{2}\,\texttt{ExpIntegralEi[-Abs[x] }\Sigma\texttt{t]}+\frac{e^{-\texttt{Abs[x]}\sqrt{3+\frac{3}{-2+\alpha}}\ \Sigma\texttt{t}}\sqrt{3+\frac{3}{-2+\alpha}}\ \alpha}{2\ (1-\alpha)}$$

Radiance (Angular Flux)

Caseology - asymptotic solution:

In[17]:=
$$\texttt{inf3Disoplaneisoscatter`LrigourousDiffusion[z\_, u\_, }\Sigma\texttt{t\_, }\alpha\texttt{\_] :=}$$
$$\frac{1}{\texttt{4 Pi}}\ \texttt{Case}\psi\texttt{0[u, \#, }\alpha\texttt{, z]}\ \frac{\texttt{E}^{-\texttt{Abs[z] }\Sigma\texttt{t/\#}}}{\texttt{CaseN0[}\alpha\texttt{, \#]}}\ \texttt{\&[Casev0[}\alpha\texttt{]]}$$

Caseology - exact solution:

In[18]:=
$$\texttt{inf3Disoplaneisoscatter`Lexact[z\_, u\_, }\Sigma\texttt{t\_, }\alpha\texttt{\_] :=}$$
$$\texttt{inf3Disoplaneisoscatter`LrigourousDiffusion[z, u, }\Sigma\texttt{t, }\alpha\texttt{] +}$$
$$\frac{1}{\texttt{4 Pi}}\Bigg(\texttt{Case}\lambda\texttt{[u, }\alpha\texttt{]}\ \frac{e^{-\frac{\texttt{Abs[z] }\Sigma\texttt{t}}{\texttt{u}}}}{\texttt{CaseN[}\alpha\texttt{, u]}}\ \texttt{HeavisideTheta[1 - u] HeavisideTheta[u]}$$
$$+\ \texttt{NIntegrate}\Big[\frac{e^{-\frac{\texttt{Abs[z] }\Sigma\texttt{t}}{\texttt{v}}}}{\texttt{CaseN[}\alpha\texttt{, v]}}\ \frac{\alpha}{2}\ \frac{\texttt{v}}{\texttt{v - u}}$$
$$\texttt{, \{v, 0, u, 1\}, Method}\rightarrow\texttt{"PrincipalValue", PrecisionGoal}\rightarrow\texttt{5}\Big]\Bigg)$$

Fourier transform exact solution (found by integrating the Green’s function of [Benoist and Kavenoky 1968] over one cosine):

In[19]:=
$$\texttt{inf3Disoplaneisoscatter`FourierLcollided[z\_, k\_, c\_, u\_] :=}$$
$$\frac{\texttt{c}\ e^{-i\,\texttt{k z}}\ \big(\texttt{Log[}i\texttt{ - k] - Log[}i\texttt{ + k]}\big)}{\texttt{16 Pi}^2\ \big(i+\texttt{k u}\big)\ \texttt{(k - c ArcTan[k])}}$$

In[37]:=
$$\texttt{inf3Disoplaneisoscatter`LexactFourier[z\_, u\_, }\Sigma\texttt{t\_, }\alpha\texttt{\_] :=}$$
$$\texttt{Re[NIntegrate[inf3Disoplaneisoscatter`FourierLcollided[z }\Sigma\texttt{t, k, }\alpha\texttt{, u] +}$$
$$\texttt{inf3Disoplaneisoscatter`FourierLcollided[z }\Sigma\texttt{t, -k, }\alpha\texttt{, u],}$$
$$\texttt{\{k, 0, Infinity\}]] +}\ \frac{e^{-\frac{\texttt{z }\Sigma\texttt{t}}{\texttt{u}}}}{\texttt{4 Pi u}}\ \texttt{Sign[z] HeavisideTheta[u Sign[z]]}$$

load MC data

In[21]:=
$$\texttt{inf3Disoplaneisoscatter`ppoints[zs\_, dz\_, maxz\_, }\Sigma\texttt{t\_] :=}$$
$$\texttt{Table}\Big[\Big\{\texttt{dz (i) - 0.5 dz - maxz, zs[[i]]}\Big/\Sigma\texttt{t}\Big\}\texttt{, \{i, 1, Length[zs]\}}\Big]\texttt{[[2 ;; -2]]}$$

