Geometrical Optics applied to 1DSR of Inhomogeneous Soil Deposits

This notebook details how the results shown in the article “Geometrical Optics applied to 1D Site Response of Inhomogeneous Soild Deposits” have been obtained. The procedure that yields the results shown in the plots included in the paper can be found herein.

This notebook is referenced with respect to the text of the article, using it during the reading is advised.

Section 5: Verification

5.1. Increasing contrast

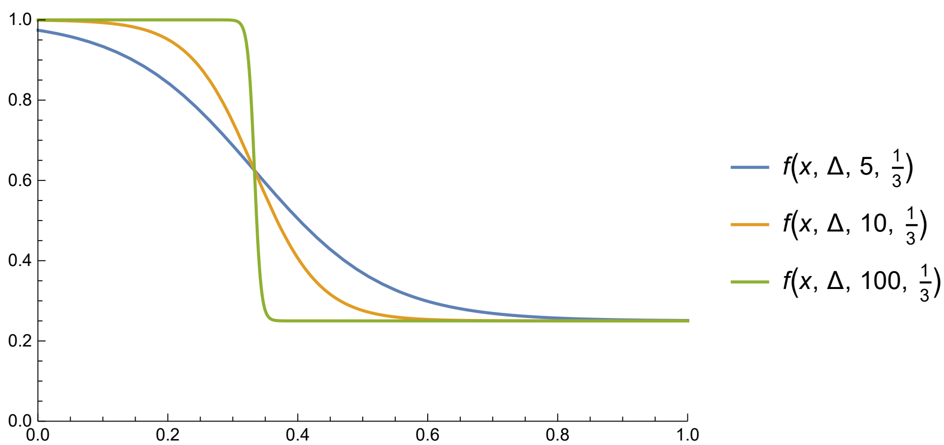
Define the velocity profiles

The following corresponds to eq.(18) in the text.

```
f[η_, Δ_, p_, x0_] := 1 - Δ/2 * (1 + Tanh[p (η - x0)])
```

Next, we show an example of the parametric change of the profile:

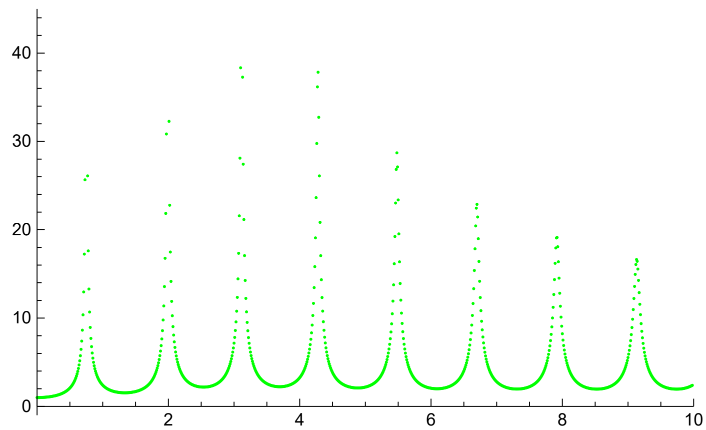
```
DE = 0.01; Δ = 0.75;
Plot[{f[x, Δ, 5, 1/3], f[x, Δ, 10, 1/3], f[x, Δ, 100, 1/3]},
 {x, 0, 1}, PlotLegends → "Expressions", PlotRange → {{0, 1}, {0, 1}}]
```



Numerical evaluation using exact profiles

The equation of motion (1), once it has been converted using relative-to-base displacement, has to be solved numerically when using the tanh profile. This is achieved using the *Mathematica* built-in function **NIntegrate** as follows:

```
DE = 0.01; Δ = 0.75;
mylist = Range[0, 10, 0.01];
L = Length[mylist];
For[i = 1, i < L - 1, i++, {solQ = NDSolve[{D[f[η, Δ, 5, 1/3]^2 * U'[η], η] + mylist[[i]]^2/(1 + ⅈ * DE) *
      U[η] == 1, U[0] == 0, U'[1] == 0}, U, {η, 0, 1}], Result[i] = Evaluate[U[1] /. solQ]}];
res1 = Table[{mylist[[j]], Abs[1 - mylist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figA = ListPlot[res1, PlotRange → {{0, 10}, {-1, 45}}, PlotStyle → {Green}]
```



Similar results are computed for other values of the jump, and then combined into Figure 10-

```
For[i = 1, i < L - 1, i++,
  {solQ = Quiet[NDSolve[{D[f[η, Δ, 10, 1/3]^2 * U'[η], η] + mylist[[i]]^2/(1 + ⅈ * DE) * U[η] == 1,
      U[0] == 0, U'[1] == 0}, U, {η, 0, 1}]], Result[i] = Evaluate[U[1] /. solQ]}];
res2 = Table[{mylist[[j]], Abs[1 - mylist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figB = ListPlot[res2, PlotRange → {{0, 10}, {-1, 45}}, PlotStyle → {Red}];
```

```
For[i = 1, i < L - 1, i++,
  {solQ = Quiet[NDSolve[{D[f[η, Δ, 100, 1/3]^2 * U'[η], η] + mylist[[i]]^2/(1 + ⅈ * DE) * U[η] == 1,
      U[0] == 0, U'[1] == 0}, U, {η, 0, 1}]], Result[i] = Evaluate[U[1] /. solQ]}];
res3 = Table[{mylist[[j]], Abs[1 - mylist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figC = ListPlot[res3, PlotRange → {{0, 10}, {-1, 45}}, PlotStyle → {Blue}];
$Aborted
```

