Solutions for each layer of the shell. For the purposes of this worksheet, allow space for up to 20 layers.

```
In[1]:= v = Table[c1[k] * r^(n) + c2[k] * r^(-(n)), {k, 1, 20}];

In[2]:= dv = D[v, r];

In[3]:= Bn = dv[[1]] / u[1] /. r → R
```

Out[3]= $\frac{n R^{-1+n} c1[1] - n R^{-1-n} c2[1]}{u[1]}$

Define a function to get B.n just inside the inner boundary of the ABC region; this is called as part of the a minimization that computes permeability.

```
In[4]:= GetBn[enn_, dee_, seed_] := Module[{qsub, csub, eqs, m, k, intf, vars},
          m = Length[seed];
          qsub = Table[u[k] → seed[[k]], {k, 1, m}];
          qsub = Join[qsub, {R → 1, n → enn, d → dee}];
          eqs = {(v[[1]] /. r → R /. qsub) == 1, (v[[m]] /. r → R + m * d /. qsub) == 0};
          (* edges of boundary region *)
          For[k = 1, k < m, k++, (* interface conditions between layers in the boundary region *)
            intf = {(dv[[k]] / u[k] /. r → R + k * d /. qsub) ==
                (dv[[k + 1]] / u[k + 1] /. r → R + k * d /. qsub),
              (v[[k]] /. r → R + k * d /. qsub) == (v[[k + 1]] /. r → R + k * d /. qsub)};
            eqs = Join[eqs, intf];
          ];
          vars = Join[Table[c1[k], {k, 1, m}], Table[c2[k], {k, 1, m}]];
          csub = Solve[eqs, vars][[1]];
          Bn /. qsub /. csub
         ]
```

Now, figure out the permeabilities that satisfy the first m asymptotic boundary conditions. Do this via an error minimization. Note that working precision, accuracy, etc., have to be crazy, becuase the calculation isn't well conditioned.

```
In[5]:= GetPerm[een_, guess_] := Module[{ee, uu, m, U, seed, Q, k, ic},
          m = Length[guess];
          ee = Rationalize[een];
          seed = Table[U[k], {k, 1, m}];
          ic = Table[{U[k], guess[[k]]}, {k, 1, m}];
          uu = FindMinimum[Total[Table[(GetBn[k, ee, seed] + k)^2, {k, 1, m}]], ic,
            WorkingPrecision → 1000, MaxIterations → 10 000, PrecisionGoal → 100, AccuracyGoal → 100];
          N[{uu[[1]], seed /. uu[[2]]}]
         ]
```

Test to see if it works. Compute the permeabilities for $\delta = 1$ for a 3 rd order BC, assuming all ones as the initial condition.

```
In[6]:= GetPerm[1, {1, 1, 1}]
```

Out[6]= $\{7.537568223668950 \times 10^{-347}, \{1.002, 0.79096, 4.76838\}\}$

In[7]:=
```
pp = {0.001, 0.0025, 0.005, 0.01, 0.025, 0.05, 0.1}
```
Out[7]=
```
{0.001, 0.0025, 0.005, 0.01, 0.025, 0.05, 0.1}
```
In[8]:=
```
Off[FindMinimum::"precw"];
```
In[9]:=
```
Off[FindMinimum::"sszero"];
```
In[10]:=
```
Off[FindMinimum::"fmgz"];
```

Churn through each order, starting at $\delta = 1$ and reducing δ in small steps until $\delta = 0.001$

In[11]:=
```
MaxOrder = 7;
iclist = {};
ttt = {};
For[nn = 1, nn ≤ MaxOrder, nn++,
 (* NewGuess=Table[1,{k,1,nn}]; *)
 NewGuess = Join[{1}, iclist];
 tt = {};
 For[k = Length[pp], k > 0, k--,
  NewGuess = GetPerm[pp[[k]], NewGuess][[2]];
  tt = Append[tt, {pp[[k]], NewGuess}];
  Print[pp[[k]], "  ", NewGuess];
  If[k == Length[pp], iclist = NewGuess];
 ];
 ttt = Append[ttt, tt];
]
```
In[15]:=
```
For[k = 1, k ≤ MaxOrder, k++, Print[MatrixForm[ttt[[k]]]]]
```

0.1	{10.5238}
0.05	{20.5122}
0.025	{40.5062}
0.01	{100.502}
0.005	{200.501}
0.0025	{400.501}
0.001	{1000.5}