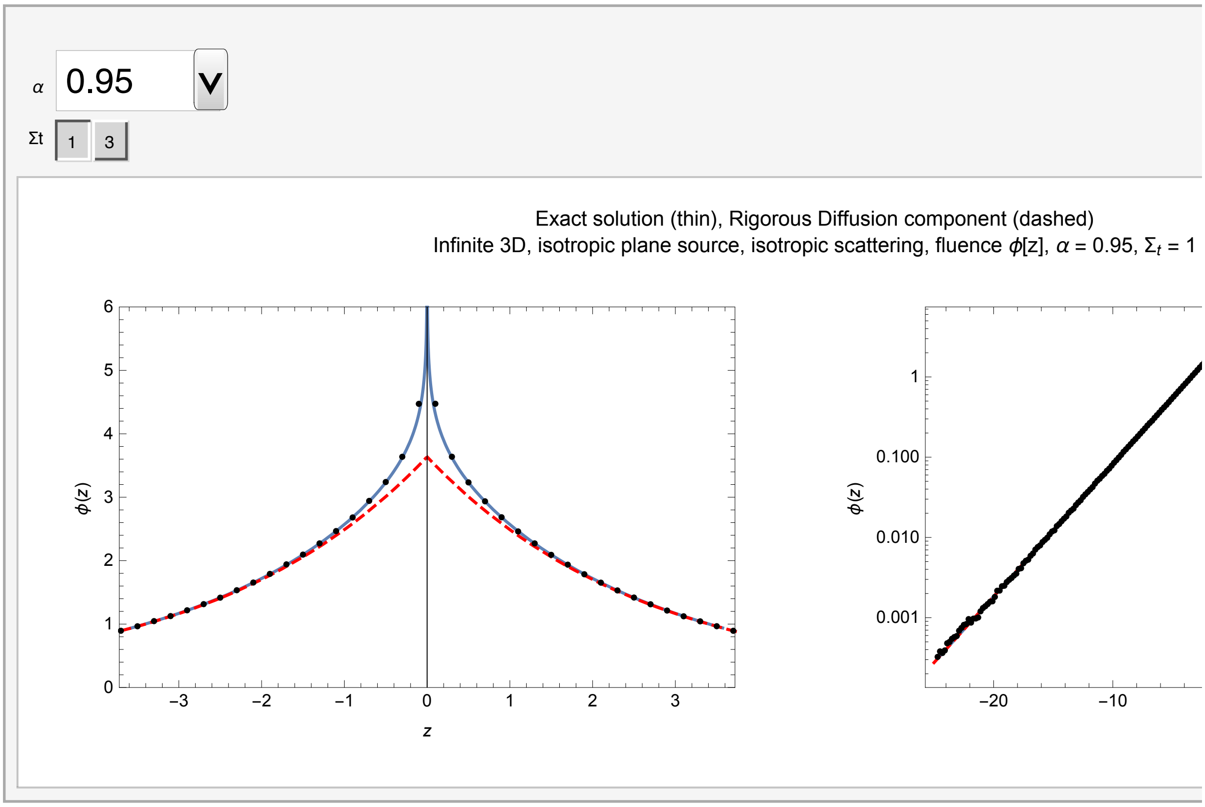
In[22]:=
$$\texttt{inf3Disoplaneisoscatter`ppointsu[xs\_, du\_, }\Sigma\texttt{t\_] :=}$$
$$\texttt{Table}\Big[\Big\{\texttt{-1.0 + du (i) - 0.5 du, xs[[i]]}\Big/\texttt{(2 }\Sigma\texttt{t)}\Big\}\texttt{, \{i, 1, Length[xs]\}}\Big]\texttt{[[1 ;; -1]]}$$

```
In[23]:= inf3Disoplaneisoscatter`fs =
           FileNames["code/3D_medium/infinite3Dmedium/Isotropicplanesource/data/
              inf3D_isotropicplane_isotropicscatter*"];

In[24]:= inf3Disoplaneisoscatter`index[x_] := Module[{data, α, Σt},
            data = Import[x, "Table"];
            Σt = data[[1, 13]];
            α = data[[2, 3]];
            {α, Σt, data}];
         inf3Disoplaneisoscatter`simulations =
           inf3Disoplaneisoscatter`index /@ inf3Disoplaneisoscatter`fs;
         inf3Disoplaneisoscatter`alphas =
          Union[#[[1]] & /@ inf3Disoplaneisoscatter`simulations]

Out[26]= {0.01, 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999}

In[27]:= inf3Disoplaneisoscatter`muts =
          Union[#[[2]] & /@ inf3Disoplaneisoscatter`simulations]

Out[27]= {1, 3}

In[28]:= inf3Disoplaneisoscatter`numcollorders =
           inf3Disoplaneisoscatter`simulations[[1]][[3]][[2, 13]];
         inf3Disoplaneisoscatter`maxz =
          inf3Disoplaneisoscatter`simulations[[1]][[3]][[2, 5]];
         inf3Disoplaneisoscatter`dz =
          inf3Disoplaneisoscatter`simulations[[1]][[3]][[2, 7]];
         inf3Disoplaneisoscatter`numz =
           Floor[2 inf3Disoplaneisoscatter`maxz / inf3Disoplaneisoscatter`dz];
```