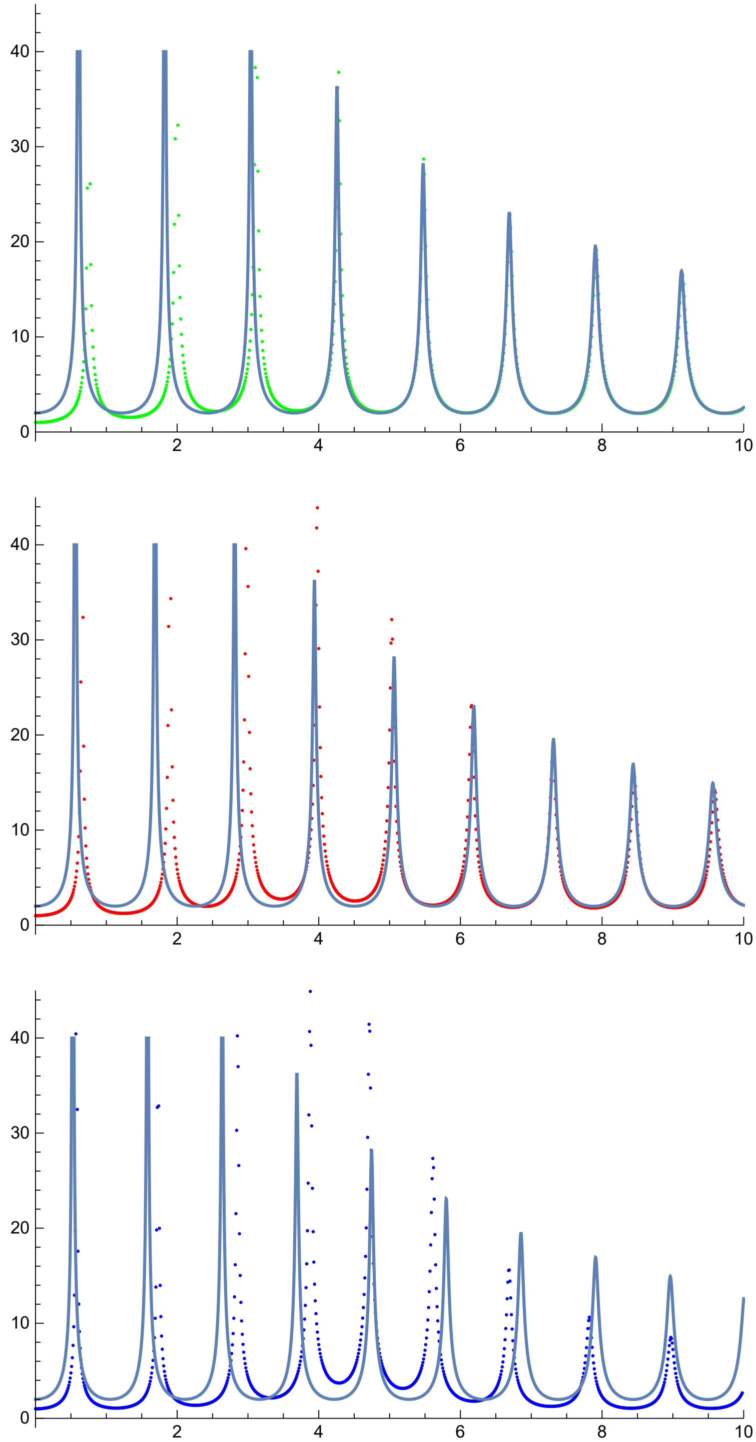
Add geometrical optics approximation to the comparison

Add geometrical optics approximation, eq.(3a):

```
figA2 = Plot[{Abs[(1/Sqrt[0.25])/Cos[NIntegrate[1/f[x,Δ,5,1/3], {x, 0, 1}] * R/Sqrt[(1+ⅈ*DE)]]]},
  {R, 0, 10}, PlotRange → {{0, 25}, {0, 40}}, PlotLegends → "Expressions"];
figB2 = Plot[{Abs[(1/Sqrt[0.25])/Cos[NIntegrate[1/f[x,Δ,10,1/3], {x, 0, 1}] * R/Sqrt[(1+ⅈ*DE)]]]},
  {R, 0, 10}, PlotRange → {{0, 25}, {0, 40}}, PlotLegends → "Expressions"];
figC2 = Plot[{Abs[(1/Sqrt[0.25])/Cos[NIntegrate[1/f[x,Δ,100,1/3], {x, 0, 1}] * R/Sqrt[(1+ⅈ*DE)]]]},
  {R, 0, 10}, PlotRange → {{0, 25}, {0, 40}}, PlotLegends → "Expressions"];
```

Display the three cases together to generate Figure 2:

```
GraphicsGrid[{{Show[figA, figA2]}, {Show[figB, figB2]}, {Show[figC, figC2]}}]
```



5.2. Profile reversal

We repeat here the same procedure assuming a sinusoidal profile.

Define the velocity profiles

The following corresponds to eq.(19) in the text.

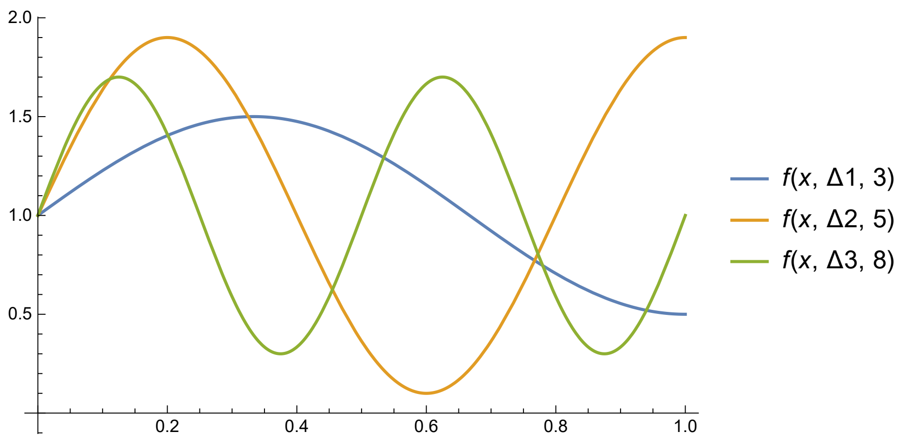
```
f[η_, Δ_, K_] := 1 + Δ * Sin[K/2 π * η]
```

Define some parameters:

```
DE = 0.01;  Δ1 = 0.5; Δ2 = 0.9; Δ3 = 0.7; K = 3;
```

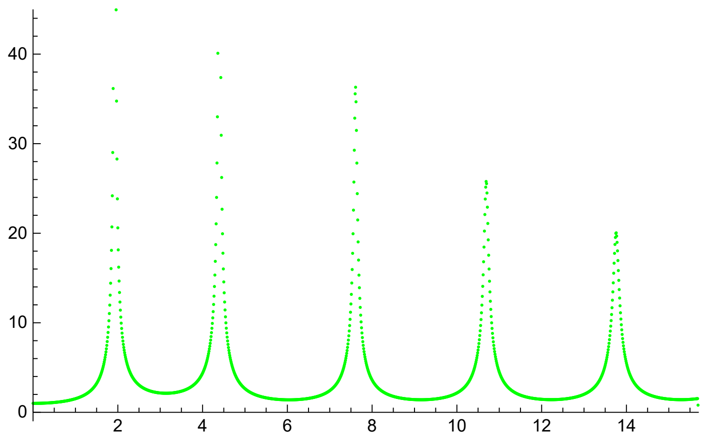
Show examples of the parametric change of the profile:

```
Plot[{f[x, Δ1, 3], f[x, Δ2, 5], f[x, Δ3, 8]}, {x, 0, 1}, PlotLegends → "Expressions"]
```



