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

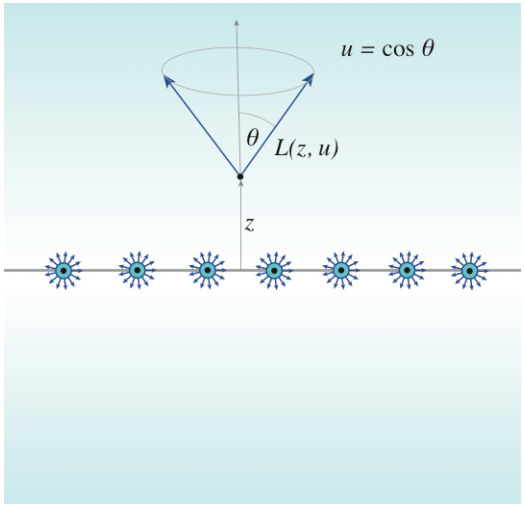
www.eugenedeon.com

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

In[3]:=
```
Casev0[c_?NumericQ] :=
 FindRoot[c v ArcTanh[1/v] - 1 == 0, {v, 1.00000000001, 10^10}, Method -> "Brent"][[1]][[
  2]]
```

In[4]:=
$$\text{CaseN}[c\_, v\_] := v\left(\text{Case}\lambda[v, c]^2 + \left(\frac{\pi\, c\, v}{2}\right)^2\right)$$

In[5]:=
$$\text{Case}\lambda[v\_, c\_] := 1 - c\, v\, \text{ArcTanh}[v]$$

In[6]:=
$$\text{Case}\psi 0[u\_, v0\_, c\_, z\_] := \frac{c}{2}\,\frac{v0}{v0 - \text{Sign}[z]\, u}$$

Rigorous diffusion approximation

In[7]:=
$$\text{inf3Disoplaneisoscatter`}\phi\text{rigourousDiffusion}[z\_, \Sigma t\_, \alpha\_] := \frac{1}{2}\,\frac{E^{-\text{Abs}[z]\,\Sigma t/\#}}{\text{CaseN0}[\alpha, \#]}\,\&[\text{Casev0}[\alpha]]$$

Fluence: exact solution

[Case and Zwiefel 1967]

In[8]:=
$$\text{inf3Disoplaneisoscatter`}\phi\text{exact}[z\_, \Sigma t\_, \alpha\_] := \text{inf3Disoplaneisoscatter`}\phi\text{rigourousDiffusion}[z, \Sigma t, \alpha] + \frac{1}{2}\,\text{NIntegrate}\left[\frac{e^{-\Sigma t\,\text{Abs}[z]/v}}{\text{CaseN}[\alpha, v]}, \{v, 0, 1\}\right]$$

Nth-scattered fluence

In[9]:=
$$\text{inf3Disoplaneisoscatter`}\phi\text{exact1}[x\_, \Sigma t\_, \alpha\_, n\_] := \frac{(\alpha\,\Sigma t)^n}{\pi}\,\text{NIntegrate}\left[\frac{\text{ArcTan}\left[\frac{z}{\Sigma t}\right]^{n+1}\text{Cos}[x\, z]}{z^{n+1}}, \{z, 0, \text{Infinity}\}, \text{Method} \to \text{"ExtrapolatingOscillatory"}\right]$$

In[10]:=
$$\text{inf3Disoplaneisoscatter`}\phi\text{exact2}[x\_, \Sigma t\_, \alpha\_, n\_] := \frac{\alpha^n}{\pi}\,\text{Chop}\left[\text{NIntegrate}\left[i\, 2^{-2-n}\, e^{-\text{Abs}[x]\, z\, \Sigma t}\left(\left(\frac{-i\,\pi + \text{Log}\left[\frac{z+1}{z-1}\right]}{z}\right)^{1+n} - \left(\frac{i\,\pi + \text{Log}\left[\frac{z+1}{z-1}\right]}{z}\right)^{1+n}\right), \{z, 1, \text{Infinity}\}\right]\right]$$

In[11]:=
$$\text{inf3Disoplaneisoscatter`}\phi\text{Gaussian}[x\_, \Sigma t\_, \alpha\_, n\_] := \frac{e^{-\frac{3\,x^2\,\Sigma t^2}{4\,(1+n)}}\sqrt{\frac{3}{\pi}}\,\alpha^n}{2\sqrt{1+n}}$$

Classical Diffusion Approximation

In[12]:=
```
inf3Disoplaneisoscatter`ϕDiffusion[x_, Σt_, α_] := 3 ℯ^(-Abs[x] Sqrt[3-3 α] Σt) / (2 Sqrt[3 - 3 α])
```