Compare Deterministic and MC

Fluence - Exact solution comparison to MC

```
Manipulate[
 If[Length[inf3Disoplaneisoscatter`simulations] > 0,
  Module[{data, maxz, dz, pointsϕ, plotpointsϕ, logplotϕ, plotϕ, exact1points},
   data = SelectFirst[
      inf3Disoplaneisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
   maxz = data[[2, 5]];
   dz = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disoplaneisoscatter`ppoints[pointsϕ, dz, maxz, Σt];

   exact1points =
    Quiet[{#[[1]], inf3Disoplaneisoscatter`ϕexact[#[[1]], Σt, α]}] & /@
     plotpointsϕ;

   numpoints = Length[plotpointsϕ];
                            6 1
   skip = Floor[numpoints   - -];
                            7 2

   plotϕ = Quiet[Show[
      (*ListPlot[exact1points[[skip;;-skip]],PlotRange→{0,6},Joined→True],*)
      Plot[inf3Disoplaneisoscatter`ϕexact[z, Σt, α],
            maxz   maxz
       {z, - ----, ----}, PlotRange → {0, 6}],
             7      7
      Plot[inf3Disoplaneisoscatter`ϕrigourousDiffusion[z, Σt, α],
       {z, -maxz, maxz}, PlotRange → All, PlotStyle → {Red, Dashed}],
      ListPlot[plotpointsϕ[[skip ;; -skip]], PlotRange → All,
       PlotStyle → {Black, PointSize[.01]}],
      Frame → True,
      FrameLabel -> {{ ϕ[z],}, {z,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      LogPlot[inf3Disoplaneisoscatter`ϕrigourousDiffusion[z, Σt, α],
       {z, -maxz, maxz}, PlotRange → All, PlotStyle → {Red, Dashed}],
      ListLogPlot[plotpointsϕ, PlotRange → All,
       PlotStyle → {Black, PointSize[.01]}],
      Frame → True,
      FrameLabel -> {{ ϕ[z],}, {z,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800], PlotLabel ->
     "Exact solution (thin), Rigorous Diffusion component (dashed)\nInfinite
        3D, isotropic plane source, isotropic scattering, fluence
        ϕ[z], 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}, inf3Disoplaneisoscatter`alphas},
 {{Σt, 1}, inf3Disoplaneisoscatter`muts}]
```