Numerical evaluation using exact profiles

```
Lint = 5 * π;
mylist = Range[0, Lint, 0.01];
L = Length[mylist];
For[i = 1, i < L - 1, i++,
  {solQ = Quiet[NDSolve[{D[f[η, Δ1, 3]^2 * U'[η], η] + mylist[[i]]^2/(1 + ⅈ * DE) * U[η] == 1,
       U[0] == 0, U'[1] == 0}, U, {η, 0, 1}]], Result[i] = Evaluate[U[1] /. solQ]}];
res1 = Table[{mylist[[j]], Abs[1 - mylist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figA = ListPlot[res1, PlotRange → {{0, 5 * π}, {-1, 45}}, PlotStyle → {Green}]
```




```
For[i = 1, i < L - 1, i++,
  {solQ = Quiet[NDSolve[{D[f[η, Δ2, 5]^2 * U'[η], η] + mylist[[i]]^2/(1 + ⅈ * DE) * U[η] == 1,
       U[0] == 0, U'[1] == 0}, U, {η, 0, 1}]], Result[i] = Evaluate[U[1] /. solQ]}];
res1 = Table[{mylist[[j]], Abs[1 - mylist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figB = ListPlot[res1, PlotRange → {{0, 4 * π}, {-1, 45}}, PlotStyle → {Red}];
For[i = 1, i < L - 1, i++,
  {solQ = Quiet[NDSolve[{D[f[η, Δ3, 8]^2 * U'[η], η] + mylist[[i]]^2/(1 + ⅈ * DE) * U[η] == 1,
       U[0] == 0, U'[1] == 0}, U, {η, 0, 1}]], Result[i] = Evaluate[U[1] /. solQ]}];
res1 = Table[{mylist[[j]], Abs[1 - mylist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figC = ListPlot[res1, PlotRange → {{0, 5 * π}, {-1, 45}}, PlotStyle → {Blue}];
```

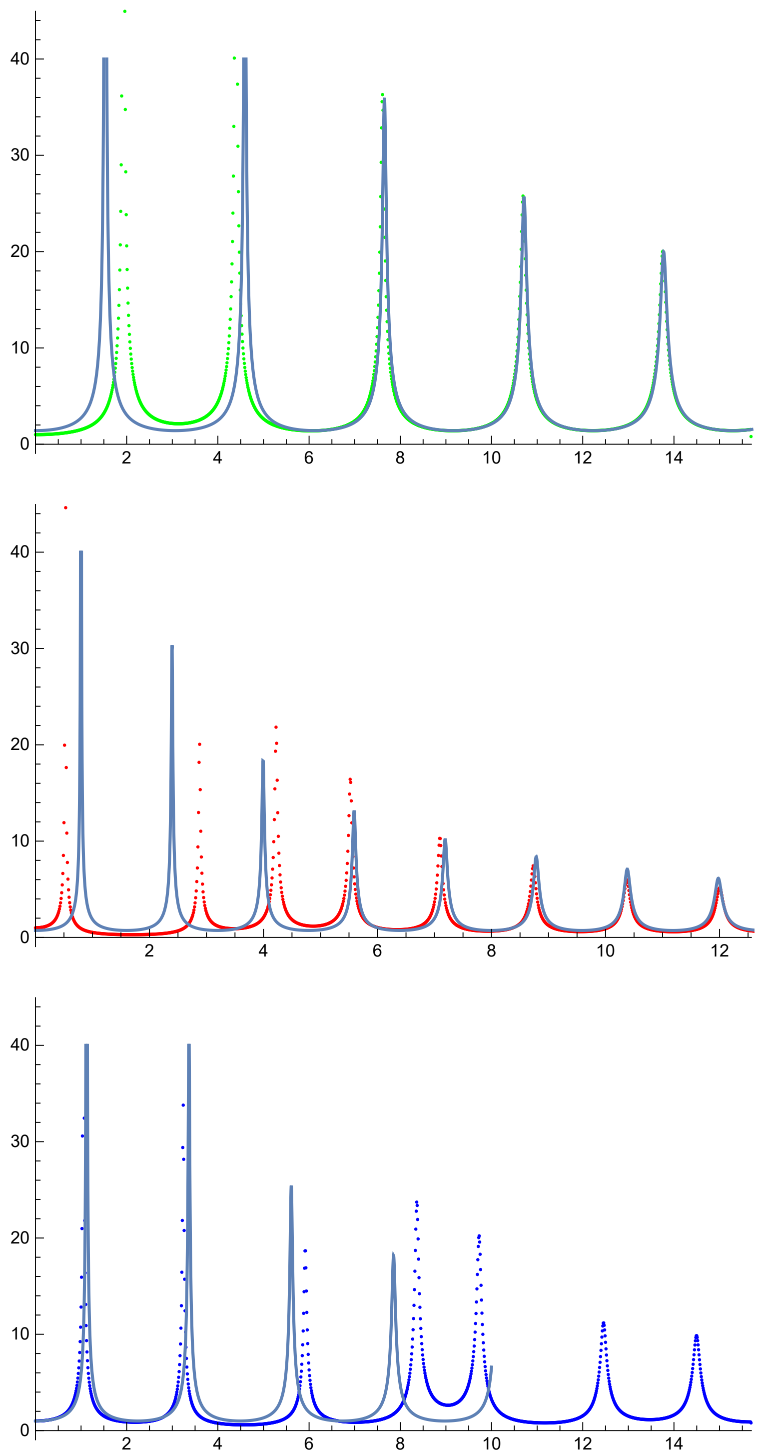
Add geometrical optics approximation to the comparison

Add geometrical optics approximation, eq.(3a):

```
figA2 = Plot[{Abs[(1/Sqrt[f[1, Δ1, 3]])/Cos[NIntegrate[1/f[x,Δ1,3], {x, 0, 1}] * R/Sqrt[(1+ⅈ*DE)]]]},
  {R, 0, Lint}, PlotRange → {{0, 5 * π}, {0, 40}}, PlotLegends → "Expressions"];
figB2 = Plot[{Abs[(1/Sqrt[f[1, Δ2, 5]])/Cos[NIntegrate[1/f[x,Δ2,5], {x, 0, 1}] * R/Sqrt[(1+ⅈ*DE)]]]},
  {R, 0, Lint}, PlotRange → {{0, 5 * π}, {0, 40}}, PlotLegends → "Expressions"];
figC2 = Plot[{Abs[(1/Sqrt[f[1, Δ3, 8]])/Cos[NIntegrate[1/f[x,Δ3,8], {x, 0, 1}] * R/Sqrt[(1+ⅈ*DE)]]]},
  {R, 0, 10}, PlotRange → {{0, 25}, {0, 40}}, PlotLegends → "Expressions"];
```

Display the three cases together to generate Figure 3:

```
GraphicsGrid[{{Show[figA, figA2]}, {Show[figB, figB2]}, {Show[figC, figC2]}}]
```



5.3. Visualizing the role of continuity

In this section we carry out the computations necessary to generate Figure 4.