Grosjean Modified Diffusion Approximation

In[13]:=
```
inf3Disoplaneisoscatter`ϕGrosjean[x_, Σt_, α_] :=
  -1/2 ExpIntegralEi[-Abs[x] Σt] + ℯ^(-Abs[x] Sqrt[3+3/(-2+α)] Σt) Sqrt[3 + 3/(-2+α)] α / (2 (1 - α))
```

Radiance (Angular Flux)

In[14]:=
```
inf3Disoplaneisoscatter`LrigourousDiffusion[z_, u_, Σt_, α_] :=
  1/(4 Pi) Caseψ0[u, #, α, z] E^(-Abs[z] Σt/#) / CaseN0[α, #] &[Casev0[α]]
```

In[15]:=
```
inf3Disoplaneisoscatter`Lexact[z_, u_, Σt_, α_] :=
 inf3Disoplaneisoscatter`LrigourousDiffusion[z, u, Σt, α] +
  1/(4 Pi) ( Caseλ[u, α] ℯ^(-Abs[z] Σt/u) / CaseN[α, u] HeavisideTheta[1 - u] HeavisideTheta[u]
    + NIntegrate[ ℯ^(-Abs[z] Σt/v) / CaseN[α, v] α/2 v/(v - u)
      , {v, 0, u, 1}, Method → "PrincipalValue", PrecisionGoal → 5] )
```

load MC data

In[16]:=
```
inf3Disoplaneisoscatter`ppoints[zs_, dz_, maxz_, Σt_] :=
 Table[{dz (i) - 0.5 dz - maxz, zs[[i]]/Σt}, {i, 1, Length[zs]}][[2 ;; -2]]
```

In[17]:=
```
inf3Disoplaneisoscatter`ppointsu[xs_, du_, Σt_] :=
 Table[{-1.0 + du (i) - 0.5 du, xs[[i]]/(2 Σt)}, {i, 1, Length[xs]}][[1 ;; -1]]
```

In[18]:=
```
inf3Disoplaneisoscatter`fs =
  FileNames["code/3D_medium/infinite3Dmedium/Isotropicplanesource/data/
     inf3D_isotropicplane_isotropicscatter*"];
```

In[19]:=
```
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[21]=
```
{0.01, 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999}
```

In[22]:=
```
inf3Disoplaneisoscatter`muts =
 Union[#[[2]] & /@ inf3Disoplaneisoscatter`simulations]
```

Out[22]=
```
{1, 3}
```

In[23]:=
```
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

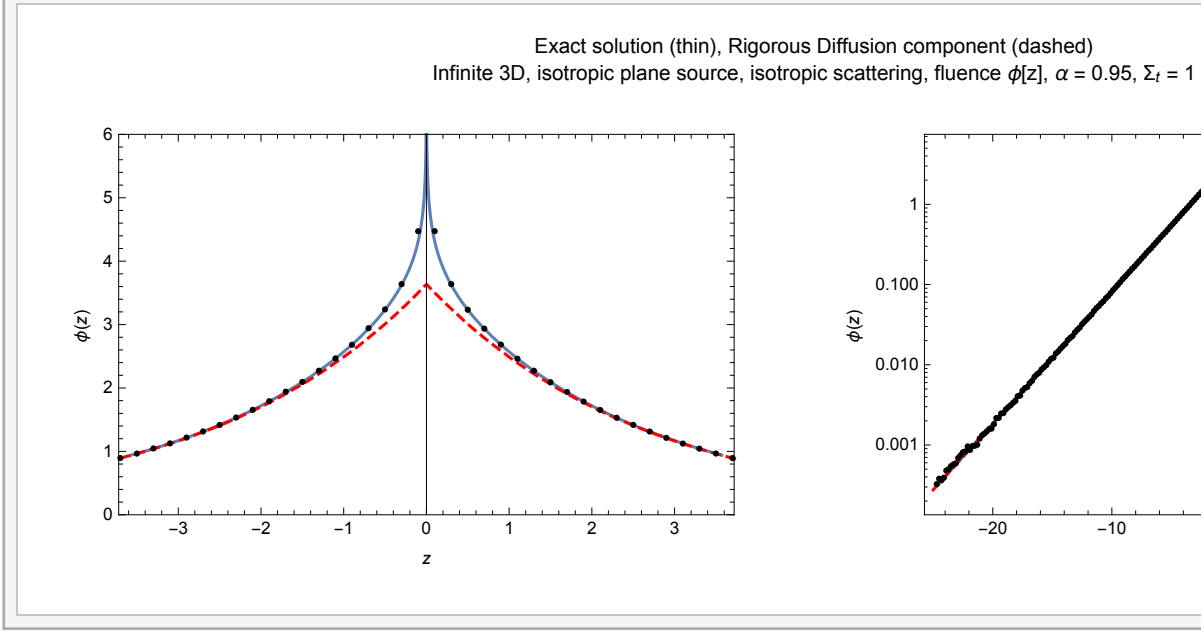
Fluence - Exact solution comparison to MC