α 0.95

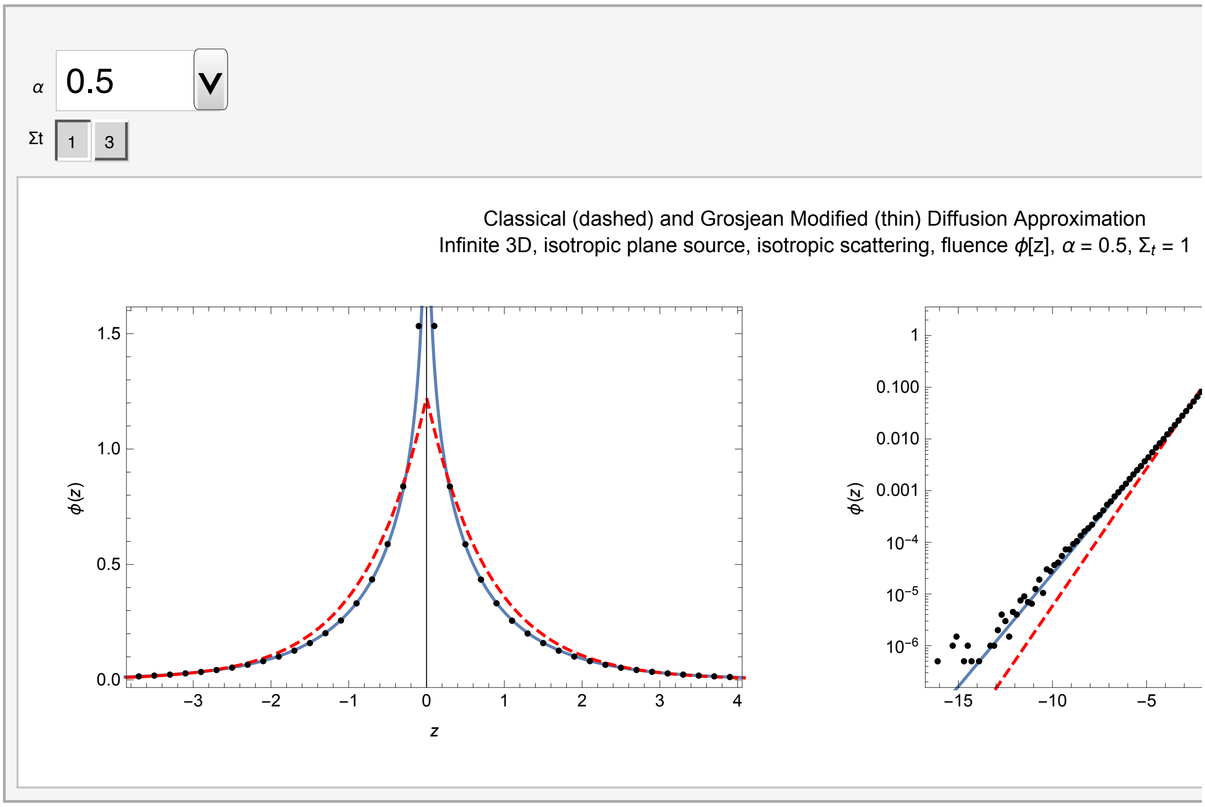
Σt [✓ 1] 3

Exact solution (thin), Rigorous Diffusion component (dashed)

Infinite 3D, isotropic plane source, isotropic scattering, fluence $\phi[z]$, α = 0.95, Σ_t = 1

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

```
Manipulate[
 If[Length[inf3Disoplaneisoscatter`simulations] > 0,
  Module[{data, maxz, dz, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numpoints, skip},
   data = SelectFirst[inf3Disoplaneisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxz = data[[2, 5]];
   dz = data[[2, 7]];

   pointsϕ = data[[4]];

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disoplaneisoscatter`ppoints[pointsϕ, dz, maxz, Σt];

   numpoints = Length[plotpointsϕ];
                          6 1
   skip = Floor[numpoints - -];
                          7 2

   plotϕ = Quiet[Show[
      ListPlot[plotpointsϕ[[skip ;; -skip]],
       PlotRange → All, PlotStyle → {PointSize[.01], Black}],
      Plot[inf3Disoplaneisoscatter`ϕGrosjean[z, Σt, α],
       {z, -maxz, maxz}, PlotRange → All],
      Plot[inf3Disoplaneisoscatter`ϕDiffusion[z, Σt, α],
       {z, -maxz, maxz}, PlotRange → All, PlotStyle → {Red, Dashed}],
      ListPlot[plotpointsϕ[[skip ;; -skip]], PlotRange → All,
       PlotStyle → {PointSize[.01], Black}],
      Frame → True,
      FrameLabel -> {{ ϕ[z],}, {z,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ,
       PlotRange → All, PlotStyle → {PointSize[.01], Black}],
      LogPlot[inf3Disoplaneisoscatter`ϕGrosjean[z, Σt, α],
       {z, -maxz, maxz}, PlotRange → All],
      LogPlot[inf3Disoplaneisoscatter`ϕDiffusion[z, Σt, α],
       {z, -maxz, maxz}, PlotRange → All, PlotStyle → {Red, Dashed}],
      ListLogPlot[plotpointsϕ, PlotRange → All,
       PlotStyle → {PointSize[.01], Black}],
      Frame → True,
      FrameLabel -> {{ ϕ[z],}, {z,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Classical (dashed) and Grosjean Modified (thin)
        Diffusion Approximation\nInfinite 3D, isotropic plane
        source, isotropic scattering, fluence ϕ[z], 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}, inf3Disoplaneisoscatter`alphas},
 {{Σt, 1}, inf3Disoplaneisoscatter`muts}]
```



α 0.5

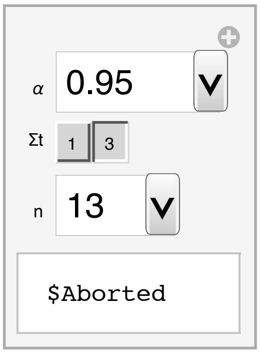
Σt 1 3

Classical (dashed) and Grosjean Modified (thin) Diffusion Approximation
Infinite 3D, isotropic plane source, isotropic scattering, fluence ϕ[z], α = 0.5, Σ_t = 1