Define the velocity profiles

Some auxiliary functions:

```
f3b[x_] := 1.2 (1 - x)^2 - 0.4 * (1 - x) + 1 / 2
```

```
f2a[x_] := 1 - 0.8 * x; f2b[x_] := 0.7 - 0.2 * x;
```

```
f[x_, a_, b_] := a + b * x;
```

The three actual distributions for this example:

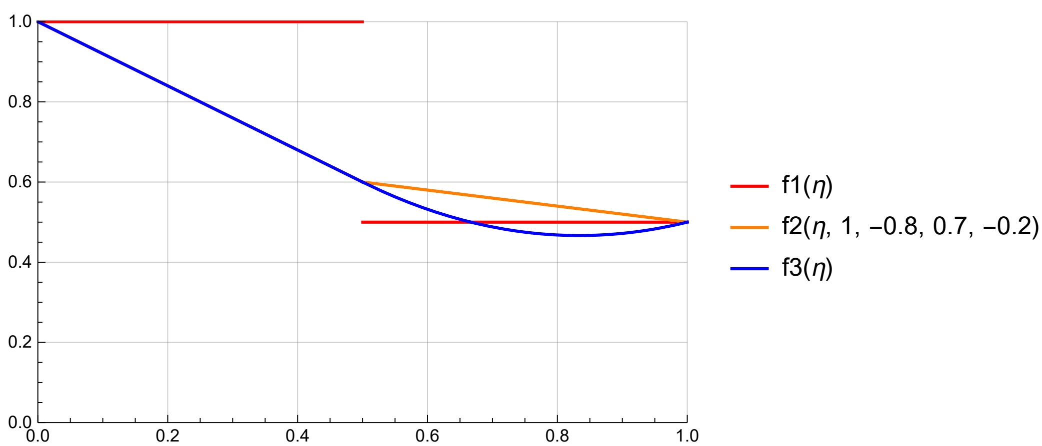
```
f1[x_] := Piecewise[{{1, x < 1 / 2}, {1 / 2, x >= 1 / 2}}]
```

```
f2[x_, a_, b_, c_, d_] := Piecewise[{{f[x, a, b], x < 1 / 2}, {f[x, c, d], x >= 1 / 2}}]
```

```
f3[x_] := Piecewise[{{f2a[x], x < 1 / 2}, {f3b[x], x >= 1 / 2}}]
```

Plot the three of them:

```
Plot[{f1[η], f2[η, 1, -0.8, 0.7, -0.2], f3[η]}, {η, 0, 1}, PlotRange → {{0, 1}, {0, 1}},
 GridLines → Automatic, PlotLegends → "Expressions", PlotStyle → {Red, Orange, Blue}]
```



Generate results of linear distribution

Displacement field in each layer

Displacement field in the first layer:

```
u1lin[η_, a_, b_, r_] = FullSimplify[
   u[η] /. First @ DSolve[{D[(a + b * η)^2 * D[u[η], η], η] + r^2 * u[η] == 1, u[0] == 0}, u, η]];
```

Derivative of the displacement field in the lower layer (for the stresses):

```
u1linP[η_, a_, b_, r_] = D[u1lin[η, a, b, r], η];
```

Displacement field in the second layer (use auxiliary to have c1 and c2 distinct):

```
u2linAux[η_, c_, d_, r_] = FullSimplify[
   u[η] /. First @ DSolve[{D[(c + d * η)^2 * D[u[η], η], η] + r^2 * u[η] == 1, u'[1] == 0}, u, η]];
```

```
u2lin[η_, c_, d_, r_] = u2linAux[η, c, d, r] /. C[1] → C[2];
```

Derivative of the displacement field in the upper layer (for the stresses):

```
u2linP[η_, c_, d_, r_] = D[u2lin[η, c, d, r], η];
```

Enforce displacement and stress continuity at the interface between layers

Find the value of the constants to satisfy these continuity constraints:

```
cs[r_, a_, b_, c_, d_] = Solve[u1lin[1/2, a, b, r] == u2lin[1/2, c, d, r] &&
     (a + b/2)^2 * u1linP[1/2, a, b, r] == (c + d/2)^2 * u2linP[1/2, c, d, r], {C[1], C[2]}];
```

Save solution for c2 (as only the upper layer is necessary for the transfer function)

```
c2sol[r_, a_, b_, c_, d_] := C[2] /. cs[r, a, b, c, d][[1]]
```

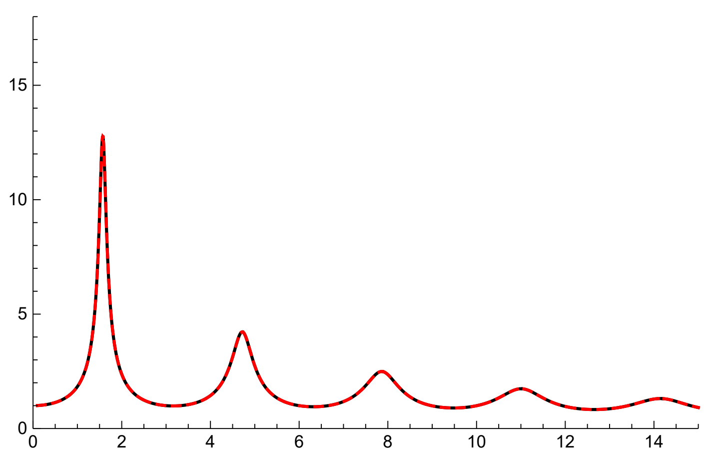
Get the transfer function

This solves the piecewise linear distribution:

```
Alin[r_, a_, b_, c_, d_] := 1 - r^2 * (u2lin[1, c, d, r] /. {C[2] → c2sol[r, a, b, c, d]})
```