In[26]:=
```
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];
   skip = Floor[numpoints 6/7 1/2];

   plotϕ = Quiet[Show[
      (*ListPlot[exact1points[[skip;;-skip]],PlotRange→{0,6},Joined→True],*)
      Plot[inf3Disoplaneisoscatter`ϕexact[z, Σt, α],
       {z, -maxz/7, maxz/7}, PlotRange → {0, 6}],
      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}]
```

Out[26]=

α 0.95

Σt 1 3



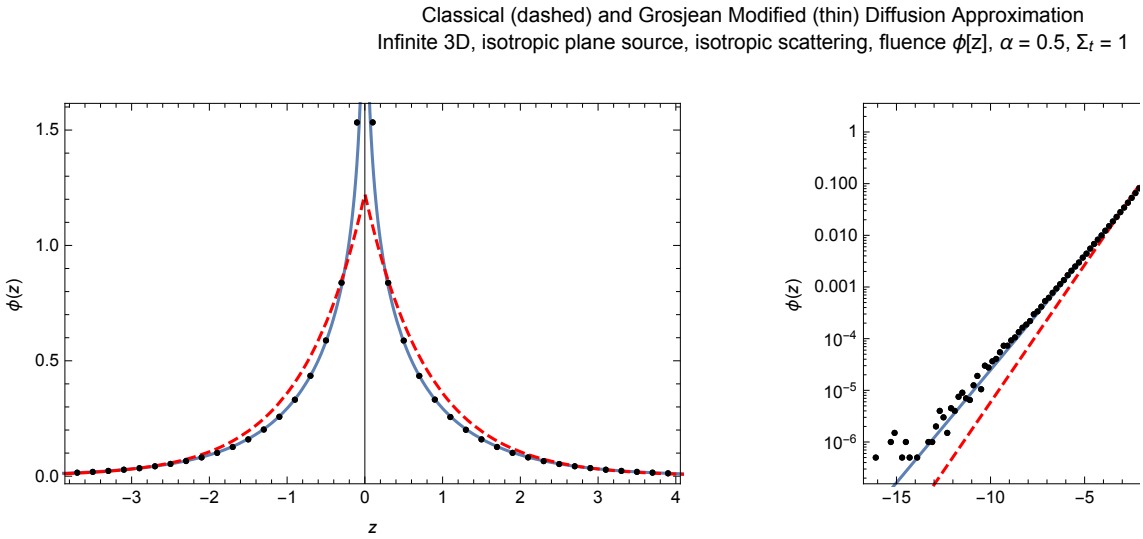
Fluence - Diffusion approximations (Classical and Grosjean) comparison to MC

In[27]:=
```
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];
   skip = Floor[numpoints 6/7 1/2];

   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}]
```