N-th order fluence / scalar flux

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

```
Manipulate[
 If[Length[inf3Disoplaneisoscatter`simulations] > 0,
  Module[{data, maxz, dz, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[inf3Disoplaneisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxz = data[[2, 5]];
   dz = data[[2, 7]];

   numorders = data[[2, 13]];

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

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disoplaneisoscatter`ppoints[pointsϕ, dz, maxz, Σt];

   exact1points =
    Quiet[{#[[1]], inf3Disoplaneisoscatter`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 -> {{ ϕ["z|" <> ToString[n]],}, {z,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{ ϕ["z|" <> ToString[n]],}, {z,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (1)\nInfinite 3D, isotropic plane source,
        isotropic scattering, n-th scattered fluence ϕ[z|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, 0.99}, inf3Disoplaneisoscatter`alphas},
 {{Σt, 3}, inf3Disoplaneisoscatter`muts},
 {{n, 13}, Range[If[NumberQ[inf3Disoplaneisoscatter`numcollorders],
    inf3Disoplaneisoscatter`numcollorders, 1]]}

]
```



α 0.95

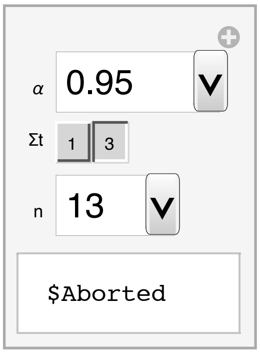
Σt 1 3

n 13

```
$Aborted
```

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

```
Manipulate[
 If[Length[inf3Disoplaneisoscatter`simulations] > 0,
  Module[{data, maxz, dz, pointsϕ,
    plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[inf3Disoplaneisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxz = data[[2, 5]];
   dz = data[[2, 7]];

   numorders = data[[2, 13]];

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

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disoplaneisoscatter`ppoints[pointsϕ, dz, maxz, Σt];

   exact1points =
    Quiet[{#[[1]], inf3Disoplaneisoscatter`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 -> {{ ϕ["z|" <> ToString[n]],}, {z,}}
     ]];
   logplotϕ = Quiet[Show[
      ListLogPlot[plotpointsϕ, PlotRange → All, PlotStyle → PointSize[.01]],
      ListLogPlot[exact1points, PlotRange → All, Joined → True],
      Frame → True,
      FrameLabel -> {{ ϕ["z|" <> ToString[n]],}, {z,}}
     ]];
   Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 800],
    PlotLabel -> "Exact solution (2)\nInfinite 3D, isotropic plane source,
        isotropic scattering, n-th scattered fluence ϕ[z|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, 0.99}, inf3Disoplaneisoscatter`alphas},
 {{Σt, 3}, inf3Disoplaneisoscatter`muts},
 {{n, 13}, Range[If[NumberQ[inf3Disoplaneisoscatter`numcollorders],
    inf3Disoplaneisoscatter`numcollorders, 1]]}

]
```