<u>Verification 1: homogeneous case</u>

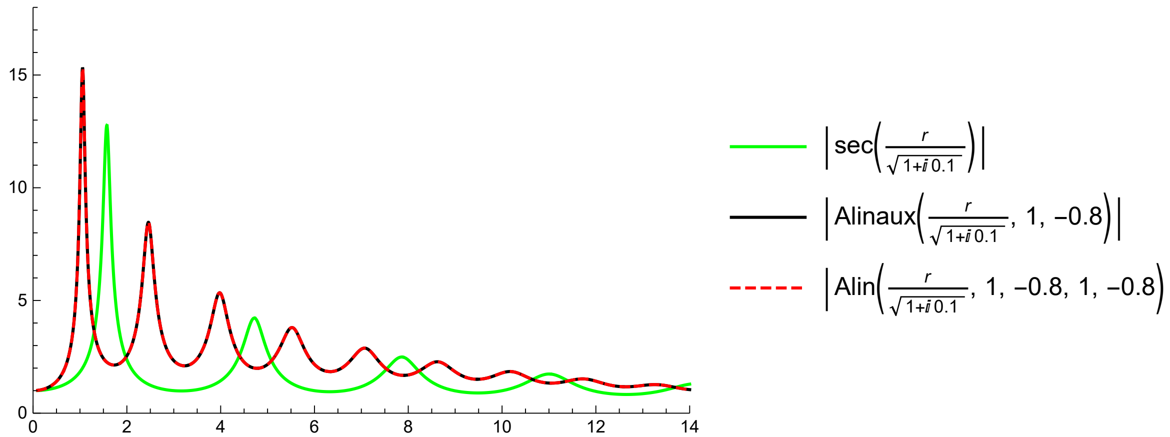
```
Plot[{Abs[Sec[r/Sqrt[1 + i * 0.1]]], Abs[Alin[r/Sqrt[1 + i * 0.1], 1, -0.00008, 1, -0.00002]]},
 {r, 0.1, 15}, PlotRange → {{0, 15}, {0, 18}}, PlotStyle → {Black, {Red, Dashed}}]
```



<u>Verification 2: continuous linear across layers</u>

```
Alinaux[r_, a_, b_] =
  DSolveValue[{D[f[x, a, b]^2 * v'[x], x] + r^2 * v[x] == 0, v[0] == 1, v'[1] == 0}, v[1], x];
```

```
Plot[{Abs[Sec[r/Sqrt[1 + i * 0.1]]], Abs[Alinaux[r/Sqrt[1 + i * 0.1], 1, -0.8]],
  Abs[Alin[r/Sqrt[1 + i * 0.1], 1, -0.8, 1, -0.8]]}, {r, 0.1, 14},
 PlotRange → {{0, 14}, {0, 18}}, PlotStyle → {Green, Black, {Red, Dashed}},
 PlotLegends → "Expressions"]
```



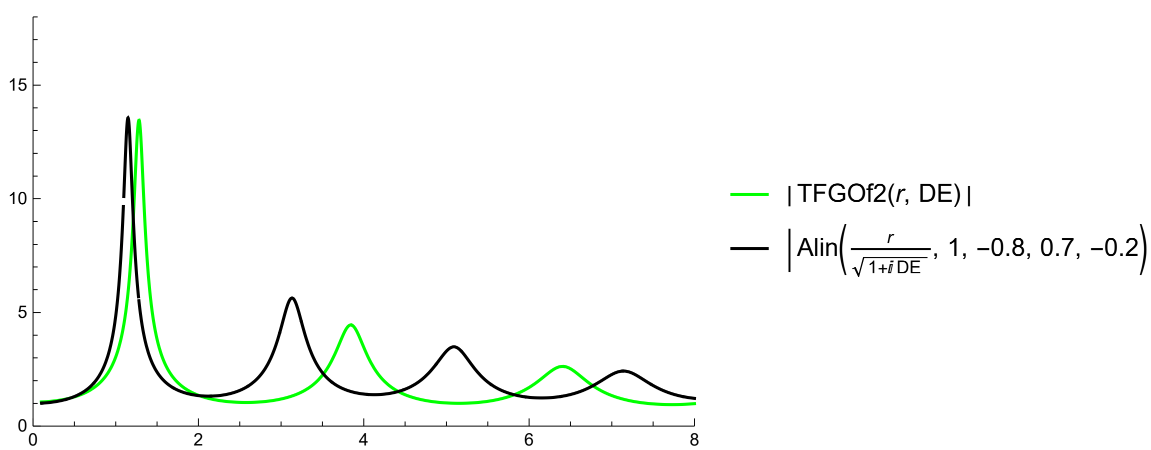
Compare transfer functions

Geometrical Optics transfer function:

```
TFGOf2[r_, δ_] := (1/Sqrt[f2[1, 1, -0.8, 0.7, 0.2]]) /
  Cos[NIntegrate[1/f2[x,1,-0.8,0.7,0.2], {x, 0, 1}] * r/Sqrt[(1+i*δ)]]
```

The following generates the middle tile of Figure 4:

```
DE = 0.1;
Plot[{Abs[TFGOf2[r, DE]], Abs[Alin[r/Sqrt[1 + i * DE], 1, -0.8, 0.7, -0.2]]},
 {r, 0.1, 8}, PlotRange → {{0, 8}, {0, 18}},
 PlotStyle → {Green, Black, {Red, Dashed}}, PlotLegends → "Expressions"]
```



Generate results for 2-layer system (over bedrock)

The following generates the result for the system split in two distinct layers, following as similar procedure as before.

Analytical solution

Solve the lower layer:

```
u1hom[η_, r_] =
  (u[η] /. First@DSolve[{D[D[u[η], η], η] + r^2 * u[η] == 1, u[0] == 0}, u, η]) /. C[2] → c1
```

$$\frac{1 - \mathrm{Cos}[r\,\eta] + c1\, r^2\, \mathrm{Sin}[r\,\eta]}{r^2}$$

Solve the upper layer:

```
u2hom[η_, r_, α_] =
  (u[η] /. First@DSolve[{α^2 * D[D[u[η], η], η] + r^2 * u[η] == 1, u'[1] == 0}, u, η]) /. C[1] → c2
```

$$\frac{1 + c2\, r^2\, \mathrm{Cos}\left[\frac{r\,\eta}{\alpha}\right] + c2\, r^2\, \mathrm{Sin}\left[\frac{r\,\eta}{\alpha}\right] \mathrm{Tan}\left[\frac{r}{\alpha}\right]}{r^2}$$