Out[27]=

α 0.5

Σt 1 3

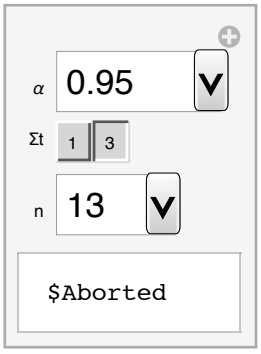


N-th order fluence / scalar flux

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

In[28]:=
```
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]]}

]
```

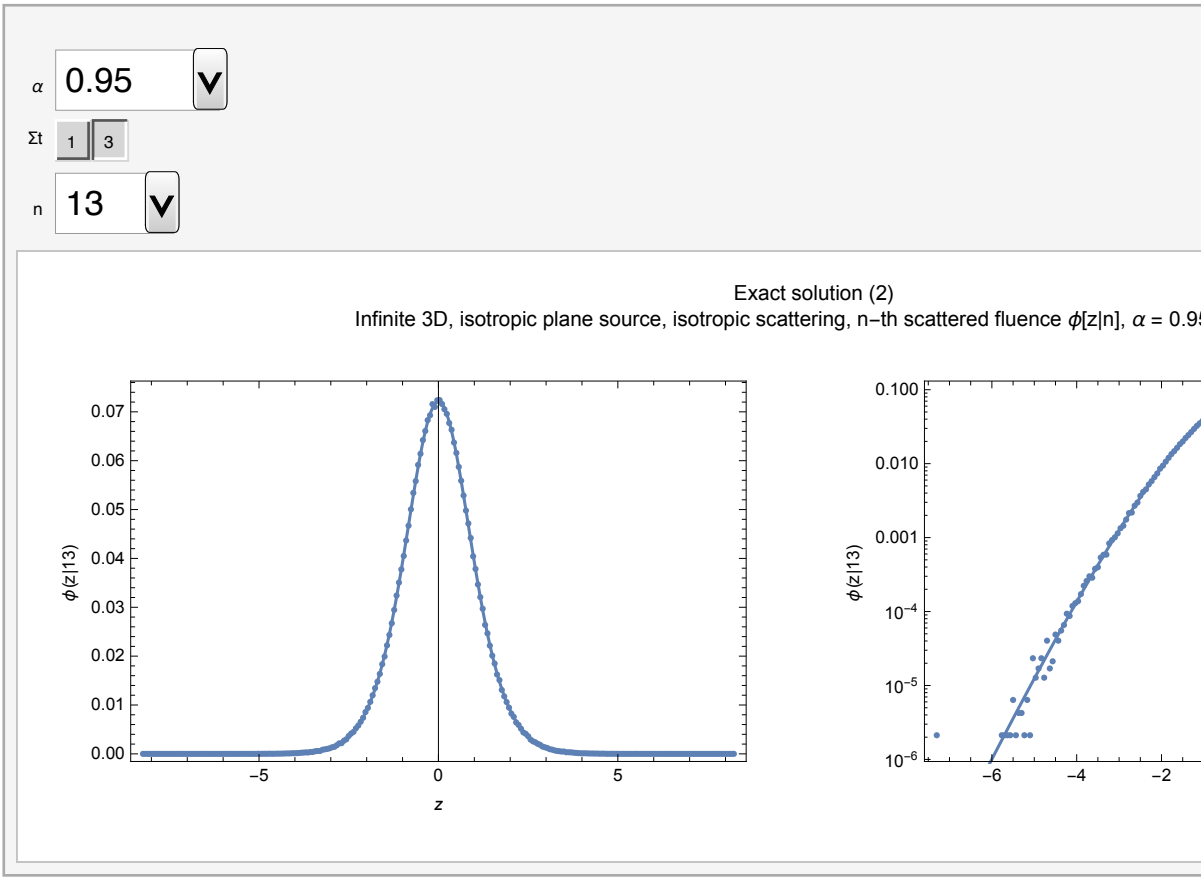
Out[28]=



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

In[29]:=
```
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]]}

]
```

Out[29]=



N-th collided Fluence - Approximations

In[30]:=

```
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]]}]
```

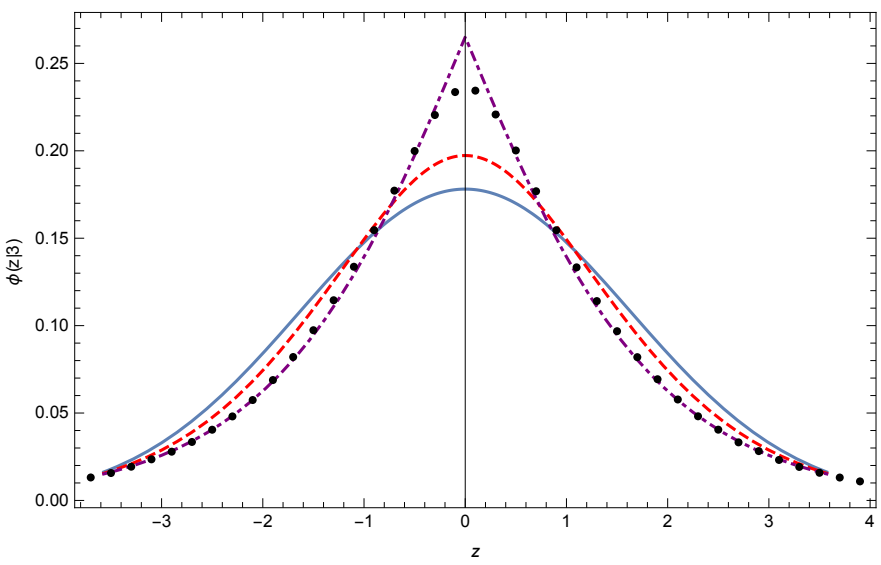
Out[30]=

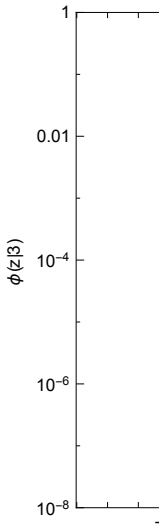
α 0.9

Σt 1 3

n 3

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





Angular Distributions

In[31]:=
```
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[31]=

α 0.9

Σt 1 3

n 127