α 0.95

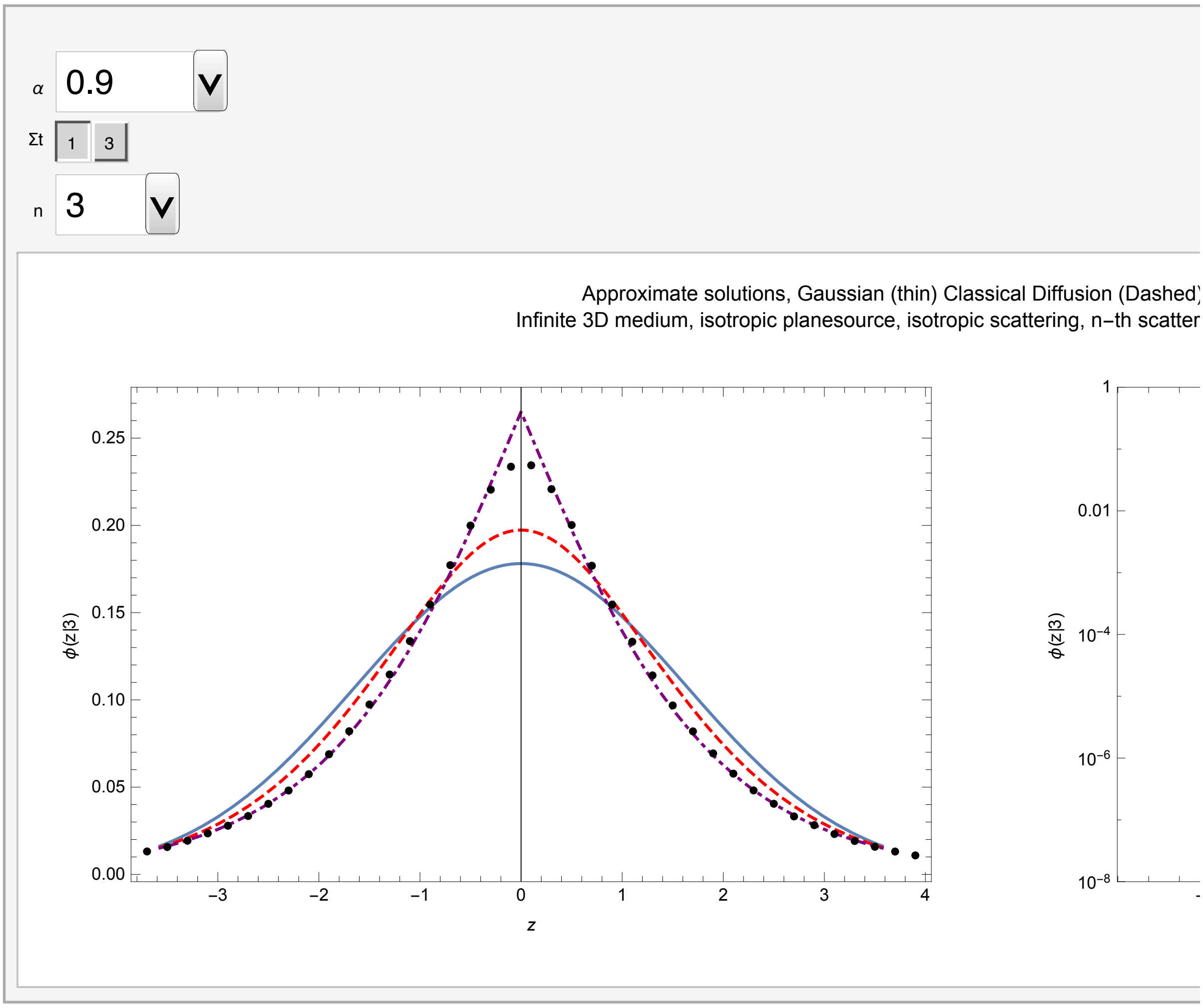
Σt 1 3

n 13

```
$Aborted
```

N-th collided Fluence - Approximations

```
Manipulate[
 If[Length[inf3Disoplaneisoscatter`simulations] > 0,
  Module[{data, maxz, dz, pointsϕ,
     plotpointsϕ, logplotϕ, plotϕ, exact1points, numorders},
   data = SelectFirst[inf3Disoplaneisoscatter`simulations,
      #[[1]] == α && #[[2]] == Σt &][[3]];
   maxz = data[[2, 5]];
   dz = data[[2, 7]];
   numorders = data[[2, 13]];

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

   (* divide by Σt to convert collision density into fluence *)
   plotpointsϕ = inf3Disoplaneisoscatter`ppoints[pointsϕ, dz, maxz, Σt];

   seriesclassical = α^n SeriesCoefficient[
      inf3Disoplaneisoscatter`ϕDiffusion[z, Σt, C], {C, 0, n}];
   seriesG = α^n SeriesCoefficient[inf3Disoplaneisoscatter`ϕGrosjean[z, Σt, C],
      {C, 0, n}];

   numpoints = Length[plotpointsϕ];
   skip = Floor[numpoints 6/7 1/2];

   plotϕ = Quiet[Show[
      Plot[inf3Disoplaneisoscatter`ϕGaussian[z, Σt, α, n],
       {z, -maxz/7, maxz/7}, PlotRange → All],
      Plot[seriesclassical, {z, -maxz/7, maxz/7}, PlotRange → All,
       PlotStyle → {Red, Dashed}],
      Plot[seriesG, {z, -maxz/7, maxz/7}, PlotRange → All,
       PlotStyle → {DotDashed, Purple}],
      ListPlot[plotpointsϕ[[skip ;; -skip]], PlotRange → All,
       PlotStyle → {Black, PointSize[.01]}],
      Frame → True,
      FrameLabel -> {{ ϕ["z|" <> ToString[n]],}, {z,}}
     ]];
   logplotϕ = Quiet[Show[
      LogPlot[inf3Disoplaneisoscatter`ϕGaussian[z, Σt, α, n],
       {z, -maxz, maxz}, PlotRange → {10^-8, 1}],
      LogPlot[seriesclassical, {z, -maxz, maxz},
        PlotRange → {10^-8, 1}, PlotStyle → {Red, Dashed}],
      LogPlot[seriesG, {z, -maxz, maxz}, PlotRange → {10^-8, 1},
        PlotStyle → {DotDashed, Purple}],
      ListLogPlot[plotpointsϕ, PlotRange → {10^-8, 1},
        PlotStyle → {Black, PointSize[.01]}],
      Frame → True,
      FrameLabel -> {{ ϕ["z|" <> ToString[n]],}, {z,}}
     ]];
  Show[GraphicsGrid[{{plotϕ, logplotϕ}}, ImageSize → 1000],
    PlotLabel -> "Approximate solutions, Gaussian (thin) Classical
        Diffusion (Dashed) Grosjean (Dot-Dashed)\nInfinite
        3D medium, isotropic planesource, isotropic
        scattering, n-th scattered fluence ϕ[z|" <>
      ToString[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}, inf3Disoplaneisoscatter`alphas},
{Σt, inf3Disoplaneisoscatter`muts},
{{n, 3}, Range[If[NumberQ[inf3Disoplaneisoscatter`numcollorders],
   inf3Disoplaneisoscatter`numcollorders, 1]]}]
```