Enforce continuity conditions at the interface (η=1/2)

```
Simplify[Solve[u2hom[1/2, r, α] == u1hom[1/2, r] &&
   α^2 ((D[u2hom[x, r, α], x]) /. x → 1/2) == ((D[u1hom[x, r], x]) /. x → 1/2), {c1, c2}]]
```

$$\left\{\left\{c1 \to -\frac{\mathrm{Cos}\left[\frac{r}{2\alpha}\right] \mathrm{Sin}\left[\frac{r}{2}\right] + \alpha\, \mathrm{Cos}\left[\frac{r}{2}\right] \mathrm{Sin}\left[\frac{r}{2\alpha}\right]}{r^2 \left(\mathrm{Cos}\left[\frac{r}{2}\right] \mathrm{Cos}\left[\frac{r}{2\alpha}\right] - \alpha\, \mathrm{Sin}\left[\frac{r}{2}\right] \mathrm{Sin}\left[\frac{r}{2\alpha}\right]\right)},\right.\right.$$
$$\left.\left. c2 \to -\frac{\mathrm{Cos}\left[\frac{r}{\alpha}\right]}{r^2 \left(\mathrm{Cos}\left[\frac{r}{2}\right] \mathrm{Cos}\left[\frac{r}{2\alpha}\right] - \alpha\, \mathrm{Sin}\left[\frac{r}{2}\right] \mathrm{Sin}\left[\frac{r}{2\alpha}\right]\right)}\right\}\right\}$$

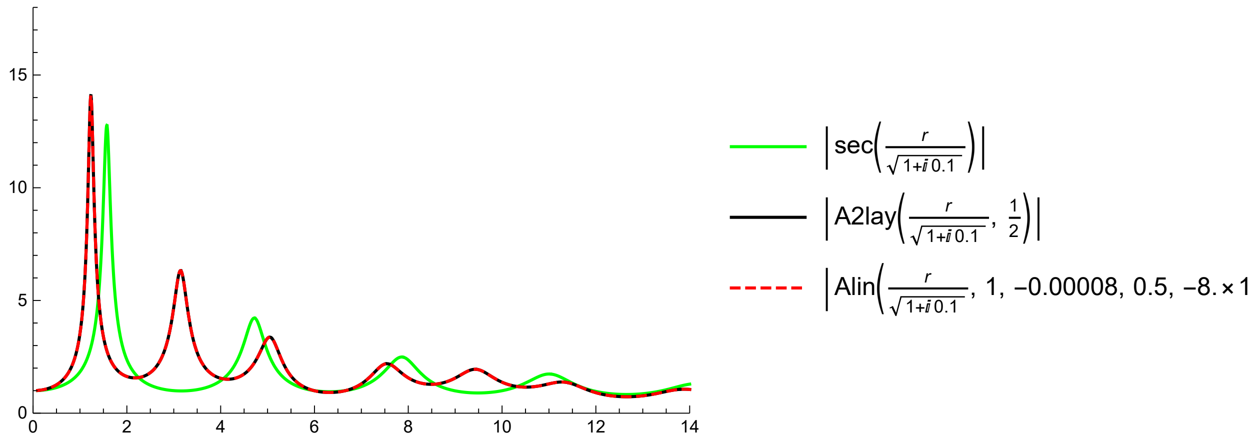
Get the transfer function:

```
A2lay[r_, α_] = 1 - r^2 * (u2hom[1, r, α] /. c2 -> - Cos[r/α] / (r^2 (Cos[r/2] Cos[r/(2 α)] - α Sin[r/2] Sin[r/(2 α)])));
Simplify[A2lay[r, α]]
```

$$\frac{1}{\mathrm{Cos}\left[\frac{r}{2}\right] \mathrm{Cos}\left[\frac{r}{2\alpha}\right] - \alpha\, \mathrm{Sin}\left[\frac{r}{2}\right] \mathrm{Sin}\left[\frac{r}{2\alpha}\right]}$$

Verification 1: compare to two-layer linear

```
Plot[{Abs[Sec[r/Sqrt[1 + ⅈ * 0.1]]], Abs[A2lay[r/Sqrt[1 + ⅈ * 0.1], 1/2]],
  Abs[Alin[r/Sqrt[1 + ⅈ * 0.1], 1, -0.00008, 0.5, -0.000008]]}, {r, 0.1, 14},
 PlotRange → {{0, 14}, {0, 18}}, PlotStyle → {Green, Black, {Red, Dashed}},
 PlotLegends → "Expressions"]
```



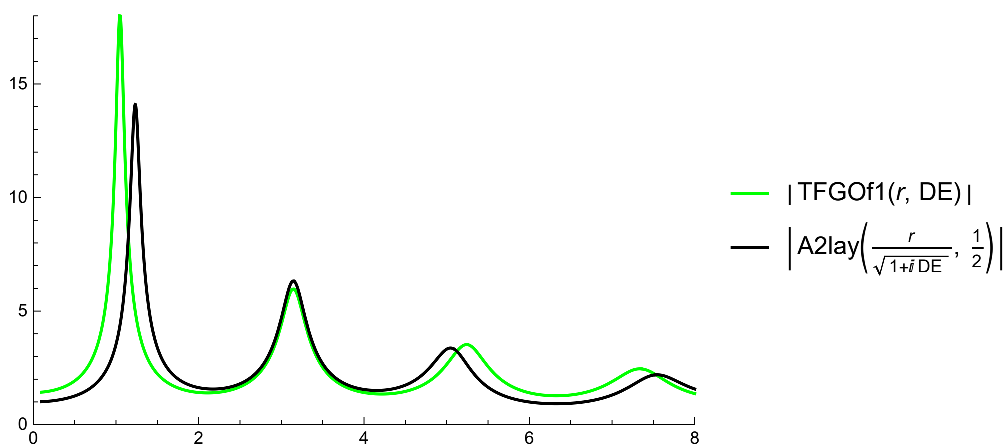
Compare transfer functions

Geometrical Optics transfer function:

```
TFGOf1[r_, δ_] := (1/Sqrt[f1[1]]) / (Cos[NIntegrate[1/f1[x], {x, 0, 1}] * r/Sqrt[(1+ⅈ*δ)]])
```

The following corresponds to the upper tile in Figure 4:

```
Plot[{Abs[TFGOf1[r, DE]], Abs[A2lay[r/Sqrt[1 + ⅈ * DE], 1/2]]},
 {r, 0.1, 8}, PlotRange → {{0, 8}, {0, 18}},
 PlotStyle → {Green, Black, {Red, Dashed}}, PlotLegends → "Expressions"]
```



Generate results for linear-quadratic connected smoothly

This one can be solved numerically automatically, since the gradient is well-defined at all points (no gradient discontinuities).