0.1	{0.278128, 17.0384}
0.05	{0.145314, 32.1429}
0.025	{0.0739412, 62.1965}
0.01	{0.0298421, 152.229}
0.005	{0.0149615, 302.239}
0.0025	{0.0074905, 602.245}
0.001	{0.00299849, 1502.25}

0.1	{1.94234, 0.152429, 20.6497}
0.05	{3.51534, 0.0786636, 37.4093}
0.025	{6.79961, 0.0403162, 70.7882}
0.01	{16.7699, 0.0164308, 170.815}
0.005	{33.4266, 0.00827262, 337.491}
0.0025	{66.755, 0.00415127, 670.829}
0.001	{166.752, 0.00166418, 1670.83}

0.1	{0.736225, 3.41246, 0.115694, 23.429}
0.05	{0.451776, 6.88636, 0.0627577, 41.0256}
0.025	{0.241961, 13.6813, 0.0330144, 76.075}
0.01	{0.0991743, 33.7712, 0.0136635, 181.105}
0.005	{0.049834, 67.1355, 0.00691422, 356.115}
0.0025	{0.0249636, 133.818, 0.00347834, 706.12}
0.001	{0.00999467, 333.827, 0.00139651, 1756.12}

0.1	{1.12973, 0.483856, 4.7899, 0.0977492, 25.8825}
0.05	{1.60697, 0.235357, 9.17218, 0.0551167, 43.9878}
0.025	{2.82388, 0.115251, 17.6107, 0.0296156, 80.0432}
0.01	{6.74992, 0.046171, 42.6698, 0.0124265, 188.077}
0.005	{13.3916, 0.0231843, 84.3558, 0.00631855, 368.089}
0.0025	{26.7125, 0.0116262, 167.699, 0.00318647, 728.094}
0.001	{66.705, 0.00465988, 417.704, 0.00128123, 1808.1}

0.1	{0.95916, 1.44817, 0.340272, 5.92399, 0.0863066, 28.1756}
0.05	{0.768586, 2.83065, 0.167703, 10.785, 0.0502436, 46.6238}
0.025	{0.476078, 5.94445, 0.085693, 20.2099, 0.0275201, 83.3525}
0.01	{0.20595, 15.0995, 0.0351873, 48.2639, 0.0116999, 193.391}
0.005	{0.104356, 30.1612, 0.0177861, 94.9486, 0.00597716, 376.737}
0.0025	{0.0523864, 60.193, 0.00894516, 188.291, 0.00302163, 743.41}
0.001	{0.0209862, 150.214, 0.00359105, 468.301, 0.00121674, 1843.42}

0.1	{1.01237, 0.852762, 1.93928, 0.26402, 6.8905, 0.0779844, 30.3781}
0.05	{1.14404, 0.496245, 4.11759, 0.138212, 12.0685, 0.0466781, 49.0737}
0.025	{1.67241, 0.233888, 8.31674, 0.0728682, 22.1585, 0.0260269, 86.2854}
0.01	{3.6822, 0.0899755, 20.4301, 0.030409, 52.2132, 0.011206, 197.757}
0.005	{7.20734, 0.0448338, 40.4653, 0.0154496, 102.232, 0.00575146, 383.485}
0.0025	{14.3269, 0.0224373, 80.4827, 0.0077897, 202.241, 0.00291454, 754.921}
0.001	{35.7415, 0.00898789, 200.493, 0.00313194, 502.246, 0.00117536, 1869.21}

Now, we can create a function that directly gets the collection of permeabilities for any δ and ABC order :

In[16]:=
```
GetPermAlt[dee_, enn_] := Module[{seed},
  seed = Interpolation[ttt[[enn]], InterpolationOrder → 1][dee];
  GetPerm[dee, seed][[2]]
 ]
```

Table of permeabilities needed for 0.1 R stackup case

In[17]:=
```
MatrixForm[Table[{1 / (10 k), GetPermAlt[1 / (10 k), k]}, {k, 1, MaxOrder}]]
```

Out[17]//MatrixForm=

$\frac{1}{10}$	{10.5238}
$\frac{1}{20}$	{0.145314, 32.1429}
$\frac{1}{30}$	{5.1494, 0.053269, 54.1064}
$\frac{1}{40}$	{0.241961, 13.6813, 0.0330144, 76.075}
$\frac{1}{50}$	{3.46611, 0.0920926, 21.7973, 0.0240632, 98.0545}
$\frac{1}{60}$	{0.333693, 9.02367, 0.057897, 29.5733, 0.0189665, 120.04}
$\frac{1}{70}$	{2.6499, 0.129533, 14.399, 0.0428826, 37.1975, 0.0156642, 142.03}