α 0.9

Σt 1 3

n 3

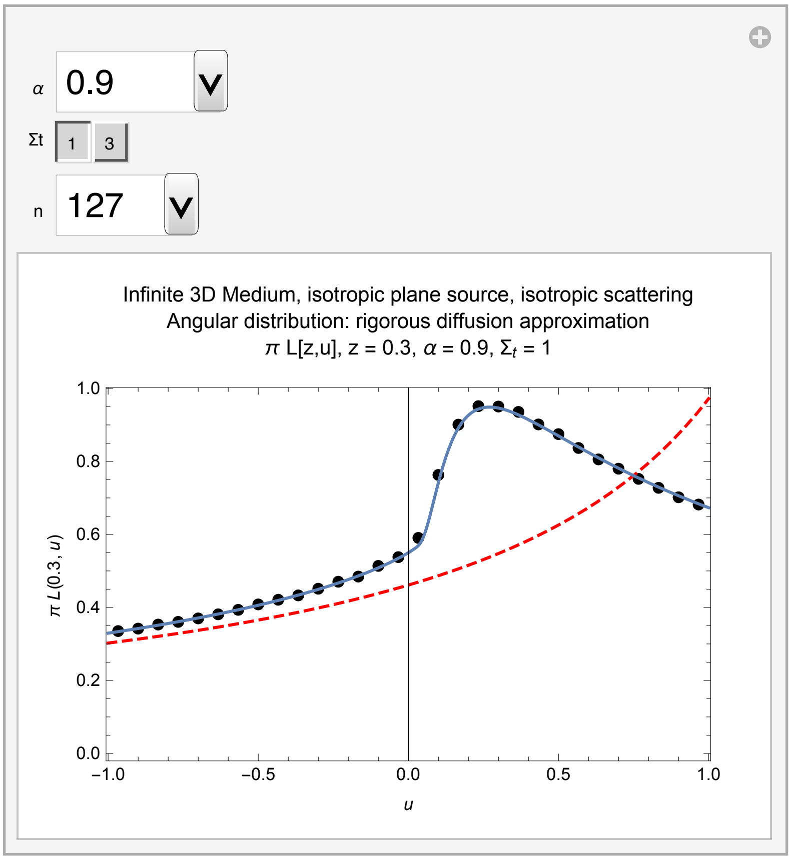
Approximate solutions, Gaussian (thin) Classical Diffusion (Dashed)
Infinite 3D medium, isotropic planesource, isotropic scattering, n−th scatter

Angular Distributions

Exact solution (Caseology)

```
In[378]:= Manipulate[
            If[Length[inf3Disoplaneisoscatter`simulations] > 0,
             Module[{data, numorders, pointsu, plotpointsu, du, r, dz, maxz, zsim},
              data = SelectFirst[
                 inf3Disoplaneisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
              numorders = data[[2, 13]];
              du = data[[2, 9]];
              dz = data[[2, 7]];
              maxz = data[[2, 5]];

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

              zsim  =  dz * n - 0.5 dz - maxz;

              plotpointsu = inf3Disoplaneisoscatter`ppointsu[pointsu, du, Σt];
              pp = Show[
                 ListPlot[plotpointsu, PlotRange → All, PlotStyle → Black,
                  Frame → True,
                  FrameLabel -> {{ Pi L[zsim, u],}, {u,}}],
                 Plot[Pi inf3Disoplaneisoscatter`LrigourousDiffusion[zsim, u, Σt, α],
                  {u, -1, 1}, PlotStyle → {Red, Dashed}
                 ],
                 Plot[Pi inf3Disoplaneisoscatter`Lexact[zsim, u, Σt, α], {u, -1, 1}
                 ],
                 PlotLabel -> "Infinite 3D Medium, isotropic plane
                     source, isotropic scattering\nAngular distribution:
                     rigorous diffusion approximation\nπ L[z,u], z = " <>
                   ToString[zsim] <> ", 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}, inf3Disoplaneisoscatter`alphas},
            {{Σt, 1}, inf3Disoplaneisoscatter`muts}, {{n, 127}, Range[If[
               NumberQ[inf3Disoplaneisoscatter`numz], inf3Disoplaneisoscatter`numz, 1]]}]
```