Parameters for the numerical simulation

First, set up the limit for the plots as we did for the first two plots:

```
rmax = 4 * π;
```

Next, generate a list of points where the solution will be obtained (“rlist”), at a distance 0.01 between points. “L” represents the number of elements in that list.

```
rlist = Range[0, rmax, 0.01]; L = Length[rlist];
```

The damping factor (which is also in the plots in the text)

```
DE = 0.1;
```

Solving the ODE and saving the results

```
For[i = 1, i < L - 1, i++,
 {solQ = Quiet[NDSolve[{D[f3[η]^2 * V'[η], η] + rlist[[i]]^2/(1 + ⅈ * DE) * V[η] == 1, V[0] == 0, V'[1] == 0},
     V, {η, 0, 1}]], Result[i] = Evaluate[V[1] /. solQ]}]
res1 = Table[{rlist[[j]], Abs[1 - rlist[[j]]^2/(1 + ⅈ * DE) * Result[j][[1]]]}, {j, L - 1}];
figA = ListPlot[res1, PlotRange → {{0, 8}, {0, 18}}, PlotStyle → {Black}];
```

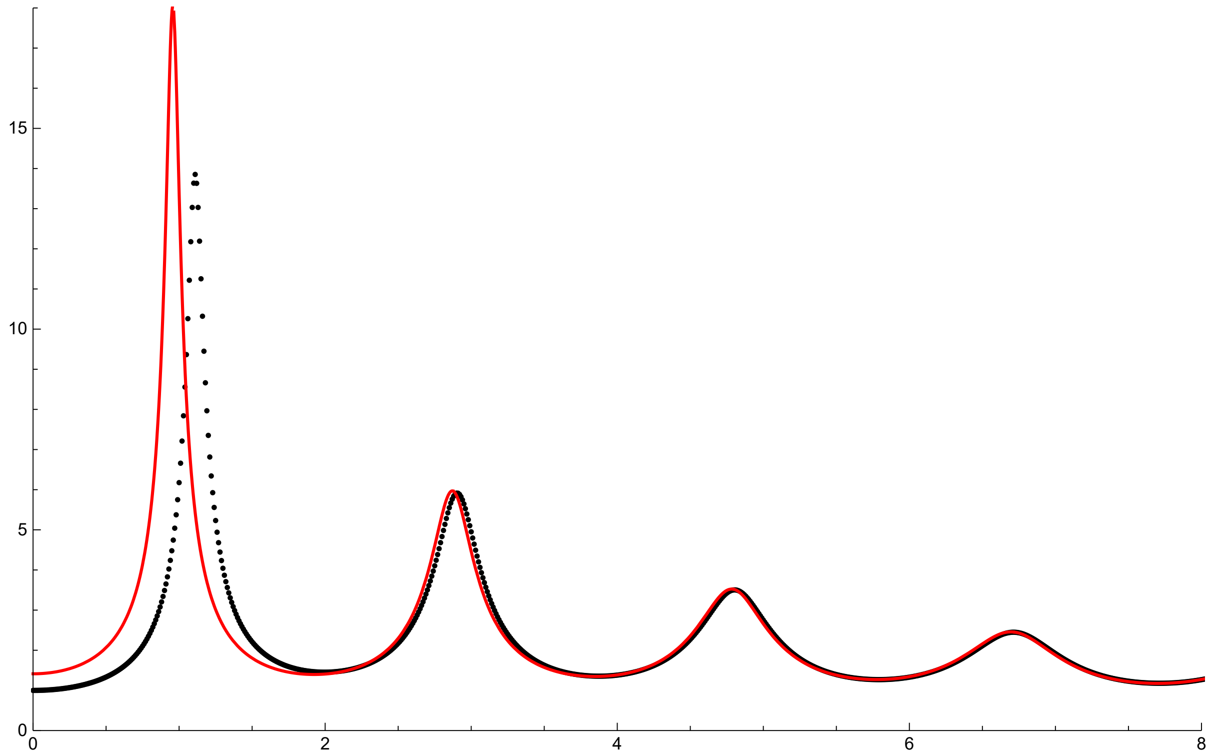
Compare Transfer Functions

Geometrical Optics transfer function:

```
TFGOf3[r_, δ_] := (1/Sqrt[f3[1]]) / Cos[NIntegrate[1/f3[x], {x, 0, 1}] * r/Sqrt[(1+ⅈ*δ)]]
```

Finally, the following corresponds to the lower tile of Figure 4.

```
figB = Plot[{Abs[TFGOf3[r, DE]]}, {r, 0, rmax},
  PlotStyle → {Red}, PlotRange → {{0, rmax}, {0, 45}}]; Show[figA, figB]
```



Section 7: Re-evaluation of current guidelines

This section includes an example of how taking only the first 30 meters to calculate the harmonic mean of the shear wave velocity can lead to errors.

We shall use the “generalized parabola” shear-wave velocity distribution studied by Rovithis et al. (2011).

Exact solution (as given in Rovithis et al. 2011)

These are the parameters that they chose to express the transfer function of the deposit having shear-wave profile given by “generalized parabola”:

$$\lambda[\alpha\_, n\_, r\_, \delta\_] = \frac{r / \mathrm{Sqrt}[(1 + i * \delta)]}{(1 - n) * \left(1 - \alpha^{\frac{1}{2*n}}\right)};$$

$$\psi[n\_] = \frac{(1 - 2 * n)}{2};$$

$$\nu[n\_] = \frac{(2 * n - 1)}{2\,(1 - n)};$$

$$\text{TFR}[\alpha\_, n\_, r\_, \delta\_] = \frac{2}{\pi} * \frac{\left(\alpha^{\frac{1}{2*n}}\right)^{\psi[n]-1+n}}{\lambda[\alpha, n, r, \delta]} * \left(\text{BesselJ}\left[\nu[n]+1, \lambda[\alpha, n, r, \delta] * \left(\alpha^{\frac{1}{2*n}}\right)^{(1-n)}\right] * \text{BesselY}[\nu[n], \lambda[\alpha, n, r, \delta]] - \text{BesselY}\left[\nu[n]+1, \lambda[\alpha, n, r, \delta] * \left(\alpha^{\frac{1}{2*n}}\right)^{(1-n)}\right] * \text{BesselJ}[\nu[n], \lambda[\alpha, n, r, \delta]]\right)^{-1};$$

Define the velocity profiles and short-wavelengths approximations

Implement the generalized parabola (distribution of shear-wave modulus):

$$\text{DE} = 0.1;\ \text{ALPHA} = 0.5^2;\ \text{NN} = 0.2;$$

$$\text{fgen}[\eta\_, n\_, \alpha\_] = \left(1 + \left(\alpha^{\frac{1}{2n}} - 1\right) * \eta\right)^n;$$

$$\text{TFGO}[\alpha\_, n\_, r\_, \delta\_] := \frac{1 / \text{Sqrt}[\text{fgen}[1, n, \alpha]]}{\text{Cos}\left[\text{NIntegrate}\left[\frac{1}{\text{fgen}[x,n,\alpha]}, \{x, 0, 1\}\right] * \frac{r}{\text{Sqrt}[(1+i*\delta)]}\right]}$$