Out[378]=



α 0.9

Σt 1 3

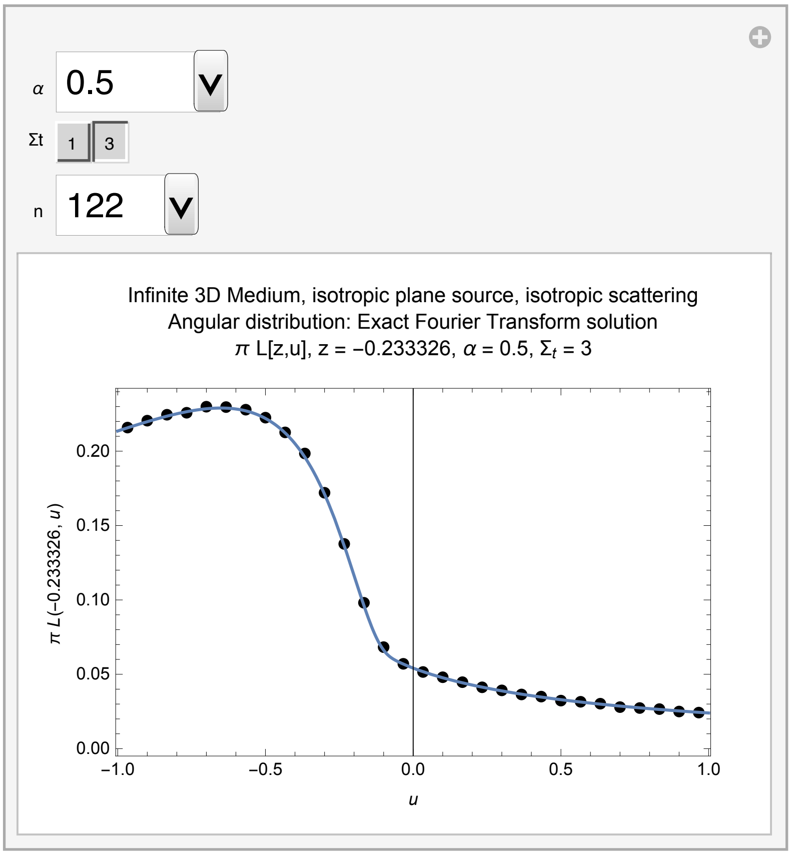
n 127

Infinite 3D Medium, isotropic plane source, isotropic scattering
Angular distribution: rigorous diffusion approximation
π L[z,u], z = 0.3, α = 0.9, Σ_t = 1
π L(0.3, u)
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
u

Exact solution (Fourier Transform)

```
In[42]:= Manipulate[
           If[Length[inf3Disoplaneisoscatter`simulations] > 0,
            Module[{data, numorders, pointsu, plotpointsu, du, r, dz, maxz, zsim},
             data = SelectFirst[
                 inf3Disoplaneisoscatter`simulations, #[[1]] == α && #[[2]] == Σt &][[3]];
             numorders = data[[2, 13]];
             du = data[[2, 9]];
             dz = data[[2, 7]];
             maxz = data[[2, 5]];

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

             zsim  =  dz * n - 0.5 dz - maxz;

             plotpointsu = inf3Disoplaneisoscatter`ppointsu[pointsu, du, Σt];
             Show[
              ListPlot[plotpointsu, PlotRange → All, PlotStyle → Black,
               Frame → True,
               FrameLabel -> {{ Pi L[zsim, u],}, {u,}}],
              Plot[Pi inf3Disoplaneisoscatter`LexactFourier[zsim, u, Σt, α], {u, -1, 1}],
              PlotLabel -> "Infinite 3D Medium, isotropic plane source,
                  isotropic scattering\nAngular distribution: Exact
                  Fourier Transform solution\nπ L[z,u], z = " <>
                ToString[zsim] <> ", 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}, inf3Disoplaneisoscatter`alphas},
           {{Σt, 3}, inf3Disoplaneisoscatter`muts}, {{n, 122}, Range[If[
              NumberQ[inf3Disoplaneisoscatter`numz], inf3Disoplaneisoscatter`numz, 1]]}]
```

Out[42]=



α 0.5

Σt 1 3

n 122

Infinite 3D Medium, isotropic plane source, isotropic scattering
Angular distribution: Exact Fourier Transform solution
π L[z,u], z = −0.233326, α = 0.5, Σ_t = 3