The V_{s30} approximations:

$$\text{TFGOVs}[\alpha\_, n\_, r\_, \delta\_, H\_] := \frac{1 / \text{Sqrt}[\text{fgen}[1, n, \alpha]]}{\text{Cos}\left[\text{NIntegrate}\left[\frac{1}{\text{fgen}[x,n,\alpha]}, \left\{x, 1 - \frac{30}{H}, 1\right\}\right] * \frac{r}{\text{Sqrt}[(1+i*\delta)]}\right]}$$

Use units for the shear-wave velocity at the base in this case (in m/s):

```
cs = 300;
```

Relevant range of frequencies

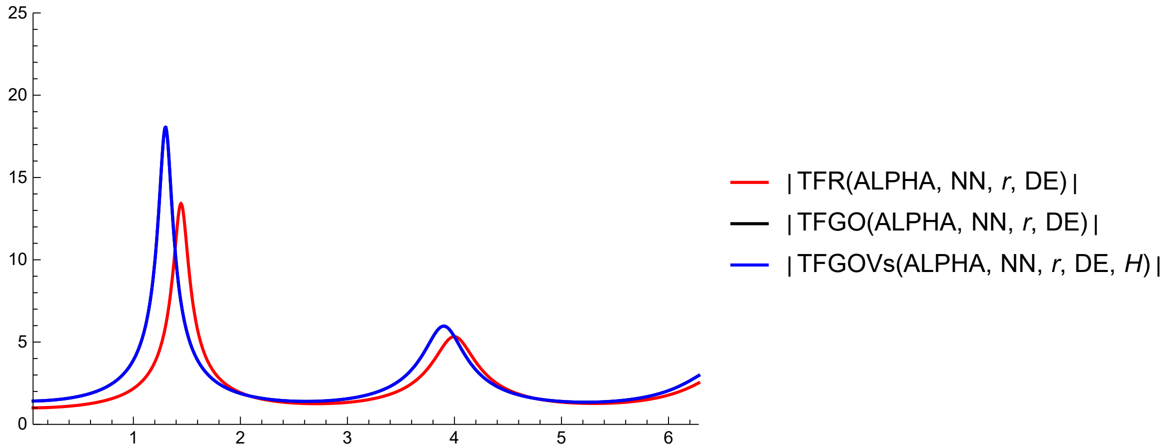
```
fmax = 10.; fmin = 0.1;
```

Upper tile Figure 5 (H=30m)

Parameters:

```
H = 30; rmin = fmin * 2 * π / cs * H; rmax = fmax * 2 * π / cs * H;
```

```
Plot[{Abs[TFR[ALPHA, NN, r, DE]], Abs[TFGO[ALPHA, NN, r, DE]],
  Abs[TFGOVs[ALPHA, NN, r, DE, H]]}, {r, rmin, rmax}, PlotRange → {{rmin, rmax}, {0, 25}},
 PlotStyle → {Red, Black, Blue}, PlotLegends → "Expressions"]
```

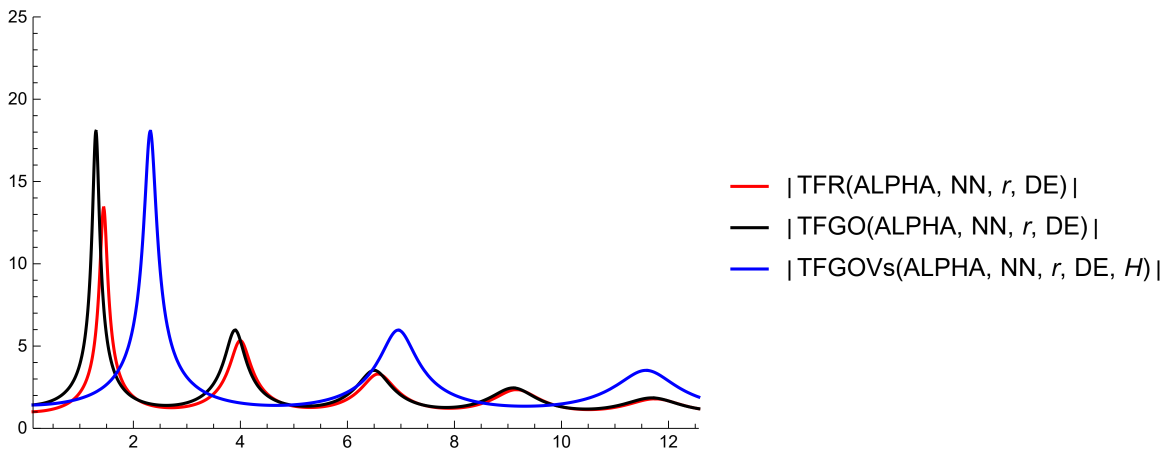


Middle tile Figure 5 (H=60m)

Parameters:

```
H = 60; rmin = fmin * 2 * π / cs * H; rmax = fmax * 2 * π / cs * H;
```

```
Plot[{Abs[TFR[ALPHA, NN, r, DE]], Abs[TFGO[ALPHA, NN, r, DE]],
  Abs[TFGOVs[ALPHA, NN, r, DE, H]]}, {r, rmin, rmax}, PlotRange → {{rmin, rmax}, {0, 25}},
 PlotStyle → {Red, Black, Blue}, PlotLegends → "Expressions"]
```




```
rlist = Range[rmin, rmax, 0.01];
```

```
Export["H60.xls", Table[{rlist[[jj]],
    Abs[TFR[ALPHA, NN, rlist[[jj]], DE]], Abs[TFGO[ALPHA, NN, rlist[[jj]], DE]],
    Abs[TFGOVs[ALPHA, NN, rlist[[jj]], DE, H]]}, {jj, 1, Length[rlist]}]];
```

Lower tile Figure 5 (H=180m)

```
H = 180; rmin = fmin * 2 * π / cs * H; rmax = fmax * 2 * π / cs * H;

Plot[{Abs[TFR[ALPHA, NN, r, DE]], Abs[TFGO[ALPHA, NN, r, DE]],
   Abs[TFGOVs[ALPHA, NN, r, DE, H]]}, {r, rmin, rmax},
  PlotRange → {{rmin, rmax}, {0, 15}}, PlotStyle → {Red, Black, Blue}]
```